

Rüzgar Hassas Seyir Fazı Uçuş Planlaması Eniyilemesine Dinamik Yaklaşım – Gerçekleştirim ve Sonuç

Güray YILDIZ, Hakan YILMAZ

Uzay Sistemleri Grup Başkanlığı, TUSAŞ, Ankara, Türkiye
gyildiz@tai.com.tr

Uzay Sistemleri Grup Başkanlığı, TUSAŞ, Ankara, Türkiye
hayilmaz@tai.com.tr

Öz: Bir Uçuş Yönetim Sistemi (UYS) 4 boyutlu uçuş planlaması gerçekleştirmektedir; Yatay Planlama (Uçuş Kontrol Noktalarının enlem ve boylamlarının hesaplanması), Dikey Planlama (Uçuş Kontrol Noktalarının yüksekliklerinin hesaplanması) ve Zaman Planlaması (Uçuş Kontrol Noktalarına tahmini varış zamanı).

4 boyutlu uçuş planlamasının doğru ve hassas olarak yapılması ve sonrasında uçağın rotayı belirli yatay, dikey ve zaman hassasiyet sınırları içinde takip etmesinde pilotu yönlendirmesi UYS'nin gerçekleştirdiği fonksiyonların kısa bir özetidir.

Planlanan girdi değerleri ile gerçekleşen değerler arasındaki en ufak bir sapma, özellikle acil durumlarda (motorlardan birinin yanması vb.), uçağın plandan sapmasına ve mevcut durumu göz önüne alacak şekilde yeniden planlama yapılmasını gerekli kılar.

Acil durumlarda, özellikle Okyanus Uçuşlarında Seyir Fazı Uçuş Rota Planlamasının Eniyilenmesi hayati bir öneme sahiptir. Havacılık alanında, özellikle rüzgâr şiddeti ve yönünün yer hızını doğrudan etkilemesi sebebiyle, “*en iyi*” “*en kısa yol*” anlamına gelmemektedir.

Bu bildiri kapsamında UYMS 2012’de analiz ve modelleme aşaması sunulan çalışmanın [1] Tasarım ve Gerçekleştirim aşamaları ve varılan sonuçlar sunulacaktır.

Anahtar Kelimeler: Rota Planlama, Dinamik Programlama, Eniyileme Yöntemi

1. Giriş

DO-236B’de [2] rota tanımlama fonksiyonunun uçulacak olan rotayı dikey, yatay ve zaman boyutlarında hesapladığı ve tanımlanan rotanın planlanan uçuşun başlangıcından bitişine yatay ve dikey boyutlarda bir bütün olarak bağlantılı olmasını sağlayacak şekilde nokta, rota kısıtları (zaman, hız, irtifa, konum) ve bacak tipleri bileşenlerinden oluştuğu belirtilmektedir.

Bu bağlamda uçuş planlaması, kalkış pisti ile iniş pisti arasında kesintisiz olarak yatay, dikey ve zamansal olarak uçuş rotasının tanımlanması şeklinde tanımlanabilir.

Uçuş Planlama ve *Rota Planlama* terimleri birbirlerinin yerine eş anlamlı olarak kullanılmakla beraber, özel olarak değerlendirildiğinde *Rota Planlama Problemi* bir aracın yolculuğunun başından sonuna belirli bir amaç ve kısıtlar doğrultusunda “*en iyi*” rotanın otomatik olarak bulunması anlamına gelmektedir.

“En iyi”nin anlamı uçuştan uçuşa değişmektedir. Arama kurtarma operasyonu icra eden bir pilot için amaç düşük seyir hızında havada kalma süresini arttırmaktır. Daha uzun mesafeli uçuşlarda ise önemli faktör gerekli yakıt miktarı veya gerekli varış zamanı olmaktadır.

Bu bildiri kapsamında, UYMS 2012’de analiz ve modelleme aşaması sunulan çalışmanın [1] Tasarım ve Gerçekleştirim aşamaları ve varılan sonuçlar sunulacaktır.

Bu çalışma sırasında vurgulanması gereken bir husus problem çözüm yaşam döngüsünde sistemli bir yaklaşımın uygulanmış olmasıdır:

- Konsept Analizi,
- Problem Modelinin Formülasyonu,
- Algoritmanın Tasarlanması,
- Algoritmanın Gerçekleştirimi

Bir diğer nokta ise bu çalışma kapsamında problem çözüm kümesinin çok boyutlu olması kaynaklı sorunu aşmak üzere etkin buluşsal yöntemlerin geliştirilmiş olmasıdır. Gerçek dünya probleminin doğru ve hassas bir şekilde modellenmesi, modelin karmaşıklığına yol açmakta ve bu durum da “eniyilik-çalışma hızı” dengesini gözecek şekilde etkin bir buluşsal yöntem kullanımını gerektirmektedir.

2. Literatür Araştırması

Rota planlama problemi pek çok özelliğe göre sınıflandırılabilir:

- Çözüm Alanı:
 - Vektörel
 - Ayrık: Yol ağ modeli/Hücresel ağ modeli/Her ikisinin birleşimi
- Amaç Fonksiyonu: Tek amaç veya çoklu amaç fonksiyonu
- Çevresel verinin (örneğin meteoroloji) bilinirliği
 - Veri kesin olarak bilinir veya belirsizlik/olasılık içerir
 - Sabit veya değişken veri
- Maliyet Fonksiyonu: Her adımda aynı veya değişken
- Sonuç: En iyi veya en iyiyi yakınsayan bir sonuç
- Algoritma: A*, Dijkstra vb.

Rota planlama problemlerinin tamamı problem çözüm kümesinin çok boyutlu olması kaynaklı etkin çözülebilirlik problemini yaşamaktadır. Daha fazla araştırmanın bu alana yoğunlaşması gerekmektedir.

Benton, Lyengar, Deng, Brener ve Subrahmanian [3] ağ modelini oluşturan benzer karakteristiklere sahip hücreleri kümeleyen bir yaklaşım tanımlamışlardır. Bu yaklaşımın hücresel ve yol ağ model yapılarını aynı anda desteklediğini belirtmişlerdir.

Yapılan araştırmaların büyük çoğunluğu Dijkstra ve A* algoritmalarını temel almakta ve iyileştirmeler yapmayı hedeflemektedir. Shi, Cao, S. Zhu ve B. Zhu [4] maliyet fonksiyonu Öklid düzlemi üzerinde mesafe/açı olarak tanımlı olan rota planlama problemlerinde alt sınır değerinde iyileştirmeler gerçekleştirmiştir. Szczerba, Galkowski ve Glicktein [5] uçağın hareketli kısıtları (dönüş çapı, iniş açısı vb.) sebebi ile erişilemeyecek olan hücre ağlarını silen bir yaklaşımla iyileşme sağlamıştır.

Sam P. Liden [6]; yatay olarak UKN’leri (Uçuş Kontrol Noktası) belirlenmiş bir uçuş planında ara tırmanma noktalarının hesaplanmasında rüzgar etkisini göz önüne alan dinamik programlama mantığına dayalı bir çalışma yapmıştır. Bu çalışmada, tek

değişkenin irtifa olması sebebi ile problem çözüm alanı tek boyutlu iken bu bildiride ele alınan modelin problem çözüm alanı ise 4 boyutludur.

3. Dinamik Programlama Algoritması

Dinamik Programlama Model Tanımı [1]'de belirtilmiştir. Dinamik Programlama en iyi çözümü garantileyen bir model teşkil etmektedir; [1]'de belirtilen model, rota planlama problemi olarak, Dijkstra'nın meşhur en kısa yol algoritması [7] ile bire bir formüle edilerek çözülebilmektedir.

Dijkstra'nın en kısa yol algoritmasında algoritmanın karmaşıklık mertebesi $O(n^2)$ 'dir ve algoritmanın pratik çözüm uygulama alanları kısıtlıdır.

Bu çalışma kapsamında, [1]'de belirtilen modeli “en iyi” çözümü sağlayacak şekilde gerçekleyen algoritma ve sonuçları sunulacaktır.

3.1. A* Algoritması

A* algoritması denenecek durum vektörlerini sıralı bir liste halinde tutar. Durum vektörleri ilk durum vektöründen başlayıp kendi üzerinden geçen ve son durum vektörüne erişen patikanın “tahmini” maliyetine ($f(\lambda_i)$ olarak tanımlanır) göre sıralanır. En az tahmini maliyete sahip olan durum vektörü ilk olarak denir.

$g(\lambda_i)$, λ_0 'ı to λ_i 'ye getiren mevcut patikanın maliyetidir; λ_0 'dan λ_i 'ye birden fazla şekilde gelinebileceğinden $g(\lambda_i)$, en az maliyetli yol $g^*(\lambda_i)$ 'ye eşit olmak zorunda değildir; $g(\lambda_i)$ algoritmanın bulunduğu aşamada o ana kadar bulunmuş olduğu en az değerdir.

$h(\lambda_i)$, λ_i 'den λ_{son} 'a ulaşmak için gerekli “tahmini” maliyeti tanımlar. Bu doğrultuda; $f(\lambda_i)$, $g(\lambda_i)$ ve $h(\lambda_i)$ toplamına eşittir, ($f(\lambda_i) = g(\lambda_i) + h(\lambda_i)$). Araştırılmak üzere ele alınan λ_i 'nin varış noktası ($\lambda_i = \lambda_{son}$) olması durumunda algoritma sonlanır.

Ele alınan λ_i 'den itibaren λ_i 'nin komşusu olan durum vektörleri ele alınır. Bu küme içerisinde yer alan $\lambda_{komşu}$ 'lar için $g(\lambda_{komşu}) = g(\lambda_i) + C_i(x_i(\lambda_i))$, incelenen patika üzerinde $\lambda_{komşu}$ 'ya ulaşmanın maliyeti λ_i 'ye ulaşmanın maliyeti ile $\lambda_{komşu} - \lambda_{komşu}$ arasındaki geçiş maliyetinin toplamıdır.

Dijkstra algoritması bütün çözüm kümesini araştırır iken, A* *tahmini* maliyetin yönlendirmesiyle çözüm kümesinin bir alt kümesini araştırmaktadır. Çözüm Alt kümesinin araştırılmasına rağmen en iyi çözümün bulunmasını garantileyen şey *tahmini* maliyetin ($h(\lambda_i)$) hesaplanmasında kullanılan *buluşsal* algoritmanın *Kabul Edilebilirlik* özelliğini sağlamasıdır. Bütün durum vektörleri için “ $h(\lambda_i) \leq g(\lambda_i, \lambda_j)$ ” eşitsizliğinin sağlanıyor olması durumunda $h(\lambda_i)$ buluşsal algoritması “*Kabul Edilebilir*” olarak değerlendirilmektedir.

Beeker'in [8] belirttiğine göre $h(\lambda_i)$ buluşsal algoritmasının aynı zamanda *Tekdüzelik* özelliğini sağlaması daha önce ele alınan bir durum vektörünün o durum vektörüne daha sonraki ziyaretlerde daima daha yüksek maliyetle erişileceğini garanti etmektedir. *Tekdüzelik* özelliği bir durum vektöründen komşu vektöre geçerek tahmini maliyeti azaltmanın mümkün olmadığı anlamına gelmektedir.

3.2. Uygulanan TahminiAltSınır Teknikleri

A* algoritmasında uygulanacak *buluşsal* algoritmanın (*TahminiAltSınır*(λ_i, λ_{son}) = $h(\lambda_i)$) daima gerçek maliyetten az sonuç verir iken aynı zamanda gerçek maliyete

yakın değerler verebilmesi, en iyi çözüme ulaşmadan önce araştırılması gerekli olan durum vektör sayısını azaltacaktır. Bu bildiri kapsamında tahmini maliyet elde edilmesinde kullanılan, iki tanesi bu çalışma kapsamında geliştirilmiş olan, üç *buluşsal* yöntem sunulacaktır:

- *TahminiAltSınır* olarak 0 kullanılması
- En Yüksek Rüzgar Değeri
- Spiral

TahminiAltSınır Olarak 0 Kullanılması: Bu yöntem A* algoritmasının Dijkstra'nın meşhur en kısa yol algoritması [7] ile bire bir aynı olmasına yol açmaktadır. Algoritmanın en iyi çözüme ulaşması garantilenmesine rağmen en iyi çözüm patikasından çok uzak noktalar bile araştırılmaktadır.

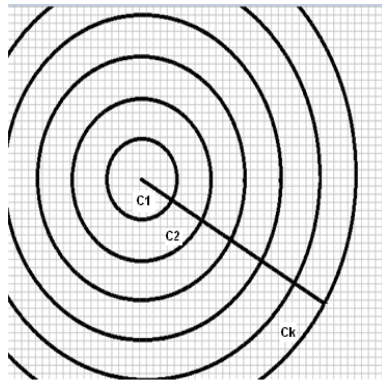
En Yüksek Rüzgar Değeri: Bu yöntemde *TahminiAltSınır*(λ_i, λ_{end}) şu şekilde hesaplanır;

- ($\lambda_i - \lambda_{end}$) arasındaki patika bu iki noktayı birbirine bağlayan uzunluk olarak alınır,
- Rüzgar istikameti ($\lambda_i - \lambda_{end}$) *doğrultusunda ve rüzgar* hız olarak problem çözüm kümesi içerisindeki en yüksek rüzgar hızı alınır (*Rüzgar_{enYüksek}* olarak tanımlanmaktadır)
- Uçak hızı olarak, uçağın mevcut ağırlığı ile ulaşabileceği uçak performans modelinin önerdiği en yüksek hız alınır (*enYüksekTAS* olarak tanımlanmaktadır)

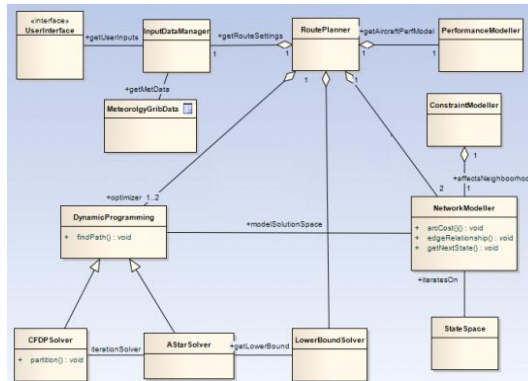
Spiral: *En Yüksek Rüzgar Değeri* buluşsal yöntemi en iyi patikadan çok uzakta ve/veya kalkış-varış doğrultusuna göre ters istikamette olan rüzgar değerinin kullanılması sebebiyle *TahminiAltSınır*'ın gerçekten çok düşük sonuçlar vermesine ve araştırılması gerekli olan durum vektör sayısının artmasına yol açmaktadır.

Spiral yönteminde Şekil 1'de gösterildiği üzere araştırılmakta olan (λ_i) vektör merkez olacak şekilde iç içe geçen daireler oluşturulmakta ve her bir daire içerisindeki en yüksek rüzgar değeri sadece o daire için hesaplanan maliyet değerinde kullanılmaktadır. *TahminiAltSınır*(λ_i, λ_{end}); en iç daireden başlamak üzere bir daireden bir dış daireye geçiş için gereken geçiş maliyetinin toplamıdır. Bu şekilde *TahminiAltSınır* gerçeğe yakın sonuçlar vermekte ve dolayısı ile araştırılması gerekli olan durum vektör sayısı azalmaktadır. Spiral yönteminin karmaşıklık mertebesi $O(n*n)$ 'dir.

En Yüksek Rüzgar Değeri ve Spiral yöntemleri *Kabul Edilebilirlik ve Tekdüzelik* özelliklerini sağlamaktadır. İspat detayları bu bildiri kapsamında belirtilmemiştir.



Şekil 1: Spiral Yöntemi



Şekil 2 Nesne Tasarımı

4. Nesne Tabanlı Tasarım ve Gerçekleştirim

Ele alınan rota planlama problemi değişen parametrelerin sonuca olan etkisini değerlendirebilmek için uygulama yazılımı olarak geliştirilmiştir. Uygulama yazılımı C# ile Microsoft Visual Studio geliştirme ortamında geliştirilmiştir. C# dilinin kullanım gerekçesi nesne tabanlı bir dil olması ve grafiksel kullanıcı arayüzü geliştirimi için hazır kütüphane yeteneklerine sahip olmasıdır. Tasarım ve geliştirme aşamalarında Nesne Tabanlı yaklaşım uygulanmıştır; Nesne Tasarımı Şekil 2’de verilmiştir.

PerformanceModeller nesnesi geliştirilen modelin farklı uçaklar üzerinde kullanılabilmesini sağlamaktadır. Yapılan test ve değerlendirmeler ise C-130 uçağına ait performans özellikleri kullanılarak gerçekleştirilmiştir.

LoverBoundSolver nesnesi paragraf 3.2’de belirtilen *TahminiAltSınır*’ları desteklemektedir. Sadece bu nesne yapılacak değişikliklerle yeni buluşsal yöntemlerin eklenmesi mümkün olabilmektedir.

Sadece *NetworkModeller* ve *ConstraintModeller* nesnelere yapılacak ilaveler ile tanımlı havayolu bağlantıları kullanarak (Hücresel Ağ Yapısının Yol Ağ Yapısına dönüştürülmesi) tanımlı YASAK SAHALAR’dan (NOTAM vb.) geçmeyecek şekilde bir uçuş planlaması gerçekleştirilmesi mümkün olabilmektedir.

5. Test Senaryo Değerlendirme

Geliştirilen matematiksel modelin çalışma etkinliği ve değişkenlerin sonuca olan etkisi çeşitli test senaryoları üzerinde denenmiştir. Test senaryolarının amacı ve kontrol değişkenleri Tablo 1’de tanımlanmıştır.

Tablo 1 Test Senaryo Tanımları

Test Senaryosu	Kontrol Değişkeni	Amaç
testSenaryo1	Buluşsal Yöntem	Farklı buluşsal yöntemlerin sonuca ulaşmadan önce araştırılan durum vektör sayısına etkisi
testSenaryo2	Bekleme Aşama Sayısı (μ)	Bekleme Aşama Sayısı (μ) kısıtı tanımlanmasının çözüme olan etkisi
testSenaryo3	Okyanus Uçuşu	EZN ¹ ’nin yeni algoritma ile hesaplanması ve klasik yöntemde bulunan değer ile kıyaslanması

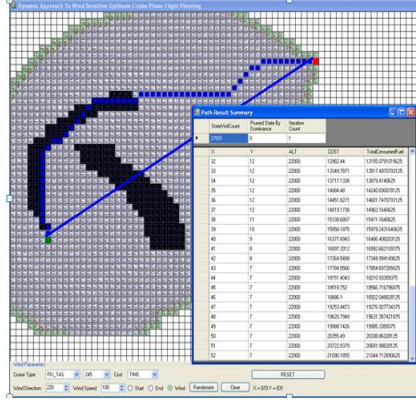
Tablo 2 TestSenaryo1 ve TestSenaryo2 Sonuçları

TestSenaryo1 Sonuçları			TestSenaryo2 Sonuçları		
Buluşsal Yöntem	Araştırılan Vektör Sayısı	Uçuş Süresi ² (saniye)	Kısıt	Araştırılan Vektör Sayısı ³	Uçuş Süresi (saniye)
Dijkstra	65866	21090	Kısıt1	42248	21840
En Yüksek Rüzgar Değeri	27551	21090	Kısıt2	39319	21671
Spiral	22378	21090	Kısıt3	103715	48660

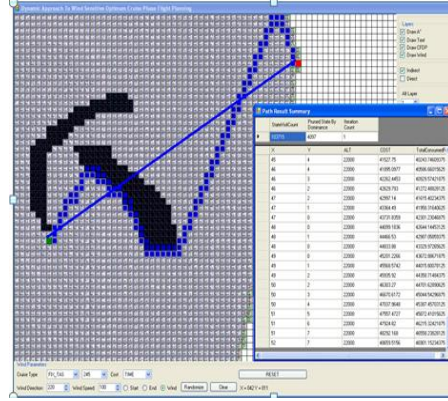
¹ Eşit Zaman Noktası (EZN): İleri doğru uçuşa devam etmek ile kalkış havalanına geri dönmenin aynı zaman süreceği nokta.

² Hız 245 Knot TAS, Mesafe 1333 NM’dir.

³ *En Yüksek Rüzgar Değeri* buluşsal yöntemi uygulanmıştır



Şekil 3 TestSenaryo1 Sonuç



Şekil 4 TestSenaryo2- Kısıt3 Sonuç

5.1. Test1 Sonuçları ve Değerlendirme

Tablo 2’de görüldüğü üzere tüm buluşsal yöntemler aynı sonucu üretmiştir. *Dijkstra*, *En Yüksek Rüzgar Değeri* ve *Spiral* buluşsal yöntemleri en iyi sonucu garantilediği için bu sonuç beklenti ile uyumludur.

Şekil 3’de En Yüksek Rüzgar Değeri yöntemi ile elde edilen sonuç gösterilmiştir; koyu alanlar rüzgarın şiddetli olduğu bölgeleri göstermektedir (yukarıda yer alan bölgede rüzgar uçuş doğrultusunda aşağıda ise ters yönde esmektedir).

Dijkstra ile kıyaslandığında diğer iki yöntemin ne kadar kazanç sağlayacağı rüzgar hız ve doğrultusunun dağılımına bağlıdır. Dağılımdan bağımsız olarak *Spiral* yönteminde araştırılacak vektör sayısı, *En Yüksek Hız Değeri* yönteminde araştırılacak vektör sayısından daima (eğer eşit değil ise) daha az olacaktır.

5.2. TestSenaryo2 Sonuçları ve Değerlendirme

Uygulanan kısıtların tanımları şu şekildedir:

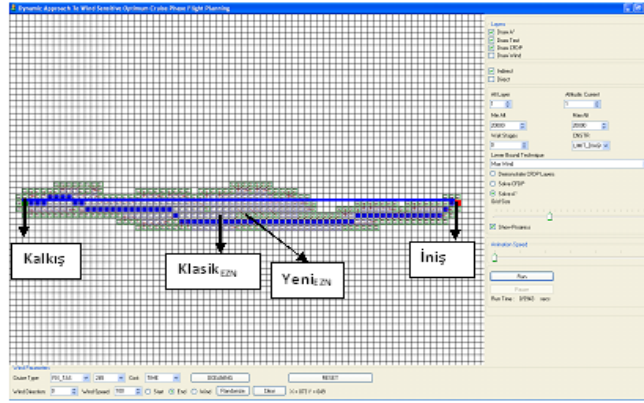
- Kısıt1: Çapraz bir hareket yaptıktan sonra tekrar bir çapraz hareket yapmak için 3 çapraz olmayan hareket yapılmalıdır
- Kısıt2: Çapraz bir hareket yaptıktan sonra tekrar bir çapraz hareket yapmak için 2 çapraz olmayan hareket yapılmalıdır
- Kısıt3: Sağa doğru bir hareket yaptıktan sonra tekrar sağa hareket yapmak için 2 sağa doğru olmayan hareket yapılmalıdır

Kısıt tanımlanması durum vektörüne bir boyut daha eklenmesine ve Tablo 2’de görüldüğü üzere araştırılan vektör sayısında artışa yol açmıştır. Kısıt1 için; Tablo 2’de belirtilen 27551’den 42248’e bir artış olmuştur.

Kısıt3’de ise çok sıkı bir kısıt tanımlanmıştır (Şekil 4’de sonuç gösterilmiştir); bu kısıt örneğin uçağın sağ kanadında bir arıza olduğu ve uzun süre sağa yatış açısı sağlanamayacak durumları modellemek için kullanılmıştır.

5.3. TestSenaryo3 Sonuçları ve Değerlendirme

Okyanus uçuşlarında EZN'nin doğru ve hassas olarak hesaplanması hayati öneme sahiptir. Klasik hesaplama yönteminde geri dönüş rotası ile ana uçuş rotası ile aynıdır.



Şekil 5 Klasik_{EZN} – Yeni_{EZN}

Hava tahmin verisinin olmadığı/göz önüne alınmadığı durumlarda EZN kalkış-iniş havaalanının tam ortasında bulunmaktadır. Ana rotanın rüzgâr hızı ve yönü göz önüne alınarak eniyendiği bir durumda ana rotanın ters doğrultusunda uçulması ise en kötü sonucu verecektir.

Klasik yöntem ile hesaplanmış olan EZN (Klasik_{EZN}) Şekil 5'de gösterilmiştir. Klasik_{EZN}'den kalkış veya iniş noktasına gitmek 3 saat 52 dakika sürmektedir (Kalkış-İniş mesafesi 1750 NM'dir).

Geri dönüşte aynı rotayı takip etmek ile ilgili kısıt kaldırıldığı zaman, geri dönüş rotası ayrıca bir eniyeme problemi olarak ele alınarak, hesaplanan Yeni_{EZN} Şekil 5'de Klasik_{EZN} ile birlikte gösterilmiştir. Yeni_{EZN}'den kalkış veya iniş noktasına gitmek 3 saat 32 dakika sürmektedir.

İki yöntem kıyaslandığında Yeni_{EZN}'nin 100 NM daha ileride olmasına rağmen 20 dakika daha az olduğu görülmektedir.

6. Sonuç

Layton, Smith, McCoy ve Bihari [9] Uçuş Planlama Sistemlerini yeteneklerine göre üç başlık altında ele almıştır:

- Birinci Seviye: Pilota kullanıcı arayüzü ile UKN'leri tanımlama imkanı veren ve sonrasında dikey uçuş planlaması ve zaman/yakıt hesabı yapan sistemler,
- İkinci Seviye: Pilotun prosedürlerden gelen yükseklik ve hız kısıtlarına ilave olarak, karşılaşılabilecek en yüksek türbülans değeri gibi, kısıtlar tanımlamasına izin veren ve Birinci Seviye yetenekler sonucunda elde edilen uçuş planlamasını tanımlanan bu kısıtları sağlayacak şekilde güncelleyen sistemler,
- Üçüncü Seviye: Tanımlanan kısıtlar doğrultusunda bir problem ile karşılaştığında hemen çözen Uzman Sistemler. Bu sistemler, kullanıcıdan iniş/kalkış pisti, uçak konfigürasyonu ve geçerli kısıtlar haricinde bir girdi beklememekte ve uçuş rota planlamasını otomatik olarak gerçekleştirmektedir.

Mevcut UYS Uçuş Planlaması yetenekleri Pilot tarafından tüm UKN'lerin tanımlanmasını takiben, UKN'lere yükseklik atanması ile yakıt ve zaman hesaplamalarının gerçekleştirilmesi şeklinde "Birinci Seviye" yetenekler sağlamaktadır. Yapılan çalışma ile UYS'nin bir karar destek sistemi olarak Seyir Fazı Uçuş Planlaması "*eniyilemesi*"ni gerekli yakıt veya zaman kısıtları doğrultusunda kullanıcı tanımlı kısıtları gerçekleştirecek şekilde otomatik olarak gerçekleyerek "Üçüncü Seviye" yeteneklere sahip olması hedeflenmiştir.

Uygulamanın geliştirilmesinde Nesne Tabanlı Analiz ve Tasarım tekniğinin uygulanması sebebi ile uygulamanın değişik amaçlarda kullanılacak şekilde değiştirilmesi kolaylıkla sağlanabilmektedir. Örneğin, ağ yapısının hücresel tabanlı değil tanımlı UKN'ler arasında olması (yol ağ yapı modeli) için sadece Ağ Modelleme sınıfının değişmesi yeterli olacaktır. Benzer şekilde; uygulama uçuş rotasını Uçuşa Yasak Bölge vb. kısıtlı alanlardan geçmeyecek şekilde oluşturabilmektedir.

Çalışma sonuçları C-130 Modernizasyon Projesi Harekat Uçuş Yazılımı UYS bileşeni ile entegre edilerek acil durum planlaması ve okyanus hesaplamalarında kullanılarak mevcut hesaplamalara olan iyileşme değerlendirilecektir.

7. Kaynakça

1. **Güray YILDIZ**, Rüzgar Hassas Seyir Fazı Uçuş Planlaması Eniyilemesine Dinamik Yaklaşım UYMS 2012
2. **DO-236B** Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation. Washington, DC 20036 USA : Radio Technical Commission for Aeronautics (RTCA) SC-181, 2003.
3. **John Benton, S. S. Lyengar, W. Deng, N. Brenner, V.S. Subrahmanian**, Tactical Route Planning: New Algorithms for Decomposing the Map, Page(s): 268-277.
4. **Hui Shi, Wen Cao, Shulong Zhu, Baoshan Zhu**, *Applications of the Improved A* Algorithm for Route Planning*, Page(s): 299-302.
5. **Szczerba R J, Galkowski P, Glicktein I S**, et al, Robust algorithm for real-time route planning, Aerospace and Electronic Systems, 2000, 36(3): pp. 869-878, doi: 10.1109/7.869506, Page(s): 869-878.
6. **Sam P. Liden**, Apparatus and Method for Computing Wind-Sensitive Optimum Altitude Steps In a Flight Management System, November 12, 1996, Patent No: 5,574,647
7. **Dijkstra**, A note on two problems in connexion with graphs, E. W. s.l. : Numerische Mathematik, 1959, Vol. 1, Page(s): 269-271.
8. **Becker, Emmet**, Potential Error in the Reuse of Nilsson's A Algorithm for Path-finding in Military Simulations, The Journal of Defense Modelling & Simulation., April 2004, Vol. 1, No.2, Page(s): 91-97.
9. **Charles F. Layton, Philip J. Smith, C. Elaine McCoy, Thomas E. Bihari** An Empirical Evaluation of Computer-based Tools to Aid in Enroute Flight Planning 0-7803-0720-8/92 IEEE 1992