

UÇAK ÇİZELGELEME PROBLEMİNİN KARINCA KOLONİLERİ OPTİMİZASYONU İLE ÇÖZÜMÜ

Sezgin KILIÇ

Endüstri Mühendisliği Bölümü
Hava Harp Okulu, Dekanlık
34149, Yeşilyurt, İstanbul
s.kilic@hho.edu.tr

Ali Rıza KAYLAN

Mühendislik Fakültesi Dekanlığı
Boğaziçi Üniversitesi
80815, Bebek, İstanbul
kaylan@boun.edu.tr

ÖZET

Havalimanına iniş yapmak üzere havada bulunan uçaklar için iniş sıra ve zamanlarının belirlenmesi özellikle trafiğin yoğun olduğu zaman periyotlarında önemli bir problemdir. Uçak çizelgeleme problemi olarak adlandırılan bu problem temel olarak iş çizelgeleme problemlerine benzemektedir. İşlem zamanları sıralamaya bağımlı olarak değişebilmektedir. Hedef zamanlarında yapılmayan işler için ek maliyet oluşmaktadır. Her uçak için önceden belirlenmiş olan ve iniş yapabileceği zaman aralığını belirleyen alt ve üst sınırlar vardır. Genelde, çizelgeleme sonucunda amaçlanan durum ise uçakların bir veya daha fazla pist için optimum iniş sıralama ve zamanlarının belirlenirken minimum takip mesafelerinin de korunmasıdır. Karınca Kolonileri Optimizasyonu (KKO) metasezgiseli kullanılarak genel amaçlı bir karar verme algoritması geliştirilmiştir. Geliştirilen algoritma tek veya çok pist kullanımında iniş ve kalkışların çizelgelenebilmesi için kullanılabilir. Test problemlerinin çözümü sonucunda elde edilen sonuçlar geçmiş çalışmalar ile karşılaştırılmıştır. Ayrıca KKO yönteminin çizelge problemleri için kullanılabilmesi için değişiklik ve yenilikler önerilmiştir.

Anahtar kelimeler: Çizelgeleme; Metasezgisel yöntemler; Karar Destek Sistemleri; Karınca algoritmaları; Uçak Çizelgeleme Problemi.

ABSTRACT

Scheduling aircraft landings is a major problem in air traffic control area of congested airports. It is a special type of machine scheduling problem; processing times are sequence dependent, and there are penalties for jobs that are not completed on target time. Each plane has an allowable predetermined time window for landing. The objective is to optimally land a set of planes on one or several runways in such a way that separation criteria between all pairs of planes are satisfied. If efficient algorithms can be developed to assist the controller who is in charge of making scheduling decisions, then more effective use of fixed runway capacity will result. We tried to solve the problem using Ant System metaheuristic, which is gained more popularity in recent years. Using Ant System metaheuristic, we present a generic decision making tool that can be used both for the single runway and the multiple runway landings and takeoffs. Computational results are presented for the standard test problems obtained from literature. Results are compared with the previous works and show that Ant System solutions can be effective in practice.

Key Words : Scheduling; Metaheuristics; Decision Support Systems; Ant algorithms; Aircraft Scheduling Problem.

1. GİRİŞ

Dönemsel dalgalanmalar olmasına rağmen hava trafik yoğunluğunun giderek arttığı bilinen bir gerçektir. Trafik yoğunluğundaki bu artışa paralel olarak mevcut sabit kapasitenin daha iyi kullanılarak kapasite kullanım oranlarını artırmaya yönelik karar destek sistemlerinin geliştirilmesine ihtiyaç duyulmaktadır. 1999 yılında Avrupa'da meydana gelen gecikmelerin havayolu şirketleri ve yolcular için toplam maliyetinin

6,6 ile 11,5 milyar Euro arasında olduğu hesaplanmıştır. Günümüzde,

hava trafik kontrolörleri ve bazı karar destek sistemleri genel olarak "İlk Gelene İlk Hizmet" prensibine göre çizelgeleme yapmaktadır. Bu prensip uygulama açısından basitlik sağlasa da özellikle trafik yoğunluğunun fazla olduğu zaman dilimlerinde önemli gecikmelere ve kapasite kullanım oranlarında düşüklüğe neden olabilmektedir. Hava trafik kontrolü ve havalimanı hizmetlerindeki önemli dar boğazlardan

bir tanesini pist kapasiteleri oluşturmaktadır. Aslında nedeni oldukça basittir, pistler üç boyutlu bir hava sahası trafik rejiminden, tek bir kulvara sahip trafik rejimine geçişte ilk adımı oluşturmaktadır. Yetersiz olan pist miktarını artırmak ise oldukça maliyetli bir iştir. Pistlerin inşaat maliyetlerinin yanısıra, havalimanları etrafındaki şehirleşme ve diğer çevresel faktörler de pist maliyetlerini aşırı ölçülerde arttırabilmektedir veya ek pist yapımını imkansız kılabilir. Bu nedenle havalimanı kapasitesinin kullanım oranını arttıracak tekniklere ihtiyaç duyulmaktadır. Kontrolöre çizelgeleme konusunda yardımcı olabilecek karar destek sistemlerinin geliştirilmesi pist kapasitelerinin kullanım oranları arttırabilecektir. Uçak çizelgeleme problemi karmaşık ve NP-zor bir problemdir, kesin çözüm metodları ile küçük boyutlu problemler çözülebilmekte iken problem boyutu büyüdükçe çözüm zamanları oldukça uzun olabilmektedir. Bu nedenle çalışmada çizelgeleme probleminin çözümü için metasezgisel bir yöntem kullanılmıştır. Kullanılan metasezgisel ise son yıllarda geliştirilmiş ve ilgi gören bir yöntem olan Karınca Sistemi'dir. Karınca Sistemi kullanılarak genel amaçlı bir karar verme algoritması geliştirilmiştir. Geliştirilen algoritma tek veya çok pist kullanımında iniş ve kalkışların çizelgelenebilmesi için kullanılabilir. Test problemlerinin çözümü neticesinde elde edilen sonuçlar geçmiş çalışmalar ile karşılaştırılmıştır.

2. UÇAK ÇİZELGELEME PROBLEMİ

Hava trafiğinin yoğun olduğu zaman aralıklarında uçakların iniş için çizelgelenmesi önemli bir hava trafik kontrol faaliyeti olmaktadır. Hava trafik kontrolörü, havada bekleyen uçakların hangi sıra ile iniş yapacağını belirlemenin ötesinde bir zaman çizelgesi de oluşturmak durumundadır. Bu çizelgede kontrolör tarafından her uçak için iniş zamanı ve iniş pisti belirlenir. Her uçağın iniş yapabileceği belirli bir zaman aralığı vardır. Bu zaman aralığının sınırlarını uçağın en geç ve en erken iniş yapabileceği zamanlar belirler. En erken iniş zamanı uçağın bulunduğu konumdan en hızlı şekilde gelerek piste en iniş yapabileceği zamandır. En geç iniş zamanını ise uçağın yakıt durumu ve varsa diğer faktörlerin etkisiyle havada bekleme yapabileceği maksimum süre belirler.

Her uçağın ekonomik uçuş hızı olarak tanımlanan ve en düşük yakıt tüketimi ile uçuş yaptığı belirli bir hızı vardır. Eğer uçak için belirlenen iniş zamanı bu hız ile mümkün olabiliyor ise belirlenen iniş zamanı aynı zamanda uçağın hedef iniş zamanı olarak da isimlendirilir. Eğer hava trafik kontrolünün belirlediği iniş zamanına uyum sağlayabilmek için uçağın ekonomik uçuş hızının üzerine çıkması veya yavaşlaması (havada bekleme yapması) gerekiyorsa ortaya ek bir maliyet çıkmaktadır. Ek maliyetin

miktarı hedef zaman ile çizelgelenen iniş zamanı arasındaki farka bağımlı olarak artmaktadır.

Ardışık olarak iniş yapan iki uçak arasındaki mesafe ayırım mesafesi olarak adlandırılan belirli bir değerin üzerinde olmalıdır. Ayırım zamanları uçakların aerodinamik yapılarına bağımlı olarak değişmektedir. Örneğin, Boeing 747 modelinde bir uçak arkasında önemli şiddetlerde hava akımları oluşturmakta ve bu hava akımları ile karşılaşan başka bir uçak aerodinamik stabilizesini kaybedebilmektedir. Bir çok uçak kazasının bu nedenle meydana geldiği belirtilmektedir [1]. Güvenli uçuş şartlarının sağlanabilmesi için uçakların arkalarında oluşturdukları türbülans miktarına göre ve türbülansın etkilenme derecelerine göre sınıflara ayrılmışlardır ve bu sınıflar arasındaki ayırım mesafeleri belirlenmiştir. Uçakların kalkış durumlarında da benzer koşullar geçerli olmaktadır. Hava trafik kontrol uygulama sahasında uçakların iniş için çizelgelenmesi (iniş sıra ve zamanlarının belirlenmesi) yukarıda belirtilen temel problemden daha karmaşık olmaktadır. Aşağıda bu karmaşıklıkları meydana getiren etkenler üzerinde durulmuştur.

Kontrol

Çalışması içerisinde incelenen ve önerilen yöntemler bir karar probleminin çözümüne yöneliktir. Yani her uçak için ayırım zamanlarına uyularak ve her uçağın önceden belirli olan en geç ve en erken iniş zaman aralığında iniş yapması sağlanmıştır. Böylece her uçak için bir iniş zamanı ve iniş sırası belirlenerek amaç fonksiyonu değerinin optimuma ulaşması sağlanmaya çalışılmıştır. Ancak verilen kararların uygulanabilmesi yani belirlenen zamanlarda ve sıralamada uçakların indirilmesinin kontrol edilebilmesi durumuyla veya bu kontrolün sağlanması durumunda oluşabilecek yeni maliyetler model içerisine dahil edilmemiştir.

Temelde karar modeli ve kontrol problemi bağlantılıdır. Ancak bir çok yöneylem araştırması modelinde olduğu gibi problemin bu iki yanının ayrı ayrı olarak ele alınmasının avantajlı olacağına inanılmaktadır. Ayrıca yaygın olarak kullanılan ve İlk Gelene İlk Hizmet (First Come First Served-FCFS) prensibiyle çalışan karar modelleri de kontrol problemini sonraki aşamalarda ele almaktadır [2,3]. Ancak kontrol probleminin bazı kısıtları basit olarak karar modeli içerisine dahil edilebilmektedir.

Ayırım Zamanları

Zorunlu minimum takip zamanlarını belirlemek için yetkili makamlar (Federal Aviation Administration-ABD, Civil Aviation Authority-İngiltere, Eurocontrol-Avrupa) uçakları temel üç veya dört kategori altında sınıflandırmış ve bu sınıflar arasındaki minimum takip zamanlarını belirlemiştir. Literatürdeki bir çok çalışma bu sınıflandırma temel alınarak yapılmıştır. Ancak uygulamada daha karmaşık durumlar ortaya çıkabilmektedir. Örneğin London Heathrow Havalimanında kalkış durumlarında minimum takip

zamanlarının belirlenmesinde, uçağın kalkıştan sonra izleyeceği rota (Standart Instrument Departure Route-SID) da etkili olmaktadır [4].

En Geç İniş Zamanı

En geç iniş zamanını uçağı yakıt durumu belirlemektedir. Uçağın sahip olduğu yakıt miktarı ile havada bekleme yapabileceği maksimum süreyi ifade eder. Literatürdeki bazı çalışmalarda en geç zaman sonsuz olarak ele alınmıştır. Bu çalışmada daha gerçekçi olması açısından sonlu bir en geç iniş zamanı kullanılmıştır.

Pist Belirleme

Eğer limanda çok sayıda pist mevcut ise iniş yapacak uçaklar için iniş pisti belirlenmelidir. Çalışma içerisinde önerilen model uçakların iniş ve kalkış durumları için ayrı ayrı veya karışık olarak çizelgelenebilmesini sağlayabilecek bir yapıda olmuştur.

Amaç Fonksiyonu

Bu çalışmada, toplam maliyetin minimizasyonu üzerinde yoğunlaşmıştır, her uçak için hedef zamandan sapma durumunda oluşacak olan maliyetin doğrusal olduğu varsayılmıştır. Çizelgelenecek iniş zamanı uçağın hedef zamanına eşit ise uçağın maliyeti sıfır olmaktadır. Aslında, kullanılan amaç fonksiyonu kullanıcıya göre değişebilmektedir. Kullanıcı havalimanı operatörü ise amaç fonksiyonu pist kapasitesi kullanım oranını artırabilir yapıda olabilir. Bir havayolu şirketi için ise amaç fonksiyonu uçaklar için oluşan maliyetin azaltılması yönünde olabilir. Veya askeri çalışmalarda stratejik ve taktik planların oluşturulmasında kullanılmak üzere farklı amaç fonksiyonları geliştirilebilir.

3. ÖNERİLEN MODEL

Her uçak için bir iniş zamanı belirlenerek uçakların hedef zamanlarına en yakın zamanlarda iniş yapmalarının sağlanması Uçak Çizelgeleme Problemi (UÇP) olarak adlandırılmaktadır. Beasley vd. (2000) çalışmasında literatürde bu konuda yapılmış olan geçmiş çalışmalar detaylı olarak aktarılmıştır. UÇP'nin çözümü için geçmiş çalışmalar incelendiğinde birçok çözüm yönteminin kullanıldığı görülmektedir. Simülasyon modeli [7], kuyruk modeli [3], sezgiseller [8,9,10], karma tamsayı doğrusal programlama [4], genetik algoritmalar [5,11] geçmişte bu konuda yapılan çalışmalardandır [12]. Bu çalışmada ise karınca kolonileri optimizasyonu yaklaşımı içerisinde önerilmiş olan Karınca Sistemi (KS) algoritması temel alınarak bir model önerilmiştir.

3.1 Karınca Sistemi

Karınca Kolonileri Optimizasyonu (KKO) ilk olarak Marco Dorigo [13,14] ortaya konulmuştur. Temel olarak karıncaların doğal davranışlarından esinlenilmiştir [15]. Karıncalar geçtikleri yollar

üzerine *pheromone* olarak isimlendirilen bir iz (madde) bırakmaktadırlar. Bu iz karıncaların birbirleri ile haberleşerek yuva ile yiyecek maddesi arasındaki en kısa yolun ortaya çıkarılmasına yardımcı olmaktadır. Karıncalar birçok yol arasından birini seçmek durumunda kaldıklarında iz miktarının yoğun olduğu yolu tercih ederler. Karıncaların yuva ile yiyecekleri arasında oluşturduğu mevcut yol üzerine bir engel konulduğunda engel ile ilk karşılaşan karıncalar eşit olasılıkla engelin sağındaki veya solundaki yolu tercih ederler. Kısa yolu tercih eden karıncalar engelin etrafından daha çabuk dolaşarak eski yola ulaşabileceğinden engel ile daha sonra karşılaşan karıncalar kısa yol üzerinde daha fazla iz bulunduğundan kısa yolu tercih etmeye başlayacaklardır. KKO algoritmalarında, yapay karıncalar (ajanlar) yukarıda açıklanan karınca davranışının yanısıra aşağıdaki özelliklere de sahip kılınarak optimizasyon problemlerinde iyi çözümlerin bulunması sağlanmaktadır:

- Yapay karıncaların yaşam alanı ayrık (kesikli,sürekli) elemanlara sahiptir.
- Geçmişte yapılan hareketler karıncaların hafızalarında tutulur.
- Yapay karıncalar geçtikleri yollar üzerine buldukları çözümün kalitesine bağımlı bir miktarda iz bırakırlar.
- Yapay karıncaların iz bırakma süreci problem yapısına bağılı olarak değişebilir. Örneğin bazı karınca algoritmalarında yapay karıncalar çözümü oluştururken değil sadece çözümü oluşturduktan sonra geçtikleri yollar üzerine iz bırakırlar.
- KKO algoritmalarının etkinliğini artırmak yapay karıncalara ileriye görebilme, yerel optimumu arama vb. gibi bir çok özellik eklenebilir.[16]

KKO yaklaşımının kullanıldığı birçok algoritma vardır. Bu algoritmalar içerdikleri farklı prosedürler nedeniyle farklı isimlerle adlandırılmışlardır. Karınca Sistemi (KS) ise KKO yaklaşımı ile önerilen ilk algoritmadır ve daha sonra önerilen birçok algoritma için de prototip olarak kullanılmıştır. Karınca sistemi gezici satıcı problemlerine çözüm üretmek amacıyla üretilmiştir. Karıncalar her iterasyonda bir şehir seçer ve seçtiği şehre gider, eğer problem N adet şehirden oluşuyor ise N iterasyon sonunda her karınca bir tam tur oluşturmuştur. Her karınca, her iterasyonda gideceği şehri seçmek için aşağıdaki olasılık fonksiyonunu (1) kullanır.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\gamma \times [\eta_{ij}]^\theta}{\sum_{j \in \text{tabu}_k} [\tau_{ij}(t)]^\gamma \times [\eta_{ij}]^\theta} & \text{eğer } j \notin \text{tabu}_k \\ 0 & \text{eğer } j \in \text{tabu}_k \end{cases} \quad (1)$$

(i=1,...,N; j=1,...,N, k=1,...,m)

$P_{ij}^k(t)$: k karıncasının t anında bulunduğu i şehirden

j şehrine gitme olasılığı

$\tau_{ij}(t)$: t anında (i,j) yolunda bulunan iz miktarı.

$\eta_{ij} = 1/d_{ij}$: j şehrinin i şehirden görülebilirliği.

d_{ij} : i ile j şehirleri arasındaki uzaklık.

tabu_k : k karıncasının tabu listesini içeren ve dinamik olarak büyüyen vektör.

γ ve θ parametreleri fonksiyon içerisinde iz ve görüş arasındaki nispi önemi belirlemektedir.

Tabu listesi her karınca için sıralamaya alınan şehirlerin listesini tutarak karıncanın aynı şehri ikinci kez sıralamaya almasını engeller, ayrıca tur tamamlandığında her karıncanın oluşturduğu sıralamayı gösterir. Her karınca t anında t+1 anında bulunacağı şehri seçmektedir. Toplam karınca sayısı m ise her (t,t+1) zaman aralığında toplam m adet seçim ve hareket gerçekleşecektir ayrıca N iterasyon sonunda bir döngü tamamlanmış olacaktır. Bir döngü tamamlandığında her karınca bir tam tur oluşturmuş olur ve şehirler arasına bırakılacak olan iz miktarları aşağıdaki (2) nolu formül ile hesaplanır

$$\tau_{ij}(t+N) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t) \quad (2)$$

ρ : t ile t+N zaman aralığında izlerin buharlaşma oranıdır ($0 < \rho < 1$).

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3)$$

$\Delta \tau_{ij}^k$: t ile t+N zaman aralığında (i,j) yoluna k karıncası tarafından bırakılacak olan iz miktarıdır.

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{Eğer k karıncası (i,j) yolunu (t,t+N) zaman aralığında kullanmışsa} \\ 0 & \text{Değilse} \end{cases} \quad (4)$$

Q sabit bir sayıdır, L_k ise k karıncasının oluşturduğu turun uzunluğudur.

Karıncı Sistemi algoritması şekilsel olarak aşağıdaki gibidir (Dorigo et al. 1996):

Adım-1. Başlangıç değerleri:

t:=0 (t zaman sayacıdır)

NC:=0 (NC döngü sayacıdır)

Her (i,j) yolu için başlangıç değeri olarak

$\tau_{ij}(t) = c$ ve $\Delta \tau_{ij} = 0$ değerini ata.

Adım-2. s:=1 (s tabu listesi indeksidir)
k:=1'den m'e kadar:

k karıncasının başlangıç şehrini $\text{tabu}_k(s)$ olarak belirle.

Adım-3. Tabu listesi dolana kadar tekrar et:

s:=s+1

k:=1'den m'e kadar:

Gidilecek olan j şehri, $P_{ij}^k(t)$

olasılık fonksiyonunu kullanarak belirle,

k karıncasını j şehrine hareket ettir, j şehri $\text{tabu}_k(s)$ olarak belirle.

Adım-4. k:=1'den m'e kadar:

k karıncasını $\text{tabu}_k(N)$ şehirden $\text{tabu}_k(1)$ şehrine götür,

k karıncası tarafından oluşturulan turun uzunluğunu (L_k) hesapla,

Bulunan en kısa yolu güncelle ve hafızaya al.

k:=1'den m'e kadar:

$\Delta \tau_{ij}^k$ değerini (4) nolu eşitliği

kullanarak hesapla.

$$\Delta \tau_{ij} = \Delta \tau_{ij} + \Delta \tau_{ij}^k$$

Adım-5. Her (i,j) yolu için $\tau_{ij}(t+N)$ değerini (2) nolu formül ile hesapla,

t:=t+N

NC :=NC+1

Her (i,j) yolu için $\Delta \tau_{ij} = 0$

Adım-6. Eğer (NC<NC_{max}) ve durgunluk hali yoksa;

Tüm tabu listelerini boşalt,

Adım-2'ye git.

Değilse;

En kısa yolu yazdır,

Dur.

3.2 Uçak Çizelgeleme Problemi için Karınca Sistemi modeli (KS-Çizelge)

Uçak çizelgeleme probleminin (UÇP) notasyonu ve formülasyonu geçmiş çalışmalarda kullanılanlar ile benzer şekilde yapılmıştır.

P : toplam uçak sayısı

(i=1,...,P; j=1,...,P; f=1,...,P)

R : toplam pist sayısı (r=1,...,R)

E_i : i uçağının en erken iniş zamanı (i=1,...,P)

L_i : i uçağının en geç iniş zamanı (i=1,...,P)

T_i : i uçağının hedef iniş zamanı (i=1,...,P)

S_{ij} : i uçağından sonra j uçağı iniş yapıyorsa aralarındaki minimum takip zamanı

($S_{ij} \geq 0, i=1, \dots, P, j=1, \dots, P$)

g_i : i uçağı T_i zamanından önce iniş yapıyorsa birim zaman başına ceza katsayısı ($g_i \geq 0, i=1, \dots, P$)

h_i : i uçağı T_i zamanından sonra iniş yapıyorsa birim zaman başına ceza katsayısı ($h_i \geq 0, i=1, \dots, P$)

x_i : i uçağı için çizelgelenen iniş zamanı ($i=1, \dots, P$)

α_i : i uçağının T_i zamanından ne kadar önce iniş yaptığı ($i=1, \dots, P$)

β_i : i uçağının T_i zamanından ne kadar sonra iniş yaptığı ($i=1, \dots, P$)

Bu çalışmada KS kullanılarak UÇP'nin çözümü önerilmiştir. Esasen en kısa yol problemlerinin çözümü için geliştirilmiş olan KS'nin bir çizelgeleme problemi olan UÇP için kullanılması üzerine yapılan denemelerde güçlü ve etkin bir çözüm algoritmasının geliştirilebilmesi için KS içerisinde bazı farklılıkların ve yeniliklerin yapılması gerekli görülmüştür. Önerilen değişiklik ve yenilikler aşağıda anlatılmıştır, yapılan değişiklik ve yenilikler sonrasında önerilen algoritma KS-Çizelgeleme olarak adlandırılmıştır.

Uçaklar arasına iz bırakılması yerine, uçakların tabu listesinde buldukları sıralama indekslerine iz bırakılması, algoritmanın daha etkin çalışmasını sağlamıştır. Aslında, önerilen bu değişiklik KS'nin diğer çizelgeleme problemlerinin çözümünde kullanılabilmesi için de mantıklı bir yaklaşımdır. Çünkü, bir karınca tarafından oluşturulan bir sıralamada, eğer uçak-A altıncı sırada ve uçak-B yedinci sırada yer alıyor ise bu sıralama için bırakılacak iz miktarının belirlenmesinde uçakların birbirlerine göre buldukları yere nispeten uçakların kendi buldukları sıralar daha fazla etkili olmaktadır. Eğer birden fazla pist için uçakların çizelgelenmesi yapılacak ise her karınca tam bir çizelgeleme yapmak üzere programlanmıştır. Bu prosedür aşağıda verilen KS-Çizelge algoritması içerisinde anlatılmaktadır. Tek pist durumu için olasılık fonksiyonu (5) içerisinde sadece uçakların buldukları sıralama indekslerine bıraktıkları izler kullanılmış, çok pist durumu için ise ilave olarak uçakların birbirleri arasına bıraktıkları iz miktarlarının da kullanılması yararlı görülmüştür. Çok pist durumunda sıralama indekslerine bırakılan izin olasılık fonksiyonu içerisindeki nispi önemi uçakların birbirleri arasına bıraktıkları iz miktarından fazla olmuştur.

Yapay karıncaların ileriye görebilme özelliklerini olasılık fonksiyonu (5) içerisinde taşımak için iki değişken kullanılmıştır. Birincisi uçakların hedef iniş zamanlarıdır ve hedef iniş zamanı daha erken olan uçağın seçilme olasılığı artırılmıştır. İkinci değişken ise ardışık olarak sıralanacak uçak arasındaki artık zamanlara bağımlı olarak değişmektedir.

$$P_{is}^{rk}(t) = \begin{cases} \frac{[\tau_{is}(t)]^r \times [\phi_{fi}(t)]^k \times [\eta_i]^\theta \times [\mu_{fi}]^\varepsilon}{\sum_{i \notin \text{tabu}_k} [\tau_{is}(t)]^r \times [\phi_{fi}(t)]^k \times [\eta_i]^\theta \times [\mu_{fi}]^\varepsilon} & \text{Eğer } i \notin \text{tabu}_k \\ 0 & \text{Eğer } i \in \text{tabu}_k \end{cases} \quad (5)$$

$P_{is}^{rk}(t)$: t anında, i uçağının, k karıncası tarafından oluşturulan sıralamanın s 'inci sırası ($\text{tabu}_k(s)$) için seçilme olasılığı

$\tau_{is}(t)$: t anında i uçağı için s sırasında bulunan iz miktarı

$\phi_{fi}(t)$: t anında f uçağı ile i uçağı arasındaki iz miktarı (f uçağı k karıncası tarafından ($s-1$)'inci sıraya yerleştirilmiş olan uçaktır)

$\eta_i = 1/T_i$: i uçağının görülebilirliği

$\mu_{fi} = [T(i)-T(f)-S(f,i)+1]^{-1}$: f uçağı ile i uçağı arasındaki artık zaman

$\text{tabu}_{kr}(s)$: k karıncasının oluşturduğu çizelgelemeyi gösteren ve dinamik olarak büyüyen bir matristir. r bu matrisin sütun indeksidir, s ise bu matrisin satır indeksidir.

γ, κ, θ and ε parametreleri ise sırayla olasılık fonksiyonu (5) içerisindeki izlerin, görüşün ve artık zamanların birbirlerine göre önem derecelerini belirleyen sabitlerdir.

Tabu matrisi her karıncanın oluşturduğu çizelgeyi hafızasında tutarak bir karıncanın daha önce çizelge içerisine dahil ettiği bir uçağı tekrar çizelge içerisine almasını engeller ve döngü tamamlandığında her karınca tarafından oluşturulan çizelgeyi gösterir. Süreç içerisinde t ile $t+1$ zaman aralığında olan olaylar iterasyon olarak adlandırılır. t anında her karınca $t+1$ anında çizelgeye eklemiş olacağı uçağı seçer ve çizelgeye ekler. Toplam karınca sayısı m ise her iterasyonda toplam m adet seçim yapılır. İterasyon sayısı uçak sayısına (P) ulaştığında ise her karınca tüm uçakları kullanarak bir çizelge oluşturmuş olur ve bu durumda bir döngü tamamlanmış olur. Bir döngü tamamlandığında aşağıdaki (6) ve (7) nolu formüller kullanılarak iz miktarları güncellenir.

$$\tau_{is}(t+P) = \rho \cdot \tau_{is}(t) + \Delta \tau_{is}(t) \quad (6)$$

$$\phi_{fi}(t+P) = \rho \phi_{fi}(t) + \Delta \phi_{fi} \quad (7)$$

$\rho = t$ ile $t+P$ zaman aralığında izlerin buharlaşma oranı ($0 < \rho < 1$).

$$\Delta \tau_{is} = \sum_{k=1}^m \Delta \tau_{is}^k \quad (8)$$

$\Delta \tau_{is}^k$: t ile $t+P$ zaman aralığında i uçağı için s sırasına, k karıncası için bırakılan iz miktarı.

$$\Delta \tau_{is}^k = \begin{cases} \frac{Q}{A_k} & \text{Eğer } i \text{ uçağı } k \text{ karıncasının, } t \text{ ile } t+P \\ & \text{zaman aralığında oluşturduğu} \\ & \text{çizelgenin } s \text{ sırasında bulunuyor ise} \\ 0 & \text{Değilse} \end{cases} \quad (9)$$

$$\Delta \phi_{fi}^k = \sum_{k=1}^m \Delta \phi_{fi}^k \quad (10)$$

$\Delta \phi_{fi}^k$: t ile $t+P$ zaman aralığında, f ile i uçağı arasına, k karıncası tarafından bırakılan iz miktarı.

$$\Delta \phi_{fi}^k = \begin{cases} \frac{Q}{A_k} & \text{Eğer } k \text{ karıncasının } t \text{ ile } t+P \text{ zaman} \\ & \text{aralığında oluşturduğu çizelgede, aynı} \\ & \text{pistte, } f \text{ uçağından hemen sonra } i \text{ uçağı} \\ & \text{yer alıyor ise} \\ 0 & \text{Değilse} \end{cases} \quad (11)$$

Q sabit bir sayıdır. k karıncasının oluşturduğu çizelgenin amaç fonksiyon değeri A_k 'dir.

KS-Çizelge algoritması şekilsel olarak aşağıdaki gibidir:

Adım-1. Başla:

$t:=0$ (t zaman sayacıdır)
 $NC:=0$ (NC döngü sayacıdır)
 Her (i,s) ve (f,i) için başlangıç değeri olarak
 $\tau_{is}(t)=c$ and $\phi_{fi}(t)=c$ değerlerini ata
 $\Delta \tau_{is}=0, \Delta \phi_{fi}=0$

Adım-2. $s:=1$ (s tabu matrisinin sıra indeksidir)

$k:=1$ 'den m 'e kadar:
 $r:=1$ 'den R 'ye kadar: (r tabu matrisinin sütun indeksidir)
 k karıncasının r pisti için başlangıç uçağını $\text{tabu}_{kr}(1)$ değerine ata

Adım-3. Tabu listeleri dolana kadar veya tüm uçaklar çizelgenene kadar tekrar et:

$s:=s+1$
 $k:=1$ 'den m 'e kadar:
 $r:=1$ 'den R 'ye kadar:

Çizelgeye alınacak j uçağını $P_{is}^{rk}(t)$ olasılık fonksiyonu ile belirle, j uçağını $\text{tabu}_{kr}(s)$ yerine yerleştir.

Adım-4. $k:=1$ 'den m 'e kadar

A_k değerini hesapla
 Minimum A_k değerini güncelle ve çizelgesini hafızaya al

$k:=1$ 'den m 'e kadar:

(9) nolu eşitliği kullanarak $\Delta \tau_{is}^k$ değerini hesapla.

$$\Delta \tau_{is} = \Delta \tau_{is} + \Delta \tau_{is}^k$$

$k:=1$ 'den m 'e kadar:

(11) nolu eşitliği kullanarak

$\Delta \phi_{fi}^k$ değerini hesapla.

$$\Delta \phi_{fi} = \Delta \phi_{fi} + \Delta \phi_{fi}^k$$

Adım-5. Her (i,s) ve (f,i) için $\tau_{is}(t+P)$ ve $\phi_{fi}(t+P)$ değerlerini hesapla (6,7).

$t:=t+P$

$NC := NC+1$

Her (i,s) ve (f,i) için $\Delta \tau_{is}=0$ ve $\Delta \phi_{fi}=0$

Adım-6. Eğer ($NC < NC_{\max}$) ve durgunluk hali yok ise
 Tüm tabu matrislerini sıfırla
 Adım-2'ye git,

Değilse

En iyi çizelgeyi yazdır,
 Dur.

Her karıncanın oluşturduğu çizelgenin optimum amaç fonksiyon değerinin (A_k) ve uçakların optimum iniş zamanlarının bulunması için doğrusal programlama modeli kullanılmıştır.

$$A_k = \min \sum_{i=1}^P (g_i \alpha_i + h_i \beta_i) \quad (12)$$

Tek pist durumunda uçakların sıralaması bilindiğinde UÇP'nin matematiksel formülasyonu aşağıda verilmiştir:

$$E_i \leq x_i \leq L_i \quad i = 1, \dots, P \quad (13)$$

$$x_j \geq x_i + S_{ij} \quad (14)$$

(j uçağı i uçağından hemen sonra iniş yapmaktadır)

$$\alpha_i \geq T_i - x_i \quad i = 1, \dots, P \quad (15)$$

$$0 \leq \alpha_i \leq T_i - E_i \quad i = 1, \dots, P \quad (16)$$

$$\beta_i \geq x_i - T_i \quad i = 1, \dots, P \quad (17)$$

$$0 \leq \beta_i \leq L_i - T_i \quad i = 1, \dots, P \quad (18)$$

$$x_i = T_i - \alpha_i + \beta_i \quad i = 1, \dots, P \quad (19)$$

Yukarıda verilen formülasyon (Beasley 2000) kullanılarak aşağıdaki doğrusal programlama modeli geliştirilmiştir:

(12) nolu eşitlik amaç fonksiyonu olmak üzere;

Kısıtlar:

$$\alpha_j - \beta_j - \alpha_i + \beta_i \leq T_j - T_i - S_{ij} \quad (20)$$

$$\alpha_i - \beta_i \leq T_i - E_i \quad (21)$$

$$-\alpha_i + \beta_i \leq L_i - T_i \quad (22)$$

$$\alpha_i, \beta_i \geq 0 \quad (23)$$

$i=1, \dots, P \quad j=1, \dots, P$

Çok pist durumda her pist için yukarıdaki model kullanılarak amaç fonksiyon değeri ve iniş zamanları belirlenmiştir, çizelgenin amaç fonksiyon değeri (iz güncelleştirmede kullanılan) ise her pistin amaç fonksiyon değerinin toplamıdır.

4. TEST PROBLEMLERİNİN ÇÖZÜMÜ

KS-Çizelge algoritması UÇP ile ilgili geçmiş çalışmalarda da kullanılmış olan test problemleri üzerinde denenmiştir [6]. Sonuçlar Tablo-1'de verilmiştir. KS-Çizelge algoritmasının etkinliğinin gösterimi için tabloya aynı test problemlerinin çözüldüğü Beasley (2000) çalışmasının sonuçları da eklenmiştir.

Parametre değerlerinin tespiti için deneysel çalışmalar yapılmış ayrıca geçmiş çalışmalardan da yararlanılmıştır. Denenen yedi adet problemin tümünde optimum sonuçlara en kısa zamanlarda ulaşılan parametre değerleri belirlenmiştir ve tablodaki sonuçlar bu parametre değerleri ile elde edilmiştir. Parametre değerleri aşağıdadır:

m = uçak sayısı (P), $\rho=0.9$, $\gamma=1$, $\theta=5$, $\varepsilon=1$, tek-pist durumunda $\kappa=0$, çok-pist durumunda $\kappa=0.5$, $Q=1$, $c=1$. Her problemin her pist durumunda 20 adet çözümü yapılmıştır. Tabloda verilen sonuçlar bu 20 denemenin ortalamasıdır. KS-Çizelge algoritması MATLAB 6.0 programı ile yazılmış ve Pentium-3 bir dizüstü bilgisayar üzerinde program çalıştırılmıştır.

5. SONUÇ VE DEĞERLENDİRME

Bu çalışmada, uçak çizelgeleme probleminin karınca sistemi algoritması ile çözülebilirliği incelenmiştir. UÇP'nin çözümü için KS algoritması ilk defa bu

çalışmada denenmiştir, ayrıca önerilen değişiklikler ve yenilikler ile KS algoritmasının diğer çizelgeleme problemlerinin çözümünde de etkin olarak kullanılabilmesi sağlanmıştır. Önerilen algoritma KS-Çizelge olarak adlandırılmıştır. Tablo-1'de verilen sonuçların başarılı olduğu görülmektedir ve önerilen algoritmanın pist kapasite kullanım oranlarını artırmak amacıyla bir karar destek aracı olarak hava trafik kontrolörleri tarafından kullanılabilmesi ortaya çıkmıştır. Amaç fonksiyonunda uygun değişiklikler yapılarak elde edilen çizelgelerin farklı amaçları gerçekleştirmesi sağlanabilir. Önerilen model, taktik veya stratejik planların ve çalışmaların yapılmasında askeri ve sivil uygulama alanlarında kullanılabilir. Bu çalışmada genel bir model önerilmiştir, uygulama alanına yönelik olarak yapılacak çalışmalarla daha iyi sonuçların elde edilebileceği öngörülmektedir.

6. KAYNAKLAR

- [1] Mullins J., Trails of Destruction, New Scientist 2056, 28-31,1996
- [2] Odoni A.R., Rousseau J.M., Wilson N.H.M., Models in Urban and Air Transportation, Handbooks in OR & MS, Vol.6, 107-150, 1994.
- [3] Milan J., The Flow Management Problem in Air Traffic Control: A Model of Assigning Priorities for Landings at a Congested Airport, Transportation Planning and Technology 20, 131-162, 1997.
- [4] Beasley J.E., Krishnamoorthy M., Sharaiha Y.M., Abramson D., Scheduling Aircraft Landings-The Static Case, Transportation Science, Vol.34, No.2, Mayıs 2000.
- [5] Beasley J.E., Sonander J., Havelock P., Scheduling Aircraft Landings at London Heathrow Using a Population Heuristic, Journal of Operational Research Society 52, 483-493, 2001.
- [6] Beasley J.E., OR-Library : Distributing Test Problems by Electronic Mail, Journal of Operational Research Society 41, 1069-1072, 1990.
- [7] Andreussi A., Bianco L., Riciardelli S., A Simulation Model of Aircraft Sequencing in the Near Terminal Area, European Journal of Operations Research 8, 345-354, 1981.
- [8] Dear R.G., The Dynamic Scheduling of Aircraft in the Near Terminal Area, Report R76-9, Flight Transportation Laboratory, MIT, Cambridge, MA, 1976
- [9] Dear R.G., Sherif Y.S., The Dynamic Scheduling of Aircraft Density Terminal Areas, Microelectronics and Reliability 29, 743-749, 1989.
- [10] Dear R.G., Sherif Y.S., An Algorithm for Computer Assisted Sequencing and Scheduling of Terminal Area Operations, Transportation Research Part A, Policy and Practise, 25, 129-139, 1991.

Tablo 1. Sonuçlar

| Problem | | | | KS-Çizelge Algoritması Sonuçları | | Beasley (2000) Sonuçları |
|--------------------------------------|-------------|-------------|---------------|----------------------------------|---|---------------------------------------|
| Problem No | Uçak Sayısı | Pist Sayısı | Optimum Değer | Optimum değer bulduğu ilk NC | İlk optimum değere ulaşma zamanı (saniye) | Optimum değere ulaşma zamanı (saniye) |
| 1 | 10 | 1 | 700 | 2,23 | 1,76 | 0.4 |
| | | 2 | 90 | 1,22 | 0,96 | 0.6 |
| | | 3 | 0 | 1 | 0,79 | 0.1 |
| 2 | 15 | 1 | 1480 | 10,1 | 17,37 | 5.2 |
| | | 2 | 210 | 1 | 1,72 | 1.8 |
| | | 3 | 0 | 1 | 1,72 | 0.1 |
| 3 | 20 | 1 | 820 | 11,2 | 35,87 | 2.7 |
| | | 2 | 60 | 8,23 | 26,33 | 3.8 |
| | | 3 | 0 | 1 | 3,2 | 0.2 |
| 4 | 20 | 1 | 2520 | 39,3 | 125,6 | 220.4 |
| | | 2 | 640 | 11,6 | 37,15 | 1919.9 |
| | | 3 | 130 | 8,9 | 28,48 | 2229.2 |
| | | 4 | 0 | 1,2 | 3,84 | 0.2 |
| 5 | 20 | 1 | 3100 | 43,8 | 140,2 | 922 |
| | | 2 | 650 | 21 | 67,04 | 11510.4 |
| | | 3 | 170 | 8,1 | 25,92 | 1655.3 |
| | | 4 | 0 | 1,71 | 5,49 | 0.2 |
| 6 | 30 | 1 | 24442 | 1 | 8,26 | 33.1 |
| | | 2 | 554 | 57,7 | 476,2 | 1568.1 |
| | | 3 | 0 | 17,9 | 147,9 | 0.1 |
| 7 | 44 | 1 | 1550 | 1 | 23,34 | 10.6 |
| | | 2 | 0 | 1 | 23,34 | 0.2 |
| Yedi problem için toplam süre (sn) | | | | 251,19 | 1202,48 | 20153 |
| Yedi problem için ortalama süre (sn) | | | | 11,42 | 54,66 | 916 |

- [11] Abela J., Abramson D., Krishnamoorthy M., De Silva A., Mills G., Computing Optimal Schedules for Landing Aircraft, Proceedings of the 12th National ASOR Conference, Adelaide, Australia, 71-90, 1993.
- [12] Ernst A.T., Krishnamoorthy M., Storer H., Heuristic and Exact Algorithms for Scheduling Aircraft Landings, Networks 34, 229-241, 1999.
- [13] Dorigo M., Optimization, Learning and Natural Algorithms, PhD thesis, Dipartimento di Eletttronica, Politecnico di Milano, IT, 1992.
- [14] Dorigo M., Maniezzo V., Colorni A., Positive Feedback as a Search Strategy, Technical Report

91-016, Dipartimento di Eletttronica, Politecnico di Milano, IT, 1991.

- [15] Deneubourg J.L., Aron S., Goss S., ve Pasteels J.M., The Self-organizing Exploratory Pattern of the Argentine Ant, Journal of Insect Behavior, 3:159-168, 1990.
- [16] Dorigo M., Caro G., Gambardella L., Ant Algorithms for Discrete Optimization, Artificial Life, Vol.5, No.3, 137-172, 1999
- [17] Dorigo M., Maniezzo V., Colorni A., The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on System, Man and Cybernetics, Part-B, Vol.26, No.1, 1996.

ÖZGEÇMİŞLER

Hv.İkm.Ütğm. Sezgin KILIÇ

1976 Turhal doğumlu olan Üsteğmen Sezgin KILIÇ ilk öğrenimini Turhal, orta ve lise öğrenimini ise Amasya, Tokat ve Ankara'da tamamlamıştır. 1994

yılında Ankara Atatürk Anadolu Lisesinden, 1998 yılında ise Hava Harp Okulu Endüstri Mühendisliği bölümünden Hava Teğmeni olarak mezun olmuştur. 1999 yılında Hv. Snf. Ok. ve Tek. Eğt. Mrk. K.lığında İkmal Subay Temel Eğitimini tamamlayarak 15nci Füze Üs İkmal Grup K.lığına Depolar Amiri olarak atanmıştır. 2001 yılında Hava Harp Okulu, Havacılık ve Uzay Teknolojileri Enstitüsü, Endüstri Mühendisliği bölümünde başladığı lisansüstü

KILIÇ, KAYLAN

eğitimini 2003 yılında tamamlayarak Hava Harp Okulu Dekanlığı, Endüstri Mühendisliği Bölüm Başkanlığına öğretim görevlisi olarak atanmış olup halen görevine devam etmektedir. 2003 yılında İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği bölümünde başladığı doktora eğitimini sürdürmektedir. Evli ve bir çocuk babası olup İngilizce bilmektedir.

Prof.Dr. Ali Rıza KAYLAN

1973 yılında Boğaziçi Üniversitesinden B.S., 1975 yılında Syracuse Üniversitesinden M.S. ve 1979 yılında yine aynı üniversiteden Ph.D. derecelerini aldı. Araştırma ve ilgi alanları; Modelleme ve Simülasyon, İşletmelerde Kalite Geliştirme Programları, Bilgisayarlı Bütünleşik İmalat Sistemleri ve Kuyruk Teorisidir. Halen Boğaziçi Üniversitesi Mühendislik Fakültesinde Dekan olarak görev yapmaktadır.