

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENİSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**AKILLI HAVA TRAFİĞİ KONTROL SİSTEMİNİN TASARIMI VE
SİMÜLASYONLARI**

Turkan Ali İbrahim AL-SALİHE

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2012**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

AKILLI HAVA TRAFİK KONTROL SİSTEMİNİN TASARIMI VE SİMÜLASYONU

Turkan Ali İbrahim AL-SALİHE

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Şahin EMRAH

Gün geçtikçe ulaşım araçlarına olan ihtiyaç artmaktadır. Şu an hava taşıtları, sağladıkları hız ve güvenilirlik açısından en önemli ulaşım araçları arasında yer almaktadır. Yapılan araştırmalara göre, özellikle uzun mesafeli yolculuklarda, hava taşıtlarının tercih edildiği saptanmıştır. Bu hava taşıtların düzenli, hızlı ve birbirleri ile uyumlu bir şekilde çalışması için hava trafik kontrol sistemi (HTKS) çok etkin bir rol oynamaktadır. Bu tezin amacı, hava trafik kontrolünün bir parçası olan uçuş sıralama ve zamanlama problemlerini ele alıp, bu nedenle olan gecikmeleri en aza indirmektir. Uçuş sıralamalarında optimal veya optimala yakın çözümler üretmek için iki farklı algoritma kullanılmıştır. Bu algoritmalar, zamanlamada olan gecikmeleri göz önünde bulundurma niteliği taşıdıklarından dolayı, akıllı yöntemler olarak belirlenmektedirler. Simülasyon sonucu elde edilen verilere göre gecikmeler ,eski yöntemle karşılaştırılarak çok daha iyi bir performans sergilediği görülmüştür. Yapılan araştırmalara göre, yalnızca ABD’de, gecikmelerin yerli havayolları şirketlerine yılda 3.5 milyar dolar gibi büyük rakamlara mal olduğu hesaplanmıştır. Dolayısıyla gelecekte bu algoritmaların hasarları önlemekte büyük katkı sağlayacağı düşünülmektedir.

Mayıs 2012 , 55 sayfa

Anahtar Kelimeler: hava trafik kontrolü, uçuş sıralama ve zamanlamaları, uçuş sıralama problemi, genetik algoritma, greedy algoritma, hava trafiği gecikme problemi.

ABSTRACT

Master Thesis

DESIGN AND SIMULATION AN INTELLIGENT AIR TRAFFIC CONTROL SYSTEM

Turkan Ali İbrahim AL-SALİHE

Ankara University
Graduate School of Natural and Applied Science
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Şahin EMRAH

The need for transportation means in whole world are increasing day after day. Factors like speed and safety found in air transportation means make them one of the most popular transportation means. Depending on the researches, especially at the long distance traveler those means become most preferred. Air traffic control system (HTKS) play active role to make those means work in systematic way and in coordination with each other. The goal from this thesis is to take a part of HTKS which is arrival sequence and scheduling problem(SDP) and minimize the delay produced because of this operation. Two kind of algorithms used to get optimal solutions or solutions close to optimal in means of loosing time. Those algorithms take in account the delay factor during the optimization of the arrival aircrafts. The result of the simulations being tested shows that those two algorithms have better performance than the classic method in means of delay. In USA alone, this has been estimated to cost domestic airlines as much as \$3.5 billion per year. So that we believe that in the near future, algorithms merging the intelligence with the reality can avoid this kind of damage.

May 2012 , 55 pages

Key Words: air traffic control, arrival sequencing and scheduling, arrival sequencing problem, genetic algorithm, greedy algorithm, minimize delay in air traffic.

TEŐEKKÖRLER

Çalıőma konusunun seèimimde, çalıőmanın planlanmasında ve deęerlendirilmesinde büyük katkılarını gördüğüm, bilgilerinden yararlandığım danışman hocam sayın Doç. Dr. Şahin EMRAH'a (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) teşekkür borçluyum.

Ayrıca çalışmam süresince ilgi ve desteklerini esirgemeyen tüm aileme teşekkür ediyorum.

Turkan Ali İbrahim AL-SALİHE
Ankara, Mayıs 2012

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER VE KISALTMALAR.....	vi
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	ix
1. GİRİŞ.....	1
2. HAVA TRAFİK KONTROLÜ	2
2.1 Hava Trafiği Kontrolü Hava Sahaları	3
2.2 HTKS'nin Genel Yapısı.....	4
3. DİNAMİK PLANLAYICI.....	7
3.1 Dinamik Planlayıcı Girişleri	7
3.1.1 Uçuş planları.....	7
3.1.2 Takip güncellemeleri.....	8
3.1.3 Tahmini varış zamanları (TVS'ler).....	8
3.1.4 Üst geçiş zamanları	9
3.1.5 Liste kısıtlamaları.....	9
3.1.6 Sıralama kısıtlamaları	10
3.2 Dinamik Planlayıcıdan Çıkışlar.....	10
3.2.1 Planlanan varış süreleri (PVS'ler).....	11
3.2.2 Uçuş pisti tahsisleri	11
3.3 SDP'nin Genel Açıklaması	12
3.4 İlk Gelene İlk Hizmet İGİH	13
4. GA İLE SDP PROBLEMİNİN OPTİMİZASYONU	16
4.1 Genetik Algoritma.....	16
4.2 Genetik Algoritma Uygulamaları.....	17
4.3 GA'nın Genel Tanımlaması	18
4.4 Genetik Algoritma'nın Aşamaları	20
4.4.1 Popülasyonun oluşması.....	21
4.4.2 Seçim	22
4.4.3 Çaprazlama (Rekombinasyon)	23
4.4.4 Mutasyon.....	24

4.4.5 Yeni kuşağın oluşması ve döngünün durdurulması.....	26
4.4.6 Aramanın bitirilmesi.....	26
5. HTK'ünde SDP PROBLEMİ.....	27
5.1 Hava Trafiği Kontrol Kısıtlamaları	27
5.2 Genetik Algoritma ile SDP Yaklaşımı.....	29
5.3 SDP Probleminin Formülasyonu.....	30
5.4 GA'nın Kromozom Yapısı.....	31
5.5 HTK'nın GA İle Uygulaması	32
5.6 GA İle Yeni SDP Yaklaşımı	35
6. DOYUMSUZLUK ALGORİTMASI (DA).....	37
6.1 DA'nın Özellikleri	37
6.2 DA'nın Türleri.....	38
6.3 DA'nın Uygulama ve Örnekleri	38
6.4 DA ile SDP Problemi.....	39
7. TARTIŞMA VE SİMİLASYONLAR	43
7.1 GA ile İGİH Karşılaştırmaları.....	43
7.2 DA ile GA Similasyonu	47
8. SONUÇ.....	52
ÖZGEÇMİŞ.....	55

SİMGELER VE KISALTMALAR

Σ	Kartezyan Toplam
μ	Mutasyon oranı
SDP	Varışların sıralama ve zamanlama
Ceil A	A sayısına en yakın pozitif sayı
CTAS	Merkez-TRACON otomasyon sistemi
DA	Doyumsuzluk Algoritması
DP	Dinamik planlayıcı
TVS	Tahmini varış süresi
İGİH	İlk gelene ilk hizmet verilir yöntemi
GA	Genetik Algoritma
HTK	Hava trafik kontrolü
LTI	İnişlerin zaman aralığı
nm	Nanometre
rand	Rastgele sayı
PVS	Planlanan varış süresi
TMA	Trafik yönetim danışmanı
TMC	Trafik yönetim koordinatörü
TRACON	Terminal radar yaklaşma kontrolü

ŞEKİLLER DİZİNİ

Şekil 2.1 Hava alanı sahasının genel yapısı.....	5
Şekil 3.1 Basit trafik varış örneği.....	11
Şekil 4.1 Genetik Algoritmanın akış diyagramı.....	21
Şekil 4.2 Popülasyon.....	22
Şekil 4.3 Seçim.....	23
Şekil 4.4 Çaprazlama işlemi.....	24
Şekil 4.5 Mutasyon işlemi.....	25
Şekil 5.1 İGİH iniş sıralaması ve yer değiştirme optimizasyonu.....	30
Şekil 5.2 Kromozumun yapısı.....	32
Şekil 5.3 Oluşturulan popülasyon örneği.....	33
Şekil 6.1 DA'nın SDP problemi için işlem adımları.....	40
Şekil 7.1 Matlab ile hazırlanan arayüz.....	44
Şekil 7.2 Uçak sıralama, uçak kategori ve uçak isimleri matrisleri.....	44
Şekil 7.3 GA ve İGİH ile 5 uçağın optimizasyonu.....	45
Şekil 7.4 GA ile çıkan sonuçlar ve her jenerasyonun ortalama gecikme değerleri.....	46
Şekil 7.5 Uçak iptal fonksiyonu.....	46
Şekil 7.5 Matlab komut penceresinde 5 farklı kategoriye sahip uçağın optimizasyon sonuçları.....	50
Şekil 7.6 Üç yöntemin graf halinde karşılaştırmaları.....	50

ÇİZELGELER DİZİNİ

Çizelge 3.1 İGİH sırlaması ve zamanlayıcı sonuçları.....	15
Çizelge 5.1 İnişler için minimum zaman aralığı.....	28
Çizelge 7.1 GA ile İGİH karşılaştırma ve performans değerlendirmesi.....	47
Çizelge 7.2 Simülasyon sonucu elde edilen MSExcel tablosunun örneği.....	49

1. GİRİŞ

Hava trafik kontrolünde (HTK), özellikle önemli olan konu, artan trafik hacmi ve uçak gecikmeleridir. Heere ve Zelenka 2000'e göre hava trafiğinin gelecek on yılda dünya çapında artması beklenmektedir: 1996'da tüm dünyada 1.7 trilyon olan yolcu mili gelirin 2016'da 4.5 trilyona ulaşması beklenmektedir. Hava trafik sistemi üzerindeki talep, rutin olarak hava alanlarının kapasitelerini aşmakta ve hava taşıtlarının yerde ve havada gecikmelerine neden olmaktadır. Yalnızca ABD'de, bu gecikmelerin yerli hava yolları şirketlerine yılda 3.5 milyar dolar gibi büyük rakamlara mal olduğu hesaplanmaktadır. Pazara göre belirlenen fiyatlar ve çok ince kâr marjlarıyla giderek daha rekabetçi hale gelen endüstride, bu gibi işletme cezaları artmaktadır.

Hava yollarında, uçuş programlarına bağlı kalınması, verimli işletim ve kârlılık açısından önemlidir. Havaalanına gelen uçak gruplarının iniş kalkışlarının merkezden kontrolüne "bank yönetimi" denir. Güvenilir ve etkin bank yönetimi hava yollarının başarısı için kritik önem taşır. Merkez kontrol şebekeleri, birbirine yakın iniş ve kalkışlar, kötü hava şartları veya havayolu tıkanıklığı nedeniyle ortaya çıkan zamanlama hatalarına karşı çok hassastır. İniş zamanı hataları, yolcu ve mürettebat bağlantılarının kaçırılmasına, yer işletmesinin verimsiz olmasına ve bazen de uçakların alternatif havaalanlarına yönlendirilmelerine yol açar. Bunlar, yolcuların rahatsız olmasına, uçuşların gecikmesine ve havayollarının gelir kaybetmesine neden olur (Hansen 2004). Akıllı Hava trafik kontrolü, yukarıda belirtilen sorunlara etkili çözüm yollarını aramaktadır. Üretilen yöntemlerin amacı, havayollarında giderek artan yoğun trafik tıkanıklığını, gelir kaybına uğratmadan ve daha emniyetli bir şekilde yönetmektir.

2. HAVA TRAFİK KONTROLÜ

Bir bölge veya ülkedeki hava trafik kontrol sistemi kendi hava sahasında uçan tüm hava trafiğini yönetir, kontrol sektörlerini belirler, farklı havaalanları arasındaki akışı yönetir ve uçakların uçuşları, kalkışları ve inişleri sırasında aralarında mesafe olmasını sağlar. Yani, her bir trafiğin bir sonraki düzeye geçişini kontrol etmek, güvenliğini sağlamak ve sınırlamak amacıyla tasarlanmış olan farklı düzeylerde çalışır. Bir havayolu uçuş planı gerçek kalkış zamanlarını yansıtabilecek şekilde güncellendiğinde, genellikle uçuşun sabit hızla uzun süre devam eden bölümü için iyi bir yaklaşım sağlar. Ancak beklenmedik gecikmeler hava taşıtlarının, zaman tahminlerini önemli derecede değiştirebilir. Hava trafik kontrolünün temel bir parçası merkez havaalanındaki tüm aktif iniş pistlerinde, iniş trafiğinin sırasının ve programının belirlenmesini sağlayan bir modül olmasıdır. Sıranın belirlenmesi, bir grup hava taşıtı için iniş sırasını, programlamayla gruptaki her hava taşıtı için piste deęme zamanını spesifik olarak belirler.

Mevcut uygulamada, hava taşıtları için varış havaalanına 200 deniz mili veya 45 dakikalık uçuş zamanı varken sıra belirleme ve programlama yapılır. Planlama sürecinin asıl amacı havadaki toplam gecikmelerin en aza indirilmesi sağlanırken tüm aktif pistlerde hava trafiğinin dengeli olmasının sağlanmasıdır. Etkin bir analiz, güvenliği teşvik eder, hava yolları için getirilen kısıtlamaların plana alınmasına yardım eder ve pistin genişletilmesinin faydalarıyla ilgili bilgi sağlar.

Kompleks hava terminali alanlarında trafik akışını optimize eden otomatize sistemlerin geliştirilmesi, havacılık arařtırmacıları ve planlayıcıları tarafından uzun zamandır hedeflenmektedir. Tüm dünyada devam eden hava trafięi artışı ve terminallerde alan gecikmelerinin giderek kötüleşmesi bu sistemlerin geliştirilmesindeki aciliyeti arttırmaktadır (Hansen 2004). Hava trafik kontrolü üzerindeki yeni arařtırmaların çoęu NASA Ames Arařtırma Merkezi tarafından yapılmaktadır. Bu arařtırmalardan varış trafiğinin planlanması ve programlanması için kullanılacak olan ve trafik yönetimine gerçek zamanlı olarak yardım eden Center-TRACON Otomasyon Sistemi ortaya çıkmıştır (CTAS). CTAS'nin kilit bileşeni kontrolörlere havaalanından yaklaşık 35-200 deniz mili uzaktan gelen hava taşıtlarının programa alınması için yardım eden TMA'dır.

TMA varış zamanlarının optimize edilmesi için terminal alanına doğru akışın düzgün olmasını sağlamak üzere sırayı, varış zamanlarını ve pist atama işlemlerini hesaplar. Bunu yaparken TMA'nın trafik akışını kısıtlayan veya hava taşıtları arasında gerekli mesafeyi etkileyen faktörleri hesaba katmak gerekir. Bu kısıtlamalar trafik yoğunluğunu, hava taşıtı tip dağılımını ve benzerlerini içerebilir.

2.1 Hava Trafiği Kontrolü Hava Sahaları

Hava trafiği akışını yönetmenin yakıt verimliliğini artırmak için gerekli olduğu ve güvenliği sağlarken aynı zamanda kontrolörler üzerindeki iş yükünü de hafiflettiği netleşmiştir. Buna ek olarak TMClere trafik akışı yönetimi sorumluluğu da verilmiştir.

TMA, TMC leri ve kontrolörleri pek çok yönden desteklemektedir. İlk olarak, TMA TMC lere bir hava aracının ne zaman hangi noktada olacağını tahmin ederek gelecekteki trafik akışını tahmin etmelerinde yardımcı olmaktadır. TMA aynı zamanda bu tahminleri istenilen sıralamaları ve genel trafik akışını geliştirmek için çeşitli referans noktalarına varış zamanlarını hesaplamak için kullanmaktadır. Varış zamanlarını etkili kılmaya yardımcı olmak için TMA her bir uçak için uçuş pisti tahsisini hesaplamaktadır. TMA trafik akışı ile ilgili analizler ve raporlar oluşturarak trafik analizine destek olmaktadır.

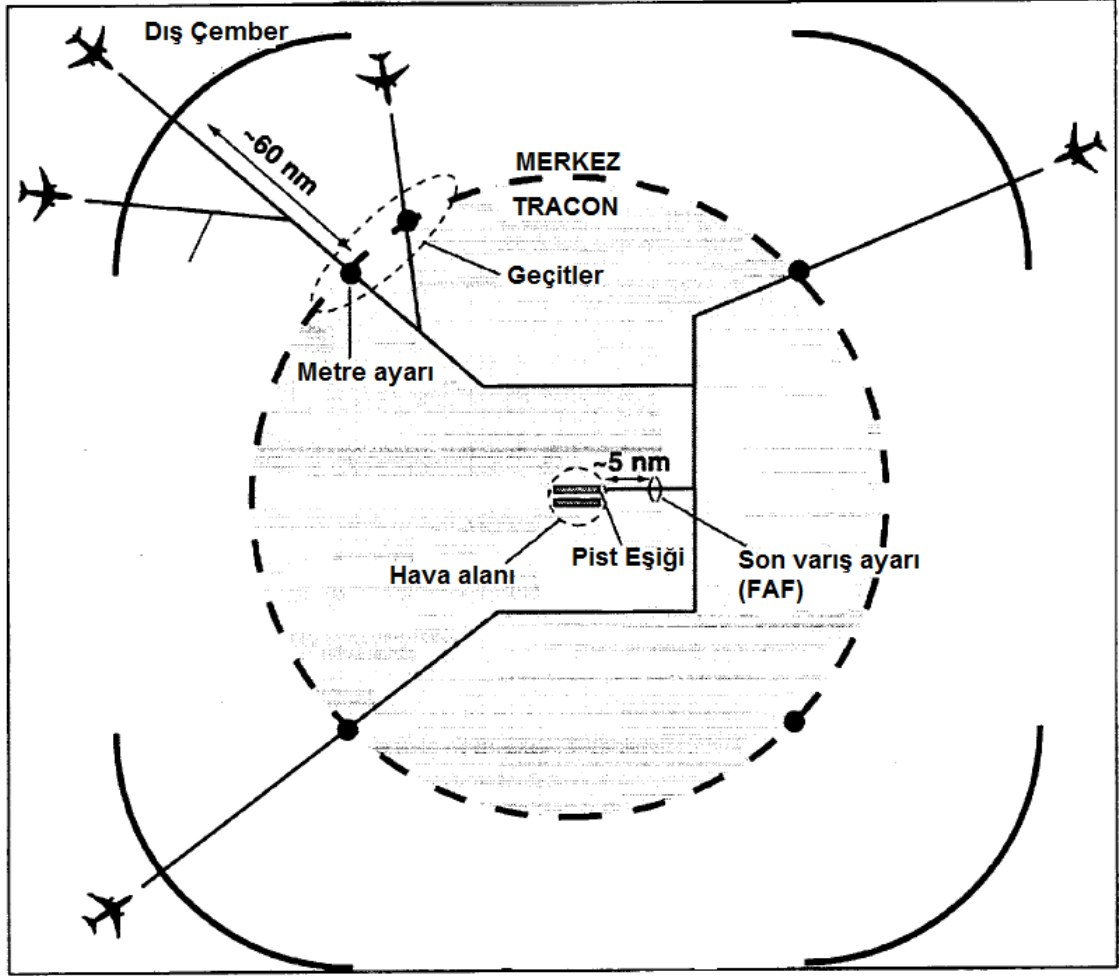
TMA içerisinde dinamik planlamaların(DP) rolü terminal alanı içerisinde trafik akışı etkililiğinden emin olmak için düzgün olan sıralamaları, varış zamanlarını ve uçuş pisti tahsislerini hesaplamaktır. Normal olarak, DP uçakları sıralamaktadır ve bu nedenle uçaklar ilk geliş – ilk hizmet(İĞİH) düzeni içinde varmaktadırlar fakat TMClere bu düzeni sıralama kısıtlamasına giderek geçersiz kılabilirler. Buna ek olarak TMC trafik akışını kısıtlayan veya uçaklar arasında gerekli olan ayrımı yapmayı etkileyen kısıtlamaların bir listesini yapmaktadır. Bu kısıtlama listeleri mevcutteki uçuş pisti kapasitesi, trafik yoğunluğu, uçak türü dağılımı, hava alanı yapılandırması ve hava durumları için gereklidir . Bu araştırma DPnin algoritmalarının ve birimlerinin detaylarını açıklamaktadır. Hava trafiği için listeleyicilerin gelişimde hatırı sayılır

derecede bir artış olmuştur. Bunların çoğu İGİH düzeninin türlerini içermektedir. Aynı zamanda özellikle maliyet etkisini en üst seviyeye taşıyan bir listeleyicinin tasarlanması için de girişimler olmuştur(Wong 2000). Bu listeleyicilerin bazıları diğerleri TMC tarafından Ft. Worth Center gibi yoğun hava trafiği kontrolü işlerinde kullanılması gerekli görülen kısıtlamalar ile zıtlık oluştururken yeterli derecede titiz olmamıştır. DP kavramsal olarak daha önceden geliştirilmiş olan bu listeleyicilerin pek çoğu ile benzerlik göstermektedir fakat aşağıdakiler için tasarlanmıştır:

1. Çeşitli özel trafik kısıtlamaları için hesapları açıklamak.
2. CTAS için rota tahmini sağlayan liste güncellemeleri sağlamak.
3. TRACON ve Merkez arasında tahsis gecikmelerini azaltmak .
4. Gelişmiş yenileme özelliklerini daha geç bir tarihe ekleme özelliklerine izin vermek.

2.2 HTKS'nin Genel Yapısı

Bazı tipik varış uçaklarının uçuş yolları şekil 2.1'de gösterilmiştir. Her bir varış uçağı Merkezin hava sahası içerisinde uçtuğundan dolayı dış çemberden geçmektedir. Dış çember kendi ortak metre ayarının yaklaşık 60 nm dışındadır. Uçak daha sonrasında Merkezin ve TRACON'un hava sahaları arasındaki sınırdaki çizgi oluşturan metre ayarından geçer. Bazı merkezler için, DP metre ayarlamalarını birlikte gruplar ve bu metre ayarlama gruplarına da geçit denilmektedir. Varış uçağı, metre ayarlamasından yararlanarak, TRACON'un hava sahasına doğru uçar, son varış ayar noktasını (FAF) geçer ve hava alanı pistine inmiş olur.



Şekil 2.1 Hava alanı sahasının genel yapısı

DP her bir uçak için dış çember, metre ayarı, FAF ve hava alanı eşiğinde varış zamanını listeler. Dış çember, metre ayarı, FAF ve hava alanı eşiğinin hepsi birlikte Referans Noktaları olarak bilinmektedir ve varış zamanlarının hesaplanmasında kullanılan her biri Listelenmiş Varış Zamanları (PVS) olarak tanımlanmaktadır. Çeşitli trafik, hava durumu ve hava alanı durumunu hesaplamak için, TMC (DP için birincil kullanıcı) mesafe ve kabul oranları gibi listelemeye sınırlama girişleri yaparak listelemeyi kontrol edebilir. DP her bir varış uçağı için PVS hesaplaması yaparken bu listeleme sınırlandırmalarına uymalıdır. Buna ek olarak DP, ilk gelene ilk hizmet verilir (İGİH) düzeni içerisinde metre ayarına varacak olan uçakları da sıralar. TMC özel sıralama sınırlamaları koyarak bu sıralamaları değiştirmektedir. Tüm bunlara ek olarak, DP, uçak pisti tahsisi olarak bilinen işlem ile genel gecikmeyi azaltan uçak pistlerine uçak tahsis

edecektir. TMC hava alanı tahsislerine önem vermemektedir ve bölgesel uçak pistlerine kendisi bizzat uçak tahsis etmektedir.

Tüm sıralama, listeleme ve uçak pisti tahsisi uçak merkezin hava sahası içerisindeyken (varış hava alanından yaklaşık 25 – 300 mil uzakta iken) yapılmaktadır. Buna ek olarak, bazı uçaklar için listeleme uçak merkezinin hava sahasına girmeden hemen önce gerçekleşmektedir. DP sadece listeleme yapmak için bir uçağın uçuş planını talep eder. Bu bir uçağın, merkezin hava sahasına girmesinden 90 dakika önce ortaya çıkar. DP trafik durumundaki, doğadaki ve TMC girişlerindeki değişikliklerden kaynaklanan değişimlere uyum sağlamak için eş zamanlı olarak bu listeleri, sıralamaları ve uçuş pisti tahsislerini günceller.

Dış çemberi oluşturan tüm noktalar TMA tarafından tek bir nokta olarak kabul edilmektedir. Böylece, dış çember varış zamanı bir uçağın, dış çemberin her hangi noktasından geçeceği zamandır.

3. DİNAMİK PLANLAYICI

3.1 Dinamik Planlayıcı Girişleri

DP'lere girişler aşağıda özetlenmiştir:

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

3.1.1 Uçuş planları

Uçuş planları içeride olan veya Merkezin hava sahasına girmek üzere olan her bir uçak için önemli bilgiler içermektedir. Uçuş planları hava yolları tarafından sunulmaktadır ve Merkezin ana bilgisayarından işleme sokulmaktadır. Uçuş planları aynı zamanda hava trafiği kontrolörleri tarafından uçak kendi talimatları içerisindeyken de düzeltilmelidir.

DP, CTAS tarafından işleme alınan her bir uçak için bir uçuş planı alır. 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 de DP'nın 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.

3.1.2 Takip güncellemeleri

CTAS Merkezin ana bilgisayarından, Merkezin hava sahasında bulunan uçakla ilgili olarak radar takiplerini alır. Radar takipleri alınan bu uçakları aktif uçaklar olarak bilinmektedir. DP aktif ve aktif olmayan uçaklar için uçuş planlarını hazırlar.

3.1.3 Tahmini varış zamanları (TVS'ler)

TMA'ya ait Rota Analizcisi (RA) ve Rota Birleştiricisi (TS) programları, Şekil 2.1'de gösterilen her bir referans noktasına Tahmini Varış Zamanlarını (TVS'leri) DP'ye sağlamak için beraber çalışırlar. RA varmak üzere olan uçağın yatay rotasını 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 mevcutteki yerinden uçuş pistindeki değme noktasına kadar 4 boyutlu bir rota hesaplar. Bu rotadan, RA referans noktaları için TVS'leri çıkartabilir ve bu TVS'leri DP'ye gönderir. TVS'leri her bir radar güncellemesi için tekrardan hesaplanır (Wong 2000).

DP'nin perspektifinden, TVS'ler hava sahası içerisinde başka bir uçak bulunmadığında her bir uçağın her bir referans noktasından ne zaman geçeceğine dair bir belirleyicidir. Bu nedenle DP en yeni yetkili PVS olarak TVS'leri etkiler. Tüm uçakları içerir biçimde bir liste inşa ederken diğer uçaklar ile karışıklığa neden olmamak için her bir uçağın PVS ertelenmektedir. Böylelikle olarak kontrolör veya TMC tarafından elle yapılan bir işlemi ortaya çıkartmak her bir uçağın PVS değeri TVS'ne eşit veya ileri olacaktır.

3.1.4 Üst geçiş zamanları

RA, DP'ye bir uçağın Merkezin hava sahasından TRACON'un hava sahasına giriş zamanını gönderir. DP ile birlikte bu zaman üst geçiş zamanı olarak tanımlanır. Genellikle bu zaman uçağın metre ayarından geçtiği zamandır. Buna rağmen, bazı uçaklar TRACON'un hava sahasına girerken metre ayarından geçmezler ve RA buna ait özel bir mantığa sahiptir. Diğer bir durumda DP üst geçiş zamanını uçağın metre ayarından geçtiği zaman olarak tanımlamaktadır.

3.1.5 Liste kısıtlamaları

DP, PVS'ları hesaplarırken liste kısıtlamalarına uyar. Bu liste kısıtlamaları TMC tarafından aktif ve gelecek hava alanı kapasitesi, trafik karışıklığı, hava durumu, personel seviyesi, uçuş pisti topolojisi ve hava trafiği kontrol prosedürlerini etkilemek için düzenlenir. Liste kısıtlaması olmadığı durumlarda her bir uçak için hesaplanan PVS kendi TVS'sı ile eşit olur. Buna rağmen trafik yeterince ağır olduğunda veya liste kısıtlamaları yeterince açık olduğunda DP liste kısıtlamalarına uymak için uçağı geciktirir. Bunun sonucu olarak uçağın PVS'ları kendi TVS'sından daha sonra olur.

Liste kısıtlamaları aşağıda listelenmiş ve yazılım uygulamaları içindeki benzerliklerine göre gruplanmıştır.

- Ayırma Mesafesi

Yoldaki miller. Bu uçaklar arasındaki bölgesel metre ayarını geçtikleri sırada izin verilen en az yatay mesafedir.

Canlı girdap ayrımı. Bu uçaklar arasındaki ayrılmış uçuş pistine girdikleri sırada izin verilen en az mesafedir.

- Kullanım süresi

Uçuş pisti kullanım zamanı. Bu ayrılmış uçuş pistine inişler arasında izin verilen en az zaman toplamıdır. TMC bu liste sınırlamasını uçuş pistinde durma durumları veya uçuş pistini boşaltmak için gerekli fazladan zamanı hesaplamak için kullanmalıdır.

- Kabul Oranı

TRACON kabul oranı. Bu metre ayarından geçen veya her saat başı TRACON'a giriş yapan uçak sayısını tanımlamaktadır.

Metre ayarı kabul oranı. Bu bölgesel metre ayarından her saatte geçen uçağın sayısını tanımlamaktadır.

Giriş kabul oranı. Bu ayrı girişler ile birlikte her saatte herhangi bir metre ayarından geçen uçak sayısını tanımlamaktadır. Aynı hava sahası bölgesinde olan metre ayarları giriş olarak birlikte gruplanmaktadır. Girişlere metre ayarlarının gruplanması CTAS'ın alan ilişkili verilerinin bir parçasıdır.

Hava alanı kabul oranı. Bu her bir saatte ayrılmış hava alanına iniş yapacak olan uçak sayısını tanımlamaktadır.

Uçuş pisti kabul oranı. Bu her bir saatte ayrılmış uçuş pistine iniş yapacak olan uçak sayısını tanımlamaktadır.

- Engellenmiş Mesafe

Metre ayarı tarafından engellenmiş mesafe. Ayrı metre ayarından hiçbir uçağın geçmesine izin verilmeyen zaman devresidir.

Uçuş pisti tarafından engellenmiş mesafe. Ayrı bir uçuş pistine hiçbir uçağın iniş yapmasına izin verilmeyen zaman devresidir.

3.1.6 Sıralama kısıtlamaları

DP normalde İGİH düzeni içerisinde metre ayarlarına göre kendi TVS'larına dayanarak uçakların metre ayarına varışlarını listeler. Buna rağmen TMC diğer bir uçaktan önce veya sonra listelenmesi gereken uçağı etkileyecek bir biçimde listeleri kısıtlayabilir. DP, metre ayarlarında sıralama yaparken bu kısıtlamaların hepsini hesaba katmaktadır.

3.2 Dinamik Planlayıcıdan Çıkışlar

DP den çıkışlar aşağıda özetlenmiştir.

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

3.2.1 Planlanan varış süreleri (PVS'ler)

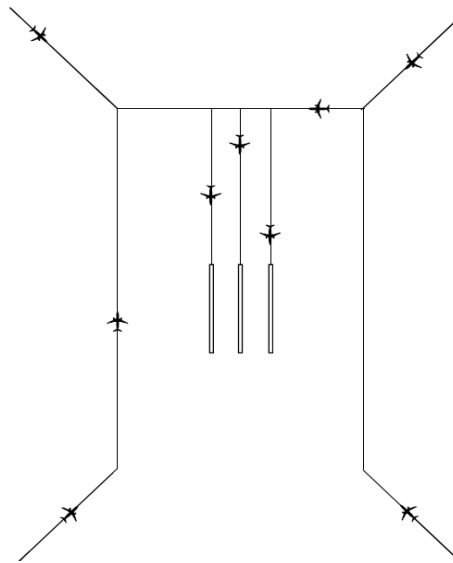
DP'nın hesaplamaları her bir uçak için aşağıdaki referans noktalarında PVS'ların bir seti ile sonuçlanır.

- Dış metre arkı
- Metre ayarı
- FAF
- Uçuş pisti eşiği

Bu PVS'lar tüm listelemeler ve sıralama kısıtlamaları ile birleştirilir. PVS'lar kontrolörlerin ve TMC'lerin çalışma istasyonlarında kendi hava trafiği kontrollerine yardımcı olmak amaçlı olarak görüntülenmektedir. Buna ek olarak, DP diğer CTAS lara kontrolör talimatlarını işleme sokmaları için STAları gönderir.

3.2.2 Uçuş pisti tahsisleri

Uçuş Pisti Tahsisi olarak bilinen işlem içerisinde DP her bir uçak için bir uçuş pisti tahsis edecektir. Bu uçuş pisti tahsisleri her bir uçağın gecikmesini azaltarak listelemeyi verimli kılacaktır. İlaveten TMC, DP tarafından hesaplanmış olan uçuş pisti tahsislerini ihlal edebilir ve bir uçağa ayrı olarak bir uçuş pisti tahsis edebilir.



Şekil 3.1 Basit trafik varış örneği

3.3 SDP'nin Genel Açıklaması

SDP problemi, uçuş pisti içerisinde n kadar varış uçuşları için uçuş pisti tahsisini, sıralamayı ve listelemeyi göz önünde bulundurarak çözümlenmiştir.

Her bir n varış uçuşu için her bir uçuş pistine tahmini varış zamanı (EST) uçuş pistine varmak için en az zaman olarak geçerlidir. Uçak performansı ve uçuş türlerinde herhangi bir ayırım bu TVS değerlerinin belirlenmesinde kullanılmaktadır. Bir uçuş pistine uçuşun listesi kesişen TVS den daha erken olamaz.

Bir uçuş pisti üzerine inmek için listelenen tüm uçuşlarda, uçak türüne göre ilk gelen ve onu takip eden uçaklar arasında özel bir ayırım yapılır. Bu kapsamda uçak türü ; Ağır (H), Geniş (L) ve Küçük (S) olarak üç çeşittir. Belirlenen ayırımlar, uçuş pisti eşliğinde belirlenen zaman yarımlarının terimlerinde verilmek üzere sayılmıştır (Hansen 2004).

Bir uçak, bir uçuş pisti için listelendiğinde listenin bir sonucu olarak ortaya çıkan gecikme o uçuş için tüm uçuş pistlerine göre listelenmiş varış zamanı (PVS) ve en erken TVS arasındaki fark olarak açıklanmaktadır. 12 uçuş ve 3 uçuş pisti ile birlikte dört şemayı değerlendirmek için keyfi olarak basit bir senaryo oluşturulmuştur. Amaçların tanımlanması için çağrı işaretleri çağrı işareti listeleri tarafından verilmektedir = (UA 138, UA 532, UA 599, NW 358, UA 2987, AA 128, UA 1482, NW 357, AA 129, UA 2408, UA 805, AA 399); ve uçak türü listesi tarafından verilen kesişen uçak türleri = (H, L, H, H, S, H, L, H, H, S, H, L). 12 uçuş ve 3 uçuş pisti için TVS ların listeleri aşağıdaki dizey yolu ile verilmiştir.

$$\text{eta} = \begin{bmatrix} 11 & 10 & 9 \\ 15 & 17 & 19 \\ 6 & 7 & 8 \\ 6 & 7 & 8 \\ 9 & 12 & 15 \\ 7 & 6 & 5 \\ 15 & 17 & 19 \\ 6 & 7 & 8 \\ 6 & 7 & 8 \\ 9 & 12 & 15 \\ 7 & 6 & 5 \\ 9 & 7 & 5 \end{bmatrix}$$

Herhangi bir iniş yapan sıralama genellikle ayırım dizeyi tarafından belirlenen uçak ayırımı için talimatlara uymalıdır. Bu problemde kullanılan dizey şöyledir:

$$\text{sep} = \begin{bmatrix} 1 & 1.5 & 2 \\ 1 & 1.5 & 1.5 \\ 1 & 1 & 1 \end{bmatrix}$$

Sıraların öncelikli gelen uçakları gösterdiği ve kolonların da arkasından gelen uçağı gösterdiği yerlerde Ağır, Geniş ve Küçük uçak türleri içerisinde düzenlenen tüm ölçütler ile birlikte. Örneğin, küçük bir uçak arkasından gelen geniş bir uçak sadece 1 zaman ünitesi ayırımı gerektirirken, ağır bir uçağı takip eden küçük uçak 2 zaman ünitelik ayırımı gerektirmektedir.

3.4 İlk Gelene İlk Hizmet İGİH

Uçaklar, minimum uzaklık kriterlerini taşıdığı sürece gerek geçitlerde olsun gerekse pistte ilk gelene ilk hizmet yöntemine göre sırlanır. Bir uçak çember geçiti geçerken en az 5nm (60 saniye) kuyruk mesafesi gerekmektedir. Pistte olması gereken uzaklık sınırlamaları uçağın ağırlık sınıfı ve FAA tarafından belirlenen güvenli iniş kurallarına göre gerçekleşir. Ayrıca TRACON'a girmelerine izin verilen uçaklar ,10 dakika zaman

aralığında sınırlayarak ve toplam gecikmeleri minimize edecek şekilde , pistlere dağıtarak havaalanı kabul oranını dikkate almaları gerekir. Varan uçakların ,İGİH sıralaması TVS'lerindeki artışa dayanarak elde edilir. İGİH sırası artan TVS'ya göre, zaman ayarlı varış uçağı tarafından kurulmaktadır.

Sıralamada birinci olan uçak ile başlayarak, her bir uçak geçici olarak uçuş pisti eşiğindeki uçaklar arasında tahmin edilen en az zaman ayırımının her biribirinin ardından gelen uçağı uyum gösterdiğinden emin olduğunda üç uçuş pistinin her birisine listelenirler. Uçuş pistindeki uçak için, listelenen en erken varış zamanı ile sonuçlanan uçuş pisti (PVS) daha sonrasında iniş için uçuş pisti olarak seçilir. Uçuş pistine yönelik listeleme otomatik olarak trafik yoğun iken (uçuş pisti dengelemesi) uçuş pistlerine uygun doğru trafiğı ve uçağı iniş yapması için en yakın ve en uygun pisti ayarlar. Besleme kapısında listelenmiş varış zamanı (PVS), önceden hesaplanmış olan PVS'den herhangi bir TRACON gecikmesinin ve TRACON geçişi toplamının çıkartılması ile belirlenir. Son olarak, eğer iki uçuş için PVS'lar gerekli olan 60 saniye az ise listelenmiş zamanlar besleme kapısında gerekli ayırımı yapmaları için uyarılırlar.

Çizelge 3.1, bir uçağın İGİH sıralamasına göre sıralandığında sonuç olarak elde edilen varış düzenini göstermektedir. Her bir bank uçağının öncelik sıralaması uçak tanımlayıcısını takip eden parantezler içerisinde gösterilmiştir. Tablodaki ikinci ve üçüncü kolonlar artan TVS'ya göre zaman ayarlı uçak ile birlikte listeye eklenen İGİH sıralamasını göstermektedir. Dördüncü ve beşinci kolonlar da artan PVS'ya göre zaman ayarlı uçaklar ile sonuçlanan listedir. Uçuş pistinde varışın sonuçlanan liste düzeninin kesin olarak listeye eklenen TVS'ya dayanan İGİH sıralamasına uymadığını not edilir(Carr, vd. 1998). Çünkü liste hem besleme kapısında hem de uçuş pisti eşiğinde sonraki ayırım kriterine uyum sağlamalıdır ve uçuş pisti eşiğinde yapılan ayırımlar uçak ağırlık sınıfı ve iniş düzeninin bir fonksiyonudur, İGİH sıralaması uçuş pistinde korunmamalıdır. İniş yapılacak olan sıraya ait olan uçak arasında AAL 1934 ve AAL 1428 uçuşları listeye eklenen sıralama içerisindeki yerlerini değiştirmiştir (varış sırasına ait olmayan DAL 431 ve AAL 410 uçaklarına sahip olarak). Bu durum, durum değişikliğı TVS'ya dayanarak girişleri yapılan İGİH varışlarının beklenen veya ideal

olan düzenine daha yakın bir biçimde uyum sağlayan liste ile sonuçlanmaktadır. Ancak sonuçlanan listenin önerilen düzene daha yakın olması durumu geçicidir ve merkeze varış hatasının önemine dayanarak listelenen düzen daha sonraki tercih edilen düzenden sapmaktadır.

Çizelge 3.1 İGİH sırlaması ve zamanlayıcı sonuçları

Sıralama Numarası	FCFS Sıralaması		Zamanlayıcı Sonuçları	
	Uçağın ID'si	TVS	Uçağın ID'si	PVS
54	DAL428	2972	DAL428	3483
55	AAL1150 (1)	3007	AAL1150 (1)	3507
56	DAL1670	3086	DAL1670	3540
57	EGF006	3089	EGF006	3593
58	DAL834	3146	DAL834	3601
59	DAL2062	3206	DAL2062	3627
60	UAL359	3206	UAL359	3695
61	ASE924	3212	ASE924	3737
62	AAL1428 (4)	3212	AAL1934 (3)	3768
63	AAL1934 (3)	3266	AAL1428 (4)	3789
64	AAL1554 (5)	3272	AAL1554 (5)	3843
65	EGF628 (2)	3300	EGF628 (2)	3895
66	AAL410	3326	DAL431	3952
67	DAL431	3441	AAL410	3953
68	DAL756	3624	DAL756	4039

4. GA İLE SDP PROBLEMİNİN OPTİMİZASYONU

Hava alanı pistlerine 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 verilir yöntemiyle belirli bir sıralamaya tabi tutlur. Fakat bu yöntem ciddi şekilde gecikmeye neden olmaktadır. En uygun ve optimal sıralamayı bulmak için farklı yöntemler kullanılabilir. Bu yöntemler arasında Genetik Algoritma bu problem için en uygun olanıdır. Çünkü bu yöntem tüm sıralama olasılıklarını değerlendirip en iyi sonuç vereni seçmektedir. Genetik Algoritma, belirlenen uygunluk fonksiyonuna göre bireyleri seçer. Çalışmamızda iki farklı uygunluk fonksiyonu test edilmektedir. Birinci uygunluk fonksiyonu, planlanan iniş süresi ile tahmin edilen iniş süresini değerlendirerek gecikmeleri minimize eden bireyleri(uçakları) seçmektedir. İkinci uygunluk fonksiyonu ise tüm sıralamada olan toplam bekleme süresini minimize etmeyi hedeflemektedir.

4.1 Genetik Algoritma

Genetik algoritma (GA), genetik ve doğal seleksiyon prensiplerine dayanan bir optimizasyon ve arama tekniğidir. GA birçok bireyden oluşan bir topluluğun belli seleksiyon kurallarına göre evrimleşerek ‘uygunluğu’ maksimize eden (yani maliyet fonksiyonunu en aza indiren) bir duruma ulaşmasına izin verir. Bu yöntem ilk olarak John Holland (1975) tarafından geliştirilmiş, 1960’lar ve 1970’ler boyunca gelişerek öğrencilerinden biri olan ve tezi için gaz boru hattıyla ilgili zor bir problemi bu şekilde çözen David Goldberg tarafından popüler hale getirilmiştir. Kitabında Holland’ın orijinal çalışması özetlenmektedir. Şema teoremi yoluyla, GA’lar için teorik bir temel geliştirmeyi ilk kez deneyen odur. De Jong’un (1975) çalışması fonksiyon optimizasyonu için GA’nın faydalı olduğunu göstermiş ve optimize GA parametrelerinin bulunması için ilk çabaları oluşturmuştur. Goldberg başarılı uygulamaları ve mükemmel kitabı sayesinde GA'nın kullanılması için çok çaba sarf etmiştir. O zamandan beri evrim programlamasının çeşitli versiyonları çeşitli başarı dereceleriyle denenmiştir. GA’nın bazı avantajları aşağıdaki gibidir:

- Sürekli veya ayrı değişkenlerle optimizasyon yapar.
- Türev bilgisi gerektirmez.
- Maliyet yüzeylerinden alınmış geniş örnekler üzerinde aynı anda arama yapar.
- Çok sayıda değişkeni ele alır.
- Paralel bilgisayarlara uygundur.
- Son derece karmaşık maliyet yüzeyleri olan değişkenleri optimize eder (bunlar yerel minimum dışına çıkabilirler).
- Tek bir çözüm değil, optimum değişkenler listesi sunar.
- Değişkenleri kodlayarak optimizasyonun kodlanmış değişkenlerle yapılmasını sağlar.
- Sayısal olarak oluşturulan verilerle, deneysel verilerle veya analitik verilerle çalışabilir.

Bu avantajlar ilgi çekicidir ve geleneksel optimizasyon tekniklerinin çok başarısız olduğu durumlarda şaşırtıcı derecede başarılı olabilirler.

4.2 Genetik Algoritma Uygulamaları

Genetik algoritmalar, NP-hard problemleri gibi zor problemlerin çözümünde, mekanik öğrenme ve basit programların geliştirilmesi için kullanılmıştır. Bunlar, ayrıca bazı sanat çalışmalarında resim ve müziğin geliştirilmesinde de kullanılmıştır (Sivanandam ve Deepa 2008). Birkaç GA uygulaması aşağıda verilmiştir:

- Nonlineer dinamik sistemler – kestirim, veri analizi.
- Robot yolu planlama.
- LISP programlarının geliştirilmesi (genetik programlama).
- Strateji planlama.
- Protein moleküllerinin biçimlerinin bulunması.
- TSP ve sıra planlama.
- Görüntü oluşturma fonksiyonları.
- Kontrol–gaz boru hattı, kutup dengeleme, füze kaçıışı, arama.
- Tasarım – yarı iletken yerleşim düzeni, hava taşıtı tasarımı, klavye biçimlendirme, iletişim şebekeleri.

- Planlama – imalat, tesis planlama, kaynak tahsisi.
- Makine Öğrenme – Sınır şebekelerinin mimari ve ağırlık olarak tasarlanması, sınıflandırma algoritmalarının iyileştirilmesi, sınıflandırıcı sistemler.
- Sinyal işleme – filtre tasarımı.
- Kombinasyon Optimizasyonu – set kapsamı, seyahat eden tüccar (TSP), Dizi planlama, yol atama, grafik renklendirme.

4.3 GA'nın Genel Tanımlaması

Algoritma problem çözümünde kullanılan bir dizi adımdır. Genetik algoritma, problem çözme modeli olarak genetiği kullanan bir yöntemdir. Optimizasyon ve arama problemlerinde, yaklaşık çözümlerin bulunmasında kullanılan bir arama tekniğidir. Temel olarak bir optimizasyon problemi gerçekten basit görünür. Spesifik bir soruya karşı gelen olası tüm çözümlerin formu bilinmektedir. Bu formu karşılayan tüm çözümlerin seti, arama uzayını oluşturur. Sorun, olası tüm çözümler arasında en iyi uyan çözümün, yani en fazla geri dönüşü sağlayan çözümün bulunmasıdır. Tüm çözümlere, hızla numara verilmesi mümkün olduğunda, sorun güçlük çıkarmaz. Ama arama alanı genişlediğinde numaralandırma çok zaman alacağından, uygulanabilir olmaktan çıkar. Burada optimal çözümün bulunabilmesi için spesifik bir tekniğin kullanılması gerekir. Genetik Algoritmalar, bu yöntemlerden biridir. Pratikte, basit genetiğin algoritmik mekanizmalara uygulandığı benzer şekillerde çalışır.

GA, olası çözümler popülasyonunu ele alır. Her çözüm soyut olan kromozomla temsil edilir. Tüm çözümler birinci bölümde bir kromozoma kodlanır. Genetik Algoritmada bir dizi reproduksiyon operatörlerinin de belirlenmesi gerekir. Reproduksiyon operatörleri, doğrudan kromozomlara uygulanır ve problemin çözümleri üzerinde mutasyonlar ve rekombinasyonlar oluşturmak için kullanılır. Uygun temsil ve reproduksiyon operatörleri gerçekten belirleyici olmaktadır, çünkü GA davranışı bunlara son derece bağlıdır. Arama alanının ve reproduksiyon operatörlerinin yapısına uyan, problemlerin özellikleriyle tutarlı ve ilgili olan bir metodun bulunması genellikle zor olmaktadır. Seçimin popülasyondaki her birey için karşılaştırma yapabilmesi beklenir. Seçim, uygunluk fonksiyonunu kullanarak gerçekleşir. Her

kromozomun temsil ettiği çözümün uygunluğuna karşı gelen bir değer vardır. Uygunluk, aday çözümün ne kadar iyi olduğunun değerlendirmesine karşı gelmelidir. Optimal çözüm, uygunluk fonksiyonunu maksimize eden çözümdür. Genetik Algoritmalar, uygunluk fonksiyonunu maksimize eden sorunlarla ilgilenir. Ama sorun bir maliyet fonksiyonunun en aza indirilmesinden oluşuyorsa, adaptasyon oldukça kolaydır. Ya maliyet fonksiyonu örneğin ters çevrilerek bir uygunluk fonksiyonuna dönüştürülür ya da seçim düşük değerlendirme fonksiyonlarının daha iyi kabul edildiği bir şekilde adapte edilir. Reprodüksiyon ve uygunluk fonksiyon uygun bir şekilde tanımlandıktan sonra aynı basit yapıya göre bir Genetik Algoritma geliştirilir. Başlangıç, kromozom popülasyonunun oluşturulmasıyla başlanır. Bu ilk popülasyon çok geniş bir çeşitlilikte genetik materyaller sunmalıdır. Gen havuzu her türlü çözümün arama alanına alınabileceği şekilde mümkün olduğu kadar büyük olmalıdır. Genellikle ilk popülasyon rastgele oluşturulur. Bundan sonra genetik algoritma popülasyonun evrimleşmesine izin vermek üzere bir yenileme sürecine girer(Sivanandam ve Deepa 2008). Her yenileme aşağıdaki adımlardan oluşur:

- SEÇİM: Birinci adım reprodüksiyon için bireylerin seçiminden oluşur. Bu seçim bireylerin bağıl uygunluklarına bağlı bir olasılıkla rastgele olarak yapılır, böylece sıklıkla zayıf bireyler değil, en iyi olanlar seçilir.
- REPRODÜKSİYON: İkinci adımda seçilmiş bireyler tarafından sonraki nesil yetiştirilir. Yeni kromozomlar yaratırken algoritma rekombinasyonu veya mutasyonu kullanabilir.
- DEĞERLENDİRME: Yeni kromozomların uygunluğu değerlendirilir.
- REPLASMAN: Son adımda önceki neslin bireyleri öldürülür ve yerlerine yenileri getirilir.

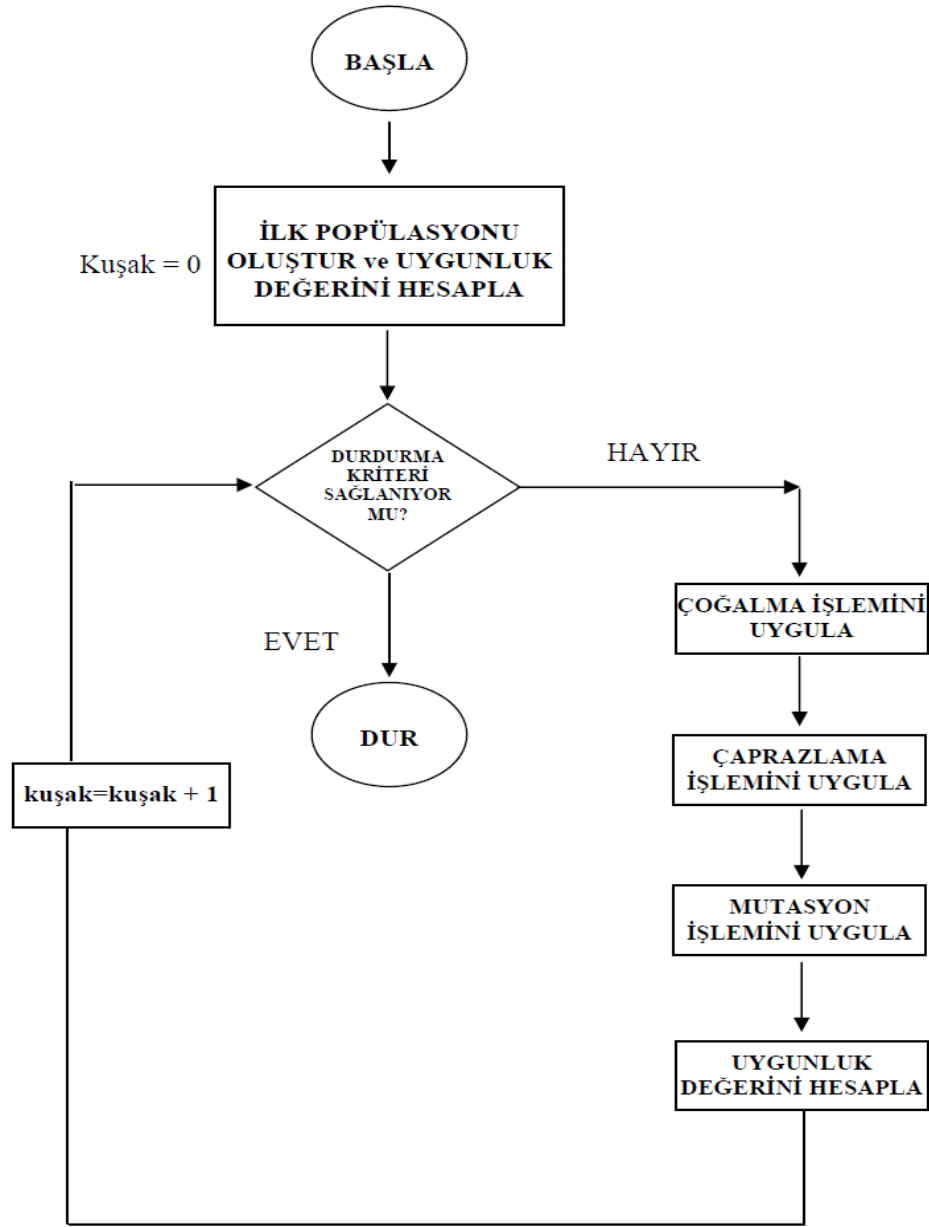
Popülasyon optimal çözüme yakınsama gösterdiğinde algoritma durdurulur. Temel genetik algoritma aşağıdaki gibidir:

- [başla] n sayıda kromozomun genetik rastgele popülasyonu (problem için uygun çözümler)
- [Uygunluk] Her kromozom x için popülasyondaki uygunluğun $f(x)$ değerlendirmesi
- [Yeni popülasyon] Yeni popülasyon tamamlanıncaya kadar aşağıdaki adımları tekrarlayarak yeni bir popülasyon yaratılır

- [seçim] popülasyon içinden uygunluklarına göre iki ana kromozom seçilir (uygunluk ne kadar fazlaysa seçilme şansı o kadar fazla olur).
- [çaprazlama] Çaprazlama olasılığında ebeveyn arasında çaprazlama yapılarak yeni nesil (çocuklar) oluşturulur. Çaprazlama yapılmazsa bunlar ebeveynlerin tam kopyaları olurlar.
- [Mutasyon] mutasyon olasılığında yeni nesilde her lokusta (kromozom pozisyonu) mutasyon yapılır
- [Kabul] Yeni nesil yeni popülasyona yerleştirilir.
- [Replasman] yeni bir algoritma toplamı elde etmek için yeni nesil kullanılır.
- [Test] Son şartlar tatmin ediciyse durulur ve mevcut popülasyondaki en iyi çözüme geri dönülür.
- [Döngü] Uygunluk değerlendirmesi için 2. Adıma geri dönülür.

4.4 Genetik Algoritma'nın Aşamaları

Bölüm 4.2'de GA operatörleri özet şekilde açıklanmaktadır, burada bu operatörlerin detaylarından bahsedilecektir. Şekil 4.1 GA genel akış diyagramını göstermektedir, bu işlemler her kromozom için kullanılmaktadır ve bir sonraki kuşağa daha iyi sonuçlar oluşturmayı amaçlamaktadır.



Şekil 4.1 Genetik Algoritmanın akış diyagramı

4.4.1 Popülasyonun oluşması

Popülasyon, bir bireyler topluluğudur. Popülasyon, test edilecek farklı kişilerden, bireyleri, tanımlayan fenotip parametrelerinden ve arama uzayı hakkındaki bazı bilgilerden oluşur. Genetik Algoritmalarda kullanılan iki önemli popülasyon konusu:

1. İlk popülasyon nesli.
2. Popülasyon boyu.

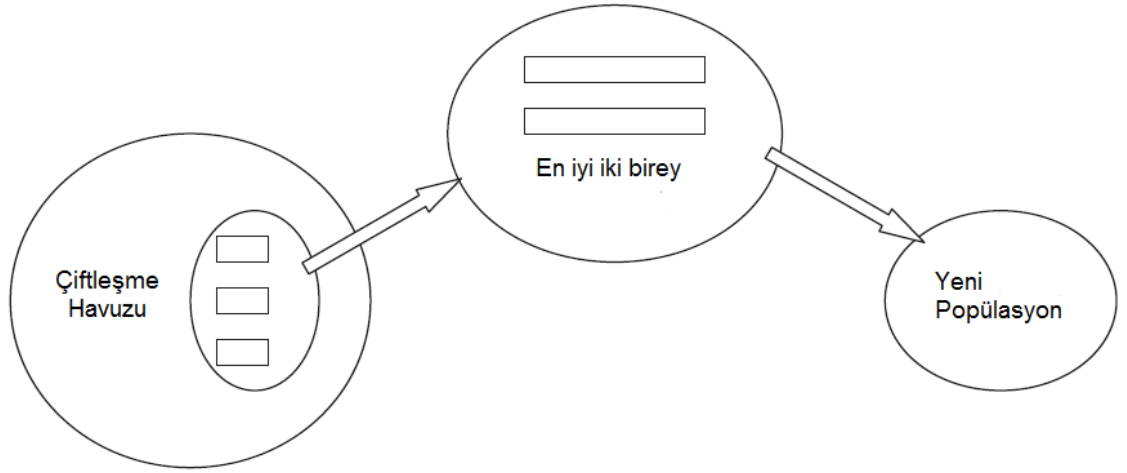
Popülasyon	Kromozom 1	1110010
	Kromozom 2	01111011
	Kromozom 3	10101010
	Kromozom 4	11001100

Şekil 4.2 Popülasyon

Şekil 4.2 popülasyon örneği gösterilmektedir. Tüm problemlerde popülasyon boyu, problemin karmaşıklığına bağlı olacaktır. Popülasyon, çoğu zaman rastgele olarak başlatılır. İkili kodlu kromozom durumunda bu her bitin sıfır veya bir olarak başlatılması anlamına gelir. Ama ilk popülasyonun bazı bilinen iyi çözümlerle başlatıldığı durumlar da vardır.

4.4.2 Seçim

Seçim, popülasyonun iki ebeveyni arasında, çaprazlama için seçim yapma işlemidir. Kodlamaya karar verdikten bir sonraki adım, seçimin nasıl yapılacağına karar vermektir, yani bir sonraki nesil için yavruları oluşturmak için bireylerin nasıl seçileceğine ve ne kadar yavru oluşturulacağına karar verilir. Seçimin amacı yavruların uygunluklarının daha fazla olacağı umuduyla popülasyondaki daha uygun bireylerin bulunmasıdır. Reprodüksiyon için, ebeveynler olmak üzere ilk popülasyondan kromozomlar seçilir. Sorun, bu kromozomların nasıl seçileceğidir. Darwin'in evrim teorisine göre en iyiler hayatta kalarak yavru yapar. Şekil 3.10'da temel seçim süreci gösterilmektedir. Seçim popülasyondan kromozomların değerlendirme fonksiyonlarına göre rastgele seçilmesi yöntemidir. Uygunluk fonksiyonu ne kadar yüksek olursa bireyin seçilme şansı o kadar artar.



Şekil 4.3 Seçim

Çeşitli seçim yöntemleri:

- RouletteWheel Seçimi
- Rastgele Seçim
- Derece Seçimi
- Turnuva Seçimi
- Boltzmann Seçimi

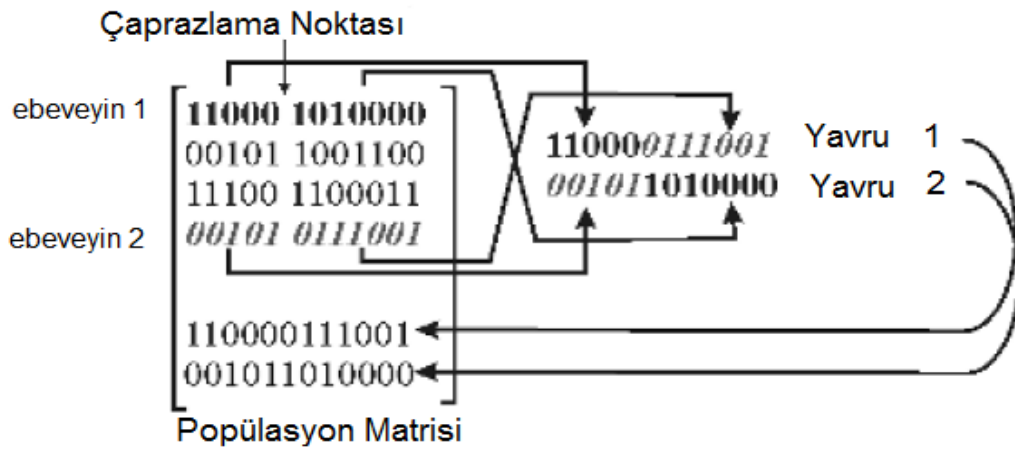
4.4.3 Çaprazlama (Rekombinasyon)

Çaprazlama ebeveyn çözümün alınması ve bunlardan bir çocuk üretilmesi işlemidir. Seçim (reproduksiyon) işleminden sonra popülasyon daha iyi bireylerle zenginleşir. Reproduksiyon iyi sıraları olan klonlar oluşturur ama yenilerini yaratmaz. Çaprazlama operatörü daha iyi yavrular elde etme umuduyla eşleştirme havuzuna uygulanır. Çaprazlama üç adımda ilerleyen bir rekombinasyon operatörüdür:

- i. Reproduksiyon operatörü iki bireysel diziyi eşleştirme için rastgele seçer.
- ii. Dizi boyunca rastgele bir çaprazlama yeri seçilir.
- iii. Son olarak çaprazlama yerini izleyen iki dizi arasında pozisyon değerleri değiştirilir.

Bunu yapmanın en basit yolu rastgele bir çaprazlama yeri seçmek ve birinci ebeveynde bu noktadan önceki her şeyi kopyalamak ve sonra diğer ebeveynde çaprazlama noktasından sonraki her şeyi kopyalamaktır. Çeşitli çaprazlama teknikleri:

- Tek Nokta Çaprazlama
- İki Nokta Çaprazlama
- Çok Nokta Çaprazlama (N-Nokta Çaprazlama)
- Üniform Çaprazlama
- Üç Ebeveyn Çaprazlama
- Azaltılmış Taşıyıcılı Çaprazlama
- Karıştırarak Çaprazlama
- Öncelik Koruyucu Çaprazlama (PPX)
- Düzenli Çaprazlama
- Kısmi Eşleştirmeli Çaprazlama (PMX)



Şekil 4.4 Çaprazlama işlemi

4.4.4 Mutasyon

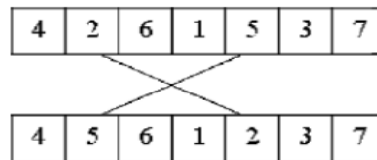
Çaprazlamadan sonra diziler mutasyona tabi tutulur. Çaprazlama neticesinde farklı çözümlere ulaşmak bazen zor olmaktadır. Mutasyon, yeni çözümleri aramanın kolaylaştırılması ve aramanın yönünü değiştirmek amacı ile bir kromozomun bir

elemanının (gen) deęiřtirilmesi iřlemidir. Bir problemin havuzu iinden ka kromozomun mutasyona uęratılacaęına mutasyon oranına gre karar verilmektedir. Kromozomlarda tesadüfi olarak belirlenen bir noktada yapılan deęiřikliklerdir. Mutasyon kaybolan genlerin geri kazanılması yanında genetik bilgilerin rastgele olarak karıřtırılmasına da yarar. Genetik Algoritmalar sadece seim ve aprazlama operatrlerinden meydana gelselerdi ok hızlı bir Őekilde yerel bir optimuma ulařabilirlerdi. Mutasyon geleneksel olarak basit bir arama operatr olarak kabul edilir. aprazlama daha iyi bireylerin bulunması iin mevcut durumun kullanılması ise, mutasyon da tm arama alanının arařtırılmasına yardım eder. Genetik algoritmalarda eřitlendirmeye gitmede mutasyondan faydalanılır. Mutasyonda genlerden biri rassal olarak deęiřtirilir. Gen havuzundaki stokları iyi durumda tutar ve bylece dinamiklięi saęlar. Herhangi bir poplasyon ařamasında bir özm oluřturma olasılıęının sıfırdan farklı olması durumunda bir arama uzayının ergodik olduęu sylenir (Haupt 2004).

Mutasyon teknięindeki nemli parametre mutasyon olasılıęıdır (Pm). Mutasyon olasılıęı kromozom paralarının ne sıklıkta mutasyona uęrayacaęını belirler. Mutasyon yoksa yavrular hibir deęiřiklik olmadan aprazlanamadan hemen sonra ortaya ıkarılır (veya doęrudan kopyalanırlar). Mutasyon yapılırsa kromozomun bir veya daha fazla parası deęiřir. Mutasyon olasılıęı %100 ise tm kromozom deęiřir, %0 ise hi deęiřmez. Mutasyon genellikle GA'nın ařırı lokal ulara kaymasını nler. Mutasyon ok sık olmamalıdır, nk bu durumda GA rastgele aramaya dnřr.

Farklı temsil trleri iin farklı mutasyon formları vardır:

- Dndrme
- Karřılıklı deęiřim
- Ters dnme



4.5 Mutasyon iřlemi

4.4.5 Yeni kuşakın oluşması ve döngünün durdurulması

Yeni kuşak çoğalma, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki kuşakın ebeveynleri olmaktadır. Süreç yeni kuşakla çoğalma için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder. İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir (Emel ve Taşkın 2002).

- Rastgele Değiştirme
- Zayıf Ebeveynin Çıkarılması
- Her İki Ebeveyn

4.4.6 Aramanın bitirilmesi

Kısaca, aramanın durdurulması için çeşitli şartlar aşağıda verilmiştir:

- Maksimum nesiller– Belli sayıda nesil ortaya çıktığında genetik algoritma durur.
- Geçen Süre –Belirlenen süre geçtiğinde genetik süreç durur.
- Uygunlukta değişiklik olmaması–Belli sayıda nesilde popülasyonun en iyi uygunluğunda hiç değişiklik olmazsa genetik süreç durur.
- Oyalanan nesiller – Bir dizi ardışık nesilde objektif fonksiyonda hiç iyileşme olmazsa algoritma durur.
- Oyalanma zamanı limiti –Saniye olarak Oyalanma zamanı limitine eşit bit zaman aralığında objektif fonksiyonda hiç iyileşme olmazsa algoritma durur. Kriterin sonlanması veya birleşme noktasının gösterilmesi aramayı durdurur. Aşağıda bazı sonlandırma teknikleri verilmiştir.

5. HTK'ünde SDP PROBLEMİ

Variş sıralarının düzenlenmesi ve programlanması (SDP), Hava Trafik Kontrolünün (HTK) standart sorunlarından biridir. Basitçe ifade edilirse, SDP havaalanındaki varışlar için etkin iniş sıraları ve zamanları oluşturma, böylece gelen hava taşıtları arasında güvenli bir aralığın garanti edilmesi, havaalanının kullanılabilir kapasitesinin verimli kullanılmasının ve havadaki gecikmelerin etkin bir şekilde azalmasının sağlanması fonksiyonudur (Pelegri, 1994). SDP işlemlerinde basit bir yol gelen hava taşıtları için, tahmin edilen piste iniş zamanına (TVS) dayanan bir sıraya göre “önce gelene önce hizmet” (İĞİH) prensibinin uygulanmasıdır. İĞİH programlaması, tahmin edilen iniş süresine göre adil bir sıra oluştursa da, havaalanı kapasitesinin verimli kullanımı, havadaki gecikmelerin azaltılması ve/veya havayollarına verilen hizmetin iyileştirilmesi için kullanılacak başka faydalı bilgileri ihmal eder. Örneğin Gregory vd. (1999) ve Gregory vd. (2000) tarafından SDP sorununa “gecikme değişimi” getirilmiş, gelen uçuşlar arasında bireysel hava yolu öncelikleri hesaba katılmış ve buna göre “öncelik programlaması” tanımlanmıştır (Hu ve Chen 2005).

SDP uygulamalarında en yaygın kabul gören kavram iki gerçeğe dayanan “pozisyon kaydırmadır” Birincisi güvenlik düzenlemeleri aynı yükseklikteki iki hava taşıtlarının aralarında “minimum yatay mesafeyi” korumalarını gerektirir, bu ise iki hava taşıtlarının tiplerinin ve bağlı pozisyonlarının bir fonksiyonudur; İkincisi, bir hava taşıtı tipi için “iniş hızı” genellikle bir başka hava taşıtı tipi için olandan farklıdır. Yukarıdaki parametrelerin değişkenliklerinin bir sonucu olarak birbiri ardına iki iniş için izin verilen minimum zaman aralığı olan iniş zaman aralığı (LTI) değişken bir değerdir.

5.1 Hava Trafiği Kontrol Kısıtlamaları

Hava trafik kontrolü dört görev düzeyinde çalışır:

(a) Hava sahası tasarımı, havayolları ve kontrol sektörleriyle ilgilidir. Bir havaalanından diğerine giden bir hava taşıtı, güzergâhı ve havaalanı işaret ışıklarını izlemelidir. Gökyüzünde aynı anda pek çok hava taşıtı bulunacağından tek bir kontrolör

bunların hepsini yönetemeyecektir. Sonuç olarak, hava sahası farklı sektörler'e bölünür ve her biri bir kontrolöre atanır. Bu iş hava şebekesinin ve ilgili sektörlerin tasarımını hedefler.

(b) Hava trafiği akış yönetimi birkaç saat öncesinden stratejik planlamaya odaklanır. Artan trafikle birlikte birçok hava yolu aynı güzergâhlara hizmet vermeye başlar, bu da potansiyel olarak sektörde aşırı yüke neden olur. Hava trafiği akış yönetimi sektörlerde tıkanma ve çakışmaları azaltmak üzere hava taşıtı güzergâhlarını ayarlamayı hedefler.

(c) Koordinasyon planlamasının birkaç 10 dakika öncesinden programlama sorumluluğu vardır. Bu görev sektörler'e giren yeni hava taşıtlarının aşırı yük yaratmamasını sağlamaya çalışır.

(d) Hava trafiği kontrol merkezlerinde klasik kontrol en fazla 20 dakikalık bir zaman dilmi içinde süreyle ilgilenir. Bu düzeyde TMA'lar yaklaşan hava taşıtları arasındaki potansiyel çakışmaları çözer.

Çizelge 5.1 İnişler için minimum zaman aralığı

S_{ij} (s)	Takip eden uçakların kategorisi : j				
	1	2	3	4	
Önde gelen uçakların : i kategorisi	1	96	200	181	228
	2	72	80	70	110
	3	72	100	70	130
	4	72	80	70	90

(1) B747; (2) B727; (3) B707; (4) DC9.

Basitlik amacıyla, Bianco vd. (1997)'nin yaptığı gibi, iniş için bekleyen hava taşıtları hız, kapasite, ağırlık ve başka teknik karakteristiklere göre nispeten küçük sayıda kategorilere ayrılır. Çizelge 5.1'de ana ticari hava taşıtı kategorilerine göre minimum LTI'ler gösterilmektedir (Hu ve Chen 2005). Özellikle 4 kategori ele alınır: 1. Kategori Boeing 747'yi tanımlar (B747), 2. Kategori Boeing 727'yekarşı gelir (B727), 3. Kategori Boeing 707yi tanımlar (B707) ve son olarak 4. Kategori Mc Donnell Douglas DC9'a karşı gelir (DC9). Çizelge 5.1'deki LTI'lerin asimetrik olduğu açıkça görülmektedir. Örneğin, B727'nin B747'yi izlemesi için minimum 200 s'lik LTI

gerekirken, aynı taşıtların ters sırada birbirlerini izlemeleri için yalnızca 72 s'lik minimum LTI gerekmektedir. LTI'lerin asimetrisinden yararlanarak, bir başka deyimle FCSC iniş sırasında hava taşıtlarının pozisyonlarını değiştirerek gecikmeleri azaltmak ve havaalanı kapasitesini iyileştirmek mümkündür.

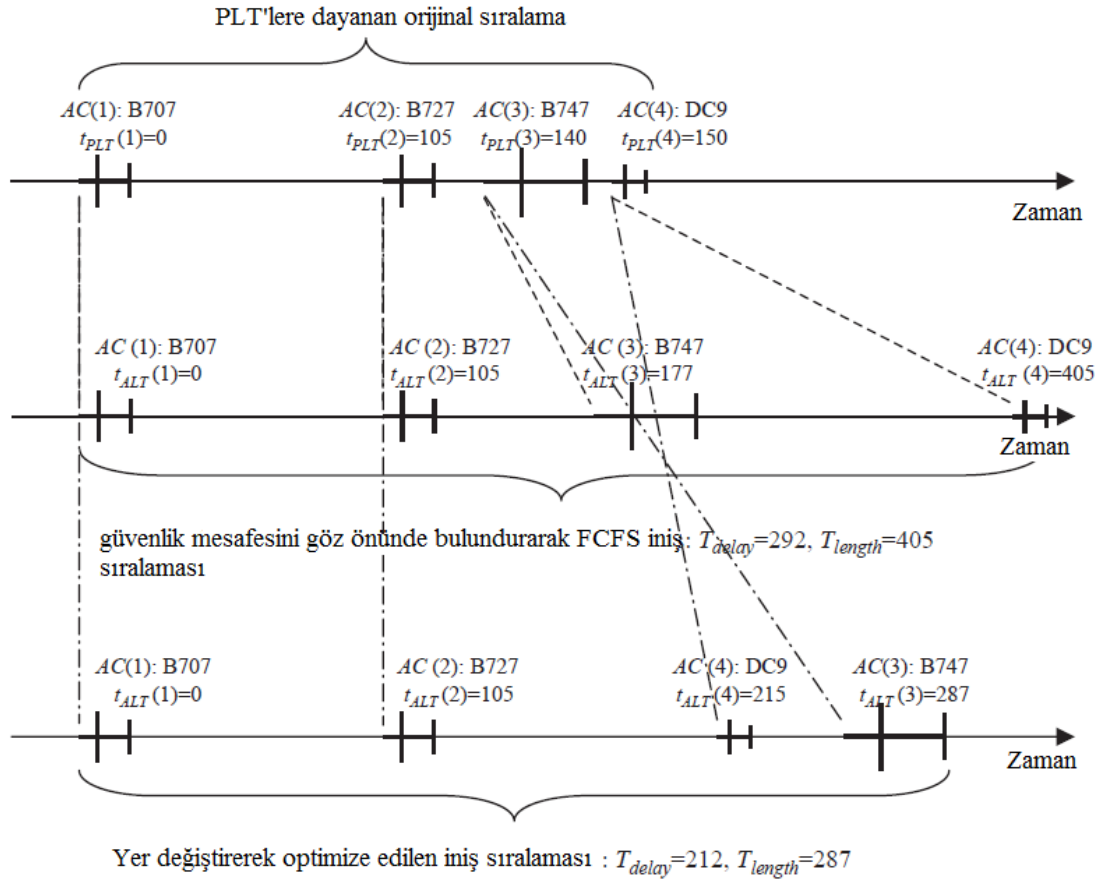
Geçen 10 yılda “pozisyon kaydırma”-SDP sorununu incelemek için çok çaba harcanmıştır, örneğin, SDP sorunu Seyahat Eden Satıcı Problemi (TSP) olarak modellenmiş ve dinamik programlama algoritmaları geliştirilmiştir, örneğin Psaraftis (1978), Psaraftis (1980), Bianco vd. (1997) ve Bianco vd. (1988). Temelde etkin statik algoritmalar geliştirilmesine odaklanmaktadır. Dinamik durum Psaraftis (1978), Psaraftis (1980) ve Bianco vd. (1997) ele alınmış olsa da, önerilen dinamik algoritmalar temelde ilgili statik sonuçların bazı kısıtlamalar getirilmesiyle yapılan basit uzantılarından ibarettir.

5.2 Genetik Algoritma ile SDP Yaklaşımı

Gerçek HTK dünyasında SDP daima dinamik bir ortamda yer alır. Bu yüzden SDP probleminin dinamik doğasına doğrudan dayanan modelleme ve geliştirme algoritmaları avantajlı olabilir. İyi bilindiği gibi, GA büyük ölçekli bir paralel stokastik arama ve optimizasyon algoritmasıdır ve SDP problemi gibi NP-tam problemlerinin çözümünde etkindir. GA'nın dinamik SDP problemi durumuna uygulanması için geleneksel yol, çalışma gününün geri kalanı, varış akışının tamamı için GA'nın tekrar tekrar uygulanmasıdır. Kalabalık bir havaalanına günlük varış sayısı çok yüksek olduğundan, geleneksel olarak kullanılan GA pratikte gerçek zamanlı özelliklerin kullanılması ihtiyacını pek karşılamaz. Diğer taraftan dinamik HTK ortamında kestirilen varış akışında kaçınılmaz olarak güvenilir bilgiler olacaktır. Örneğin, bazı uçuşlar iptal edilirken bazı beklenmeyen hava taşıtları acil iniş isteyebilir. Tüm uçakların geleneksel şekilde hesaba katılması optimal veya hatta suboptimal çözümler bile sağlamayabilir.

5.3 SDP Probleminin Formülasyonu

İşletme günü içinde havaalanının aynı pistine, bir dizi uçağın inmesi beklenir. İlgili hava taşıtı sayısının N_{AC} olduğunu ve işletim gününün T_{range} -dakika uzunluğunda olduğunu kabul edelim. N_{AC} ve T_{range} havaalanında olan pistin yoğunluk derecesini elde etmek için kullanılabilir. Orijinal sıralamada, $i = 1, \dots, N_{AC}$; i 'nci uçak için $AC(i)$, piste bir tahmini iniş süresi(PLT) vardır, buna $t_{PLT(i)}$ denir. Bu PLT'leri göz önünde bulundurularak ve gereken güvenlik mesafesini de katarak İĞİH sıralaması elde edilir. Elde edilen İĞİH iniş sıralamasında uçakların 'yer değiştirme' fonksiyonunu ve minimum uzaklık değerindeki LTI asimetrik özelliklerden yararlanarak havaalanındaki gecikmeler azaltılabilir ve mevcut olan kapasite daha etkili bir şekilde kullanılır. Şekil 5.1'de 'yer değiştirme' işleminden sonra gecikmede olan olumlu fark açıkça görülmektedir.



Şekil 5.1 İĞİH iniş sıralaması ve yer değiştirme optimizasyonu

GA verilen orjinal sıralamayı esas alarak, rastgele bir takım sıralama üretir. Üretilen sıralamalar, uçakların yerlerini değiştirerek tüm olasılıkları kapsamaktadır. Daha sonra bu olasılıkların uygunluğu ölçülecektir ve iyi olan sıralamalar seçilerek, daha da iyi sonuçlar üretmek için GA işlemlerinden geçmektedir.

5.4 GA'nın Kromozom Yapısı

GA stratejisinde genelde mevcut varış akışı içindeki uçakların tamamı optimizasyon sürecine dâhil edilmez. Ama kromozomdaki genler ve (.|k) ilgili değişkenin k sırasındaki zaman aralığında kestirildiğini veya hesaplandığını gösterir. Bir kromozomun elemanlarından her birisi çözümün bir özelliğini göstermektedir. Bunlara da “Gen” adı verilmektedir Kromozoma ve çizelge 5.1’de verilen listeye göre iniş sırasına dayanarak olası iniş sırasında her uçak için PVS aşağıdaki İĞİH prensibine göre hesaplanır if $k = 0$, $t_{PVS}(a1|k) = t_{TVS}(a1|k)$ aksi takdirde $t_{PVS}(a1|k),(k-1)$ zaman aralığında $AC(a1|k)$ ile pisti en son terk eden uçak arasındaki LTI’nin yerine gelmesini sağlamalıdır, Şekil 5.2 kromozom yapısını göstermektedir. PVS’lerle olası iniş sırasının hava gecikmeleri aşağıdaki adımlara göre hesaplanır:

1. Dizide bulunan her uçağın kategorisini (büyüklüğünü ifade etmek) belirlemek.
2. Bu uçakların arasındaki minimum uzaklık değerini belirlemek LTI
3. Her uçak için planlanan iniş zamanını TVS değerini belirlemek.
4. LTI kullanılarak uçağa verilen yeni iniş zamanını belirlemek PVS.

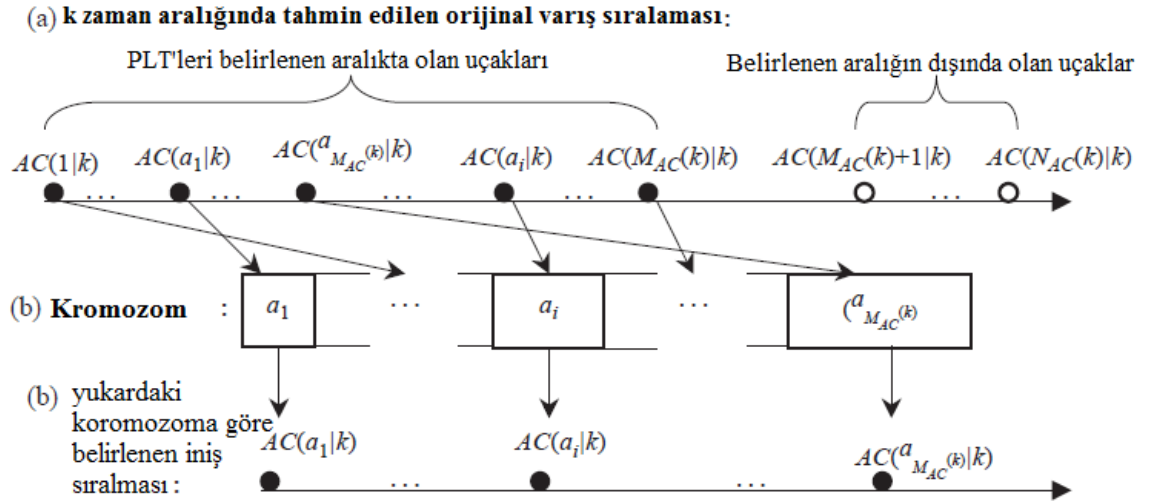
$$t_{PVS}(i) = \sum t_{PVS}(i-1) + LTI(ij) \quad (5.1)$$

5. Her uçak için TVS ve PVS değerlerini kullanarak COST fonksiyonu kullanarak T delayi yani gecikme süresi bulunur(dizideki tüm uçaklar için) :

$$T_{delay} = \sum_{i=1}^{N_{AC}} (t_{PVS}(i) - t_{TVS}(i)), \quad (5.2)$$

5. Her dizi için 1 tane (COST) gecikme süresi elde edilir,son olarak optimal (en az COST’a sahip) dizi belirlenir.

6. Bu işlemler sisteme gelen yeni bilgileri göz önünde bulundurmak için belirli bir süre içerisinde sürekli güncellenmek zorundadır.



Şekil 5.2 Kromozumun yapısı

5.5 HTK'nın GA İle Uygulaması

Hava trafik kontrolünde GA'nın etkinliğini göstermek için Matlab simülasyon ve programlama dilinde bir yazılım hazırlanmıştır. Bu yazılım, havaalanına iniş yapan uçakların sıralama problemini SDP gecikmeleri en aza indirecek şekilde ayarlamaktadır. Uygulamanın detaylı, nasıl çalıştığını öğrenmek için aşağıda geçtiği aşamalar anlatılmaktadır:

1. Gelen Uçaklar, TVS'lere göre dizilir ve bu dizi GA'nın girişi olacaktır.
2. Daha sonra GA, 1.aşamadan gelen verileri ilk popülasyonu oluşturmak için kullanacaktır. Popülasyon, İGİH uçak dizisindeki pozisyonlarda değişiklik yaparak farklı dizilerden oluşur. Popülasyondaki her kromozom bir farklı sıralamayı temsil etmektedir. Popülasyonun boyutu ilk olarak 20 kez kullanıldı, sonra deneyerek 30 kez daha uygun olarak belirlendi. Şekil 5.1 oluşturulan popülasyonun bir örneğini göstermektedir.

Kromozom1			1	2	3	4
Kromozom2	2	1	3	4		
Kromozom3	2	3	1	4		
.		2	3	4	1	
Pop(30) =				
.					
Kromozom30					

Şekil 5.3 Oluşturulan popülasyon örneği

3. Popülasyon oluşturulduktan sonra, içinde mevcut olan her bir kromozomun (sıralamanın) aşağıdaki uygunluk fonksiyonuna tabi tutlur:

$$T_{\text{delay}} = \sum_{i=1}^{N_{AC}} (t_{\text{PVS}}(i) - t_{\text{TVS}}(i)), \quad (5.3)$$

Uygunluk fonksiyonu her bir kromozom için bir değer belirlemektedir, bu değer o sıralamada olan toplam gecikmeyi saniye birimi ile ifade etmektedir. Çıkan sonuçları daha da iyileştirmek için GA çaprazlama ve mutasyon işlemlerini uygular fakat bunları uygulamadan önce yeni doğacak olan yavrulara yer ayrılmalıdır. Bu nedenle, popüasyondan çıkan kötü sonuçlar elenir böylece yeni yavrular için yer açılır. Elenecek olan koromozomların sayılarını belirlemek için aşağıdaki fonksiyon uygulanır:

$$N_{\text{keep}} = X_{\text{rate}} N_{\text{pop}} \quad (5.4)$$

Burada N_{keep} çiftleşmeye kalacak olan koromozomların sayısı, X_{rate} , bırakmak istediğimiz kromozomların yüzdesini ifade etmektedir. Çok fazla kromozomun kalması, bir sonraki kuşağa kötü performansa neden olabilir. N_{pop} ise popülasoynumuzun sayısıdır.

4. Bir önceki adımdan kalan kromozomları çiftleştirmeden önce ebeveynleri seçiyoruz. Bu işlemi gerçekleştirmek için aşağıdaki fonksiyonu kullanılacaktır:

$$\begin{aligned} ma &= \text{ceil}(N_{keep} * \text{rand}(1, N_{keep})) \\ pa &= \text{ceil}(N_{keep} * \text{rand}(1, N_{keep})) \end{aligned} \quad (5.5)$$

Bu fonksiyon ebeveynleri rastgele seçecektir, ceil çıkan sonucu en yakın tam sayıya yaklaştırır (Haupt 2005).

5. Seçilen ebeveynler bu aşamada çaprazlama işlemine tabi tutlur. Daha önce belirttiğimiz gibi birçok çaprazlama yöntemi vardır bunlardan en yaygın olarak kullanılan yöntem, rastsal tek noktalı yöntemdir. Bu yöntemde, koromozomu ikiye bölecek şekilde bir rastsal sayı seçilir ve o noktadan sonraki bireyler iki kromozom arasında yer değiştirir. Fakat bu yöntem SDP probleminde sorun yaratmaktadır, bunun nedeni aşağıdaki örnekte açıklanmaktadır.

$$\begin{array}{l} \text{Çaprazlama noktası} \\ \uparrow \\ ma = \text{kromozom2} = [1 \ 2 \ 4 \ 3] \\ pa = \text{kromozom5} = [4 \ 2 \ 1 \ 3] \\ \text{çıkan sonuç : yavru1} = [1 \ 2 \ 1 \ 3] \\ \text{yavru2} = [4 \ 2 \ 4 \ 3] \end{array}$$

üretilen yavrulardaki sorun, aynı yavruda iki uçağın tekrar edilmesidir. Bu sorunu gidermek için yine birçok yöntem önerilmektedir. Bizim kullandığımız yöntem Döngü Çiftleşme yöntemidir. Bu yöntemde tekrarlanan genleri teker teker yavrudaki pozisyonu ile değiştirerek oluşan tekrar ortadan kalkmış olmaktadır.

6. Mutasyon: Mutasyon işlemi çaprazlama işlemi ile aynı mantığa sahip fakat mutasyon işlemi tek kromozom üzerinde gerçekleşir. Mutasyon işlemi iki kromozom arasındaki birkaç gen değişimi yapmaktadır. Kaç tane genin yerini değiştirmek için aşağıdaki hesaplama yapılır :

$$\# \text{mutations} = \mu \times (N_{pop} - 1) \times N_{bits} \quad (5.6)$$

Burada μ değeri kullanıcı tarafında verilen bir değerdir, bu değer poplasyonun yüzde kaçına mutasyon işlemini uygulamak istediğimiz açıklamaktadır. Örnek olarak %20 olursa $\mu = 0.2$ olur, %40 olursa $\mu = 0.4$. N_{pop} değeri ise popülasyonun boyutudur, burada 20 olarak belirlenmiştir. N_{bits} ise uçak sayısını temsil etmektedir o da 4'tür. Çıkan sonuç tüm popülasyondaki mutasyon sayısını temsil eder. Bu sayı belirlendikten sonra işlem uygulanmaya başlar. Çıkan sayıya göre rastsal çiftler seçilir ve yerleri değiştirilir. Fakat bu yöntem aynı çaprazlama yönteminde olduğu gibi mutasyon işleminde de tekrar çıkma olasılığı yüksektir. Bu durumu çözmek için kullanılan basit yöntemlerden biri, aynı kromozom içinde iki rastsal değeri seçip yerlerini değiştirmek. Buna örnek olarak aşağıdaki mutasyon işlemi gösterilmektedir.

$$\begin{aligned} \text{chromosome} &= [6 \mathbf{1} 5 3 \mathbf{2} 4] \\ &\Downarrow \\ \text{chromosome} &= [6 \mathbf{2} 5 3 \mathbf{1} 4] \end{aligned}$$

Bu örnekte rastsal olarak 2 ile 5 pozisyonları seçilerek gen değişimi yapılmaktadır. Bu işlemin sonunda her kromozom için uygunluk fonksiyonu tekrar yeniden hesaplanır ve bir sonraki generasyonda aynı işlemler devam eder. İstenilen generasyon sayısına ulaşıncaya işlemler biter ve en az cost 'a sahip kromozom seçilir.

5.6 GA İle Yeni SDP Yaklaşımı

GA ile SDP probleminin optimizasyonu daha önce de bahsedildiği gibi 2 farklı yaklaşımla test edilmiştir. 5.4 ve 5.5 bölümlerinde gerçekleştirdiğimiz SDP optimizasyonu, TVS ve PVS değerlerini kullanarak, her uçakta olan gecikme sürelerini hesaplamaktaydı ve bu hesaplama sonucunda elde edilen değeri, uygun uçağı bulmak için kullanmaktaydı. Kıscası GA'nın kullandığı uygunluk fonksiyonu gecikmeleri azaltmayı hedefliyordu ve bunu LTI sürelerindeki alternatif seçenekleri kullanarak gerçekleştiriyordu. 2.Yaklaşım yine aynı LTI prensibini kullanarak farklı uygunluk

fonksiyonu geliştirilmiştir. Yeni uygunluk fonksiyonu, belirli bir süre içerisinde talep edilen uçak sıralamalarının aralarındaki toplam LTI süresini azaltmayı amaçlamaktadır. Kromozom yapısı ve diğer GA işlemleri 5.4 bölümündeki gibi uygulanmaktadır. Uygunluk fonksiyonu ise aşağıdaki gibi olacaktır:

$$\text{Toplam bekleme süresi} = \sum_m^i [LTI(AC(i), AC(i + 1)) \dots LTI(AC(i + 1), AC(m))] \quad (5.7)$$

Yukarda denklemden AC(i), sıralamadaki ilk uçağı, m ise son uçağı, LTI ise iki uçak arasındaki uzaklığı temsil etmektedir. GA'nın uygulanmasında çıkacak en büyük sorunlardan biri uygulama süresidir. Çok geniş bir arama özelliğine sahip olması ve tüm olasılıkları teker teker değerlendirmesi nedenlerinden dolayı GA işlemleri optimal sonuca varıncaya dek uzun süreler almaktadır. Yukardaki uygunluk fonksiyonu sözkonusu probleme, performansı etkilemeden çözüm getirmek için geliştirilmiştir. Bir önceki uygunluk fonksiyonunda var olan karmaşık hesaplamaları sadeleştirerek, optimizasyon için gereken süreyi düşürmeyi hedeflemektedir. Son bölümde yapılan test ve simülasyonlarda iki fonksiyon arasındaki fark gerek performans açısından gerekse uygulama süresi açısından gösterilmiştir.

6. DOYUMSUZLUK ALGORİTMASI (DA)

DA yöntemi, çizelgeleme üzerinde en iyi çözümü bulmak için dolaşma yapılırken bir sonraki düğümü belirlemek için kullanılan bir yöntemdir. “O anki durumlar içinde en iyi olanını seç” kriteri yerel değerlendirmeye göre yapılmakta olup, bunun tüm sistem için en iyi seçim olacağı düşünülerek karar verilir.

DA, çalışırken algoritmanın her adımında birden çok alternatif çıkabilir. Algoritma, her adımda bu alternatiflerden en iyi olanını seçer. Daima en iyiyi seçerek ilerlemek, sonuçta optimum çözümün elde edilmesini sağlayabilir. Ama bu her zaman için geçerli olmayabilir. Esnek akış tipi problemleri çözmek için kullanılan Doyumsuzluk Algoritma paralel algoritmadır. Yıkım ve yapım evreleri olmak üzere iki aşamada uygulanır. Yıkım aşamasında, bazı çözüm bileşenleri önceden yapılandırılan aday çözümden çıkarılır. Sonra yapı prosedürü eksiksiz bir aday çözümü tekrar yapılandırmak için Doyumsuzluk yapısal heuristik uygulamaktadır. Aday çözüm tamamlandığında kabul kriteri yeniden yapılandırılan çözümün şimdiki çözümü üretip üretmeyeceğine karar vermektedir.

6.1 DA'nın Özellikleri

Genel olarak DA 5 sütundan oluşmaktadır:

- 1- Aday seti, üretilen çözüm hangisindedir.
- 2- Seçim fonksiyonu, en iyi adayı seçip çözüme ilave etmektedir.
- 3- Uygulanabilirlik fonksiyonu, adayın çözüme katkı sağlayacağını belirlemek için kullanılır.
- 4- Amaç fonksiyonu, çözüme veya kısmi çözüme değer katmaktadır.
- 5- Çözüm fonksiyonu, kesin çözüm keşfedildiği zaman onu bildirir.

DA, bazı matematiksel problemler için en iyi çözümler üretebilir fakat bazıları için de üretemez. Çalıştığı birçok problem için iki özelliğe sahip:

Doyumsuzluk seçme özelliği: şimdiki durum için en iyi görünen çözüm seçilir sonrasında ise ortaya çıkan alt problemler çözülür. DA tarafından gerçekleştirilen yeni

seçimler, gelecekteki seçimler veya alt problemin tüm çözümlerini değil de daha önce seçimleri dikkate alınarak yapılır. Verilen her problemi küçük problemlere dönüştürene kadar tekrarlamalı olarak seçimleri ard arda gerçekleştirir. Başka bir deyişle, DA hiç bir zaman yaptığı seçimleri tekrar gözden geçirmez. Bu özellik Dinamik Programlama ile DA arasında olan en belirgin farktır ve bu fark DP'yi çözümün bulunmasında daha kapsamlı kılmaktadır.

Optimal yapı: şayet problemin optimal çözümü alt problemler için optimal çözümler içeriyorsa, problem optimal bir tavır sergiler.

6.2 DA'nın Türleri

DA, 'kısa görüşlü' ve 'geri dönüşü olmayan' algoritmalar kategorisine dahil olmaktadır. Sadece optimal yapıya sahip problemler için idealdir. Buna rağmen DA, arama seçeneklerinde olan öncelik, branş veya algoritma sınırlarını belirlemek için kullanılır. DA birkaç varyasyonları vardır:

- Kuramsal DA
- Ortogonal DA
- Relaks DA

6.3 DA'nın Uygulama ve Örnekleri

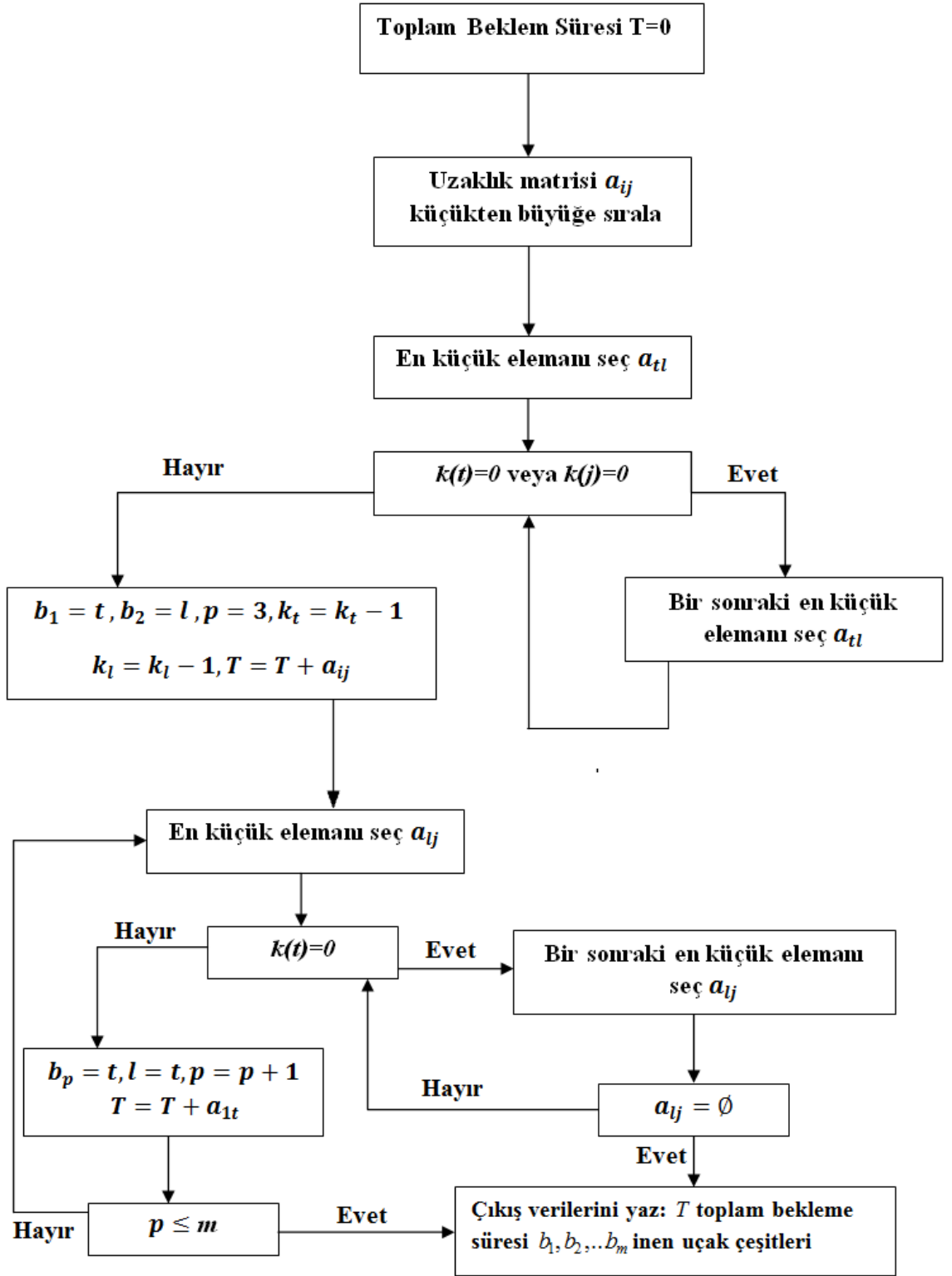
DA genelde (ama her zaman değil) optimal çözümü bulmakta başarısız olur çünkü tüm veriler üzerinde en ince ayrıntısına kadar çalışmaz. Belirli seçimleri çok erkenden belirler bu da sonradan çıkabilecek daha iyi seçimleri görmesine engel olmaktadır. Şayet DA verilen bir problem için optimal üretim alanını genişletebilirse doğal olarak en iyi seçim metodu olarak kullanılabilir çünkü diğer optimizasyon yöntemlerinden(örneğin DP'den) daha hızlıdır. Bu gibi DA örnekleri Kruskal algoritması, Prim algoritması, Dijkstra algoritması ve Huffman algoritması.

Ayrıca DA aktivite seçim probleminde kullanılmaktadır, bu problemde amaç, birbiriyle çakışmayan maksimum aktivite sayısını seçmektir. Bir diğer uygulama ise, Macintosh

Computer tarafından geliştirilen Crystal Quest oyunudur. Bu oyun, gezgin satıcı yaklaşımına benzer biçimde kristal toplamayı amaçlamaktadır. Buna ilaveten, DA Malfatti problemi için en optimal çözümü bulabilmektedir, bu problem bir üçgen içinde toplam çember alanını maksimuma çıkaran birbirinden ayrı üç çemberi bulmaya çalışmaktadır.

6.4 DA ile SDP Problemi

SDP problemini çözmek için farklı yaklaşımlar vardır. Daha önceki bölümlerde GA'nın bu problem üzerinde olan çalışma şeklini göstermiştik. GA'nın bu konudaki performansını daha açık ve net bir şekilde görebilmek için başka bir yöntemle karşılaştırmak gerekmektedir. DA yöntemi, sıralama problemleri için oldukça iyi sonuçlar üretmektedir. DA kullanarak, 5.6.bölümde kullanılan GA şekli ile bir SDP algoritması yazılmıştır. Bu algoritma (5.7)'den elde edilen toplam bekleme süresini (TBS) başka bir metotla minimize etmeye çalışmaktadır. Şekil 6.1 DA'nın SDP problemi için işlem adımlarını göstermektedir.



Şekil 6.1 DA'nın SDP problemi için işlem adımları

Dinamik gerçek zamanlı uçak sıralama ve zamanlama sorusu matematiksel olarak aşağıdaki gibi ifade edilebilir. n çeşit uçak vardır. i . çeşit uçak indikten sonra j . çeşit uçağın aynı piste güvenli bir biçimde inebilmesi için pozitif bir a_{ij} bekleme süresine ihtiyaç vardır. Genel durumda $a_{ij} \neq a_{ji}$ dir, yani bekleme süreleri matrisi $A = (a_{ij}), i = 1, n; j = 1, n$ matrisi simetrik değildir.

i . çeşit bir uçak indikten sonra o ana kadar geçen toplam bekleme süresinin T olduğunu ve havada inmeyi bekleyen m tane uçak olduğunu varsayalım. Bu uçaklardan k_1 tanesi 1. çeşitten, k_2 tanesi 2. çeşitten, ... k_n tanesi n . çeşitten olsun, yani $k_1 + k_2 + \dots + k_n = n$ dir. Her i için $k_i \geq 0$ dır, yani özel durumda $k_i = 0$ da olabilir. Her uçağın özelliklerine göre havada bekleyebilme süreleri vardır, havadaki uçakları geliş sıralarına göre numaralandırırsak, 1. uçağın havada bekleyebilme süresi b_1 , çeşidi ise $s(1)$ olsun. Her l için $b_l \geq a_{is(l)}$ olduğunu varsayalım. Bir zaman anında i . çeşit uçaktan sonra j . çeşit bir uçağın indiğini ve bu uçağın da kuyruktaki 1. uçak olduğunu varsayalım. Bu durumda tüm uçaklar için a_{ij} bekleme süresi geçecektir.

Bu sırada hava sahasına r tane daha uçağın dahil olduğunu var sayalım ve bunlardan q_1 tanesi 1. çeşit, q_2 tanesi 2. çeşit, ... q_n tanesi n . çeşit olsun. Yani $q_1 + q_2 + \dots + q_n = r$ dir. Her i için $q_i \geq 0$ dır, yani özel durumda $q_i = 0$ da olabilir. Şimdi havada $m+r-1$ tane uçak olacaktır. Yine de bu uçakları geliş sıralarına göre numaralandırırsak ilk $m-1$ uçağın havada bekleme süresi değişecektir ve ilk $l-1$ uçak için, yani $j=1, \dots, l-1$ için $b_j = b_j - a_{ij}$, sonraki $m-1$ uçak için ise, yani $j=l, \dots, m-1$ için ise $b_j = b_{j+1} - a_{ij}$ olacaktır. Yeni gelen r tane uçak için havada bekleme süreleri $b_m, b_{m+1}, \dots, b_{m+r-1}$ olsun. Yine de her l için $b_l \geq a_{js(l)}$ sınırlaması doğru olmalıdır. Yeni bekleme süresi $T = T + a_{ij}$ olacaktır. Amacımız bir işletme günü içinde verilen sınırlamalar dahilinde toplam bekleme süresi olan T yi minimize etmektir. Algoritmanın uygulamalı şekli aşağıdaki örnekte açıklanmıştır.

Örnek:

Sadece 2 çeşit uçak olsun ve bekleme süreleri matrisi olsun. Havadaki uçakların geliş sıralarına göre çeşitleri 1,1,2,1,2,1,1 olsun. Uçaklar geliş sırasına göre indirilirse $T=1+15+2+15+2+1=34$ bekleme süresi olacaktır. Yukarıda anlatılan Doyumsuzluk Algoritmaya göre matrisin en küçük değeri olduğundan önce 1.çeşit uçaklar, daha sonra 2.çeşit uçaklar indirilecektir yani iniş 1,1,1,1,1,2,2 olarak gerçekleştirilecektir ve toplam bekleme süresi $T= 1+1+1+1+15+3=22$ olacaktır. Oysaki 2,2,1,1,1,1,1 iniş sırasında $T=3+2+1+1+1+1=9$ olacaktır.

Bu örnek statik ALS sorusu için bile greedy algoritmanın optimal çözümü vermediğini göstermektedir ama ALS probleminin NP-hard olması nedeniyle optimal çözümü veren algoritmalar gerçek zamanda çalışmayacaktır. Optimal çözümü değil, iyi bir çözümü bulan algoritmalar arasında ise greedy algoritma en hızlı çalışacaktır.

İlk uçağın inişinde bekleme süresi 0 alındığından greedy algoritmanın ilk adımları bir az iyileştirilebilir. Örneğin, matrisin önce ve koşuluna uyan en büyük elemanı bulunur ve ilk olarak l çeşit bir uçak indirilir. Bundan sonra ise matrisin l. satırında bulunan ve koşulunu sağlayan en küçük elemanı bulunur ve t. çeşit bir uçak indirilir. İyileştirilmiş algoritmayı bizim örneğe uyguladığımızda iniş sırası 2,1,1,1,1,1,2 olacaktır ve bu soru için toplam bekleme süresi $T=2+1+1+1+1+15=21$ olacaktır.

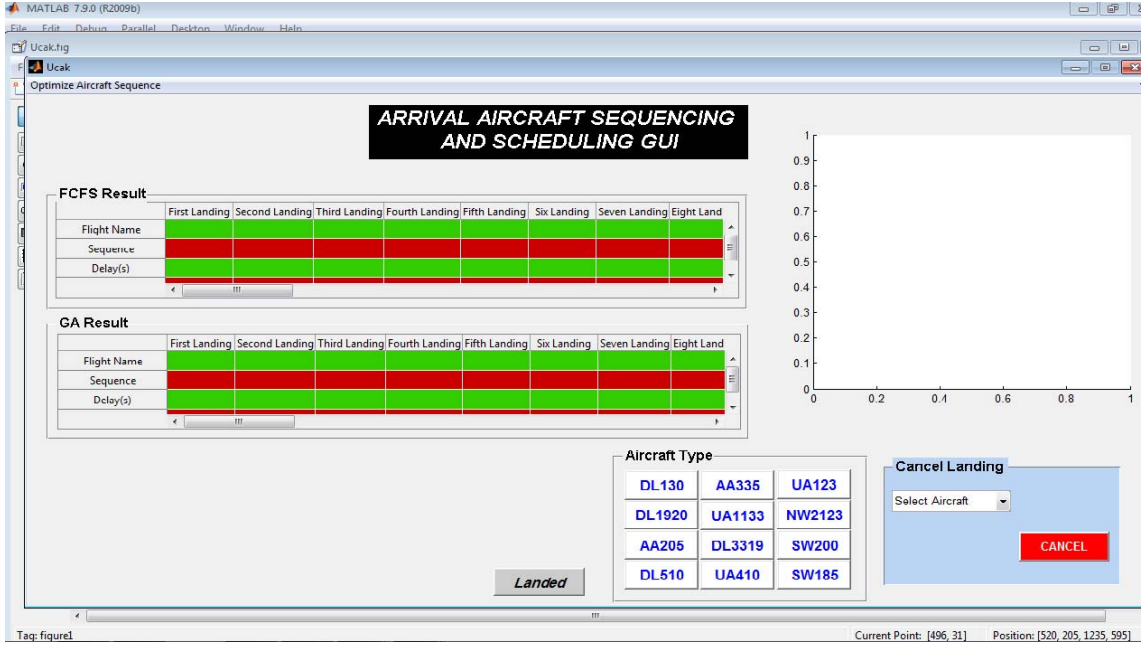
7. TARTIŞMA VE SİMİLASYONLAR

Bu bölüm, GA'nın iki yaklaşımını ve DA'nın performansını SDP problemi için test ederek çıkan sonuçları değerlendirmektedir. Similasyonlar MATLAB programlama ve similasyon dili ile yazılmıştır. Similasyonlar üç bölümden oluşmaktadır ve her birinde şu an kullanılmakta olan ilk gelene ilk hizmet yöntemi İGİH ile karşılaştırmalar gerçekleştirilmektedir. Karşılaştırmalar gecikme, toplam bekleme süresi ve uygulama süresi gibi faktörler değerlendirilmektedir.

7.1 GA ile İGİH Karşılaştırmaları

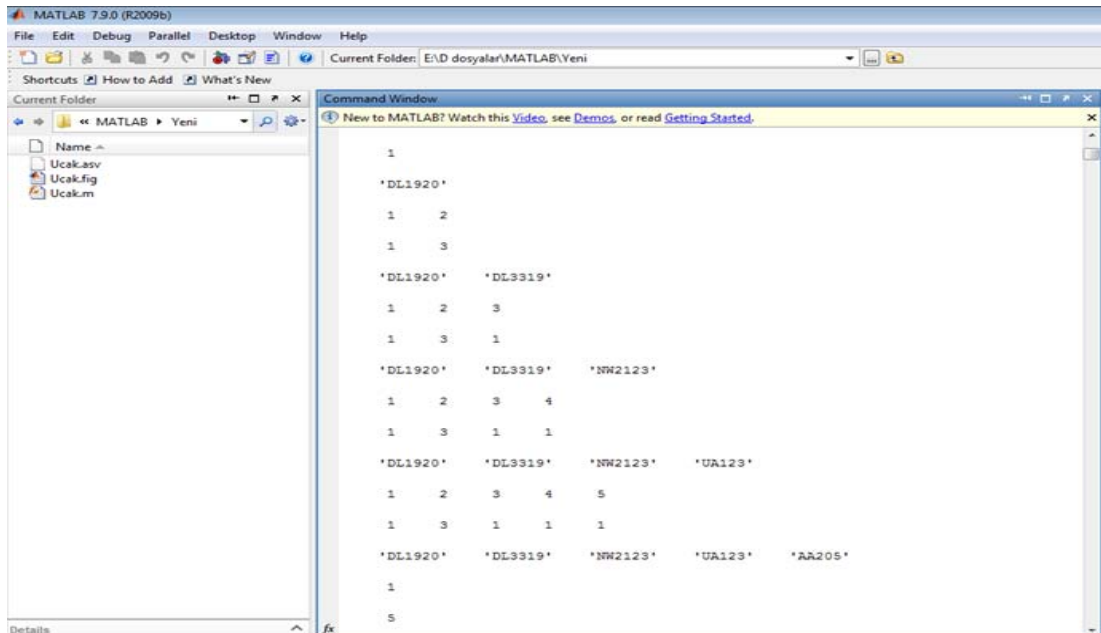
Bu bölümde gerçekleştirilen similasyonlar 5.3.bölümündeki prensipleri esas almaktadır. PVS ve TVS değerlerini kullanarak gecikmeler hesaplanır ve optimal iniş sıralaması elde edilir. MATLAB Guide ile hazırlanan arayüz statik olarak çalışmaktadır. Kullanıcı tarafından belirlenen veriler arayüze giriş olarak kullanılır ve o durum için en uygun çözüm bulunur. Programı çalıştırmak için öncelikle iniş yapacak olan uçakların verilerini uygulamaya girmemiz gerekmektedir. Bu veriler arasında yer alan tahmini iniş süreleri TVS[4]'ten alınmıştır. Şekil 7.1'in aşağı sol kısmında bu verileri girmek için (Aircraft Type) adı altında 12 tuş bulunmaktadır. Bu tuşlara bastığımız zaman aşağıdaki verileri sonradan optimize edecek yöntemle verilecektir :

1. Uçak sıralaması
2. Uçağın hangi kategoriye ait olma bilgisi
3. Uçağın kısaltılmış adı (örnek olarak : Boeing747 kısaltması B747 oluyor)



Şekil 7.1 Matlab ile hazırlanan arayüz

Bu veriler 3 tane matris halinde kaydedilmektedir. Bu durum şekil 7.2'de gösterilmektedir.

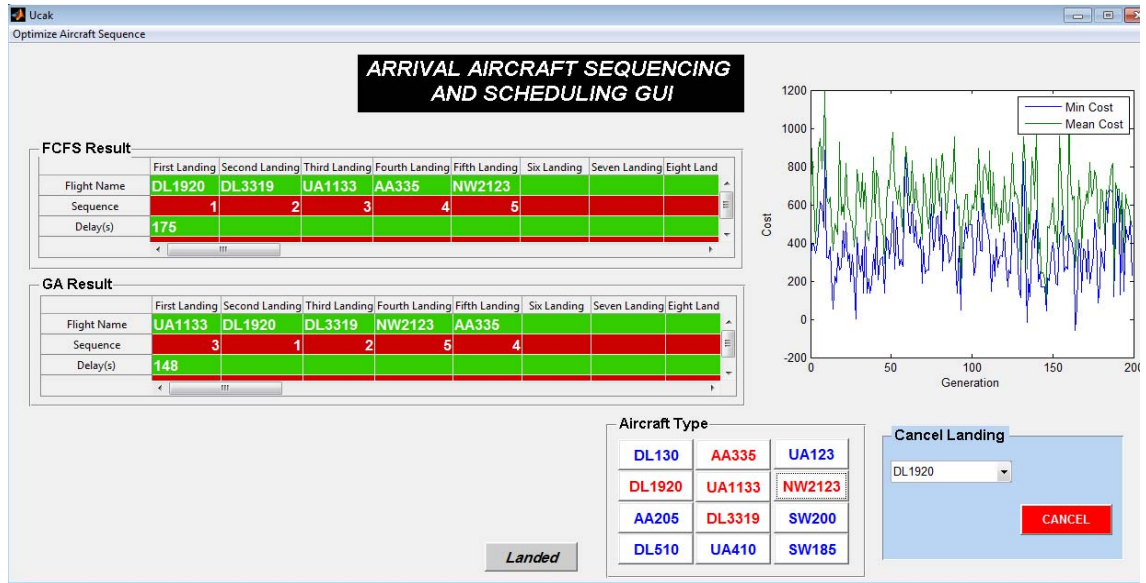


Şekil 7.2 Uçak sıralama, uçak kategori ve uçak isimleri matrisleri

Verileri girdikten sonra, verilen verilerin optimizasyonunu gerçekleştirmek için uygulamanın sağ üst köşesinde bulunan (Optimize Aircraft Sequence) menüsüne gidiyoruz ve oradan da yöntemi seçiyoruz. By Genetic Algorithm yöntemini seçtiğimiz zaman, uygulamanın orta kısmında olan (GA Result Tablosunda) çıkan sonuçlar gösterilir. Tablo üç satırdan oluşmaktadır bunlar :

1. Flight Name : uçağın adı
2. Sequence : orijinal sıralamadaki yeri
3. Delay(s) : belirtilen yöntemle elde edilen gecikme süresi.

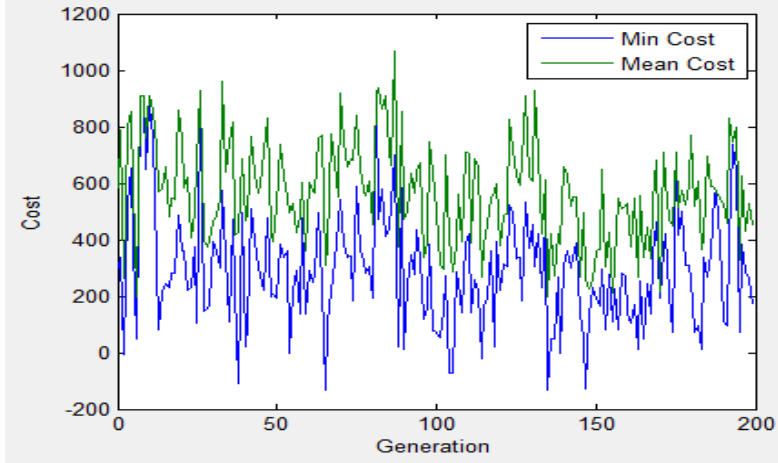
By İĞİH yöntemde ise aynı şekilde İĞİH Result tablosunda yöntemin sonuçları gösterilmektedir. Tablolarda olan her sütun (column), bir uçağın durumu ifade etmektedir ve iniş yapacak kaçınıcı uçak olduğunu açıklamaktadır (First Landing, Second Landing, Third Landing, ..., etc.). Şekil 7.3, 5 farklı türden uçağın iki yöntemle çıkan sonuçlarını göstermektedir.



Şekil 7.3 GA ve İĞİH ile 5 uçağın optimizasyonu

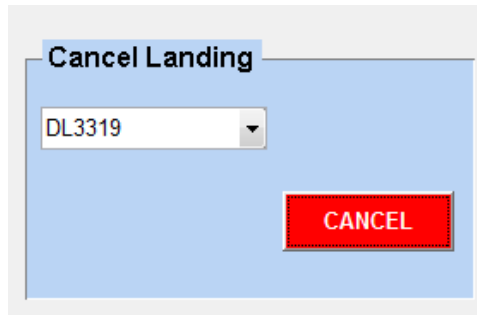
Uygulamada sonuçları sayısal olarak gösteren tablolara ilaveten bir de sonuçların graf şeklinde çizilmiş hali mevcuttur. Şekil 7.4 bu çizimi göstermektedir. Çizimin X-ekseni, jenerasyon sayısını temsil ediyor (kodlamayı yapan kişi tarafından belirlenir burada 50

olarak belirlenmiştir). Y-ekseni ise her jenerasyonda çıkan Cost yani Delay'i(Gecikme) temsil etmektedir. Çizimde Min Cost (mavi çizgi) ve Mean Cost (yeşil çizgi) var. Min Cost, jenerasyonlar sırasında çıkan en küçük değerlerdir. Mean Cost, çıkan sonuçların ortalaması(mean) oluyor.



Şekil 7.4 GA ile çıkan sonuçlar ve her jenerasyonun ortalama gecikme değerleri

Şekil 7.5'te uygulamanın sağ alt köşesinde Cancel Landing bölümü görülmektedir. Bu bölümde girilen verileri (güncelleme veya yanlış girildiğini düşünerek) iptal edilebilir. Bir uçağı iptal etmek için açılan Pop-Up menu'dan ismini seçerek Cancel tuşuna basılır, onay sorusuna Yes verilince uçak iptal olur ve yeni sıralamada yer almaz.



Şekil 7.5 Uçak iptal fonksiyonu

İlk aşamada GA yöntemi İĞİH yöntemine karşı test edilmiştir. Karşılaştırmalar Matlab R2011a versiyonunda Intel Pentium Dual-Core 1.78Ghz işlemciye sahip bir bilgisayarda gerçekleştirilmiştir. Çizelge 7.1'de program tarafından elde edilen sonuçları

göstermektedir, başlangıçta farklı kategorilere ait birkaç uçaktan oluşan küçük sıralamalar test edilmiştir. Daha sonra uçak sayısı adım adım artırılarak sergilenen performans değerlendirilmiştir. Test edilen farklı 8 durum için GA'nın gecikmeleri İĞİH yöntemine göre ortalama olarak 13%-56% oranlarında bir azalma kaydedilmiştir. İyileşme oranları baskın olmasına rağmen hata payı çok ender görülmektedir. Bu hata payı iyi kromozomların kötü kromozomlarla çiftleşmesi ve iyi kromozomların yavaş yavaş özelliklerini kaybetmekten kaynaklanmaktadır. Bir diğer önemli faktör ise uygulama süresidir. GA'nın uygulama süresi 7.34 saniye sürerken İĞİH sadece 0.056 saniye sürmektedir.

Çizelge 7.1 GA ile İĞİH karşılaştırma ve performans değerlendirmesi

Sıralamadaki Uçak sayısı	Toplam gecikme(s) İĞİH kullanarak	Toplam gecikme(s) GA kullanarak	GA hesaplanan gecikmelerin azalma oranı %
3	158	119	24.68
5	540	406	24.81
6	387	308	20.41
9	348	199	42.81
11	676	297	56.06
14	1120	486	56.6
18	3774	3255	13.75
20	4373	3424	21.7

7.2 DA ile GA Similasyonu

Similasyonların ikinci bölümü GA ile DA arasındaki farkı görmek için gerçekleştirilmiştir. Uçaklar arasındaki güvenlik mesafesi LTI değeri esas alınarak her sıralama için toplam bekleme süresi (5.7) formüldeki göre hesaplanır. Daha önceki bölümlerde de belirtildiği gibi, GA tüm olasılıkları tek tek değerlendirip içlerinden minimum TBS sahip sıralamayı seçmektedir. DA ise minimum LTI değerine sahip iki uçak bularak, verilen sıralama kategorilerini tamamlamaya çalışmaktadır. Her iki yöntem için başlangıçta sıralamaya dahil olacak uçak sayısı ve kategorileri

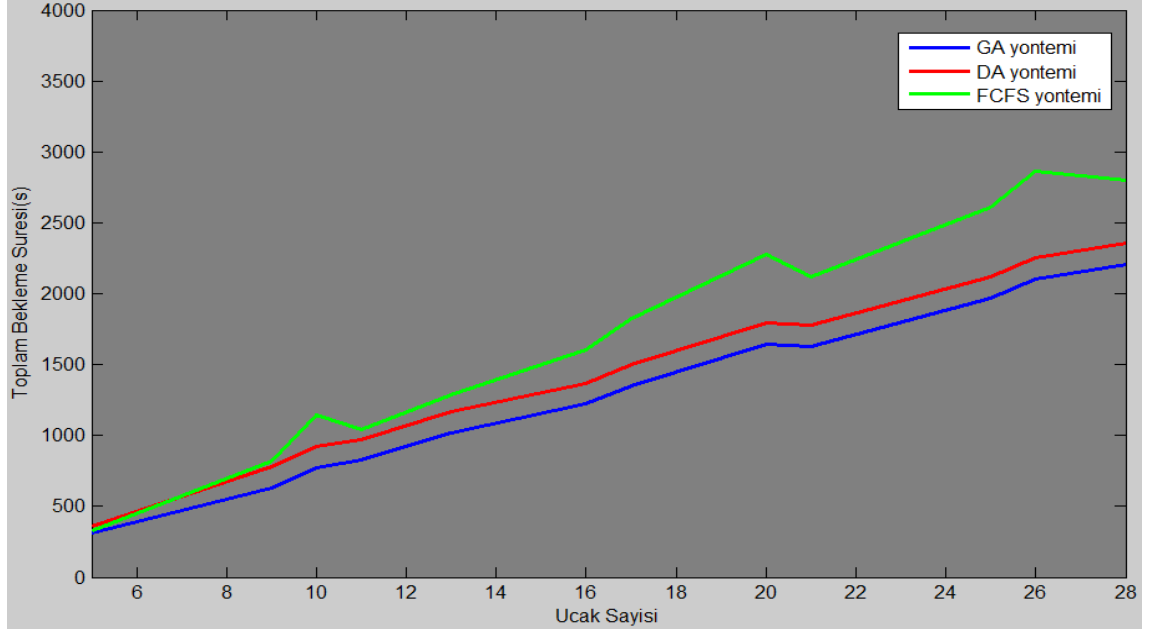
belirlenmelidir. Similasyonda yazılan kod, sırasıyla DA, İGİH ve GA yöntemlerini verilen sırlamaya uygulanmaktadır.

Geliştirilen kod, bu yöneme giriş olarak kullanılacak uçak sayısı ve kategorilerini rastgele seçmektedir. Fakat rastgele olmasına rağmen belirli kısıtlamalar çerçevesinde (örneğin, sadece 4 uçak tipi vardır ve sıralama mantıksal olarak maksimum 30 uçaktan oluşur (Dynammic planner)). Optimizasyon sonucunda, DA, GA, İGİH yöntemleri ile elde edilen TBS ve her bir yöntemin herhalde süre otomatik olarak MS Excel dosyasını yazılmaktadır. Çizelge 7.2 her üç yöntem için gerçekleştirilen similasyon sonuçlarını göstermektedir. Çıkan sonuçlardan açıkça görülmektedir ki DA ve GA yöntemleri İGİH yönteminden daha üstün performans sergilemektedir. Similasyon, uçak sayılarını hem kısa sıramalar hem de uzun sıramaları rastgele seçip değerlendirmektedir. Çizlge 7.2'den görüldüğü gibi DA, çok kısa süre içinde optimal çözüme yaklaşabilmektedir fakat neredeyse hiç bir zaman optimal çözüme ulaşmamaktadır. Diğer yandan GA geniş arama sayesinde her defasında optimal çözümü bulabilmektedir fakat bunu gerçekleştirmek için epey bir süre kaybı yaşanmaktadır. DA, uygulama süresi maksimum 0.042 saniyeyken GA'nın uygulama süresi ise 9 saniye gibi ciddi zaman kaybına uğramaktadır.

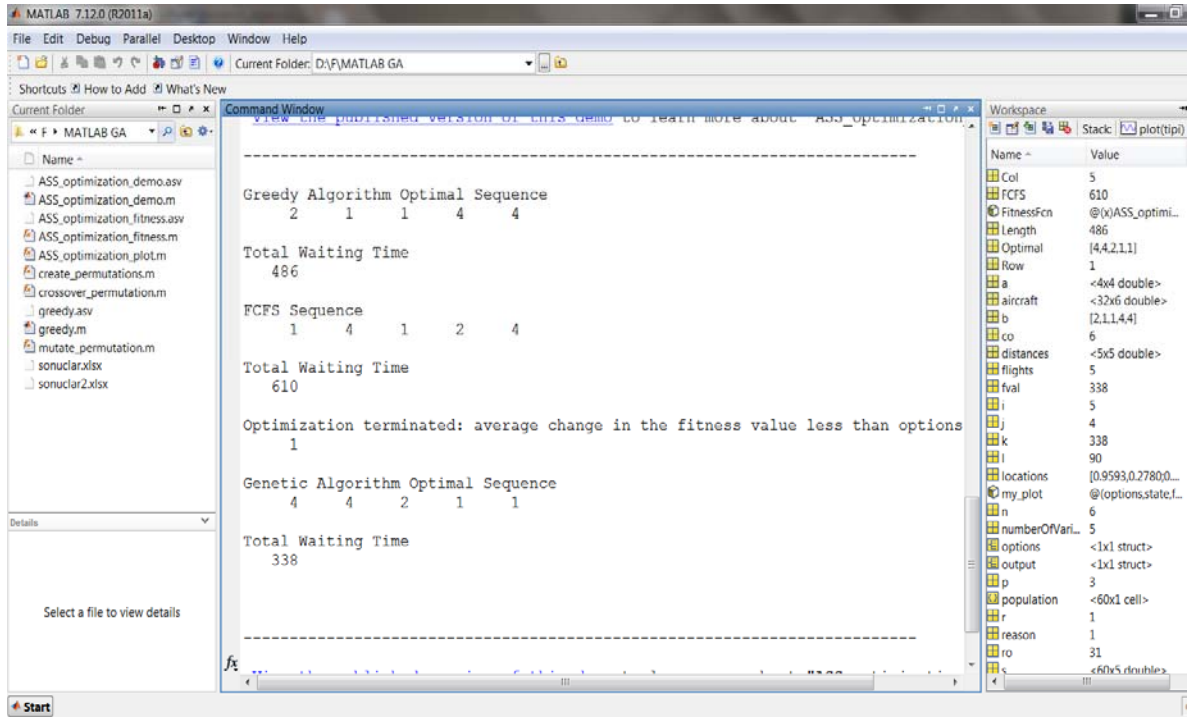
Çizelge 7.2 Similasyon sonucu elde edilen MS Excel tablosunun örneği

Uçak sıralaması	Toplam bekleme süresi FCFS(s) yöntemiyle	Toplam bekleme süresi DA(s) yöntemiyle	Toplam bekleme süresi GA(s) yöntemiyle	DA uygulama süresi(s)	GA uygulama süresi(s)
27	2880	2354	2204	0.039602441	8.970340926
26	2863	2250	2100	0.03625169	8.931380809
9	822	778	628	0.036708135	8.744084078
26	2301	2084	1934	0.037939439	8.941235637
10	1144	926	776	0.034442955	9.460793512
21	2119	1780	1630	0.042125541	8.828406652
28	2803	2356	2206	0.037633125	8.939466497
25	2612	2120	1970	0.037796456	9.02422462
20	2276	1794	1644	0.03792954	9.156931541
40	4099	3378	3248	0.038514671	9.008635073
33	3466	2888	2738	0.059130644	6.08915345
40	4515	3442	3292	0.017234569	3.9958859
33	3466	2888	2738	0.034078863	5.293394929
33	3466	2888	2738	0.023809665	7.046271501
40	4515	3442	3292	0.020672825	4.878389854
36	3667	2992	2842	0.020145774	4.563235428
11	1044	974	824	0.016709877	4.353006143
17	1825	1504	1354	0.020296191	4.581569687
35	3826	3026	2876	0.019771509	4.56038284
38	3927	3248	3098	0.019381846	4.99398037
28	2677	2348	2198	0.021308247	4.830122667
13	1287	1166	1016	0.020870617	4.543234741
16	1567	1344	1194	0.023234647	4.416394357
18	1901	1572	1422	0.019218402	4.698773334
12	1092	1038	888	0.018189764	4.432557641
27	2778	2322	2172	0.020708356	4.566062552
10	1016	896	738	0.017937491	4.396062635
39	4008	3408	3258	0.020482731	4.719905699

Şekil 7.5 Matlab komut penceresinde her üç yöntem kullanılarak 5 uçaklı küçük bir sıralama sonuçları gösterilmektedir, uçakların farklı kategorilerini gösteren 1,2,3,4 numaraları her yöntemde farklı şekilde dizilerek TBS nasıl etkilediğini izlenmektedir. DA ile GA'dan elde edilen toplam bekleme sürelerini karşılaştıracak olursak çıkan sonuçlar birbirine çok yakınlık göstermektedir örneğin sıralama 9 uçaktan oluştuğu halde DA'nın TBS 778 saniyeyken GA'nın TBS 628 saniyedir, diğer durumlarda ise ortalama olarak 200 saniye bir fark saptanmıştır. Buna rağmen SDP probleminin optimizasyonunda GA'yı kullanmak daha avantajlı görülmektedir çünkü optimal sıralamayı bulmak için harcanan süre GA ile DA arasındaki TBS'nden çok daha azdır.



Şekil 7.6 Üç yöntemin graf halinde karşılaştırmaları



Şekil 7.5 Matlab komut penceresinde 5 farklı kategoriye sahip uçağın optimizasyon Sonuçları

Şekil 7.6 veçizlge 7.2'deki bazı sonuçlar graf halinde gösterilmiştir. Burada x-ekseni uçak sayısını y-ekseni ise toplam bekleme süresini ifade etmektedir. Şekilden de

görüldüğü DA ve GA yöntemlerinin İGİH yönteminden olan farkı uçak sayısı artınca büyümektedir fakat DA ile GA arasındaki TBS farkı hemen hemen sabit bir şekilde ilerlemektedir. Diğer yandan uçak sayısının az olduğu bazı durumlarda İGİH yöntemi DA yönteminden daha düşük TBS'ler üretebilmektedir hatta GA bile çok yakın olabilmektedir. Fakat bu durum ancak uçak sayısı 5'i geçmeyen durumlarda ortaya çıkabilir.

8. SONUÇ

Hava trafik kontrol sistemi çok detaylı bir konu olmakla beraber bu tez bu sistemin çok önemli bir parçasını ele almıştır. SDP problemi çok karmaşık bir problem olmamasına rağmen yapılan araştırmalar gösteriyor ki bu probleme daha önce optimal çözümler bulunamamıştır. Çalışmamızda iki önemli arama ve optimizasyon algoritması olan Genetik Algoritma ve Doyumsuzluk Algoritması sözkonusu probleme etkin çözümler bulmak için test edilmiştir. Her yöntemde olduğu gibi bu yöntemlerde de ne yazık ki olumsuz yönler bulunmaktadır, DA olumsuz yönü çoğu zaman optimal çözüme değil ona yakın olan çözümlere ulaşabilmesi GA'nın olumsuz yönü ise optimal çözüme ulaşmak için zaman kaybına uğramasıdır. Buna rağmen gerçekleştirilen simülasyonlar gösteriyorki bu iki yöntem eski İGİH yöntemine karşı gerek sıralamalarda olan toplam gecikme süresi açısından gerekse sıralamadaki toplam bekleme süresi açısından çok daha iyi bir performans sergilemektedir. Sonuç olarak bu yöntemler HTKS'nde kullanıldıkları zaman trafikte yaşanan yoğunluk ve ciddi maddi kayıplara neden olan gecikmelere etkili çözümler getireceği kanıtlanmıştır. Gelecekte bu yöntemler geliştirilerek uçuş kontrol sistemlerinde fiilen kullanılabilir. Bunun yanı sıra problemin ileri aşaması olan farklı pistlere etkili uçak dağıtım problemini de kapsayabilir.

KAYNAKLAR

- Alliot, J.M, Gruber, H., Joly, G. and Schoenauer, M. 1993. Genetic Algorithms For Solving; Air Traffic Control Conflicts, IEEE, pp.338-344.
- Beasley, J.E. 2000. Scheduling Aircraft Landings-The Static Case. Transport Science, Vol. 34, Number. 2, pp.180-197.
- Bianco, L. Bielli, M.M. 1993. System aspects and optimization models in HTK planning. In: Bianco, L., Odoni, A.R. (Eds.), Large Scale Computation and Information Processing in Air Traffic Control. Springer, Berlin, pp. 47-99.
- Bianco, L., Dell'Olmo, P. and Giordani, S. 1997. Scheduling models and algorithms for TMA traffic management. In: Bianco, L., Dell'Olmo, P., Odoni, A.R. (Eds.), Modelling and Simulation in Air Traffic Management. Springer, Berlin, pp. 139-167.
- Capri S. and Ignaccolo, M. 2004. Genetic algorithms for solving the aircraft-sequencing problem:the introduction of departures into the dynamic model. Journal of Air Transport Management 10, pp.345-351.
- Carr G., Erzberger H. and Neuman F.1998. Airline Arrival Prioritization In Sequencing And Scheduling, 2nd Usa/Europe Air Traffic Management R&D Seminar Orlando, pp.1-11,USA.
- Cheng, V., Crawford L. and Menon P. 1999. Air traffic control using genetic search techniques. Proceedings of the IEEE International Conference on Control Applications.
- De Schutter, B. and van den Boom, T. 2001. Model predictive control for max-plus-linear discrete event systems. Automatica Vol. 37 (7), pp.1049-1056.
- Erzberger, H. and Davis TT. 1993. Green S. Design of Centre-TRACON automation system. Proceedings of the 56th symposium on machine intelligence in air traffic management, pp. 11-1-12.
- 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.
- Farley, T, Foster, J, Hoang, T. and Lee, K. 2001. A time-based approach to metering arrival traffic to Philadelphia. Proceedings of AIAA 2001-5241.
- Gianazza, D. and Alliot, J.M. 2002. Optimization Of Air Traffic Control Sector Configurations Using Tree Search Methods And Genetic Algorithms, C. E. N. A., Toulouse, France, IEEE, pp. 2.A.5.1-8.

- Hansen JV. 2004, Genetic search methods for air traffic control. *Computers and Operations Research*. 31:445–459.
- Hu X.B., Di Paolo E. 2007. An Efficient Genetic Algorithm With Uniform Crossover For Air Traffic Control. *Computers & Operations Research*, Elsevier, pp.1-15, UK.
- Hu X.B., Chen WH. 2005. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. *Engineering Applications of Artificial Intelligence*,18(5):633–642.
- Neuman F., Erzberger H. 1990. Analysis of Sequencing and Scheduling Methods for Arrival Traffic. NASA Technical Memorandum 102795, pp.1-45, California.
- Oussedik S., Delahaye D., Schoenauer M. 1999. Dynamic Air Traffic Planning by Genetic Algorithms, *IEEE*, pp.1110-1117.
- Haupt R.L., Haupt S.E. 2004. *Practical Genetic Algorithms (Second Edition)*. John Wiley & Sons, Inc., Hoboken, New Jersey, Canada.
- Sivanandam S.N., Deepa S.N. 2008. *Introduction to Genetic Algorithms*. Springer-Verlag Berlin Heidelberg, Germany.
- Shi Y., Shi Z. 2002. Compression-Coded Genetic Algorithms And It's Application In Traffic Control, *Proceedings of the First International Conference on Machine Learning and Cybernetics*, pp.906-909, Beijing.
- Şahin Emrah Amrahov and Turkan Ali Ibrahim Alsalihe, Greedy Algorithm for Scheduling Aircrafts Landings, *Proceedings of the 5th International Conference "Application of Information and Communication Technologies" (AICT2011)*, (ISBN: 978-1-61284-830-3), pp. 747-749, Baku, Azerbaijan, October 12-14, (2011).
- Wang S. 2009. Solving Aircraft-sequencing problem Based on Bee Evolutionary Genetic Algorithm and Clustering method, *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 157-161, China.
- Wong G. 2000. The dynamic planner: sequencer, scheduler, and runway allocator for air traffic control automation, NASA/TM-2000-209586.
- Zhang X., Zhang X., Zhang J., Liu B. 2007. Optimization of Sequencing for Aircraft Arrival Based on Approach Routes. *IEEE Intelligent Transportation Systems Conference*, pp.592-596, USA.
- Zhang X., Zhou Y., Liu B., Wang Z. 2007. The Air Traffic Flow Management with Dynamic Capacity and Co-evolutionary Genetic Algorithm. *IEEE Intelligent Transportation Systems Conference*, pp.580-585.

ÖZGEÇMİŞ

Adı Soyadı : TURKAN ALI İBRAHİM ALSALIHE

Doğum Yeri : Kerkük_Iraq

Doğum Tarihi : 05.10.1984

Medeni Hali :Bekar

Yabancı Dili : Türkce-İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Alhuda kız lisesi2003

Lisans : Kerkük Teknik Fakültesi2007

Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği
Anabilim Dalı (Eylül 2008-Haziran 2012)

Yayınları (SCI ve diğer)

Şahin Emrah Amrahov and Turkan Ali Ibrahim Alsalihe, Greedy Algorithm for Scheduling Aircrafts Landings, Proceedings of the 5th International Conference “Application of Information and Communication Technologies” (AICT2011), (ISBN: 978-1-61284-830-3), pp. 747-749, Baku, Azerbaijan, October 12-14, (2011).