



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**ALÇAK YÖRÜNGEDEKİ UYDU SİSTEMLERİNDE YENİ
BİR YÖNLENDİRME ALGORİTMASININ TASARIMI**

Bilg.Bil.Yük.Müh. Derya YILTAŞ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Doç.Dr. A.Halim ZAIM

Ekim, 2007

İSTANBUL



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**ALÇAK YÖRÜNGEDEKİ UYDU SİSTEMLERİNDE YENİ
BİR YÖNLENDİRME ALGORİTMASININ TASARIMI**

Bilg.Bil.Yük.Müh. Derya YILTAŞ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Doç.Dr. A.Halim ZAIM

Ekim, 2007

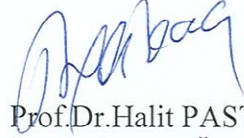
İSTANBUL

Bu çalışma 15/10/2007 tarihinde ařağıdaki jüri tarafından Bilgisayar Mühendisliğı Anabilim Dalı Bilgisayar Mühendisliğı programında Doktora Tezi olarak kabul edilmiştir.

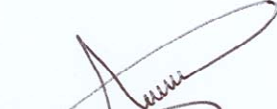
Tez Jürisi



Doç.Dr.A.Halim ZAİM (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi



Prof.Dr.Halit PASTACI
Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi



Prof.Dr.Ahmet SERTBAŞ
İstanbul Üniversitesi
Mühendislik Fakültesi



Prof.Dr.Mahmut ÜN
Beykent Üniversitesi
Mühendislik-Mimarlık Fakültesi



Doç.Dr.Ataç BAŞÇETİN
İstanbul Üniversitesi
Mühendislik Fakültesi

Bu alıřma İstanbul Üniversitesi Bilimsel Arařtırma Projeleri Yürütücü Sekreterlięinin
T-739/13092005 numaralı projesi ile desteklenmiřtir.

ÖNSÖZ

Lisansüstü öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç.Dr.A.Halim Zaim'e en içten dileklerimle teşekkür ederim. Tez izleme komitemde olan ve iyi bir çalışmanın ortaya çıkması için fikirlerini esirgemeyen Prof.Dr.Ahmet Sertbaş'a ve Prof.Dr.Halit Pastacı'ya teşekkürlerimi sunarım.

Doktora öğrenimime ve çalışmalarına destek sağlayan İstanbul Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne ve TÜBİTAK'a teşekkürü borç bilirim.

Her zaman yanımda olan ve her güçlüğü atlatmamda bana destek olan aileme sonsuz teşekkür ederim. Bu çalışmamı çok değerli anneme ve babama armağan ediyorum.

Ekim, 2007

Derya YILTAŞ

İÇİNDEKİLER

ÖNSÖZ.....	I
İÇİNDEKİLER	II
ŞEKİL LİSTESİ.....	V
TABLO LİSTESİ	VII
SEMBOL LİSTESİ	VIII
ÖZET.....	X
SUMMARY	XI
1. GİRİŞ	1
2. GENEL KISIMLAR	4
2.1. UYDULARLA İLGİLİ BİLGİLER.....	4
2.1.1. Temel Kavramlar	8
2.1.2. Yörünge Çeşitleri.....	11
2.1.2.1. Alçak Yörüngelerin Seçilme Amacı.....	15
2.2. ÇEŞİTLİ YÖNLENDİRME ALGORİTMALARI.....	16
2.2.1. Veri Paketi (Datagram) Yönlendirme Algoritması.....	16
2.2.2. Sonlu Durum Otomasyonu (SDO) – Tabanlı Yönlendirme Algoritması	18
2.2.3. Lojiksel Topoloji Tabanlı Yönlendirme.....	18
2.2.4. Coğrafik Tabanlı Paket Yönlendirmesi.....	20
2.2.5. Minimum Atlama/Maliyet Algoritması (MAA, MMA).....	22
2.2.6. Ağ Algoritması	23
2.2.7. Tahmini Servis Kalitesi (SK) Yönlendirme Metodu.....	23
2.2.8. Yaşam Süresini İçeren SK-Tabanlı Yönlendirme.....	24
2.2.9. Yönlendirme İçin Sinir Ağı Yaklaşımı.....	26
2.2.10. ATM Sanal Yol Yönlendirmesi	26
2.3. OPTİMİZASYON PROBLEMLERİNE GİRİŞ.....	27
2.4. EVRİMSEL HESAPLAMA TEKNİKLERİ.....	29
2.4.1. Evrimsel Algoritmalar.....	30
2.4.1.1. Evrim Stratejileri	30

2.4.1.2. Evrimsel Programlama.....	31
2.4.2. Genetik Programlama.....	31
2.4.3. Genetik Algoritmalar	32
2.5. GENETİK ALGORİTMALARDAKİ İŞLEMLER VE GÖSTERİMLER.....	38
2.5.1. Yeniden Üretim (Regeneration)	38
2.5.1.1. Rulet Çarkı Seçim Yöntemi	38
2.5.1.2. Sıralama Yöntemi.....	39
2.5.1.3. Seçme (Elitist) Yöntemi.....	40
2.5.1.4. Turnuva Yöntemi.....	40
2.5.1.5. Sabit Durum (Steady State) Yöntemi.....	40
2.5.1.6. Tahmini Evrensel Seçim Yöntemi.....	40
2.5.2. Çaprazlama Teknikleri	41
2.5.3. Ters Alma ve Yeniden Düzenleme	42
2.5.4. Mutasyon ve Naive Evrimi.....	43
2.5.5. İkili Olmayan Gösterimler.....	43
2.5.6. Kodlama ve Hesaplama.....	44
3. MALZEME VE YÖNTEM.....	45
3.1. STATİK UYDU AĞININ OLUŞTURULMASI.....	46
3.2. DİNAMİK UYDU AĞINA GEÇİŞ	51
3.3. DİNAMİK YÖNLENDİRME ALGORİTMASINDA YOL SEÇİMİ.....	53
3.3.1. Uygunluk Değerlerinin Hesaplanması.....	55
3.3.2. Dinamik Topolojilerin Elde Edilmesi.....	59
3.3.3. Dayanıklılık Kriterinin Uygunluk Değerine Etkisi.....	62
3.4. RULET ÇARKI YÖNTEMİNİN YOL SEÇİMİNDE KULLANIMI.....	63
3.5. ÇAĞRILARIN BLOKE OLMA OLASILIKLARI.....	66
3.5.1. Olasılık Süreçleri ve Dağılımlarıyla İlgili Bazı Önemli Kavramlar	67
3.5.1.1. Poisson Dağılımı.....	67
3.5.1.2. Markov Modellemesi.....	68
3.5.1.3. Yarı-Markov Süreci.....	69
3.5.1.4. Doğma-Ölme Süreci.....	70
3.5.2. Uydu Ağındaki Herbir Linkin ÇBO Değerinin Hesaplanması.....	73
3.6. ÇBO DEĞERLERİNİN ZAMAN VE KONUMA BAĞLI HESABI	76
3.6.1. Konum ve Kapsama Alanıyla İlgili Kavramlar	77
3.6.2. ÇBO Değerlerinin Analitik Çözümü.....	81
4. BULGULAR.....	92
4.1. STATİK ALGORİTMANIN PERFORMANS ÖLÇÜMÜ	92
4.1.1. Gecikme Sürelerinin Hesabı.....	92

4.1.2. Aktarım Sayısının Bulunması.....	94
4.2. STATİK VE DİNAMİK ALGORİTMALARIN KARŞILAŞTIRILMASI.....	98
4.3. STATİK VE DİNAMİK ALGORİTMALARIN DEĞİŞTİRİLMİŞ DIJKSTRA ALGORİTMASIYLA KARŞILAŞTIRILMASI	103
4.4. ANALİTİK ÇBO ÇÖZÜMÜNÜN KARŞILAŞTIRMA SONUÇLARI.....	106
4.5. ANALİTİK ÇBO ÇÖZÜMÜNÜN FARKLI FONKSİYONLARA GÖRE SONUÇLARI	111
5. TARTIŞMA VE SONUÇ.....	117
KAYNAKLAR	119
EKLER.....	125
EK-A ALGORİTMADAKİ ADIMLARIN ÖZETİ.....	125
EK-B SİMÜLASYON ÇIKTILARI.....	125
ÖZGEÇMİŞ.....	130

ŞEKİL LİSTESİ

Şekil 1.1	: YAL ve UAL gösterimleri (Franck ve Maral, 2000).....	2
Şekil 2.1	: Araç Çizgisi.....	10
Şekil 2.2	: Kaplama Yolları.....	11
Şekil 2.3	: Bazı uydu çeşitlerinin görünüşü (Inetdaemon.com, 2005).....	14
Şekil 2.4	: HEY'deki bir uydunun görünüşü (The Tech Museum of Innovation, 2006).....	15
Şekil 2.5	: Lojiksel bir ağ topolojisi (Henderson ve Katz, 2000).....	20
Şekil 2.6	: Coğrafi paket gönderimi ve yerel (locally-scoped) en kısa yol yönlendirmesi (Henderson ve Katz, 2000).....	21
Şekil 2.7	: Coğrafi gönderim ile minimum-gecikmeli en kısa yol yönlendirmesi arasındaki ortalama ve maksimum gecikme farkları (Henderson ve Katz, 2000).....	22
Şekil 2.8	: Arama Teknikleri.....	30
Şekil 2.9	: Şimdiki jenerasyondan başlayarak yeni jenerasyonlar oluşturma....	35
Şekil 2.10	: Rulet Çarkı.....	38
Şekil 2.11	: Sıralamadan önceki durum.....	39
Şekil 2.12	: Sıralamadan sonraki durum.....	39
Şekil 2.13	: Tahmini Evrensel Seçim yöntemine göre bireylerin seçilmesi.....	41
Şekil 2.14	: Döngü şeklindeki bir kromozom yapısı.....	41
Şekil 3.1	: Yüz tane düğümden oluşan bir AY uydu ağı topolojisi.....	47
Şekil 3.2	: Kromozomların uygunluk değerlerine göre dağılımı.....	50
Şekil 3.3	: Tüm linkleri aktif olan bir uydu topolojisi örneği.....	54
Şekil 3.4	: Çağrı süresince topoloji değişikliklerinin olduğu aralıklar.....	54
Şekil 3.5	: Bir yol üzerindeki tüm linklerin ÇBO gösterimi.....	57
Şekil 3.6	: Bir uydu sisteminde zamanla farklı topolojilerin ortaya çıkması.....	60
Şekil 3.7	: Dört kromozomdan oluşan Rulet Çarkı.....	63
Şekil 3.8	: Yeniden Üretim alt fonksiyonu.....	64
Şekil 3.9	: Uygulamadaki 5 yol için elde edilen Rulet Çarkı ve yol seçimi için belirlenen rastgele sayılar.....	65
Şekil 3.10	: M/M/m/m yapısına sahip m-sunuculu sistemin durum geçiş diyagramı.....	70
Şekil 3.11	: Linklerin ve yolların sunucu sayıları.....	75
Şekil 3.12	: Bir yörüngedeki uyduların ayak izleri.....	78
Şekil 3.13	: Bir uyduya ait ayak izinin hücrel gösterimi.....	80
Şekil 3.14	: Çağrının iletildiği örnek bir yol.....	82
Şekil 3.15	: Bir yol boyunca uydular arası geçiş.....	82
Şekil 3.16	: Bir uydunun belirli bir zamanda dünya yüzeyinde kapladığı alan ...	83
Şekil 3.17	: Dünyanın ve herhangi bir yörüngenin birbirlerine göre konumu.....	84
Şekil 3.18	: Kapsama alanının hesaplanması.....	87
Şekil 4.1	: İşlem sürelerinin grafiği.....	93

Şekil 4.2	: Birbirinden farklı 11 topoloji arasından seçilen yollar.....	95
Şekil 4.3	: Aktarım sayılarının karşılaştırılması.....	96
Şekil 4.4	: Aktarım sayısındaki iyileşmenin % hesabı.....	97
Şekil 4.5	: Aktarım sayısındaki iyileşmenin % hesabı (gecikme süresi kısıtı % 50 iken).....	98
Şekil 4.6	: SA ve DA işlem sürelerinin karşılaştırması.....	99
Şekil 4.7	: Uydu sayısı 60 olan bir topoloji.....	100
Şekil 4.8	: SA uygulaması sonucu elde edilen yollar.....	101
Şekil 4.9	: DA uygulaması sonucu elde edilen yollar.....	101
Şekil 4.10	: SA, DA ve D_Dijkstra işlem sürelerinin karşılaştırması.....	105
Şekil 4.11	: Kapsama alanına (konuma) göre ÇBO karşılaştırmasının grafiksel dağılımı.....	107
Şekil 4.12	: Kapsama alanına (konuma) göre ÇBO karşılaştırmasının grafiksel dağılımı (aralıklı α artımlarıyla).....	108
Şekil 4.13	: $b=1.25$ ve $\alpha=75$ değerleri için ÇBO sonucu ($ETY_j=10$ ve $m_j=65$).....	108
Şekil 4.14	: AY uzunlukları için gerekli değişkenler.....	109
Şekil 4.15	: Zamana göre ÇBO karşılaştırmasının grafiksel dağılımı.....	110
Şekil 4.16	: $t=5$ için ÇBO sonucu ($\alpha=10$, $ETY_j=10$ ve $m_j=65$).....	111
Şekil 4.17	: $f(x) = k x $ için konuma göre YTY karşılaştırmasının grafiksel dağılımı.....	115
Şekil 4.18	: $f(x) = -e^{k x }$ için konuma göre YTY karşılaştırmasının grafiksel dağılımı.....	116

TABLO LİSTESİ

Tablo 2.1	: Bazı AY sistemleri ve bunların özellikleri.....	16
Tablo 3.1	: Uygunluk fonksiyon değerlerine karşılık gelen Uygunluk değerleri	58
Tablo 3.2	: Dört farklı topolojiden elde edilen yollar.....	60
Tablo 3.3	: İlk topolojideki yollar ve bunların uygunluk değerleri	61
Tablo 3.4	: Dayanıklılık kriterinin hesabı.....	62
Tablo 3.5	: Dayanıklılık kriteriyle birlikte hesaplanan yeni uygunluk değerleri	63
Tablo 3.6	: Yeniden Üretim işlemi sonucu elde edilen jenerasyon.....	66
Tablo 4.1	: Genetik Algoritma ile Dijkstra Algoritmasının ortalama işlem süreleri	93
Tablo 4.2	: Hedef düğümlere karşılık gelen x değerleri.....	96
Tablo 4.3	: Algoritmanın statik ve dinamik durumlarında elde edilen ortalama çalışma süreleri	99
Tablo 4.4	: SA, DA ve D_Dijkstra Algoritması sonucunda elde edilen ortalama çalışma süreleri	105
Tablo 4.5	: $ETY_j=10$ ve $m_j=65$ durumunda b değerlerine göre ÇBO karşılaştırması	106
Tablo 4.6	: $ETY_j=10$ ve $m_j=65$ durumunda b değerlerine göre ÇBO karşılaştırması (aralıklı α artımlarıyla)	107
Tablo 4.7	: $ETY_j=10$ ve $m_j=65$ durumunda t değerlerine göre ÇBO karşılaştırması	110
Tablo 4.8	: $f(x) = k x $ için b değerlerine göre YTY karşılaştırması	114
Tablo 4.9	: $f(x) = -e^{k x }$ için b değerlerine göre YTY karşılaştırması	115

SEMBOL LİSTESİ

a	: Kapsama bölgesindeki karesel alanın kenar uzunluğu
AY	: Alçak yörünge
b	: Uydunun belirli zamanda gittiği yolun dünya üzerindeki izdüşümü
B	: Bir çağrının başlangıç zamanı
c	: ϵ_t 'ye bağlı (0,1) aralığındaki bir sayı
ÇBO	: Çağrı bloke olma olasılığı
ÇKN	: Çok katmanlı nöron
d_{x,y}	: x-y uyduları arasındaki linkin gecikme süresi
D_{Yi}	: i. yolun toplam gecikme süresi
DA	: Dinamik algoritma
D_Dijkstra	: Değiştirilmiş (modified) Dijkstra
DK	: Yolun dayanıklılık kriteri
e	: E (Euler) sayısı
E	: Ekvator uzunluğu
ETY_j	: j.linke bir önceki linkten geçerek gelen eski paketlerin trafik yoğunluğu
F_{Yi}	: i. yolun uygunluk fonksiyonu
g	: Yerçekimi ivmesi
GP	: Genetik Programlama
h	: Uydunun dünya yüzeyinden yüksekliği
h₁	: En alçak AY'nin yeryüzünden yüksekliği
h₂	: En alçak ve en yüksek AY'ler arasındaki mesafe
HEY	: Hayli eliptik yörünge
JOY	: Jeostasyoner olmayan yörünge
JY	: Jeostasyoner yörünge
k	: Trafik yoğunluğu hesabında kullanılan fonksiyondaki sabit sayı
k_i	: i. kromozomun uzunluğu
l	: Bir yolun sahip olduğu link sayısı
l_{ij}	: i ve j uyduları arasındaki linkin aktifliğini gösteren değer
L_{ij}	: i ve j uyduları arasındaki link
LD	: Lojiksel düğüm
MAA	: Minimum atlama algoritması
MMA	: Minimum maliyet algoritması
m_j	: j. linke ait sunucu sayısı
n	: İlgili popülasyonun nüfusu
N	: Bir yolda bulunan uydu sayısı
OY	: Orta yükseklikteki yörünge
P_{Yi}	: i. yolun bloke olmama olasılığı
R	: Dünyanın yarıçapı
S	: Bir çağrının bitiş zamanı
SA	: Statik algoritma

SDO	: Sonlu Durum Otomasyonu
SK	: Servis kalitesi
SYB	: Sanal Yol Bağlantısı
t	: ϵ_t 'ye bağlı $[1,6]$ aralığındaki bir sayı
T	: Bir uydunun yörünge periyodu
T1	: 1.topoloji
u_{i1}	: i. yolun 1.uydusu
U₁	: Örnek yol üzerindeki birinci uydu
UAL	: Uydular arası Link
UD	: Yolun uygunluk değeri
v	: Ağ topolojisini gösteren vektör
V	: Uydunun yörünge üzerindeki hızı
VSAT	: Çok Küçük Açıklıklı Terminal
w	: İlgili linkin aktif olup olmadığını gösteren ağırlık değeri
x	: Δt zamanda bir uydunun yörünge üzerinde gittiği mesafe
x_y	: Yörünge uzunluğu
x₁	: En alçak AY'nin uzunluğu
x₂	: En yüksek AY'nin uzunluğu
Y_i	: Bir paketi kaynaktan hedefine ulaştıran i. yol (kromozom)
YAL	: Yukarı ve Aşağı Link
YTY	: Bir linke kapsama alanından gelen yeni paketlerin trafik yoğunluğu
α	: Trafik yoğunluğu hesabında kullanılan fonksiyondaki sabit sayı
λ	: Paketlerin uydu ağındaki bir linke geliş oranı
Δt	: Bir yol boyunca iki uydu arasında geçen süre
ϵ	: $(0,1)$ aralığındaki bir sayı
ϵ_t	: $(0,1)$ aralığındaki bir sayı
ϵ_x	: $(0,1)$ aralığındaki bir sayı
μ	: Paketlerin uydu ağındaki bir linkten çıkış oranı

ÖZET

ALÇAK YÖRÜNGEDEKİ UYDU SİSTEMLERİNDE YENİ BİR YÖNLENDİRME ALGORİTMASININ TASARIMI

Bu çalışmada AY uydu sistemleri için yeni bir yönlendirme algoritması geliştirilmiştir. İlk bölümlerde uydular, çeşitli yönlendirme algoritmaları, optimizasyon problemleri ve özellikle genetik algoritmalar incelenmiştir.

Daha sonra yeni yönlendirme algoritması ayrıntılı şekilde açıklanmaktadır. Bu kısımlarla ilgili çalışma birkaç aşamadan oluşmaktadır. İlk aşamada statik uydu ağları üzerinde yönlendirme işleminin temeli oluşturulmuştur. Yönlendirme algoritmasında yararlanılan işlemler genetik algoritmada sıkça kullanılmaktadır. Bu işlemler için yeni bir uygunluk fonksiyonu tasarlanmıştır. Uygunluk fonksiyonu, temel olarak gecikme süresi, dayanıklılık kriteri ve çağrı bloke olma olasılığı şeklinde üç kavrama dayanmaktadır.

Statik yapılarda algoritma tamamlandıktan sonra dinamik uydu ağları için tasarıma geçilmiştir. Sürekli hareket halindeki uyduların bulunduğu ağlarda yönlendirme işleminin gerçekleşmesi için algoritma genişletilmiştir. Paketlerin ağa geliş ve ağdan çıkış oranları sabit sayı olarak düşünülerek Erlang-B formülüne göre bloke olma olasılıkları hesaplanmaktadır.

Son aşamada analitik olarak sistemin çözümü elde edilerek paketlerin gerçek trafik yoğunlukları sisteme katılmaktadır. Bu trafik yoğunluklarına göre değişen bloke olma olasılıkları yeniden hesaplanmaktadır. Böylece gecikme süresi, dayanıklılık kriteri ve bloke olma durumuna bağlı olarak en uygun yolun seçilmesi işlemi tamamlanmaktadır.

Diğer bölümlerde, yeni algoritmanın Microsoft Visual C++ 6.0 programıyla hazırlanmış olan simülasyonunun sonuçları gösterilip değerlendirilmektedir. Ayrıca çalışmayla ilgili sonuç değerlendirmeleri ve yeni çalışmaların neler olabileceği konusunda bilgi verilmektedir.

SUMMARY

DESIGN OF A NEW ROUTING ALGORITHM FOR LOW EARTH ORBIT SATELLITE SYSTEMS

In this study, a new routing algorithm for LEO satellite systems has been designed. In the first few sections, different routing algorithms, optimization problems and especially genetic algorithms have been explained.

Afterward the new routing algorithm has been clearly explained. The study about this part consists of several stages. In the first stage, the base of the routing algorithm has been constructed on static satellite networks. The operations included in this routing algorithm are frequently used in genetic algorithms. A new fitness function has been designed for these operations. This fitness function fundamentally consists of three concepts as delay, aging factor and call blocking probability of the paths.

After completion of the static algorithm, a new design has been started for dynamic satellite systems. The algorithm has been enhanced through that the routing operation can actualize on the systems comprised continuously moving satellites. Arrival and departure rates of the packets on the network have been chosen constants. Blocking probabilities have been computed by Erlang-B formula.

In the last stage, analytical solution of the system has been observed and the real traffic densities of the packets have been considered. Blocking probabilities have changed by the effect of these densities, so that the new probabilities have been evaluated. Thus the selection of the most suitable path according to delay value, aging factor and blocking situation has been completed.

In the last two sections, results of the simulation prepared by Microsoft Visual C++ 6.0 have been presented. Besides, some information has been given about the result evaluations related to this study and probable future works.

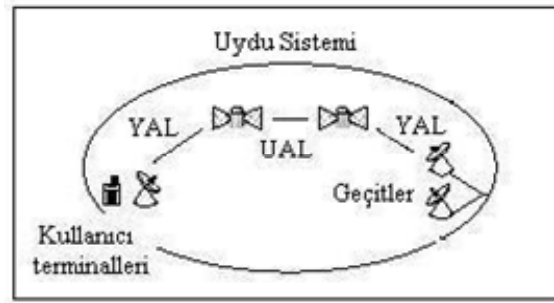
1. GİRİŞ

Uydu ağlarında kaynak düğümden hedef düğüme doğru çağrı iletilmektedir. Bu sırada birçok link üzerinden geçiş yapılır. Linklerin aktif olup olmamasına göre çok sayıda yol elde edilebilmektedir. Bu işlem için matematiksel algoritmalar çıkarılarak çözüm bulunur. Böylece bir başlangıç noktasından yola çıkılarak, uygun linkler üzerinden hedefe ulaşan yolları ortaya çıkarmak için kullanılan matematiksel işlemlerin bütünü, bir yönlendirme algoritmasını oluşturmaktadır. Özellikle şirketlerde, hükümetlerde ve kişisel alanda uydu haberleşmesinin önemi büyüktür. Ticari anlamda uydu ağlarının kullanılabilir hale gelmesinin önünde birtakım zorluklar vardır. Aşılması gereken bu zorluklardan biri etkin bir yönlendirme algoritmasının bulunmasıdır. Uydu hareketlerinin, ağ topolojisinin değişimine neden olmasından dolayı algoritmaların işlem sürecinde önceden kurulmuş bağlantıların korunması da gereklidir. Başka bir deyişle iyi bir yönlendirme algoritması, gecikme süresi açısından en avantajlı yolu bulmanın yanısıra bağlantı boyunca da yolu korumak zorundadır.

Yönlendirme algoritmaları statik ve dinamik algoritmalar olmak üzere iki kısma ayrılabilir. Statik algortmada sistemin ilk yapısında iletimi sağlayacak yollar belirlenir. Bunlar yönlendirme tablosuna aktarılır. İlerleyen zamanlarda sistemde meydana gelen herhangi bir aksaklıkta bu tablodan alternatif farklı yollar seçilir. Öte yandan dinamik yönlendirme algoritmalarında topolojide meydana gelen bir değişikliğe anında çözüm üretilmektedir. Böylece yol seçimleri her zaman yapılabilir. Sistemdeki elemanlar, işlemlerini yenilenen zamana ve duruma göre ayarlayabildiği için anlık yol seçimlerinin yapılmasına imkan verilmiş olur. Böylelikle önceki mevcut yolda bir link sorunu oluşursa, sistem yeni yolları hemen belirleyip devreye sokmaktadır.

Yönlendirme algoritmalarını başka bir açıdan da Yukarı ve Aşağı Link (YAL, Up-and-Downlink:UDL) yönlendirmesi ve Uydular arası Link (UAL, Inter-satellite Link:ISL) yönlendirmesi ile ilgili olmak üzere ikiye ayırmak mümkündür. YAL yönlendirmesinde bir kaynak ve bir hedef düğüm arasında bağlantıyı tutma işlemi düşünülür. UAL

yönlendirmesinde ise kaynaktan hedefe doğru birçok link üzerinden yönlendirme yapılarak bağlantının dışardan sonlandırılmasına engel olunmaya çalışılır. Bu yöntem öncekine göre daha çok tercih edilmektedir. Şekil 1.1’de bu iki kavram arasındaki fark daha iyi görülmektedir.



Şekil 1.1: YAL ve UAL gösterimleri (Franck ve Maral, 2000)

Yönlendirme algoritmalarında ele alınan başlıca karşılaştırma kriterleri gecikme süresi ve aktarım sayısıdır. Sinyalin bir başlangıç uydusundan başlayarak hedef uydusuna gelene kadar geçen süre gecikme süresini vermektedir. Aktarım sayısı ise sinyallerin gidiş yolu üzerinde bir uydudan başka uyduya geçme sayılarıdır. Bugüne kadar birçok metoda dayanan farklı algoritmalar tasarlanmıştır. Bunların çoğunluğu statik yönlendirme algoritmalarıdır. Algoritmalarda kullanılan kriterler farklı olabilmektedir. Birçok algoritma gecikme kriterini işleme koymaktadır. Bunun yanısıra aktarım sayısına göre değerlendirme yapan algoritmalar da mevcuttur.

Bu tezde, Alçak Yörüngedeki Uydu Sistemlerinde çalışan yeni bir yönlendirme algoritması tasarlanacaktır. Bunun için öncelikle mevcut farklı yönlendirme algoritmaları incelenecektir. Bu algoritmaların performanslarını analiz edebilmek için uygun kriterler belirlenerek analitik bir çözümü de içeren yeni algoritma oluşturulacaktır. Günümüze kadar birçok yönlendirme algoritması bulunmuştur. Bunların birbirine göre farkları göz önünde bulundurularak en etkili yöntemi ortaya çıkarmak, hangi kriterlerin kullanılacağına karar vermek çok önemlidir. Algoritmaların etkinlikleri arttıkça hem maliyet, hem zaman açısından çok fayda sağlanmış olmaktadır. Böylece önceden uzun süreler alan ve buna karşılık büyük maliyetler gerektiren işlemler daha kısa sürede, daha az maliyetle yapılarak teknolojik açıdan uygun konuma

ulařılabilmektedir. Geleceęin mobil uygulamalarının g n m zden  ok daha ileriye gideceęi d ř n l rse uyduların ve uydularla ilgili t m algoritmaların  nemi de anlařılmaktadır.      uydular artık geleneksel televizyon ve telefon end strisindeki kullanımlarını ařarak kiřisel tabanlı veri servislerine girmektedirler. Akıllı uydular sayesinde sadece iřaret  retimi deęil, anahtarlama ve tamponlama gibi iřlemler de yapılabilir. Bu řekilde teknolojik ve bilimsel geliřmeler sayesinde karasal sistemlerle oluřan rekabet ortamında uydular  ne  ıkmaya ve her a ıdan daha avantajlı olduęunu g stermeye devam etmektedir.

2. GENEL KISIMLAR

2.1. UYDULARLA İLGİLİ BİLGİLER

Uydu denildiği zaman akla bir cismin yörüngesinde hareket eden bir başka cisim gelmektedir. Elektronik, haberleşme ve uzay teknolojisinde meydana gelen en önemli gelişmelerden biri de dünya çevresindeki yörüngelere yerleştirilerek çalıştırılan uydulardır. Bunun nedeni teknolojide kullanılan bu uyduların, oldukça büyük kapsama alanına sahip olması, yayın kapasitesi genişliği, bant genişliğindeki esneklik, taşınabilirliğe imkan vermesi gibi avantajlarının olmasıdır. Özellikle farklı bölgelerdeki eş kuruluşlarla sürekli iletişimde olması gereken finans, eğitim, basın gibi sektörlerde uyduların kullanımı kaçınılmaz olmuştur. Bu sayede karasal sistemlerin tamamıyla sağlayamadığı sürekli iletişim imkanını uydular geliştirmiş bir şekilde vermektedir.

Uydular genel olarak kontrol ünitesi, güç kaynağı, anten ve alıcı gibi birimlerle oluşturulan bağlantılar sayesinde iletişim kurarlar. Uydu elemanlarını iki ana grupta toplamak mümkündür. Bunların birincisi, anten, kamera, radar ve elektronik aygıtları kapsayan ve uydunun işlem yapmasını sağlayan yararlı yüküdür. Bu kısım farklı görevlere sahip uydularda birbirinden farklıdır. Örneğin hava tahminini yapan uyduyla haberleşmeyi sağlayan uydunun bu parçaları aynı olmamaktadır. İkinci eleman ise uydunun yararlı yükünü ve tüm donanımı uzayda tutan veri yolu bölümüdür (The Tech Museum of Innovation, 2006). Bu kısım, uydunun tüm parçalarını bir arada tutan, bunlara elektriksel ve itici güç sağlayan parçadır. Uyduların çalışmaları sırasında ağ tıkanıklığına engel olmak için kontrol mekanizmaları da mevcuttur. Bu amaçla gerekli olan yük kontrollerini yapmak ve tıkanıklığı engellemek için protokoller görev yapmaktadırlar.

Uyduların birçok çeşidi ve buna bağlı olarak farklı görevleri bulunmaktadır. Uyduların kullanım alanları aşağıdaki şekilde özetlenebilir (The Tech Museum of Innovation, 2006):

- Haberleşme
- Uzaktan Algılama
- Hava Durumu
- Global Konumlanma Sistemi
- Bilimsel Araştırma

Haberleşme uyduları ilk defa 1960'ların başında kullanılmışlardır. Bu uyduların birkaç erişim çeşidi vardır:

Noktadan noktaya: Uydu topolojisindeki tüm terminaller merkeze doğrudan bağlantılı olmaktadır. Ara bağlantı görevini yapan bir kontrol istasyonuna ihtiyaç duyulmamaktadır. Başka bir deyişle tek bir kablosuz bağlantı üzerinden birbirine bağlanan iki ayrı ağ ortamı, ortak kaynakları paylaşmaktadırlar.

Noktadan çok noktaya: Tek bir merkezden birçok terminale aynı anda hizmet yapılabilir. Yayın görevi yapan uydularda bu yöntem kullanılmaktadır.

Çok noktadan noktaya: Üç veya daha fazla kablosuz bağlantı sayesinde birbirinden uzak ağlar iletişim kurmaktadırlar.

Çok noktadan çok noktaya: Bir merkeze gelen bir veya daha fazla kanal yine en az bir kanalla bağlantı kurmaktadır.

Bir uydu servisinin temel elemanları uzay segmenti ve yer segmenti olmak üzere iki kısma ayrılır. Uzay segmentinde uzay mekiği ve fırlatma mekanizması bulunur. Yer segmentinde ise yer istasyonları ve tüm uydu sisteminin ağ kontrol merkezi bulunur. Bir uyduya erişim imkanı sadece yetkili kişilere verilir. Uyduların haberleşme alanında kullanılma nedenleri şunlardır (Morgan ve Gordon, 1989):

- Uydular için mesafenin önemi yoktur. Uydu aracılığıyla birbirinden 2000 km uzaklıktaki iki yer arası mesaj iletimi ile 10 km mesafeli iki yer arasındaki mesaj iletimi aynı maliyete sahiptir.
- Uydular geniş bantlı aygıtlardır. Herbir kanalda onlarca MHz mevcuttur. Herbir kanal, uydunun kapsama alanındaki herhangi iki ya da daha fazla noktaya servis verir. Karasal radyo linklerinde ise kapasite daha azdır.
- Uydu, antenin görüş alanındaki herhangi bir istasyona işlem yapar.

-Karasal sistemlerde teknolojinin kısıtlarından dolayı her yere eşit imkanlar gitmeyebilir. Uyduların ise dağlar, şehirler, çöller, okyanuslar gibi doğal sınırlandırmaları yoktur. Böylece coğrafik olarak heterojen yapıdaki bir bölgede haberleşme imkanını oluşturmak için bütün coğrafik koşullara uygun olabilen uydu yapılarından yararlanmak gereklidir. Amerika Birleşik Devletleri'nde Hawaii ve Alaska buna birer örnektir.

-Bir uydu sistemi büyük ve küçük şehirlere aynı tip servis sunabilir. Buna karşılık karasal sistemlerde büyük ve kalabalık şehirlere gelişmiş ekipmanla, تنها ve küçük şehirlere ise daha eski teçhizatla servis verilmektedir.

-Geniş bant uydu iletişimi sayesinde daha önceden kullanılmayan linkler de tahsis edilmeye başlanarak kullanıma sunulmuştur.

-Uyduların kullanımıyla birlikte tüm dünya pazarında yeni ortak taşıyıcılar ortaya çıkmaktadır. Böylece oluşan rekabet ortamı sayesinde daha iyi hizmet verilir.

-Uydular sayesinde yeni haberleşme kavramları ortaya çıkmıştır. Örneğin bir bölgeyle ilgili verileri alan uydu, bunu daha sonra birtakım alanlarda kullanır. Deprem uyarıları bu tip bilgiler sonucu oluşturulur.

-VSAT (Very Small Aperture Earth Station Terminal-Çok Küçük Açıklıklı Terminal) gibi sistemler sayesinde uydu haberleşmesi kullanılarak veritabanlarına, şube ofislere ve yönetim bilgi sistemlerine hızlı erişim sağlanabilmektedir.

Haberleşme uydularının avantajlarının yanısıra geliştirdiği iki temel alan şu şekilde özetlenebilmektedir:

a. Geleneksel Piyasalar

1960'lı ve 1970'li yılların karasal taşıyıcıları, yer altında, yer üstünde ve denizin altında bakır kablo ile mikrodalga kuleler kullanmışlardır. 1970'li yılların sonunda mevcut mikrodalga linklerin üzerine dijital servisler eklenmiştir.

1980'lerde geliştirilen fiber optik ağlar, telefon ve veri devrelerini tutan uydu trafiğinin çoğunda kullanılmıştır. Uydular uluslar arası haberleşme piyasasına 1960'larda, telefon ve telgraf piyasalarındaki denizaltı kabloları rekabeti içerisinde girmişlerdir. 1960'lı yıllarda ve 1970'lerin başlarında uydular alternatif bir yol olarak kullanılmıştır. 1970'lerin ortalarında bölgesel uydulara acil ihtiyaç belirlemiştir. Bu uydular sayesinde

noktadan noktaya (bazen çok noktaya) linkler sağlanmaktadır. Bu linkler bir ofisle bir üretim fabrikası, deposu, vb. arası iletişimi sağlamaktadır. Uydular, çok noktalı geniş alan ağları sayesinde düşük hızlı veri iletimine de olanak sağlamaktadır.

b. Yeni Piyasalar

Uyduların geliştirdiği ilk yeni piyasalardan biri uluslar arası televizyondur. Haberleşme uydularından önce televizyon için denizaltından geçen telefon kablolarının kullanılması konusunda girişimlerde bulunulmuştur. Birkaç dakikalık bir olayın iletimi saatler almıştır. Sonuçta sessiz sinema gibi ham bir durum oluşmuştur.

Uluslar arası haberleşme uydusu bu durumu oldukça hızlı bir şekilde değiştirmiştir. Böylece yüksek kalitede, tam renkli televizyon ortaya çıkmıştır. Bu başarıdan sonra uyduların televizyon alanında tamamen, telefon/telgraf trafiğinde de kısmen rol alacağı düşünülmüştür. 1988 yılıyla birlikte uluslar arası uydu gelirinde televizyon sektörünün payı oldukça fazla olmuştur.

Uydu aracılığıyla ulusal televizyon konusundaki çalışmalar bölgelere göre farklılık göstermektedir. Örneğin Kanada ve Amerika gibi gelişmiş ağa sahip yerlerde uydular, daha uzak alanlara ulaşmak için kullanılırken Brezilya, Cezayir ve Endonezya gibi daha az gelişmiş video yapılarına sahip yerlerde ağ genişletilmesinde kullanılmıştır. Uydular, alternatif televizyon programcılığının gelişmesine de neden olmuşlardır. Uydu aracılığıyla ödemeli TV programları sayesinde kablolu televizyon sistemleri çıkmıştır. Böylece uydular yeni özelliklerin ortaya çıkması için tetikleyici olmuşlardır.

Bilgisayarlardaki yüksek hızlı veri, uydular için yeni bir hizmettir. Uydu alıcıları saniyede birçok megabit kapasiteli haberleşmeye sahip geniş bant aygıtlardır. VSAT'lar alıcı kaynaklarını daha küçük parçalara ayırarak düşük veri hızına sahip servisleri oluşturmaktadırlar. Uyduların karasal sistemlere üstünlükleri mobil iletişimde öne çıkmaktadır. Uydu ağlarındaki hızlılık, iletişime oldukça büyük avantaj sağlamaktadır. Böylece tıkanıklık gibi durumlara az rastlanmaktadır.

Bir uydunun başarısına bakmak için taşıdığı trafik miktarına ve ekonomik olarak sağladığı geri dönüşüme bakılır. Bunun dışında haberleşme uydularının avantajları aşağıdaki şekilde özetlenebilmektedir:

- Geniş bant kapasitesi
- Geniş alanda kapsama
- Mesafeden bağımsız maliyetler
- Tüm kullanıcıların (büyük, küçük, kırsal, kentsel yerlerde yaşayanların) eşit erişim olanaklarına sahip olması
- Noktadan noktaya, noktadan çok noktaya (yayın) ve çok noktadan noktaya (veri koleksiyonu) erişim çeşitlerinin tümünün olması
- Mobil uygulamalar için uygun olması
- Yeni teknolojiyle uyumluluk (örneğin dijital bilgisayarlar ile analog yüksek tanımlı TV ile)
- İhtiyacın üzerinde sunulan servis kalitesi
- Yeni servisler ve son kullanıcıya hizmet.

Buna karşın uyduların şu şekilde kısıtlamaları da bulunmaktadır:

- Uzay segmentinde normalde gereksinim duyulan yüksek başlangıç yatırımı
- Yer istasyonlarında yeni yatırıma ihtiyaç duyulabilmesi
- Kısa uydu yaşam süreleri (7-10 yıl)
- Yörünge/spektrum kalabalığı, frekans paylaşımı
- Mühendislik veya düzenleme nedenlerinden dolayı son kullanıcıya erişimin zor olabilmesi
- Kurumsal/yasal/düzenleme prosedürleri
- Bakım maliyeti ve zorluğu
- Fırlatma aygıt güvenilirliği (Morgan ve Gordon, 1989).

2.1.1. Temel Kavramlar

Haberleşme alanına uyduların girmesiyle birlikte birçok yeni kavram da eklenmiştir. Bunlardan bazıları aşağıdaki şekildedir:

Ayak izi (Footprint): Uydunun dünya yüzeyinde gördüğü bölgedir. Örneğin Türkiye’yi kapsayan bir ayak izinde bulunan bir kişi, aynı ayak izinde bulunan bir başka kişiyle o uydu aracılığıyla iletişim kurabilmektedir (The Tech Museum of Innovation, 2006).

UAL (Inter-satellite Link-ISL): Uydular bir ağ ortamında birbiriyle iletişim kurmak için direk bağlantı yollarını kullanırlar. Bu yollar UAL’ler sayesinde oluşturulur. UAL’ler, somut olarak mevcut olmadıkları halde komşu uyduları birbirine bağlama görevini görmektedirler. Uydular direk görmedikleri alanlara da bu linkler yardımıyla yönlendirme işlemi sayesinde ulaşabilirler. UAL’ler karasal sistemlere bağlı olmadan işlev görmektedirler. UAL’leri iki kısma ayırmak mümkündür (Chen, 2003):

-Düzlem içi UAL (Intra-plane ISL): Aynı düzlemdeki uyduları birbirine bağlamaktadır. Bağlı durumları sabit olduğu için bu linkler kalıcı olmaktadır.

-Düzlemler arası UAL (Inter-plane ISL): Farklı düzlemde bulunan uydular arasında bulunmaktadır. Kutuplarda işlem görmemektedirler. Uydular arasındaki mesafe ve görüş açısı değiştiği zaman yeniden düzenlenmeleri gerekir.

Uydu takımyıldızı (Constellation): Geniş bir alana yayılmış olan bir uygulama için tek bir uydu yeterli gelmeyebilir. Bir uydunun etki alanı sona erdiği zaman görevini bir diğer uyduya devretmesi gerekmektedir. Uyduların yörüngedeki pozisyonları için matematiksel algoritmalar kullanılmaktadır. Bu nedenle uyduların aynı tip ve fonksiyona sahip olanları bir araya getirilerek gerekli kapsama alanı oluşturulmuş olur. Oluşan bu uydu topluluğuna “uydu takımyıldızı” adı verilmektedir. Bunların uydular arası linklerle bağlanmış şekline de “uydu ağ takımyıldızı” denir. Uydu takımyıldızları sayesinde tek bir uydunun başaramayacağı kapsama imkanını birçok uydu bir araya gelerek sağlayabilmektedir. Bunların kullanım alanları genellikle ses ve veri iletimi şeklindedir. Özellikle 1990’lı yıllarda uydu takımyıldızlarının kullanımı oldukça yaygınlaşmıştır.

Hücre (Cell): Uydu ayak izleri parçalara ayrıldığında herbir parça bir hücreyi göstermektedir. Bu durumun amacı, bant genişliği ve frekanstan kazanç sağlamaktır.

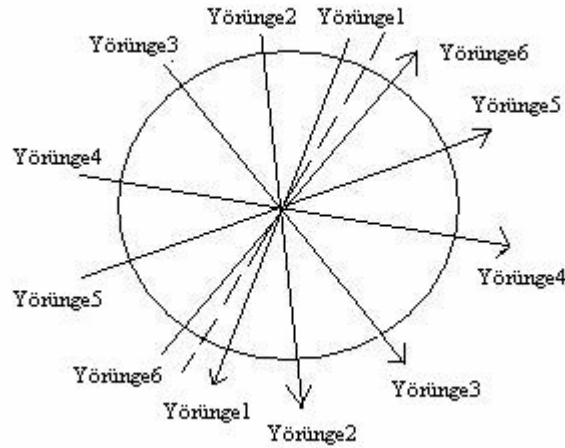
Yörünge Periyodu (Orbital period): Bir uydunun dünya çevresinde bir tam dönüşü için kendi yörüngesinde geçirdiği süredir.

Sistem Periyodu (System period): Bir uydu takımıyıldızında uyduların bulundukları konum, belirli bir zaman sonra yine aynı yer olur. Birkaç gün olabilen bu süreye Sistem Periyodu denir. Bu değer, yörünge periyoduyla dünya periyodunun (23 saat, 56 dakika, 4.1 saniye) en küçük ortak çarpanına eşittir.

Sabit Uydulu Hücresel Kaplama (Satellite-fixed Cell Coverage): Bu tür sistemlerde uydunun anteni sabit olup ayak izi ve buna bağlı olarak da ayak izindeki hücreler dinamiktir. Böylece bir çağrı için uydular arasında aktarım işlemi olmaktadır.

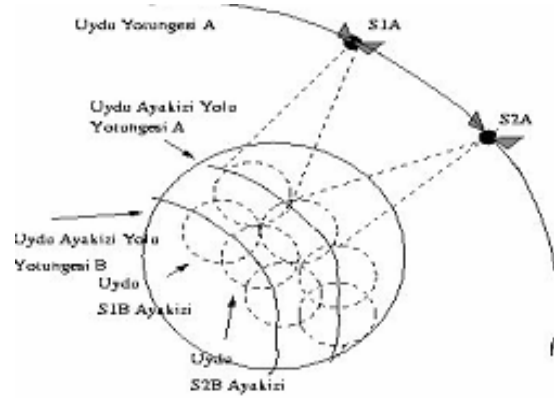
Sabit Hücreli Kaplama (Earth-fixed Cell Coverage): Uyduların antenleri hareket ettirilebilmektedir. Bir alanı kapsayan uydu belirli bir açının altına düşüncüye kadar antenin o bölgede işlem yapmasını sağlar. Sınır açuya gelindiği zaman anten bir sonraki bölgeye yönlendirilir.

Ayraç Çizgisi (Seam): Şekil 2.1’de görüldüğü gibi kuzey kutbundan bakıldığında Yörünge 1 ile Yörünge 6’nın hareket yönlerinin zıt olduğu ortaya çıkmaktadır. Buna göre bu yörüngelerdeki uydular arasındaki iletişim kopukluğunu göstermek için kullanılan sanal çizgiye “ayraç çizgisi” denir.



Şekil 2.1: Ayraç Çizgisi

Kaplama Yolları (Streets of Coverage): Aynı yörüngedeki uyduların ayak izlerinin bir araya gelmesiyle oluşan yola denir. Şekil 2.2’de görüldüğü gibi farklı yörüngelerin kaplama yolları birleşerek global alanda kapsama oluştururlar.



Şekil 2.2: Kaplama Yolları

Yükseltme Açısı (Elevation angle): Dünya yüzeyinden geçen teğetle yüzeyden uyduya uzanan çizgi arasındaki açıya denir. Uydu yeryüzünden uzaklaştıkça bu açı küçülür, bununla birlikte uydudan gelen sinyallerin gücü de az olur.

2.1.2. Yörünge Çeşitleri

Uydular, bulundukları yörüngelere göre farklılık göstermektedir. Dünya yüzeyinden 5000-8000 km yükseklikte bir tane, 15000-25000 km yükseklikte de bir tane olmak üzere toplam iki Van Allen radyasyon kemeri bulunmaktadır. Bu kemerlerin bulundukları konuma uydu yerleştirmek çok güç olduğu için uydular bunların dışındaki farklı yörüngelere yerleştirilirler. Böylece bu kemerlerin alt ve üst kısımlarındaki radyasyonsuz bölgeler uydu yörüngeleri için seçilen yerler olmaktadır. Aslında daha fazla yörünge tipi olmakla birlikte temel olarak 4 çeşidi düşünülmektedir. Bu yörünge çeşitleri de iki ana kısım içerisinde yer almaktadır:

Jeostasyoner yörünge (JY/GEO)

Jeostasyoner olmayan yörünge (JOY/NGEO).

1) Jeostasyoner yörünge

Günümüzde uyduların, özellikle de haberleşme uydularının çoğunluğu bu yörüngede bulunmaktadır. Bu yörüngedeki uydular ses, veri ve video servislerinin yanısıra sabit bir alana servis sunma ve televizyon yayını gibi görevlerde de bulunmaktadır. Ekvatordan uzaklığı 36000 km olan JY'ler yükseklikten dolayı dünyanın 1/3 kadarını görmektedirler. Yani ekvator hizasında bir JY'de bulunan uydudan, dünya yüzeyinin maksimum %30'u görülebilmektedir. Bu yörüngede bulunan bir uydu dünyayla birlikte hareket ettiği için dünyaya göre hareketsiz görünmektedir. Bu nedenle böyle bir uydu her zaman dünyanın aynı bölgesini görür. Uyduların bir sefer dönmeleri için geçen zaman 24 saattir. JY sistemlerinin bant genişlikleri, diğer yörünge sistemlerine göre büyüktür. Bundan dolayı çift yönlü veri, ses ve geniş bant servislerde kullanımları yaygındır. Yüksek yoğunluktaki trafik yükünü kaldırmaları da mümkündür. JY'lerin dezavantajları ise büyük gecikme süresine (250 ms) sahip olmaları, yüksek iletim gücü gereksinimleri, büyük anten ihtiyacı ve uyduların yerleşim maliyetinin yüksekliğidir. Ayrıca karasal antenle uydu arasında görüş alanı dahilindeki haberleşme linklerinin olması gerekmektedir. Özellikle gecikme süresinin fazlalığı yüzünden jeostasyoner uydu, gerçek zamanlı interaktif trafikte etkin olmamaktadır.

2) Jeostasyoner olmayan yörünge

JOY'lerin dünyadan uzaklıkları 700-20000 km arasında değişmektedir. Bunların ayak izi JY'lere göre oldukça küçüktür. Ayrıca bu uyduların dünyaya göre bağlı hareketleri sürekli değişir. Bundan dolayı haberleşmenin kesintisiz ve büyük bir kapsama alanında olması için aktarım (hand-off) işlemi gerekmektedir. Bu işlem, görüş alanından çıkan bir yörüngenin yerine başka bir yörüngenin gelmesini ifade etmektedir.

JOY'lerin düşük güç ve küçük anten ihtiyaçları ile daha kısa gecikme süresine sahip olmaları ağ uygulamaları için etkin olmalarını sağlamaktadır. Bu durum özellikle erişim ağları ve interaktif iletişim için faydalıdır.

JOY'ler üç gruba ayrılır:

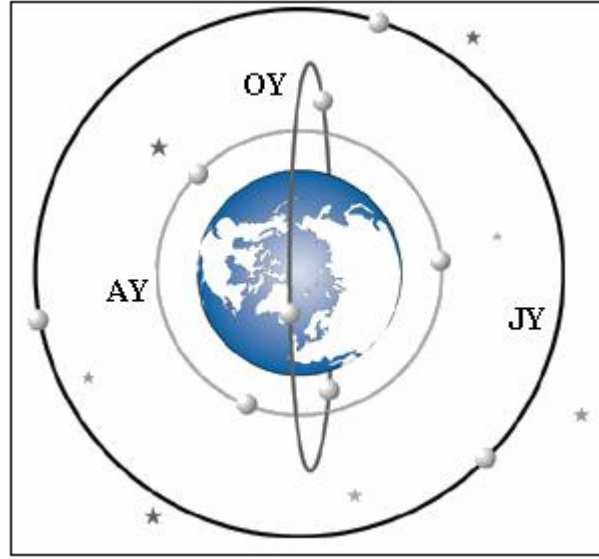
Alçak yörünge (low earth orbit=LEO-AY)

Orta yükseklikteki yörünge (medium earth orbit=MEO-OY)

Hayli eliptik yörünge (highly elliptical orbit=HEO-HEY).

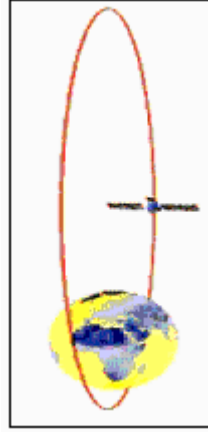
AY'ler 700-2000 km yükseklikte konumlanırlar ve dünyanın çevresinde bir sefer dönmeleri için geçen süre yaklaşık olarak 100 dakikadır. Bu sürenin kısa olmasının nedeni, AY'deki uyduların yere çok yakın olmasından dolayı yerçekiminin etkisiyle atmosferden çıkmamaları için çok hızlı dönmelerinin gerekmesidir. Bu hız yaklaşık olarak 27500 km/saat'tir. Yerçekimi etkisinden ve batarya sürelerinin kısa olmasından dolayı bu uyduların yaşam süreleri diğer yörüngedeki uydulara göre daha kısa olabilmektedir. AY uyduların dünyaya yakın olmalarının avantajı ise dünya yüzeyinin görüntülerini net şekilde alabilmeleridir. Ayrıca bu yakınlıktan dolayı kullanılan antenlerin ve diğer makinelerin büyük olmasına gerek duyulmamaktadır. Maliyet açısından avantaj sağlayan bu durumun zor tarafı da uyduların tüm dünya üzerinde bir kapsama alanı oluşturması için sayılarının fazla olmasının gerekliliğidir. Bu uyduların gecikme süreleri de 20 ms'dir. AY'deki uydular dünyanın belirli bir bölgesine 1'den fazla uydu gelecek şekilde yerleştirilirler. Bu durum da uydu ağındaki iletim için faydalıdır, çünkü uyduların birbirine aktarımı sayesinde bir çağrının bloke olma ihtimali azaltılmış olur.

OY'ler 13000 km yükseklikte olup gecikme süreleri ~100ms'dir. Uyduların dünya etrafında bir sefer dönmeleri için geçen süre 6 saattir (Rad, 2004). OY'deki birçok uydu, telefon servislerinde kullanılmaktadır. Karmaşıklık ve maliyet açısından avantajlıdır. Büyük kapasite imkanı sayesinde AY sistemleri gibi ses ve veri iletiminde kullanılabilir. Uyduların yaşam süreleri fazla değildir ve karasal sistemlere göre uydulardan giden işaretler zayıf olmaktadır. Şekil 2.3'te bu uyduların JY ve AY uydu sistemleriyle birlikte görünüşü bulunmaktadır.



Şekil 2.3: Bazı yörünge çeşitlerinin görünüşü (Inetdaemon.com, 2005)

AY'de ve JY'de bulunan uyduların çembersel hareketine karşılık HEY'lerin hareketleri eliptiktir. Şekil 2.4'te gösterilen bu yapıya göre bir uydu dünyaya yakın yerlere gelince çok hızlı hareket ederken dünyadan uzaklaştıkça da hareketini yavaşlatır. Bu özellik, Kepler'in ikinci kanunundan çıkarılabilmektedir. HEY'deki uyduların dünya etrafındaki bir tam dönüşleri 1 günden uzun sürmektedir. AY'deki uydulardan farklı olarak bu tip yörüngede sürekli bir kapsama imkanı sağlanmamaktadır. Ama uydular kalabalık yerlerde daha uzun zaman geçirecek şekilde konumlandırılmaktadır. Bu sistemlerde genelde tüm uyduların HEY'de olması değil de diğer yörüngelerle hibrit olacak şekilde bulunması sağlanmaktadır. Yani bir HEY sisteminde OY ve JY uydu sistemleri de yer alabilmektedir. HEY yapısı, kalabalık bölgeler üzerinde uyduların çok vakit geçirmesini sağladığı için diğer yörünge çeşitlerine göre daha az uyduya gereksinim sağlamaktadır. Bununla birlikte görüş açısı da daha iyi olmaktadır. Kalabalık bölgelerdeki kapsama olanağına karşılık yukarıda bahsedildiği gibi genel olarak kapsama sürekliliğinin bulunmaması dezavantajdır (Compassrose, 1999).



Şekil 2.4: HEY'deki bir uydunun görünüşü (The Tech Museum of Innovation, 2006)

2.1.2.1. Alçak Yörüngelerin Seçilme Amacı

Yukarıda bahsedildiği gibi AY'deki uyduların diğer yörüngedeki uydulara göre farkı dünyaya yakın olması, bu nedenden dolayı da gecikme sürelerinin kısa olmasıdır. AY'deki uyduların bir başka özelliği de yüksek sistem kapasitesini sağlamasıdır. Ayrıca bu uydular, ticari amaçla kullanıldıkları zaman önceden tahsis edilmiş olan frekans bantlarının yeniden kullanılmasına olanak tanımaktadırlar. Böylece maliyet açısından da oldukça iyi sonuçlar çıkmaktadır. Bu nedenle AY'deki uydular gittikçe yaygınlaşmaktadır. Günümüzde üzerinde araştırma yapılan uydu sistemlerinin çoğunluğu AY'de bulunmaktadır. Gelecekte daha önemli konuma geçerek, daha çok kullanılabilir bu uydu yörüngesi, üzerinde en çok çalışılan uydu yörüngeleridir. Bu yörüngelerde bulunan ve yaygın olarak bilinen uydu sistemleri ile bazı özellikler Tablo 2.1'de gösterilmektedir (Taleb ve diğ., 2005). Burada sistem özelliklerinden biri olan yükseklik değeri, yeryüzüne göre sistemin bulunduğu konumu göstermektedir. Yükselme açısı; bir uydu anteninin, düşey doğrultuda bulunan ve başka bir uyduya doğru oluşturduğu bakış açısını ifade etmektedir. Kapsama ise yeryüzünün % kaçının kapsandığını belirtmektedir.

AY'deki uyduların incelenmesi sırasında karşılaşılan iki ana problem, link tahsisi ve yönlendirme problemleridir. Karasal sistemlerle karşılaştırıldıkları zaman AY uydu topolojilerinin zamanla değişmesinden kaynaklanan zorluklar mevcuttur. Uydular arasındaki görünürlük durumu zamanla doğru orantılı olarak değişmektedir. Bu nedenle iyi bir yönlendirme algoritmasının oluşturulması güçleşmektedir. Bundan dolayı

AY'deki uyduların yönlendirme konusu, buna bağlı olarak da geliştirilen algoritmalar büyük önem taşımaktadır. Etkin ve iyi bir algoritmanın tasarımı çok önemli bir yer tutmaktadır. Algoritmalar sayesinde büyük oranda veri, düşük gecikme süreleriyle kaynak uydudan hedef uyduya doğru gönderilmektedir. Bu gönderim işlemi sırasında çağrının bloke olmamasına dikkat edilerek gerekli tedbirler alınmalı, hedefe ulaşım yolunda harcanan sürenin de kısaltılmasına çalışılmalıdır. Aksi halde oluşturulan yönlendirme algoritmasının hiçbir avantajı olmayacak ve gelişmiş bir işlem döngüsü kurulamayacaktır.

Tablo 2.1: Bazı AY sistemleri ve bunların özellikleri

Sistem	Kuruluş	Uydu sayısı	Yükseklik (km)	En küçük yükselme açısı	Kapsama (%)	UAL'ler
NeLS	Japan NiCT	120	1200	20	79	Var
Iridium	Motorola	66	780	8.2	100	Var
Teledesic	Teledesic	288	1375	40	100	Var
Globalstar	Global Star	48	1406	10	83	Var
Skybridge	Alcatel	80	1469	10	86	Yok
Celestri	Motorola	63	1400	16	73	Var

2.2. ÇEŞİTLİ YÖNLENDİRME ALGORİTMALARI

AY uydu sistemleri için tasarlanmış olan birçok yönlendirme algoritması mevcuttur. Bunlardan başlıcaları şöyle özetlenebilmektedir:

2.2.1. Veri Paketi (Datagram) Yönlendirme Algoritması

Bu algoritmada ana amaç, minimum iletim gecikmesine sahip yollar oluşturmaktır. Burada uydulardaki anahtarlar IP anahtarları gibi düşünülmektedir. Bu algoritma, dağıtılmış algoritma olduğundan dolayı her bir paket için yönlendirme kararı diğerlerinden bağımsızdır. Tıkanıklık ve ağın durumu hakkındaki kararlar da yerel olarak alınmaktadır. Algoritma veri paketi tabanlı olduğu için, bağlantı-tabanlı algoritmalarda gereken bağlantı yolu aktarımına burada ihtiyaç duyulmaz.

Bu yöntemde yörünge ve uydu sayısı belli olan bir uydu ağı ele alınır. Her bir uydunun konumu enlem ve boylam şeklinde verilir. Uyduların bağlantı yapısı ve Dünya

çevresindeki deterministik hareketleri, paket trafiği için yönlendirme algoritmalarının etkinlik oluşumunu kolaylaştırır. Algoritmada aradaki anahtarlama süreleri ihmal edilebilecek kadar küçük olduğundan dolayı uçtan uca gecikmeyi sadece uydular arasındaki iletim gecikmesi oluşturur. Bir uydu kendisine gelen paket için ilgili şemaya bakarak en uygun yolu seçer. Burada ana amaç, iletim gecikmesini minimize etmektir. Teorem ve varsayımlar aracılığıyla uydular arasındaki konum ilişkilerinden yola çıkılarak atlama sayıları ve minimum iletim gecikme yolu hesaplanır.

Bir uyduya paket geldiği zaman uydu bunun minimum gecikme yolundan geldiğine karar verirse sonraki adıma geçer. Bu adımda iki aşama söz konusudur:

1)Yön Tahmin Aşaması: Öncelikle o andaki uydu ile hedef uydunun konumları hesaplanır. Bunun için uydu numarası ve bir düzlemdeki toplam uydu sayısından yararlanılır. Daha sonra tüm UAL'lerin aynı uzunlukta olduğu düşünülerek atlama yolları bulunur. Uyduların, ayrıca çizgisinin aynı ya da farklı tarafında bulunmalarına göre yatay ve dikey olarak atlama sayıları hesaplanır. Bu yollardan toplam minimum atlama sayısına sahip olanın yatay ve dikey atlama sayıları saklanır. Minimum atlama yolunun yatay ve dikey atlama sayıları tablolardan bulunur.

2)Yön İyileştirme Aşaması: Bu aşamada uydu, paketi bir sonraki aşamada hangi komşusuna göndereceğini kararlaştırır. Bu nedenle yön tahminiyle karar planları birlikte kullanılır. Uyduların hangi bölgede olduklarına göre, yatay ve dikey atlamalara göre paketin iletileceği taraf bulunur. İletim kararları planlar yardımıyla yapıldığı için ağdaki uydular bu planları saklarlar.

Bu algoritmada tıkanıklık kontrolü için uydulara bakılması gerekmez. Bu bilgi çıkış tamponunun durumuna göre belirlenir. Yani çıkış tamponunda belirli bir limitin üzerinde yük varsa tıkanıklık oluşmuş demektir. Bir uydunun başarısızlığı durumunda paketler ikinci bir yöne iletilirler. Sadece kutup bölgesinde paketler geriye de gönderilebilirler.

Bu algoritmanın performans ölçümü için üç işlem yapılmıştır:

- 1) Algoritmada oluşan yolların uzunlukları Bellman En Kısa Yol Algoritmasındakiyle karşılaştırılmıştır. Algoritmada paket işleme karmaşıklığı $O(1)$ 'dir ve uydu sayısına bağlı değildir. Bellman'ın algoritmasında ise bu değer, toplam uydu sayısı olan $N \times M$ değerine bağlıdır ve ortalama $O(N \times M \log(N \times M))$, en kötü durumda da $O((N \times M)^3)$ şeklindedir. Bu algoritmada daha kısa paket işleme gecikmesi elde edilir.
- 2) Yön İyileştirme aşamasının etkisine bakılır. Bu aşama sayesinde daha küçük ortalama sapma değerleri elde edilmektedir. En kötü durumda bu değer %0.3'ten azdır. Buna karşılık sadece yön tahmin aşaması kullanılırsa bu değer %8.1'den fazla olur.
- 3) Uydu başarısızlıkları karşısında algoritma incelenir. Bir uyduda hata çıkarsa bu uyduya bağlı tüm minimum iletim gecikme yolları yeniden oluşturulur. Hatanın olduğu uydunun enlemine göre farklılık vardır. Örneğin kutup bölgesindeki ortalama sapmayla kutupların dışındaki farklıdır (Ekici ve Akyıldız, 2001).

2.2.2. Sonlu Durum Otomasyonu (SDO) – Tabanlı Yönlendirme Algoritması

Bu algoritmada kullanılan SDO'da herbir durum, AY uydu ağının sistem periyodundaki eşit uzunluklu aralıkları gösterir. Yani sistem periyodu eşit zaman aralıklarına bölünerek herbir aralık, bir duruma karşılık getirilmektedir. Başka bir deyişle zamanla değişen bir topolojiye sahip olan dinamik uydu sistemi, birçok statik topolojiye bölünerek herbir statik topoloji diğerinden ayrı düşünülmektedir. Algoritmayı oluşturan işlemler, çeşitli link tahsisi ve yönlendirme aşamalarını içermektedir. Bu link tahsisi de SDO-tabanlı olmaktadır. Uyduların birbirleriyle herbir aralıktaki görünürlükleri sonucunda link tahsis tabloları elde edilir. SDO'nun herbir aralığındaki görünürlük matrisi, uyduların kararlı kapsama alanlarını gösterir. Ayrıca link tahsisi için tavlama benzetimi (simulated annealing) gibi bir optimizasyon yöntemi kullanılarak problem basit hale getirilerek çözümlenir. Bu algoritmada yayılma gecikmesi, kuyruklama ve yayılma gecikmeleri ile tüm uyduların kaynaktan hedefe doğru iletim zamanlarının toplamı şeklinde hesaplanmaktadır (Chang ve diğ., 1996a).

2.2.3. Lojiksel Topoloji Tabanlı Yönlendirme

Bu çalışmada fiziksel yapı üzerine statik lojiksel topolojiyi oluşturarak bu probleme bir çözüm getirilmektedir. Algoritma sonucunda optimale yakın en kısa yollar oluşturulur. Uyduların değişebilen yapısına karşın bu algoritma, AY'deki yapılar üzerinde statik lojiksel bir topoloji oluşturur ve dinamik özelliği, bu topoloji üzerinde çalışan

yönlendirme işleminden gizler. Dünyanın dönüş etkilerinin önemsiz olduğu yapıda yörüngeler kutupsal ve çemberseldir. M tane yörünge 0'dan $M-1$ 'e kadar numaralandırılmıştır. Herbir yörüngeye N tane uydu dizilmiştir. Herbir uydu UAL'lerle en yakın 8 komşusuna bağlıdır, bunların 4'ü aynı yörüngede, 2'si komşu yörüngelerde, 2'si de bir sonraki komşu yörüngelerdedir.

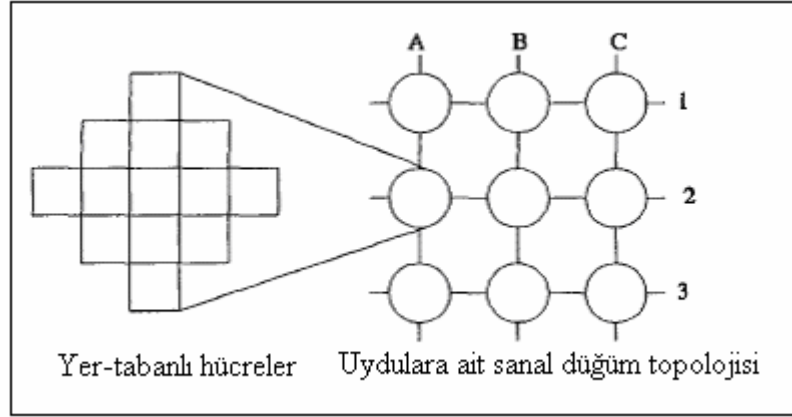
Uydular statik lojiksel topolojiye belirli bir düzene göre yerleştirilir. Yörüngeler kuzey ve güney kutbunda olmak üzere iki parçaya ayrılır. Buralarda uydular belirli bir periyot boyunca bir lojiksel düğümü (LD) oluşturur. En kötü durumda bu periyot boyunca birbirinden farklı üç statik topoloji elde edilir.

Topoloji statik olunca statik yönlendirme kolaylıkla uygulanabilir. Herbir düğümde yönlendirme bilgileri saklanmaktadır. Bir düğüm, paketi aldığı zaman direk hedefe giden UAL'e gönderir veya yönlendirme tablosundan bakarak nereye göndereceğini bulur. Modelin gerçekleşme aşamasındaki tek dezavantaj, yönlendirme tablolarının saklanma ihtiyacından dolayı oluşan karmaşıklıktır. Herbir uydu, sırasıyla N tane LD'yi göstermektedir. Herbir LD'de üç adet yönlendirme tablosu kümesi bulunmaktadır. Sonuç olarak herbir uydunun da en fazla $3N$ (N : LD sayısı) tane yönlendirme tablosu mevcuttur. Tabloların fazlalığı uygulamayı zorlaştırmaktadır. Uydulardan en üst seviyede yararlanmak ve bu karmaşıklığı engellemek için algoritmanın uygulama aşaması gerçekleştirilmiştir. Fiziksel ve lojiksel topolojiler bu amaca göre düzenlenmiştir.

Statik lojiksel topoloji üzerinde hesaplanan yönlendirme algoritmaları nedeniyle, LD'ler arasındaki en kısa yol, statik uzunlukta bulunur. Uyduların konumlarındaki sapmalardan dolayı bulunan bu uzunluklar gerçek uzunluk olmayabilir. Çeşitli hesaplamalarla düzlemler arası UAL uzunluğu hesaplanır. Ayrıca yatay lojiksel linkler birkaç çeşide ayrılarak bunların uzunlukları da hesaplanır. Uydularla LD'ler bire bir eşleşmediği zaman bulunan en kısa yollar yanıltıcı olabilir. Bu nedenle herbir periyotta farklı lojiksel topolojiler ele alınır. Fiziksel ve lojiksel konum farkının belirli bir aralıkta değiştiği sürece yolun optimalitesi kontrol edilir. En kötü durumda bile algoritma, yollar arasındaki ortalama uzunluk farkı açısından optimale yakın sonuçlar üretmektedir (Hu ve Li, 2001).

2.2.4. Coğrafi Tabanlı Paket Yönlendirmesi

Dünya yüzeyi hücre kümelerine bölündüğü zaman her bir hücreye servis yapan uydu düğüme lojikel bir adres verilir. Farklı fizikel uydular, sabit lojikel bir düğümü kapsar. Bu sanal düğüm kavramı Şekil 2.5'te görölmektedir.



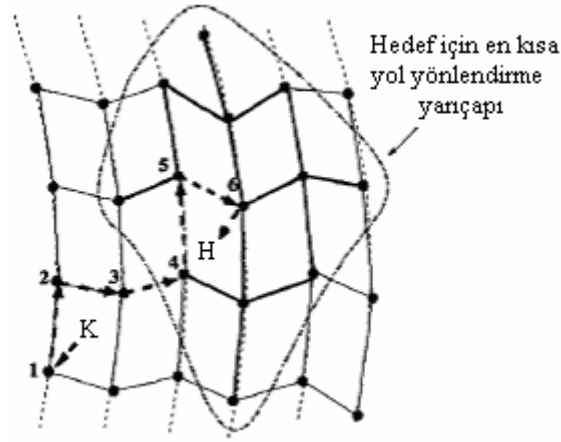
Şekil 2.5: Lojikel bir ağ topolojisi (Henderson ve Katz, 2000)

Terminalle uydu sanal düğümü arasında bire bir ilişki olmadığı zaman dinamik yönlendirme algoritmasına ihtiyaç vardır.

Bir uydu ayak izinin çok sayıdaki küçük hücrelere bölünmesinden sonra bu hücrelerden oluşturulan yarı-sabit ayak izleri oluşturulabilir. Böylece sistem uygunluğu maksimize hale getirilmiş olur.

Bu algoritmada bir uydu, komşularının ve kendinin hedefe uzaklığını tablodan çıkarır. Hedefe en yakın komşusuna paketi yönlendirir. Yönlendirme protokolünün performansını ölçmek için rastgele iki nokta alınır. Bunlar arasında iki paketin iletimine başlanır. İlk paket, iletim gecikmesini minimize eden global bir en kısa yol algoritmasıyla yönlendirilir. İkinci paket dağıtık bir protokol ile yönlendirilir. Performans kriteri olarak iki kavram kullanılır. Bunların ilki, sağlamlıktır. Bunun anlamı, paket düşmelerinden ve kutuplar gibi ölü noktalara yönlendirmekten kaçınma özelliğidir. İkinci kriter, gecikmeyi düşürmedir. Bu da optimalle karşılaştırıldığı zaman coğrafi tabanlı yolun gecikmesini azaltmaktır.

Algoritmada doğru düğümü bulmak için yönlendirme işlemi yapılırken yerel olarak en kısa yol algoritması kullanılabilir. Bu amaçla basit bir link durumu (link-state) yönlendirme algoritması çalıştırılır. Paket numaraları sayesinde bir paketin iki kere gönderilmesine engel olunur. Herbir uydunun kendisini merkez alan bir haritası mevcuttur. Örneğin Şekil 2.6'daki uydu sisteminde 6 numaralı uydu hedef uydu olarak merkezde bulunmak üzere en kısa yol yönlendirme yarıçapı bulunmuştur. Bir paket coğrafi bilgiye göre 4. uyduya gönderilir. Daha sonra yerel olarak en kısa yol yönlendirme bilgisi kullanılarak hedefi gösteren 6. uyduya ulaştırılır.

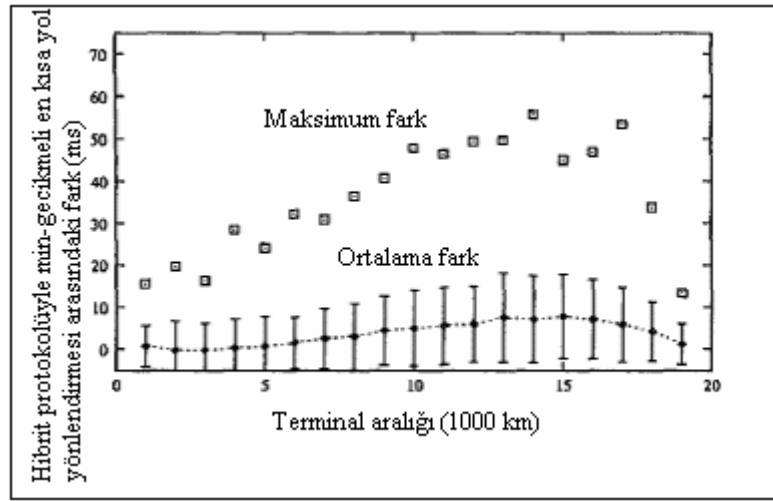


Şekil 2.6: Coğrafi paket gönderimi ve yerel (locally-scoped) en kısa yol yönlendirmesi (Henderson ve Katz, 2000)

Bir terminalin yükseltilmiş halinde yer alan uydular yönlendirme yarıçapında bulunurlar. Topolojide eksiklikler olsa bile yarıçap devam eder. Büyük enlemler kısımlarda UAL'ler aktif olmayabilir, bu nedenle paketler aynı düzlemdeki daha düşük enlemler bir uyduya yönlendirilir. Bu şekilde hedefin dışında birçok atlama yapılmış olur. Böylece yarıçapın büyütülmesi gerekir. Yarıçap büyüyünce her bir uydudaki yönlendirme bilgisi de artmış olur. Bu nedenle özel bir yönlendirme bölgesi oluşturulmuştur. Bunun için kutup bölgesi iyi bir dinamik tanımla belirtilmelidir. Kutup bölgesindeki tüm uyduların en az bir tane işlem dışı UAL'si olacak şekilde tasarım yapılır. Kutup bölgesindeki ve köşelerdeki uydular yardımıyla yönlendirme işlemi yapılır. Kutup bölgesine özel yönlendirme yarıçapı tanımlanmasıyla birlikte yüksek enlemlerdeki uydularda bulunan durum sayısı ve mesaj oranı artar.

Ayraç çizgilerindeki UAL'lerin link bulması ve senkronizasyonu düzlem içi UAL'lere göre zordur. Ayrıca bu bölgede ağıın şekli bozukluğa uğramıştır. Farklı yönlerdeki iki yörüngenin çakışması diğer yörüngelere göre daha alt enlemlerde olmaktadır. Ayrıca uydu bozulmalarından dolayı da buna benzer açıklar oluşmaktadır. Sonuç olarak ayraç çizgisi olduğu zaman coğrafik tabanlı yönlendirmenin sağlamlığı konusunda algoritmada başarı elde edilememektedir.

Bu algoritmada ayraç çizgisindeki başarısızlığa rağmen gecikme performansı yeterli ölçüdedir. Şekil 2.7'de hibrit protokolü ile minimum-gecikmeli en kısa yol yönlendirmesi arasındaki ortalama ve maksimum gecikme farkları gösterilmiştir. Şekilde görüldüğü gibi coğrafik algoritmanın, 10 ms'den daha kısa bir süre geride kaldığı görülür. Buna karşılık maksimum gecikme farkları oldukça büyüktür (Henderson ve Katz, 2000).



Şekil 2.7: Coğrafik gönderim ile minimum-gecikmeli en kısa yol yönlendirmesi arasındaki ortalama ve maksimum gecikme farkları (Henderson ve Katz, 2000)

2.2.5. Minimum Atlama/Maliyet Algoritması (MAA, MMA)

MAA, kaynak-hedef uydu çiftleri arasından minimum atlama sayılı yolu bulur. Herbir linkin maliyetini 1 düşünerek Dijkstra en kısa yol algoritmasıyla bunu örneklemek mümkündür. MMA ise minimum UAL toplam maliyetine sahip bir yolu bulur.

UAL'teki boş kanallara x denirse UAL'in maliyeti $\frac{1}{x}$ şeklinde hesaplanır (Lui ve diğ., 2000).

2.2.6. Ağ Algoritması

Maksimum-kullanım maliyetinin minimum olanına sahip bir minimum-atlama yolunu bulur. Kısaca bu algoritma, verilmiş kaynak-hedef uydu çiftleri arasından tüm minimum-atlama yollarını bulur. Bunlar arasından en az tıkanık olanı seçer.

Bu algoritma iki kavram üzerine odaklanmıştır:

- 1) uydu topolojilerinin düzenliliği
- 2) bir yol üzerinde en tıkanık UAL'ler. Bu UAL, yolun aktif noktası olarak adlandırılır.

Ağ algoritması iki kısımdan oluşur. Öncelikle kaynak ve hedef arasındaki minimum-atlama yolları bulunur. Sonra da her bir yolun maksimum-kullanım maliyeti bulunur, bu maliyetler karşılaştırılır ve bunlar arasından minimum olanı seçilir. Bu yollar arasından uygun olanı bulunamazsa yeni bağlantı kurulmaz ve yeni çağrılar bloke olur. Bu yolların dışındaki bir yola ihtiyaç duyulma durumundan dolayı başka algoritmalar tercih edilir (Lui ve diğ., 2000).

2.2.7. Tahmini Servis Kalitesi (SK) Yönlendirme Metodu

Algoritma, sistem tarafından servisi yapılan kullanıcı sayısını arttırırken her bir kullanıcının SK gereksinimlerini korur. AY uydu topolojisinin deterministik özelliği ve kullanıcı konum bilgisi kullanılarak kısa bir zaman süresi için UAL'lerdeki kullanıcı trafik yükü tahmin edilir. Bunun yardımıyla bir bağlantı için tıkanıklıktan uzak çoklu yollar bulunur. Böylece karasal uydu sistemlerine göre farklı bir yol izlenmiş olur. Karasal sistemlerde çağrı başlangıcında belirlenen yol, zamanla linklerdeki yük artışıyla birlikte SK'yı sağlayamayabilir. Bu algortmada amaç ise her bir çağrının gereksinim duyduğu SK ihtiyacını korurken toplam çağrı sayısını maksimize etmektir. Buna çözüm, linklerdeki tıkanmalar karşısında eskisinin başarısız olma durumuna karşı yeni çağrılarda yeni yol bulmaktır.

AY'lerde tıkanıklıkları engellemek için UAL'lerdeki bant genişliğinin bir bölümü, aktarılan çağrılar için ayrılabilir. Bunun da dezavantajı ayrılan kısmın kullanım dışı

olmasıdır. Bu dezavantajı kaldırmak için AY uydu topolojisinin önceden tahmin edilme özelliğinden yararlanılır. Önceden trafik yoğunluğu tahmin edilerek sadece gereken kadar bant genişliği tahsis edilir.

Paketlerin iletiminde iki tür gecikme vardır. Bunların birincisi sabit gecikmedir ve karasal sistemlerdeki iletim ve gönderme gecikmeleridir. Diğer ise değişken olan kuyruklama gecikmesidir. Bu da uydu ağları arasındaki iletim gecikmesiyle paketlerin karşılaştığı kuyruklama gecikmesidir. AY sistemlerinde garantili iletişim doğrultusunda SK'ni sağlamak için sabit gecikmeler dikkate alınmalıdır.

Algoritmanın performansını ölçmek için yeni çağrının bloke olma olasılığı ve paket düşme olasılığı olmak üzere iki kriterden yararlanılmıştır. Bir paket, çağrının gecikme gereksinimini aşıyorsa düşer. Link dolu olduğu zaman paket süreyi geçirmişse düşer. Yapılan teste göre herhangi bir tıkama (bloke) durumunda paketin ortalama düşme olasılığı %2'dir. Gecikme ihtiyacı arttığı zaman servis yapılan çağrı sayısı da artar. Bloke olma olasılığı artınca da sistemdeki servis sayısı azalır.

Bu algoritmanın dayandığı protokol, tahmini olmayan protokole göre daha yüksek bloke olma olasılığına sahiptir, çünkü burada daha zor SK garantisi yerine getirilmektedir. Optimal yollar hesaplanırken de uyduların hareketliliği sayesinde oluşan dinamik trafik değişimleri de dikkate alınır. Buna karşılık tahmini olmayan protokol yönteminde paket düşürme olasılığı daha yüksektir (Ercetin ve diğ., 2000).

2.2.8. Yaşam Süresini İçeren SK-Tabanlı Yönlendirme

Birçok atlamaya sahip AY uydu ağları için önerilmiş bir yönlendirme algoritmasıdır. SK gereksinimlerini sağlamakla birlikte link aktarımlarını minimize etmek için kullanılmaktadır. Burada UAL'ler için "yaşam süresi" isminde yeni bir kavram tanımlanmaktadır. Bir linkin aktif durumda olduğu toplam süreyi ifade eden bu kavram, bağlantının değişmesi karşısında yeniden yönlendirmeyi minimize etmek için düşünülmüştür. Ayrıca bu süreyi maksimize olarak düşünen çeşitli SK gereksinimlerinin bulunduğu AY uydu ağları için dağıtık bir SK yönlendirme metodu önerilmiştir.

Algoritmada öncelikle aktarım sayısının minimize edilmesine çalışılır. Sonra da bunun uygulanabilir bir çözüm olduğu gösterilir. Dağıtık SK yönlendirmesi uygulanır. Bir link aktarımının minimize edilmesi için UAL'in yaşam süresi önemlidir, çünkü yüksek yaşam süresi, aktarım olasılığını azaltır. Aktarım olasılığının düşük olması durumunda bir grafik metodolojisi tanımlanmıştır. Bu metodolojide yörüngeler arası linklere göre yörünge içi linklerin kullanımı daha çok tercih edilir. Ayrıca yörüngeler arası linklerin daha yüksek yaşam süresiyle kullanımı mevcuttur. Bu linklerin ikisi de dahil olmak üzere ağın şekli oluşturulur. Bütün şekil oluşturulduktan sonra kaynak-hedef düğümler arasındaki uygun yol bulunmaya çalışılır. Link aktarımlarını minimize etmek amacıyla uydular arası hareketin maliyetleri hesaplanır. Bu hesaplara göre UAL yaşam süresi ne kadar büyükse ağ grafiğinde buna karşılık gelen kenarın maliyeti o kadar düşüktür.

Link aktarımlarının minimize edilmesi SK garantisi için yeterli değildir. Çünkü bulunan yol sonucunda kabul edilemeyecek sayıda atlama sayısı ortaya çıkabilir. Bu nedenle UAL'lerin yol oluşturmadaki yaşam süresini maksimize etmek ve aynı zamanda SK gereksinimlerini karşılamak için bir yönlendirme metodu geliştirilmiştir:

SK-tabanlı yol yönlendirme stratejisi

Bir ağ tasarımından sonra yaşam süreleriyle birlikte UAL'ler belirlenir. Sonra kaynak ve hedef uydular için bir SK gereksinimine sahip bağlantı isteği alınır. Kaynaktan başlamak üzere bu istek, SK'sini sağlayan tüm UAL'ler üzerinden iletilerek hedef düğüme ulaştırılır. Gecikme gibi parametreler her düğümden güncellenir. Hedef düğüm, küçük atlama sayısı ve maksimum yol yaşam süresini işaretler. Seçilen yol üzerinden kaynağa onay gönderilir ve böylece bağlantı kurulur.

Dijkstra algoritmasıyla karşılaştırıldığı zaman bu algoritmanın daha iyi performans gösterdiği görülmektedir. Bunun nedeni Dijkstra algoritmasında sadece 1 yolun sonuçta elde edilmesi, dağıtılmış algoritmada ise sonuçta çıkan birkaç yoldan birinin seçilebilmesidir. Başka bir neden ise Dijkstra'da seçilen yolların atlama sayısının yüksek olması, bu nedenle de ağın kaynaklarının büyük kısmının kullanılmasıdır. Ağ yükü artınca da bloke olma olasılığı artar. Buna karşın Dijkstra'nın hesaplama karmaşıklığı daha azdır, çünkü yol sadece bir kere taranır (Jukan ve diğ., 2000).

2.2.9. Yönlendirme İçin Sinir Ağı Yaklaşımı

Bu algoritma, dağıtık yönlendirmeye ilgilidir ve deterministik olanlara göre daha iyi sonuç verebilmektedir. Algoritma sayesinde gecikme problemleri azalmıştır. Çünkü ağdan kalıcı bilgi edinilmesine değil de “gizli zeka” kavramına dayanılmaktadır. Burada deterministik düzendeki güncelleme aralıklarından kaynaklanan yönlendirme, bilgi gecikmesini önlediği için hızlı yönlendirme kararlarına olanak tanımaktadır. Geri yayılım öğrenmesine sahip dağıtık çok katmanlı nöronlara dayanmaktadır. Herbir uyduda bir Çok Katmanlı Nöron (ÇKN) bulunur. Bunların görevi ağdaki trafik dağılımıyla lokal trafik düzenleri arasındaki bağlantı bilgisini öğrenmektir. ÇKN’ların öğrenme aşamasında deterministik alıştırma yönlendirmesi yöntemi kullanılır. Gerekli bilginin öğrenilmesinden sonra link yükünü azaltıp kullanımını arttırmak için yönlendirme işlemi gerçekleştirilir.

Yönlendirme işlemi, kolaylık olması açısından alt kısımlara ayrılabilir. Bu alt kısımlar, UAL Yönlendirmesi, Yukarı ve Aşağı Link Yönlendirmesi ve Karasal Yönlendirme biçimindedir. Başlangıçta herbir zaman aralığı için sistemin sanal bir topolojisi oluşturulur. Bunun için birtakım özellikler düşünülür. Örneğin sistemin servis durumu olarak geliş ve sistemde kalış zamanları vardır. Kullanıcıların coğrafik dağılımı göz önünde bulundurulur. İkinci kısımda çağrının kurulum aşamasındaki trafik yük durumuna bakılarak 4 alternatif yoldan biri seçilir. Bu sırada da bölgelerdeki trafik akışı, belirli bir zaman diliminde iki uydudan birisi üzerinden düşünülür. Böylece N uydulu bir yapıdan NxN boyutlu bir trafik matrisi elde edilir. Önceden bulunan UAL trafik yükleri, yayılım gecikme süresinin yanındaki link maliyet fonksiyonuna ek giriş olarak kullanılır. Tüm yollar hesaplandıktan sonra minimum yol maliyeti Dijkstra En Kısa Yol Algoritması ile bulunur (Werner ve diğ., 1998).

2.2.10. ATM Sanal Yol Yönlendirmesi

Önceden bahsedildiği gibi yönlendirme problemleri başlıca iki kısma ayrılmaktadır. Birincisi, uydular ve kullanıcılar arasındaki bağıl hareketten dolayı oluşan YAL Yönlendirmesidir. İkincisi de uydular arasında bir yol oluşturulmasıyla ilgili olan UAL Yönlendirmesidir. Bu yönlendirme içerisine topoloji değişimiyle ortaya çıkan yeniden yönlendirme işlemi de alınır. ATM Sanal Yol Yönlendirmesinde UAL Yönlendirmesi ele alınmaktadır.

Herbir uydu bir ATM düğümü olarak düşünülmekte ve bir Sanal Yol (SY), iki ardışık ATM düğümü arasındaki lojiksel linki göstermektedir. İki ATM düğümü arasındaki fiziksel kanal kullanılarak Sanal Kanal aracılığıyla iki kullanıcı arasında bağlantı oluşturulur. Bir veya daha fazla Sanal Kanal, bir Sanal Yol Bağlantısını (SYB) oluşturmaktadır.

Burada UAL alt ağındaki tüm uç nokta çiftlerini bağlamak için SYB kullanılır. SYB, bir başlangıç uydusundan bir son uyduya giden yoldur. Uygun uçtan uca yollar bulmak için Dijkstra En Kısa Yol Algoritması kullanılır. Uç noktalar, uyduların hareketli olmasından dolayı zamanla değişir. Sistemdeki zaman aralıklarına ve tüm uydu çiftlerine göre en az maliyetli yollar bulunur. Bunlar bulunurken değişik maliyet ölçütlerinden yararlanılır. Sonra en iyi yol bulunur.

Herbir sanal link yolunun bir Link Ölçüsü özelliği vardır. Bunlar tanımlandıktan sonra bir SYB için Toplam Link Ölçüsü hesaplanır. Dijkstra algoritmasıyla bulunan yollar arasından belirli bir zaman süresince konuşmayı tutabilecek uygun olan biri seçilir. Seçilen yolun maliyeti iyi olduğu sürece elde tutulur. Fakat maliyeti iyi olmayan bir yol seçilirse SYB aktarımı yapılır ve Dijkstra Algoritması kullanılarak yeni bir yol seçilir. Aktarım işlemi gecikmeye sebep vereceği ve trafik yükünü değiştireceği için tercih edilmez (Werner ve diğ., 1996).

2.3. OPTİMİZASYON PROBLEMLERİNE GİRİŞ

Bilimsel ve mühendislik alanlarında sıkça karşılaşılan optimizasyon kavramının ana hedefi, kaynaklar için etkin performans değerlerini sağlamaktır. Günlük yaşamın birçok alanında optimizasyon örnekleriyle karşılaşılabilmektedir. Örneğin bir çiftçi tarlasına ektiği ürünlerden en iyi verimi almak ister. Bir mühendis bir makinenin güç tüketimini optimize etmeye çalışır.

Optimizasyon işleminin başlangıç aşamasında, verilen problemler düşünülerek bunları çözüme götürecek uygun algoritma elde edilmeye çalışılır. Geleneksel yöntemlerle çözümü mümkün olmayan problemler, bu algoritmalar yardımıyla sonuçlanabilmektedir. Her problem sınıfının kendine özgü çözüm teknikleri mevcuttur.

Yani genel olarak tüm sınıflara uygulanan bir algoritma yoktur. Ancak genelde birçok problem sınıfına çözüm getirebilen algoritma seçilip kullanılabilir. Bu durum olduğunda ise her zaman kullanılabilen ve yararlı olan bu algoritmanın, optimal sonucu üretme garantisi yoktur.

Problemlerle uğraşıldığı zaman karşılaşılan en önemli kavram, çözüm için harcanan süredir. Bilimsel alanda bir iş yapılırken geçen süre oldukça önemli bir zaman dilimidir. Bir işin elle veya bilgisayar aracılığıyla yapılması ise burada karşımıza çıkabilecek farklı bir konudur. Optimizasyon problemlerinde de hem zaman, hem de maliyet tasarrufu için bilgisayarlardan yararlanılması önemli bir avantaj sağlamaktadır.

Bir problem ele alındığı zaman öncelikle bunun matematiksel modelinin oluşturulması gereklidir. Modelin çözümünü elde etmek için de optimizasyon yöntemleri kullanılarak en uygun sonuç elde edilmeye çalışılır. Genetik algoritma da günümüzde bu işlem için kullanılan başlıca algoritmalar arasında yer almaktadır. Bu sayede probleme uyarlanan algoritmanın yardımıyla en uygun çözüm bulunabilmektedir.

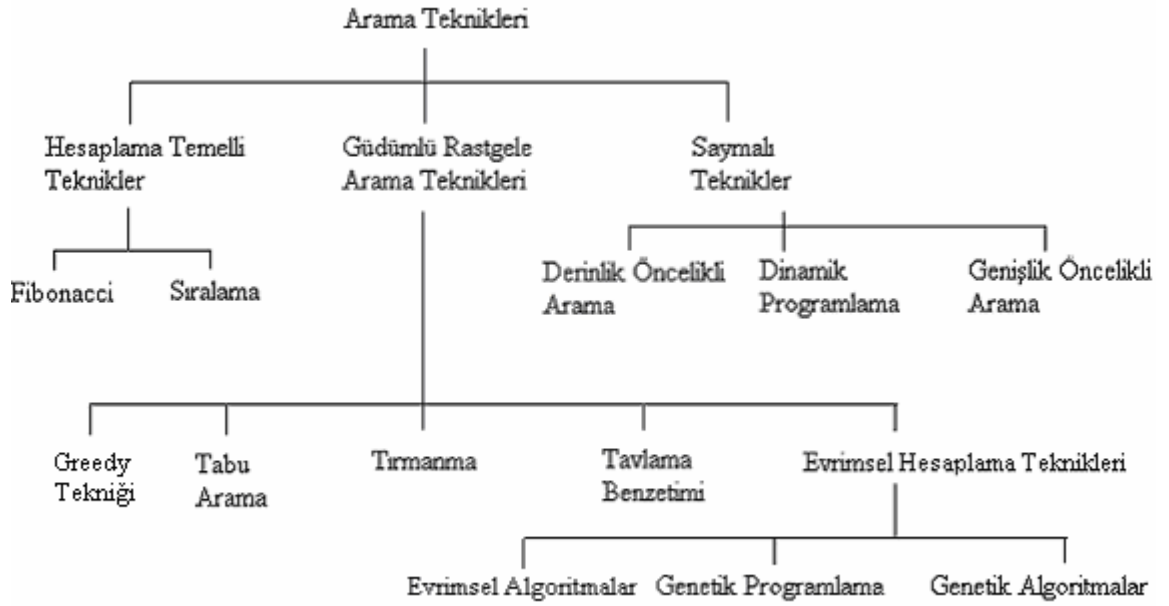
En yaygın optimizasyon problemlerinden biri Satıcı Problemidir. Burada bir grafik üzerindeki noktalar veya bir haritadaki şehirler ele alınmaktadır. Her bir noktadan bir kere geçilmek şartıyla tüm noktaların gezilmesi şartıyla en kısa dolaşma yolu elde edilir. Sonuçta da başlangıç noktasına gelinmesi gerekmektedir. n tane nokta olduğu düşünüldüğü zaman olası tüm yolların sayısı $(n-1)!/2$ olarak hesaplanmaktadır. Böylece 8 noktalı bir yapı için 2520 farklı sonuç, 16 nokta için de 653,837,184,000 farklı sonuç çıkabilmektedir. Bu yapıda problemin uzunluğu, üzerinden geçilen noktaların sayısına eşit olmaktadır. Olası çözüm kümesi, geçilen noktalardan oluşan yolları içermektedir. Maliyet fonksiyonu ise her bir yolun uzunluğudur. Burada problem çözümüne başlanırken öncelikle olası yollardan oluşan yapılar elde edilir. Daha sonra bu yapılar içerisinde en kısa yola sahip olanlar bulunur. Yol uzunluğu ne kadar kısa olursa uygunluk değeri (optimizasyon problemlerinde çözümün kalitesini gösteren ve uygunluk fonksiyonu ile elde edilen değer) de o kadar iyi olmaktadır. Başlangıçta oluşturulan yapılar üzerinde değişiklikler uygulanarak yeni popülasyonlar da elde edilir ve uygunluk değerinin yüksek bulunmasına çalışılır. Böylece zayıf bireyler hayatta kalmaya devam etmezler.

Bir diğerk problem tipi de Yer Tahsis Problemidir. Burada bir grafikte düğümle ve bunlara servis veren noktalar bulunmaktadır. Her bir düğümün bir başka düğümle ve servis veren elemanla arasındaki ortalama uzaklığın minimum olması gerekmektedir. Bu problemde ise uzunluk, noktaların ve servis veren elemanların sayısına eşittir. Çözüm kümesi, servis veren elemanların yolları ile noktaların bunlarla olan bağlantı yollarıdır. Maliyet fonksiyonu ise her bir düğümün kendi servis noktasına olan ortalama uzaklığıdır. Bu şekilde problem tanımları ve gösterimleri yapıldıktan sonraki aşamada bir mühendis, uygun bir zaman süresinde optimal çözümü üretmek için hesaplama tekniklerini uygulamaya başlar (Carnahan ve Simha, 2001).

Optimizasyon işlemlerinin kullanıldığı yönlendirme algoritmaları da mevcuttur. Yapılan bir çalışmada uyduların paketleri kaynak-hedef uydular arasında iletim işlemi olmaktadır. Uydular arasındaki aktarım işlemi için Greedy optimizasyon yöntemiyle elde edilen değerler yer almaktadır (Zaim, 2001).

2.4. EVRİMSEL HESAPLAMA TEKNİKLERİ

Evrimsel hesaplama tekniklerinde bir “birey”, bir “kromozom” şeklinde gösterilmektedir. Bir kromozom da “gen”lerin bir araya gelmesinden oluşan yapıdır. Başka bir şekilde ifade edilecek olursa bir problemin çözümüne karşılık gelen yapı “kromozom”dur (Fisssus, 2000). Evrim kavramı, kromozom yapısının çevreye uyum sağlayıp sonraki kuşaklara geçmesini içermektedir. Bir kromozomun sonraki kuşağa geçmesi için başka yapılarla çiftleşmesi, başka bir deyişle de hayatta kalması gerekmektedir. Evrimsel Hesaplama Teknikleri yardımıyla problemler bu tip kromozom yapılarına aktarıldıktan sonra çözüm kümesinin bulunmaya çalışılması sağlanmaktadır. Bu tekniklerin kullanım alanları arasında mühendislik, finans, ekonomi, endüstri, tarım ve üretim bulunmaktadır. Evrimsel yöntemlerin de içerisinde bulunduğu Arama Teknikleri Şekil 2.8’de görülebilmektedir. Şekilde olduğu gibi Evrimsel Hesaplama Teknikleri; Evrimsel Algoritmalar, Genetik Programlama ve Genetik Algoritmalar olmak üzere 3 kısma ayrılmaktadır:



Şekil 2.8: Arama Teknikleri

2.4.1. Evrimsel Algoritmalar

Evrimsel Algoritmalarla kullanılan gösterimler, standart olmaktan çok probleme özgü olmaktadır. Evrimsel Algoritmalar kendi aralarında Evrim Stratejileri ve Evrimsel Programlama olmak üzere iki kısımdan oluşmaktadır. Genel olarak bu kavramlar şu şekilde açıklanabilmektedir:

2.4.1.1. Evrim Stratejileri

Ingo Rechenberg ve Hans-Paul Schwefel isimli Alman araştırmacılar tarafından ortaya çıkarılmıştır. Burada gerçek değerli parametre kodlaması kullanılmaktadır. Yani parametre optimizasyon probleminde sayılar, gerçek değerleriyle işleme girerler. Yeniden birleşim yönteminden çok mutasyon işlemi kullanılmaktadır (Whitley, 2002).

Genetik algoritmalarla sonra en yaygın olan yöntemdir. Genetik algoritmalar gibi optimizasyon problemleri için kullanılmaktadır. Fakat genetik algoritmalar, araştırma, optimizasyon, tasarım ve makine dili konularında çok amaçlı bir araç olarak kullanılmaktadır. Buna karşın evrimsel stratejiler, sürekli fonksiyonların optimizasyon konusuna odaklanmıştır. Son yıllarda ise bu iki kavram birbirini etkileyerek çeşitli alanlara adapte olmuşlardır (Whitley, 2001).

2.4.1.2. Evrimsel Programlama

Evrimsel algoritmaların diğer alt grubu olan bu yöntem ise 1960'lı yıllarda Lawrence Fogel, A.J.Owens ve M.J.Walsh tarafından geliştirilmiştir. Bu çalışmalarda sonlu durum makinelerinin evrim süreci ele alınmıştır. Çeşitli amaçları gerçekleştirmek için de mutasyon işleminden yararlanılmıştır. Fakat sonradan bu yöntem işlev görmemiştir. 1990'larda yeniden ortaya çıkan bu grup, evrimsel stratejilere benzemektedir (Whitley, 2001). Bu programlamanın en önemli özellikleri, gen davranışlarını dolaylı incelemekten (kod çözüm ve geliştirme aşamaları sonucundaki davranışlara bakarak incelemek) çok direk incelemesi ve yeniden birleşim özelliğine ihtiyaç duymamasıdır. 1990'lı yıllarda David Fogel, evrimsel programlamaya genel parametre optimizasyonunu kullanarak yaklaşmıştır (Whitley, 2002). Aynı yıllarda John Koza ve öğrencileri bu programlamayı, işlem kontrolü, veri analizi ve bilgisayar modelleme konularını kapsayan problemleri çözmede kullanmışlardır. Oldukça geniş ölçüdeki problemlerin çözümünde kullanılan bu yöntemin etkin işlemesi için bilgisayar donanımının gelişmiş olması gerekmektedir (IEEE, 1995).

2.4.2. Genetik Programlama

Genetik Programlama (GP) konusu da son yıllarda evrimsel hesaplama tekniklerinin bir alt konusu olarak karşımıza çıkmaktadır. GP, bir parametre optimizasyon metodu değildir ancak otomatik bir programlamadır ve otomatik programlama ile makine dilinde kullanılmaktadır. 1980'li yıllarda kurallara dayalı sistemleri geliştirmeye yönelik olmuştur (Whitley, 2001). Bilgisayar programlarının geliştirilmesi için önemli bir alandır. Bir başka açıdan GP, geleneksel genetik algoritmanın geliştirilmiş bir halidir. Genetik algoritmalarındaki popülasyonun her bir bireyinin yerini, burada bir bilgisayar programı almıştır. GP'nin çalışma uzayı, probleme uygun terminal ve fonksiyonlardan oluşan tüm olası bilgisayar programlarıdır. Fonksiyonlar arasında standart aritmetik işlemler, standart programlama işlemleri, matematiksel fonksiyonlar, mantıksal fonksiyonlar veya alana özel fonksiyonlar bulunmaktadır (Koza, 1995). 1990'lı yıllarda da genetik programlama, Stanford Üniversitesi'nden John Koza tarafından geliştirilmeye devam etmiştir. Genetik algoritmanın değişik bir şekli olup kullanılan yapının ölçüsü belirli bir limite dayalı değildir ve hiyerarşiktir (IEEE, 1995). John Koza'nın geliştirdiği GP algoritması, kontrol, robotik, oyunlar ve sınıflandırma

alanlarına uygulanmıştır. Bu algorithmada geçen temel kavramlar şunlardır (Willis ve diğ., 1997):

Terminal Kümesi: Girdi olarak alınan değişken veya sabitlerdir.

Fonksiyon Kümesi: Terminal kümesiyle birlikte kullanılıp uygun çözümleri verecek olan özel fonksiyonlardır. Örneğin basit matematiksel fonksiyonlar bunlar arasındadır.

Uygunluk Fonksiyonu: Popülasyondaki herbir bireyin sahip olduğu sayısal bir değerdir. Bir çözümün probleme uygunluk derecesini ölçer.

Algoritma Kontrol Parametreleri: Popülasyon ölçüsünü, çaprazlama ve mutasyon olasılıklarını içerir.

Bitiş Kriteri: Önceden belirlenmiş jenerasyon sayısı veya uygunluk üzerinde belirlenen hata toleransının miktarıdır.

GP ile genetik algoritma arasındaki farklar şu şekilde açıklanabilmektedir (Willis ve diğ., 1997):

-GP’de çözümler ağaç yapısı şeklinde, değişken uzunluklu kromozomlar olarak gösterilmektedir. Genetik algorithmada ise kromozom uzunlukları sabittir.

-GP’de kromozomlar alana özel olarak belirli bir sentaks yapıda gösterilirken genetik algorithmalarda kromozomların sentaksa bağımlılığı yoktur.

-GP, yeniden üretim sırasında ağaç yapısındaki kromozomlarının sentaksını korumak için genetik operatörler kullanmaktadır.

-GP, kromozomların direk çalışması için uygun bir yorumlayıcı sağlayacak şekilde kodlanır. Genetik algorithma ise direk çalışacak formda çok fazla kodlanmamaktadır.

2.4.3. Genetik Algoritmalar

Genetik algorithmlar, genetik ve evrimin birleşiminden oluşan metotlardır. Bunların temel özellikleri, yaygın bir şekilde evrimsel hesaplama tekniği olarak kullanılmalarıdır. “Genetik algorithma” terim olarak iki farklı anlam içerebilmektedir. Bunların birincisi 1970’li yıllarda Amerika’da John Holland tarafından ortaya konulmuş olan, David Goldberg isimli öğrencisi tarafından da geliştirilen modeldir ve birçok çalışmada bundan faydalanılmaktadır. Bunun temelini Darwin’in Evrim Teorisi oluşturmaktadır. Diğer anlamda ise genetik algorithma popülasyon tabanlı bir modeldir ve seçme, yeniden birleşim, mutasyon gibi işlemleri kullanarak, incelenen uzayda örnek noktalar oluşturur.

Evrimsel tekniklerde çözümleri bulmak için matematiksel fonksiyonlar ve bilgisayar simülasyonları kullanılmaktadır. Örneğin genetik algoritmalar uçak motorlarında kullanıldığı zaman yakıt tasarrufu için, farklı tasarımlarda kullanılabilen bir bilgisayar simülasyonuna ihtiyaç duyulur. Başka matematik yaklaşımlarının kullanılmadığı uygulamalarda evrimsel ve genetik algoritmalar kullanılmaktadır.

Genetik algoritmanın kullanım yerleri parametre optimizasyon problemleri, tasarım uygulamaları ve optimizasyondur. Bir ürünü uygun hale getirene kadar gerekli optimizasyon işlemleri de bu sayede yapılabilmektedir. Genel olarak bakıldığı zaman genetik algoritmaların ayrık ve sürekli problemlerin çözümleri için kullanıldığı görülmektedir. Hesaplama açısından zor olmalarına karşın, algoritmada ufak değişiklikler yapılarak çeşitli problemlere uygulanabilir hale getirilmektedir.

Genetik ve evrimsel hesaplamalar, belirli bir probleme potansiyel bir çözüm üretmek için kullanılır. Burada biyolojideki kromozoma benzer bir veri yapısı kullanılır. Bu yapılar içerisindeki her bir gen, değişkenlere karşılık gelmektedir. Bunlar üzerine bilgiyi işlemek ve bu bilgileri yapı içerisinde tutmak için çaprazlama ve mutasyon işlemleri uygulanır (Whitley, 2002). En sonda kalan uygun kromozomlar da problemin çözümünü oluşturmaktadır. Genetik algoritmanın karşılaşılabileceği bir zorluk ise çaprazlama işlemi için kromozomların (yani problem çözümlerinin) uygun yapıda olmamaları durumudur.

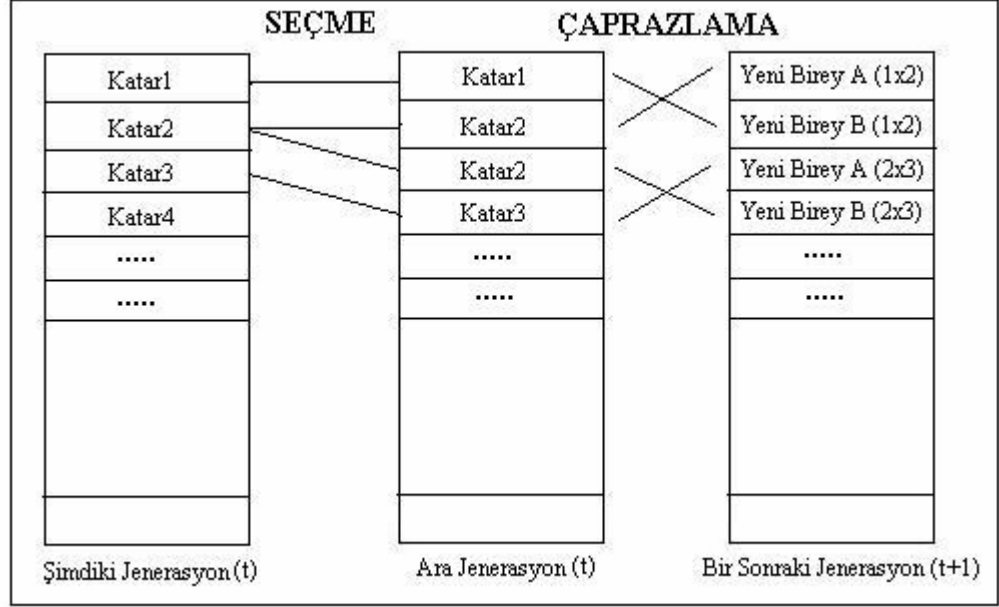
Genetik algoritmayla bir problem çözümüne girilmeden önce aşağıdaki konuların belirlenmesi gerekmektedir (Koza, 1995):

1. Gösterim şekli bilinmelidir.
2. Uygunluk için gerekli olan ölçü saptanmalıdır.
3. Algoritmanın kontrolü sırasında gerekli olan parametre ve değişkenler belirtilmelidir.
4. Sonucu oluşturacak gidiş yolunu çıkarmak ve bitişi belirleyecek parametreyi belirlemek gereklidir.

Genetik algoritmalar, diğer rassal (stochastic) arama yöntemlerinden farklı olarak tek bir çözüm yerine bir popülasyonla işleme başlamaktadır. Uygulamaya başlanırken ilk başta rastgele veya sezgisel yöntemle kromozomlardan oluşan bir popülasyon ele alınır. Burada çözümleri gösteren ifade, kromozom yapılarıdır. En iyi çözümü ortaya çıkaracak kromozomun, zayıf çözümü veren yapılara göre daha çok ortaya çıkması sağlanmaktadır. Bir çözümün iyi olup olmadığı o andaki popülasyona bağlıdır (Whitley, 1993). Popülasyondaki kromozom yapılarının hayatta kalması veya eşleşmeleri için gerekli kritere uygunluk değeri (fitness value) adı verilmektedir. Bu değer, genetik algoritmayla yapılan optimizasyon çözümlerinde ölçülmesi gereken ve kromozomlar arasındaki farklılığı gösterecek olan değişkendir. Kromozomlar, uygunluk değerlerine göre eşleşme işlemine girebilirler. Başka bir deyişle uygunluk değeri büyük olan kromozomların yeni bir kromozom oluşturma olasılıkları çok daha fazladır. Bu aşamadan sonraki çözüm kümesine çaprazlama, mutasyon veya ters alma işlemi uygulanarak en uygun çözüm elde edilmeye çalışılır. Zamanla zayıf olan kromozomların yok olduğu görülür. Belirli bir tekrar sayısı elde edildiğinde veya popülasyon bir değere yakınsadığı zaman işlem sonlandırılır. Popülasyonun bir değere yakınsaması da birkaç jenerasyon (kromozomların başarılı döngüler sonucunda oluşturmuş olduğu topluluğa jenerasyon denmektedir (Fisssus, 2000)) sonrasında uygunluk değeri yüksek olan kromozomun hayatta kalmaya devam etmesi ve böylece optimal çözümü vermesiyle olmaktadır.

Şekil 2.9’da genetik algoritmanın işleyiş biçimi görülmektedir. Bu yapıda ilk jenerasyondan başlayarak seçme işlemi ile yeni bir jenerasyon elde edilir. Burada kromozom yapılarının uygunluk derecelerine göre yeni jenerasyonda bulunup bulunmayacakları çıkarılabilmektedir. İlk jenerasyondaki çok iyi kromozomların diğer jenerasyonlarda görülebilme olasılığı fazladır. Bu şekildeki kromozomlar birkaç kere tekrarlanmış olarak ortaya çıkabilmektedirler. Kötü kromozomlar ise ara jenerasyona geçmeden ortadan kaybolurlar. Çaprazlama sonucunda da iki ebeveyn kromozomdan yola çıkılarak iki yeni kromozom yapısı elde edilir (Whitley, 2001). Bu işlem yapılırken dikkat edilen özellik, ortaya çıkacak kromozomların ebeveyn kromozomlardan daha iyi özelliklere sahip olmalarıdır. Bundan sonraki aşamada yapılabilecek mutasyon işlemleri aracılığıyla kromozom üzerindeki kötü genler devreden çıkarılarak eskisine göre çok daha iyi kromozomlar elde edilmektedir. Bütün bu yapılan işlemler popülasyonun

kromozom sayısını etkilemez. Bu sayı hep sabit olarak kalmaktadır (Correa ve diğ., 1999). Önemli bir nokta da popülasyon ölçüsünün ve oluşabilecek maksimum jenerasyon sayısının kontrol edilebilen, ilk değer alan parametreler olmalarıdır. Bir genetik algoritma uygulamasında bu değerlerin ikisi de sayıca yüzlerce, binlere ulaşabilmektedir (Koza, 1995).



Şekil 2.9: Şimdiki jenerasyondan başlayarak yeni jenerasyonlar oluşturma

Genel olarak genetik algoritmanın performans artımını sağlamak için, kullanılan kromozom yapılarının, popülasyon ölçüsünün, uygunluk fonksiyonunun ve kodlama metotlarının önemi büyüktür (Chou ve diğ., 2001).

Genetik algoritmaların temel prensibi şu şekilde olmaktadır:

1. Problem kromozom yapısı şeklinde (bit stringi, dizi, ağaç, liste vb.) ifade edilir.
2. Olası çözümleri gösteren bir başlangıç popülasyonu rastgele oluşturulur.
3. Herbir kromozom için bir uygunluk değeri hesaplanır. Bu değer direk olarak optimuma uzaklıkla ilgilidir ve hedef fonksiyonun sonucundan oluşmaktadır.
4. Uygunluk değerlerine göre popülasyondaki kromozomların seçimi yapılır. Kromozomlar arasında çiftleşmeler yapılır.
5. Kromozomların çaprazlama ve mutasyon işlemleri gerçekleştirilir. Bu işlemler, kullanılan kromozom yapısına göre tanımlanmaktadır.

6. Yeni kromozomlardan oluşan popülasyon elde edilir.
7. Bitiş kriteri sağlanmamışsa yeniden 3. adıma dönülür.

Genetik algoritmaların sahip olduğu avantajlar şu şekilde sıralanabilmektedir (Haupt ve Haupt, 1998):

1. Sürekli veya ayırık parametrelerle optimize işlemi yapılabilir.
2. Ek bilgiye ihtiyaç duyulmamaktadır.
3. Geniş bir alanda maliyet hesabı yapılmaktadır.
4. Çok sayıda parametreye hakim olunmaktadır.
5. Paralel bilgisayarlar için uygundur.
6. Oldukça karışık maliyetlerde bile parametreleri optimize edebilmektedirler.
7. Çözüm kümesinde birkaç değer olduğu için tek bir çözüm değerine bağlı kalma gibi bir durum olmamaktadır.
8. Parametrelerin kodlanmış şekliyle işlemler yapılmaktadır.
9. Sayısal veri, üssel veri ve analitik fonksiyonlarla çalışılabilir.

Genetik algoritma kullanılarak karmaşık mühendislik problemleri kolaylıkla çözüme kavuşabilmektedir. Bu konuda yapılan çalışmalar aracılığıyla çeşitli fonksiyonların optimize edildiği görülmektedir. Örneğin bir çalışmada önceden belirlenmiş bir şarkıyı tahmin etmeye yarayan bir simülasyon mevcuttur. Çeşitli notalar bir araya gelerek bir kromozom yapısını oluşturmaktadır. Buradaki optimizasyon işleminin amacı, birçok kromozom arasında şarkıyı doğru şekilde çıkaracak nota grubunu veren kromozomun bulunmasını sağlamaktır. Aynı çalışmanın ikinci kısmında da geleneksel ikili gösterim yerine sürekli parametreler kullanılarak doğrusal olmayan kısmi diferansiyel denklemler çözümlenmektedir (Haupt ve Haupt, 1998).

Bir başka çalışmada ise bir haberleşme ağı için gerekli olan yapının tasarımı gerçekleştirilmektedir. Burada oluşturulacak yapı oldukça karmaşık ve kısıtlamalıdır. Bu nedenle optimizasyon işlemi kullanılmaktadır. Genetik algoritma bu adımda devreye girer. Genetik algoritma kullanılarak çok nesneli fonksiyonlar, ayırık değerler ve çok kısıtlı problemler elde edilmektedir. Optimal topolojik bağlantılar, yönlendirme ve link kapasite tahsisi üzerinde tasarım yapılmaktadır. Başka bir deyişle topolojinin yanı sıra

yönlendirme ve kapasite optimizasyonu da yapılmaktadır. Ağ için gerekli trafiği ve minimum gecikme süresini sağlamakla birlikte, düşük maliyet değerini bulmak da gerekmektedir. Örnek olarak düşünülen 10 şehir arasındaki trafik miktarı hesaba katılarak bu akışın kapasite sınırını aşmamasına çalışılmaktadır. Aynı şekilde gecikme süresi de izin verilen süreyi aşmamalıdır (Ko ve diğ., 1997).

Ağ ortamında uygulanan genetik algoritma örneklerinden birinde de istemci-sunucu mimarisi kullanılmıştır. Böylece paralel genetik algoritma tabanlı bir yapı ortaya çıkmıştır. Burada sunucu, alt popülasyonlar arasında katar alışverişini yönetme görevini yapmaktadır. Bu alt popülasyonlar, diğerleriyle iletişime geçtiği anda “Yaşam Parametresi” denilen özel bir değişken, iletişim zamanını ayarlamaktadır. Ağ yapısında Satıcı Probleminin 100 düğüm için uygulandığı görülmektedir. Hesaplamalar sonucunda da istemcilerin birbirleriyle iletişimi gerçekleştiği sırada çalışma süreleri ve maksimum uygunluk değeri gibi ölçümler ortaya çıkmıştır. İstemci sayısı ile maksimum uygunluk ortalamalarının birbirleriyle ilişkisi tablo şeklinde ifade edilmiştir (Kojima ve diğ., 2000).

Bir başka ağ örneğinde dalga boyu tahsisinde en optimal sonucu bulmak için tavlama benzetimi yöntemi ile genetik algoritma kullanılmaktadır. Dalga boyu bölme çoğullama (wavelength division multiplex-WDM) metodu kullanılarak tüm optik iletim ağlarında optimizasyon yapılmaktadır. Bu tip ağlarda hem anahtarlama, hem iletim işlemi optik alanda olmaktadır. Böylece uzun mesafelerde çok büyük kapasiteli iletim işlemi gerçekleştirilmektedir. Bir dalga boyunun taşıdığı kanal, optik hattaki bir fibere karşılık gelmektedir. Böylece tüm ağdaki toplam fiber uzunluğu düşüktür. Fiberlerin en yüksek kullanım oranlarının oluşmasıyla elde edilen fiber-dalga boyu tahsisini yapmak için optimizasyon işleminden yararlanılır. Olası çözümler tamsayı dizisi şeklinde elde edilmektedir. Her bir dizi elemanı, dalga boyu kanalındaki bir dalga boyunu göstermektedir. Genetik algoritmada her bir çözümü bir kromozom göstermektedir. Uygunluk fonksiyonu olarak ele alınan denklem, kullanılan optik fiber sayısı ile hat uzunluklarını ve yüksek kullanım oranını içermektedir. Böylece elde edilen değerlerden en uygun dalga boyu tahsisi yapılmaktadır (Mikac ve diğ., 1998).

2.5. GENETİK ALGORİTMALARDAKİ İŞLEMLER VE GÖSTERİMLER

Genetik algoritmalarda yapıları oluşturmak ve işlemleri gerçekleştirmek için gerekli olan yöntemleri bilmek gerekmektedir. Önemli kavramlar aşağıda sıralanmıştır:

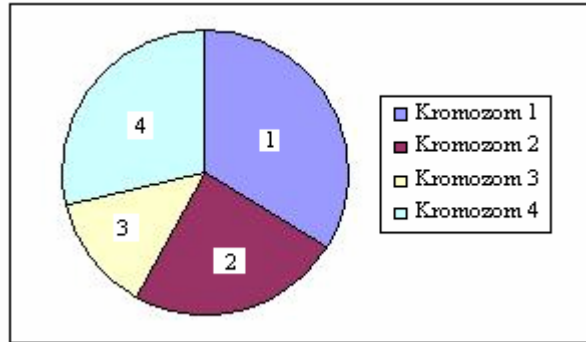
2.5.1. Yeniden Üretim (Regeneration)

Bu adımda Seçim(selection) işlemi uygulanmaktadır. Seçim metodu yeniden üretim işlemi için çiftleri ve bir sonraki jenerasyonda kalacak uygun kromozomları belirlemek amacıyla kullanılır. Bu işlem, uygunluk değerlerine göre yapılır. Uygun olan kromozomlar yeniden üretim işlemi için seçilirken zayıf olanlar elenir. Birçok seçim metodu mevcuttur. Bunlar arasında en yaygın kullanılanları şu şekilde sıralamak mümkündür:

2.5.1.1. Rulet Çarkı Seçim Yöntemi

Bireylerin seçilme olasılığı $\frac{f(i)}{\sum_i f(i)}$ 'dir. Burada $f(i)$ değeri ilgili kromozomun uygunluk

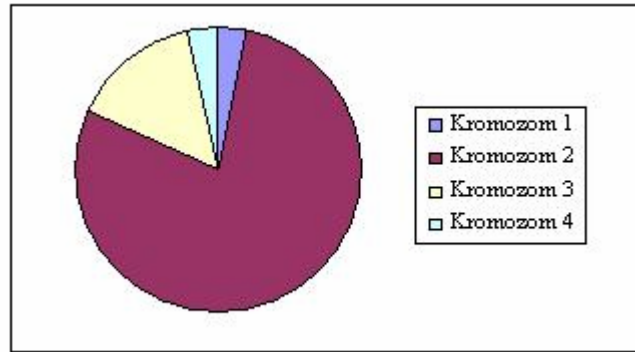
değerini, $\sum f(i)$ ise tüm kromozomların uygunluk değerlerinin toplamını gösterir. Olasılık oranı, rastgele seçilen bir sayı ile karşılaştırılarak bu sayıya yakın olan orana sahip kromozom seçilir. Şekil 2.10'da görüldüğü gibi kromozomlar çarkın üzerinde belirli dilimlere karşılık gelmektedir. 1.kromozomdan başlanarak uygunluk değerleri üst üste eklenip çark elde edilmektedir. Bundan sonra rastgele seçilen sayının hangi dilime karşılık geldiğine bakılmaktadır. Örneğin Rastgele seçilen sayı 3. dilim içerisine denk gelirse Kromozom 3 seçilmiş olur.



Şekil 2.10: Rulet Çarkı

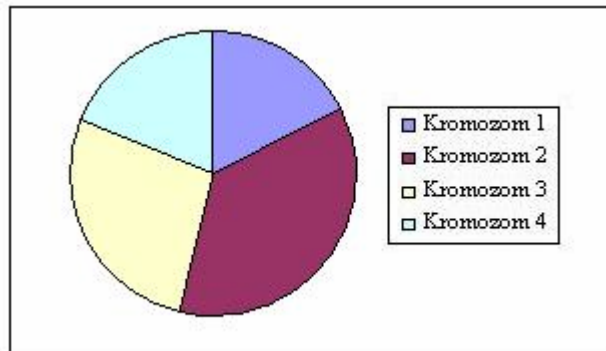
2.5.1.2. Sıralama Yöntemi

Kromozomların uygunluk değerleri arasında büyük farklar olduğu zaman Rulet Çarkının kullanılması avantajlı değildir. Çünkü uygunluk değeri büyük olan kromozom sürekli seçilebilir. Bu durum karşısında Sıralama Yöntemini kullanmak yararlı olmaktadır. Kromozomların uygunluk değerleri büyükten küçüğe doğru sıralanır. Daha sonra ilk değerden başlanarak seçim işlemi gerçekleştirilir. Böylece uygunluk değeri daha büyük olan kromozomların seçilmesiyle yeni jenerasyon belirlenmektedir. Şekil 2.11 ve Şekil 2.12’de bu yöntemin kullanılmasıyla ortaya çıkan durum farklılıkları gözlemlenebilmektedir (Marek, 1998). Şekil 2.11’de uygunluk değerleri göz önüne alınarak kromozomların dizildiği görülmektedir. Burada Kromozom 2’nin en yüksek uygunluk değerine sahip olduğu ortaya çıkmaktadır. Böyle bir durumda seçim işlemi uygulanırsa Kromozom 2 birçok kez seçilme şansına sahip olmaktadır.



Şekil 2.11: Sıralamadan önceki durum

Şekil 2.12’de ise kromozomlar sıralandıktan sonraki sıra numaraları düşünülerek grafik çizilmiştir. Böylece diğer kromozomların da seçilme ihtimalleri arttırılmış olmaktadır.



Şekil 2.12: Sıralamadan sonraki durum

2.5.1.3. Seçme (Elitist) Yöntemi

Uygunluk değeri en büyük olan kromozomlar çaprazlama işlemi için seçilirler. Kalan kromozomlar arasından diğer yöntemler vasıtasıyla seçim işlemi yapılır.

2.5.1.4. Turnuva Yöntemi

Bu yöntem iki aşamadan oluşmaktadır. İlk aşamada N ($N \geq 2$) bireyden oluşan bir grup seçilir. Daha sonra bu grupta en yüksek uygunluk değerine sahip olan birey alınır diğerleri devre dışı bırakılır (Burgess, 1999).

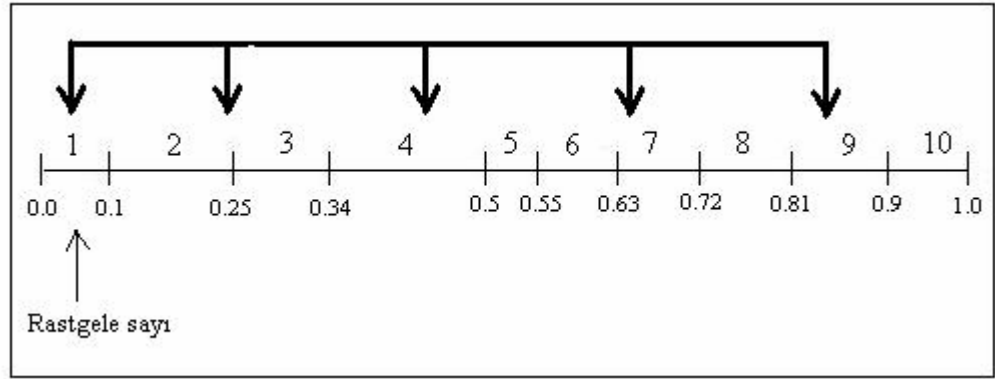
2.5.1.5. Sabit Durum (Steady State) Yöntemi

Bu yöntemde kromozomların büyük çoğunluğu bir sonraki jenerasyona geçebilmektedir. Yeni bireyi oluşturmak için her jenerasyonda daha iyi (uygunluk değeri daha yüksek olan) kromozomlar seçilir. Daha düşük uygunluk değerine sahip bazı kötü kromozomlar da atılır ve bunların yerine önceden seçilen iyi kromozomlar konulur. Popülasyonun geri kalan kısmı yeni jenerasyona aynen aktarılır (Marek, 1998).

2.5.1.6. Tahmini Evrensel Seçim Yöntemi

Tüm popülasyondaki bireylerin uygunluk değerleri 0-1 aralığındaki bir sayı doğrusundaki her bölüme karşılık gelecek şekilde dizilir. Rastgele bir sayı bulunarak bu sayıdan başlanacak şekilde kaç tane birey seçilecekse belirlenir. Rastgele sayı taban alınacak şekilde “1/Birey sayısı” miktarı kadar aralıklarla ilerlenir. Yani eşit uzunluklu belirlenecek bölmelerin aralık uzunluğu 1/Birey sayısı kadardır. Başlangıç değerinin ve ilerleme sırasında elde edilen değerlerin hizzasına gelen bireyler seçilir.

Şekil 2.13’te görüldüğü gibi 10 bireyden oluşan bir popülasyon düşünölsün. Herbir bireyin uygunluk değerine göre karşılık geldiğı bölge bulunduktan sonra rastgele bir sayı belirlenir. Bu sayıdan başlayarak 5 birey seçileceğı düşünölsünse aralıklar $1/5=0.2$ birim olacaktır. Bu aralıkların sınırları, kalın çizgiyle belirtilen oklarla belirlenmektedir. Bu örneğe göre sonuçta seçilecek bireylerin numaraları 1, 2, 4, 7 ve 9’dur (Pohlheim, 2005).

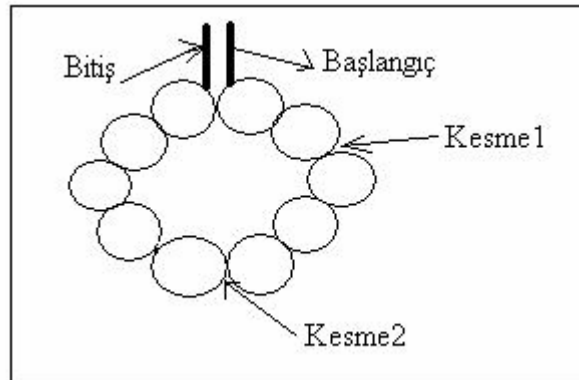


Şekil 2.13: Tahmini Evrensel Seçim yöntemine göre bireylerin seçilmesi

2.5.2. Çaprazlama Teknikleri

Geleneksel genetik algoritma konularında çaprazlama yapılırken iki ayrı kromozomun birer noktası ele alınır. Bu nokta uygun bir yerden seçilir. Böylece kesme noktalarından başlanarak çaprazlama işlemi gerçekleştirilir. Birçok çaprazlama yönteminde 1’den çok nokta ele alınarak işlem gerçekleştirilmektedir. Aslında çaprazlama noktalarının sayısındaki artış, genetik algoritmanın performansını düşürebilmektedir.

1. 2-noktalı çaprazlama: 2-noktalı ve çok noktalı tüm çaprazlama işlemlerinde katarlar lineer olarak değil de yuvarlak döngü yapıları olarak düşünülmektedir. İki döngü yapısı arasında bir parçanın değiş tokuş işleminde iki noktadan kesim yapılması gerekmektedir. Şekil 2.14’te görüldüğü gibi 2 ayrı kesme noktasıyla elde edilen segment çaprazlama işlemine girer.



Şekil 2.14: Döngü şeklindeki bir kromozom yapısı

1. Tek biçimli (uniform) çaprazlama: Bu çaprazlama türünde bir çaprazlama maskesi rastgele belirlenir. Ebeveyn kromozom yapısı tek tek kopyalanarak yeni bir kromozom elde edilir. Çaprazlama maskesi ise hangi ebeveynin seçileceği konusunda kullanılır. Maskede 1 olarak karşılaşılan kısma karşılık ilk ebeveyninden, 0 olarak karşılaşılan kısımda ise ikinci ebeveyninden gen alınır. Aşağıdaki örnekte bu işlem açıkça görülmektedir.

Çaprazlama Maskesi: 1 0 0 1 0 1 1 1 0 0

Ebeveyn 1: 1 0 1 0 0 0 1 1 1 0

Ebeveyn 2: 0 1 0 1 0 1 0 0 1 1

Yeni Kromozom: 1 1 0 0 0 0 1 1 1 1

Böylece yeni kromozomun genleri ebeveyn kromozomlardaki genlerin karışımından elde edilmektedir. Etkin çaprazlama noktası sayı olarak $L/2$ değerine yaklaşık elde edilmektedir (burada L kromozom uzunluğudur).

Çaprazlama çeşitleri birbirleriyle karşılaştırıldığı zaman aralarında performans açısından %20'den fazla bir fark olmadığı görülmüştür (Beasley ve diğ., 1993).

2.5.3. Ters Alma ve Yeniden Düzenleme

Kromozomlar arasında seçili bir noktadan başlayarak genlerin değiştirilmesi durumuna “ters alma” denir. “Yeniden düzenleme” işleminde ise çevreye en uygun potansiyele sahip kromozom yapısının bulunması için genlerin yerleriyle oynanmaktadır. Bu yöntem için genlerin dizilimlerinde bir lineerlik olması gerekmektedir. Bunun yanı sıra tek biçimli çaprazlama kuralı uygulandığı zaman genlerin dizilimi önemli olmamaktadır. Bu durum, Yeniden Düzenleme tekniğinin kullanılmasına ihtiyaç sağlamayacaktır. Ayrıca düzenleme işlemi için araştırma alanı fazladan kullanılmış olur. Genetik algoritmada genlerin değerlerinin iyi olmasının yanı sıra gen dizilimlerinin de uygun olmasına çalışılmaktadır. Bu nedenle problem karmaşık bir yapıya gelmektedir.

Organizmaların çevre kromozom dizilimleri önemli bir yer tutmaktadır. Buna “karyotipik evrim” adı da verilmektedir. Kısa dönemli düşünüldüğü zaman bir organizmanın çevreye iyi uyum sağlaması güzel bir özelliktir. Uzun bir zaman süresi

boyunca ele alınan karyotipik evrimde ise çevrede oluşan yeni koşullara adapte olabilen organizma hayatta kalmaktadır. Her bir jenerasyonda oluşan genlerin tiplerinin hesabı hemen devreye girmektedir. Karyotip evrimi ise çok uzun bir zaman süresince, belki binlerce yıl sürebilmektedir.

Birçok genetik algoritma uygulamasında uygunluk fonksiyonu içerisinde kapsanan çevre statiktir. Bu tip uygulamalarda karyotipik evrim çok fazla önemli olmaz. Buna karşılık zamanla birlikte uygunluk fonksiyonunun değiştiği ve genetik algoritmanın değişen çevreye uygun çözümler üretmek zorunda olduğu uygulamalarda karyotipik evrim önemli olmaktadır.

Yapılan çalışmalarda, statik bir çevrede iyi gen dizilimlerini bulmak için güzel genetik algoritma parametrelerini içeren kalıplardan yararlanmak gerekmektedir. Bu yapıların içerisinde her bir eleman bir genetik algoritma yapısını kullanır. Bu tür kullanımlarda hesaplama açısından maliyet söz konusudur (Beasley ve diğ., 1993).

2.5.4. Mutasyon ve Naive Evrimi

Mutasyon işlemi, kromozom yapısını oluşturan genler üzerinde iyileşme yapmak amacıyla değişiklik uygulamaktır. Böylece genin değerinde farklılık olmaktadır. Buradaki temel amaç, daha iyi yapılara sahip olan kromozomlar elde etmektir. Naive Evrimi denilen kavram ise seçme ve mutasyon işlemlerinin bütününden oluşur. Çaprazlama işleminden ayrı olarak genetik algoritmada uygulanan bir kuraldır. Çaprazlama işlemi evrimsel açıdan genlerde daha fazla değişiklik yapar ama mutasyon daha uygun sonuçlar elde edebilmektedir. İki işlemin karşılaştırılması yapıldığı zaman bazı görüşlere göre mutasyon, çaprazlamanın yaptığı her işi yapabilmektedir (Beasley ve diğ., 1993).

2.5.5. İkili Olmayan Gösterimler

Bir kromozom, sembollerin birleşiminden oluşmaktadır. Semboller ikili basamaklardan oluşur. Bazı çalışmalarda kullanılan semboller tamsayı ya da kayan noktalı sayılardan oluşur. Bu tip çalışmaları yapanlar, sembollerin sayı olarak gösterilmelerinin, bit gösteriminden daha avantajlı olduğunu belirtmektedirler (Beasley ve diğ., 1993).

2.5.6. Kodlama ve Hesaplama

Genetik algoritmaların probleme bağı olan kodlama ve hesaplama fonksiyonu olmak üzere iki ana bileşeni mevcuttur. Bu fonksiyonlar yardımıyla en iyi mutasyon ve yeniden birleşim operatörleri bulunur. Bir fonksiyonu minimize veya maksimize etmek için kullanılan kodlama şekli ikili veya gerçek sayı sistemi olabilir. Buna karşılık bir üretim uygulamasında görevlerin dağılımını gösteren bir permütasyon yapısı kullanılır. Bir problemin çözümüne başlamadan önce kodlamaya karar verilir. Tamsayı ve gerçek sayıların ikili gösterimini oluşturmak için farklı yöntemler mevcuttur. Bunlar İkili Kodlanmış Ondalık (Binary Coded Decimal) denilen kodlama tekniğiyle ikili sisteme çevrilebilirler. Daha sonra uygun mutasyon ve yeniden birleşim operatörleri seçilir. Mutasyon işleminin derecesi, değişikliğe uğrayan karakter sayısı ile doğru orantılı olmaktadır. Mesela 1 karakterin mutasyona uğradığı yerde derece düşüktür (Whitley, 2002).

Örnek optimizasyon problemi olarak maliyetin düşürülmesi, bir kar oranının maksimize edilmesi veya başka bir parametrenin durumu ele alınsın. Farklı parametreleri gösteren kontrol grupları düşünülerek problem çözümüne gidilir ve sonuçta tek bir çıktı elde edilir. Bu çıktı da hesaplama fonksiyonunun ortaya çıkardığı değerdir. Buradaki ana amaç, çıktıyı optimize etmek için değişik parametrelerle oynamaktır.

Genetik algoritmaların içerisine lineer olmayan problem çeşitleri de girmektedir. Bu problemlerde bir parametre diğerlerinden bağımsız olarak kendi başına çözümlenemez. Çıktıyı elde etmek için parametrelerin birleşik etkileri göz önünde bulundurulur. Bazı durumlarda hesaplama fonksiyonunu oluşturmak için bir bilgisayar simülasyonuna ihtiyaç duyulabilir. Hesaplama fonksiyonu bazen büyük bir yapının sadece bir kısmını oluşturur. Mesela bir üretim işinin aşamalarını gösteren yapıda performansı ölçen kontrol birimi böyle bir aşamaya karşılık gelebilmektedir. Hesaplama fonksiyonu aynı zamanda hızlı olmak zorundadır (Whitley, 1993).

3. MALZEME VE YÖNTEM

Bu tezde yeni bir yönlendirme algoritması oluşturulmaktadır. Bunun yanısıra algoritmanın statik halinin, Dijkstra algoritmasına göre performans ölçümü yapılmaktadır. İncelenen birçok yönlendirme algoritmasına göre performans kriterleri belirlenmektedir. Yeni yöntemde bu kriterlere ek olarak yolun kalıcılığını ölçen değer ile uydu sistemindeki linklerin trafik yoğunluklarını da hesaba katan analitik bir çözüm üretilmektedir. Yeni algoritmanın temeli genetik algoritmaya dayanmaktadır. Genetik algoritma, normal yöntemlerle çözülmesi zor veya imkansız olan problemlerde kullanılmaktadır. Optimizasyon problemlerinde uygulanan bir yöntem olan genetik algoritmalar sayesinde doğal seçim yöntemiyle kromozomların iyi olanları en sona kalan bireyleri oluşturmaktadır. İlk bireyler, birtakım evrimsel değişimlerden geçirilerek yeni bireyler oluşturulmaktadır. Uydu ağlarında da bu durum, sanal yolla yeni jenerasyonlar üretmeye dayanmaktadır. Bu şekilde düşünülerek bir uydu ağında uygun yollar elde edildikten sonra bu yollar genetik algoritmadaki işlemlere konularak sonuçta en uygun yollar ortaya çıkmaktadır. Bu yollar, kriterleri minimum değerde sağlamaktadırlar. Böylece avantajlı yeni bir yöntem bulunmuş olmaktadır.

Literatürdeki yönlendirme algoritmalarında uygunluk fonksiyonu olarak gecikme süresi veya aktarım sayısından biri seçilmektedir. Bu tezde bu iki kavramın da kullanıldığı ortak bir uygunluk fonksiyonu kullanılmaktadır. Yani analitik yöntemler sayesinde hem gecikme süresinin, hem de aktarım sayısının kullanıldığı yeni bir uygunluk fonksiyonu oluşturulmaktadır. Yönlendirme algoritmasıyla seçilen herbir ağ topolojisi $v=(l_{12}, ..., l_{1N}, l_{23}, l_{24}, ..., l_{2N}, ..., l_{N-1N})$ şeklindeki bir vektörle gösterilmektedir. Burada l_{ij} , i ve j uyduları arasındaki linkin kullanılıp kullanılmadığını göstermektedir. Uydu topolojisinin gökyüzündeki sabit duruma geldiği M tane farklı zaman aralığı olduğu düşünölsün. Bu algoritmada genetik yöntem kullanılacağı için herbir kromozom, M farklı yönlendirme vektörünün bir bütününden oluşur. Uygunluk fonksiyonu olarak da aktarım sayısı ve gecikme süresinin bir fonksiyonu ele alınmaktadır. Böylece aktarım sayısı ve gecikme süresi minimum olacak şekilde eski yollar seçilmektedir. Çok sayıda topoloji

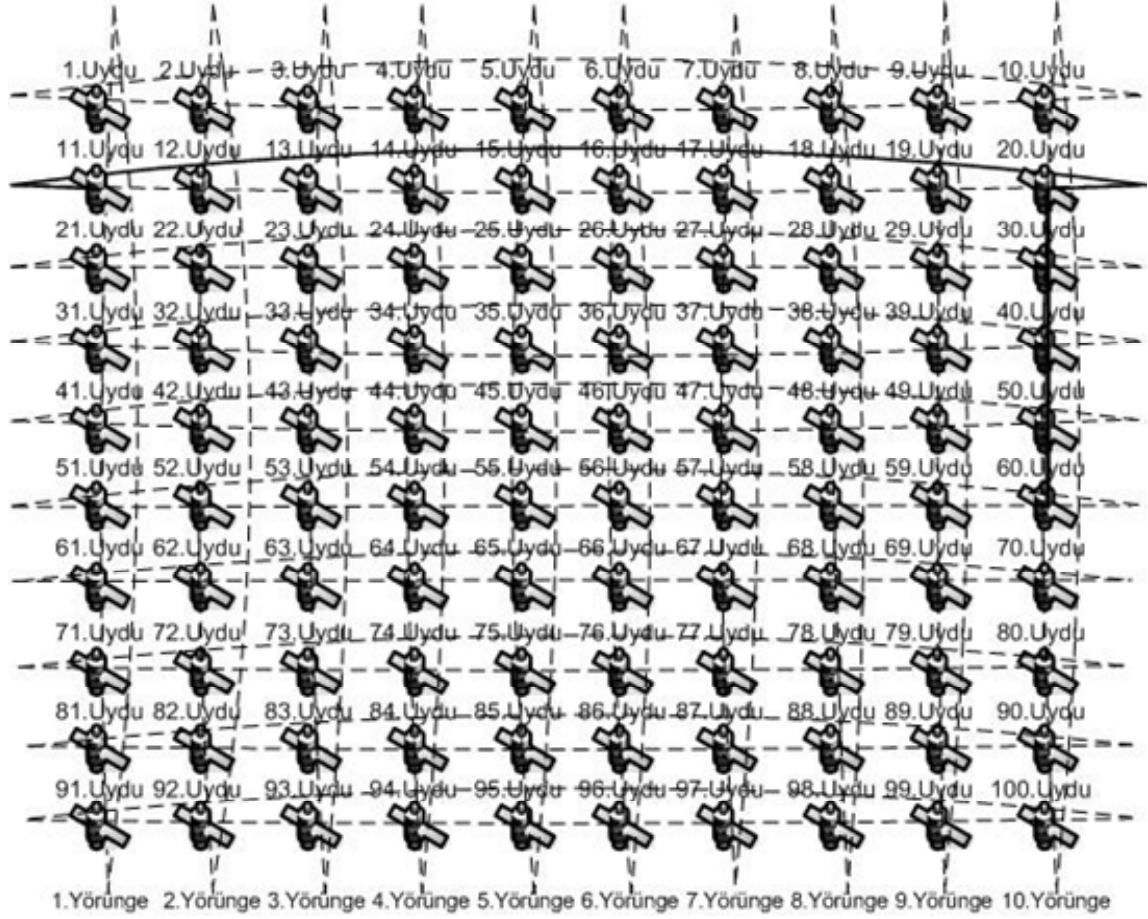
düşünülerek performans analizi yapılmaktadır. Uydu ağındaki her bir yol, kromozom yapısıyla gösterilerek uygunluk fonksiyonuna göre en uygun yollar elde edilmeye çalışılmaktadır. Öncelikle kromozomlar doygunluk değerlerine göre yeniden üretilirler. Sonra bu kromozomlara çok sayıda çaprazlama işlemi uygulanır. Son olarak da yolların en son hali elde edilir. Ayrıca farklı topolojilerden elde edilen yollar karşılaştırılarak da bir topolojiden diğerine geçerken yapılan aktarımların sayısı hesaplanabilmektedir. Bu şekilde gecikme süresinin minimum olduğu yolları ortaya çıkaran ve topoloji değişimlerinde karşılaşılan aktarım sayısını da küçük bir değer olarak bulan etkin bir yöntem elde edilmiş olmaktadır.

3.1. STATİK UYDU AĞININ OLUŞTURULMASI

Uydular arasında bilgi alışverişini sağlamak için bilgi paketlerinin bir başlangıç uydusundan yola çıkması gerekmektedir. Bir paket bu şekilde gönderildikten sonra ağıdaki uygun linklerden geçirilerek hedef uyduya yetiştirilmeye çalışılır. Buradaki işlemlerin olması için uyduların, yönlendirme yaparken bir sonraki uyduyu seçmeleri gerekmektedir. Bu seçim işlemi için de en uygun yolların bulunması amacıyla hareket edilmektedir. Kullanılan algoritma statik yönlendirme algoritması olduğu zaman uydu ağının o andaki konumunun, belirli bir sistem periyodu sonrasında tekrar edileceği düşünülür. Bu çalışmada oluşturulan ilk yapıda uydu ağ topolojisinin konum olarak sabit olduğu düşünülmektedir. Kullanılan yapıda sistemin sürekli aynı pozisyonda durduğu düşünülerek yönlendirme işlemi gerçekleştirilmektedir. Bu işlemler için temel kriter, gecikme süresi en düşük yolu seçmektir. Ayrıca aktarım sayısı en düşük olan yolu bulmak için statik birkaç sistemin ardarda geldiği düşünülmektedir. Böylece ardışık topolojiler arasındaki yol değişikliklerinden aktarım sayıları hesap edilebilmektedir.

Günümüze kadar birçok yönlendirme algoritması geliştirilmiştir. Bu algoritmalar birbirinden farklı işlem yapılarına sahiptirler. Birçoğunda kullanılan uygunluk fonksiyonu, gecikme süresini minimize etmeyi amaçlamaktadır. Bunun yanı sıra aktarım sayısını minimize etmek de mümkündür. Bu şekilde belirlenen kriterler yardımıyla en uygun minimum yol için optimizasyon işlemine başlanır. Bu çalışmada uygunluk fonksiyonu olarak hem gecikme süresi, hem de aktarım sayısı

düşünülmektedir. Bu iki kavramdan yola çıkılarak bir paketin uydu ağı yapısındaki yönlendirmede izlediği yollar, kromozom denilen genetik yapılarla gösterilmektedir. Her bir yol, bir paketi başlangıç uydusundan alarak hedef uyduya götüren bir yapıdır.



Şekil 3.1: Yüz tane düğümden oluşan bir AY uydu ağı topolojisi

Şekil 3.1’de 100 düğümden oluşan bir AY uydu ağı görülmektedir. Dünyanın çevresini saran yörünge yapılarının açılarak uyduların düzenlenmesi sonucu bu yapı elde edilmektedir. AY uydu sistemlerinde bir uydu, alt ve üst konumdaki uydulara bağlı olduğu gibi, sağ ve sol uydulara da linklerle bağlıdır. Şekilde görüldüğü gibi her bir düğümün bağlı olabildiği 4 komşu düğümü mevcuttur. Kenarlardaki uydularda ise kesik çizgilerle gösterilen çevrimsel bağlantılar sağlanmaktadır. Örneğin yukarıdaki Şekil 3.1’deki topolojide 11. uydu, 20. uyduya bağlıdır. Aralarında link bulunmayan uyduların ise birbirlerine bağlantı kuramadıkları düşünülmektedir. 11. uydudan 60. uyduya paket gönderimi için uygun bir yol olarak 11-20-30-40-50-60 elde

edilebilmektedir. Her bir linkin bir maliyet değeri olduğu bilindiğinden dolayı, bu maliyet durumlarına göre en düşük maliyetli yollar hesaplanmaktadır.

Genetik algoritma yardımıyla elde edilen yolların herbirinin bir kromozom yapısıyla gösterilmesi sağlanmaktadır. Bu kromozomların sayısı, popülasyonun nüfusunu göstermektedir. Popülasyonun nüfusu her zaman sabit olarak tutulmaktadır. Örneğin kendi topolojimizde 20 uzunluklu bir kromozom yapısı için $Y_i = u_{i1}, u_{i2}, u_{i3}, u_{i4}, u_{i5}, u_{i6}, u_{i7}, u_{i8}, u_{i9}, u_{i10}, u_{i11}, u_{i12}, u_{i13}, u_{i14}, u_{i15}, u_{i16}, u_{i17}, u_{i18}, u_{i19}, u_{i20}$ şeklinde yollar elde edilmektedir. Burada u_{i1} ifadesi paketin gittiği i. yol üzerindeki 1. uyduyu göstermektedir. Her bir yol, bu şekilde uydu numaralarından oluşmaktadır. Örneğin bir paket 5 uyduyu geçerek hedefe ulaşabiliyorsa elde edilen yol, “ $u_{i1}, u_{i2}, u_{i3}, u_{i4}, u_{i5}$ ” şeklinde olur ve diğer değişkenler boş kabul edilir. Başka bir deyişle kromozom uzunluğu 20 olarak düşünülse bile, elde edilen yollara göre bu değer değişebilmektedir. Sonuçta oluşan yolların sayısı ise popülasyonun nüfusu kadar olmaktadır. Algoritmada takip edilen adımlar şu şekilde özetlenebilmektedir:

1. Linklerin gerçek yapıdaki durumları düşünülerek bazıları aktif, bazıları pasif yapılır. Gerçekte olduğu gibi sistemdeki linklerin çoğu aktiftir.
2. Bir kaynak düğümle bir hedef düğüm belirlenir.
3. Linklerin maliyetleri hesaplanır.
4. Kaynak düğümünden yola çıkarak aktif olan linkler üzerinden geçen yollar bulunur.
5. Tüm yolların uygunluk değerleri hesaplanır.
6. Yeni bir jenerasyonun üretimi yapılır.
7. Genetik algoritma yardımıyla çaprazlama işlemi yapılır. Yeni gecikme değerleri bulunur.
8. Aktarım sayıları hesaplanır.
9. En kısa ve uygun yol elde edilir.

Bu adımları, matematiksel ifadeleri de göz önüne alarak incelemek mümkündür. Öncelikle i. kromozomun uzunluğu k_i , popülasyonun nüfusu ise n olarak düşünölsün. Algoritmada öncelikle herbir uydunun komşu tüm uydularla arasındaki 4 link aktif hale getirilir. Daha sonra belirlenen bazı linkler pasifleştirilir. Kullanıcıdan kaynak ve hedef

düğümünün girilmesi istendikten sonra aktif linkler üzerinden giden yollar elde edilir. Yolların gösterimi aşağıdaki şekilde olmaktadır:

$$Y_1 = \begin{bmatrix} u_{11} \\ u_{12} \\ \cdot \\ \cdot \\ u_{1k_1} \end{bmatrix} \quad Y_2 = \begin{bmatrix} u_{21} \\ u_{22} \\ \cdot \\ \cdot \\ u_{2k_2} \end{bmatrix} \quad \dots\dots\dots Y_n = \begin{bmatrix} u_{n1} \\ u_{n2} \\ \cdot \\ \cdot \\ u_{nk_n} \end{bmatrix} \quad (3.1)$$

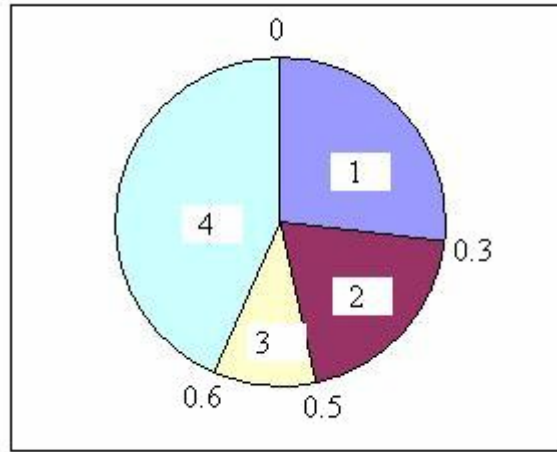
Bu ifadelerde Y_i , i. yolu; u_{ij} ise i. yolun j. uydusunu göstermektedir. k_i terimi de i. yolun sahip olduğu uydu sayısını, başka bir deyişle kromozom uzunluğunu belirtmektedir. Herbir yol bir kromozom olarak düşünülmektedir. Genetik algorithmada kullanılacak gösterimler şöyle olmaktadır:

$$\begin{aligned} Y_1 &= u_{11}, u_{12}, \dots, u_{1k_1} \\ Y_2 &= u_{21}, u_{22}, \dots, u_{2k_2} \\ &\cdot \\ &\cdot \\ &\cdot \\ Y_n &= u_{n1}, u_{n2}, \dots, u_{nk_n} \end{aligned} \quad (3.2)$$

Bu adımdan sonra Rulet Çarkı seçim yöntemi kullanılarak yeni jenerasyonda hayatta kalacak kromozomlar bulunur. Bunun için öncelikle her bir yolun toplam gecikme değeri hesaplanır. Her bir yol için aşağıdaki uygunluk değeri hesaplanır:

Uygunluk değeri= Yolun gecikme değeri / Tüm yolların gecikme değerlerinin toplamı.

Bu orana bakılarak bir yolun diğer yollara göre etkinliği elde edilmiş olur. Eğer bu oranlar üzerinde optimize işlemi gerekiyorsa yeniden düzenleme yapılarak yeni oranlar elde edilebilmektedir. Bu işlem için örneğin 4 tane yol olsun ve bunların tüm gecikme değerlerinin toplamı 1000 birim olsun. Uygunluk değerleri de $f_1=300/1000=0.3$; $f_2=200/1000=0.2$; $f_3=100/1000=0.1$; $f_4=400/1000=0.4$ olarak bulunmaktadır. Şekil 3.2’de bu 4 kromozomun uygunluk değerleri grafiksel olarak gösterilmektedir. Bu aşamada 0-1 değer aralığından rastgele bir yeniden üretme sayısı seçilir. Bu sayı, grafikteki hangi aralığa gelirse o indise sahip olan yol yeniden üretilen jenerasyonda yer alır. Örneğin rastgele bulunan sayı 0.35 ise 2. yol yeni jenerasyona geçer.



Şekil 3.2: Kromozomların uygunluk değerlerine göre dağılımı

Yeniden üretilen jenerasyon üzerinde gerekli olan çaprazlama işlemleri yapılır. Son olarak gecikme süresi ve aktarım sayısı hesaplanarak en uygun yollar elde edilir. Bir yolun gecikme süresi bulunurken aşağıdaki formül kullanılır:

$$D_{Y_i} = \sum_{j=1}^{k_i-1} d_{u_{ij}, u_{i(j+1)}} \quad (i=1,2,\dots,n) \quad (3.3)$$

Bu denklemde, $d_{u_{ij}, u_{i(j+1)}}$ ifadesi, ilgili yolda bulunan u_{ij} and $u_{i(j+1)}$ uydularının arasındaki linkin gecikme süresini ifade etmektedir. Böylece herbir yol üzerindeki tüm linklerin

gecikme süreleri toplanmaktadır. Bütün yolların gecikme süreleri aynı şekilde hesaplandıktan sonra bunlar arasında karşılaştırma işlemi yapılır. $D_{Y_{optimal}} = \min \{D_{Y_i}\}_{i=1,\dots,n}$ denklemi kullanılarak minimum yola sahip olan en optimal yol elde edilir ($Y_{optimal} = Y_i$). Uydu sisteminde çeşitli zaman aralıklarına farklı topolojilerin karşılık geldiği düşünüldüğü zaman, uygulamaya birçok farklı topoloji de eklenmektedir. Bölüm 4.1.2’de belirtilen şekilde her bir topolojiden seçilen uygun yollar arasındaki aktarım sayıları hesaplanır. Son olarak bir sistemin farklı zaman aralıklarında meydana gelen topoloji değişiklikleri de hesaba katılarak tek 1 paketin hem gecikme değeri, hem de aktarım sayısına bağlı olan yönlendirme işlemi gerçekleştirilmektedir.

3.2. DİNAMİK UYDU AĞINA GEÇİŞ

Uydu sistemlerinde yörüngelerdeki uydular sürekli hareket ederler. Bununla birlikte dünya kendi eksenini etrafında sürekli döner. Bu dinamik yapıdan dolayı uydu sistemlerinin yapısında zamana bağlı bir değişiklik olmaktadır. Böylece sürekli aynı yapının olması ve linklerin aynı şekilde işlem görmesi de mümkün olmamaktadır. Zamanla kaynak-hedef uydular arasında belirlenen yollarda kopan linkler veya aktif olan linkler olabilmektedir. Linklerin kopması, pasif hale gelmesi demektir. Bu durumların sonucunda da bir önceki sistem yapısında elde edilen yol, bir sonraki yapıda elde edilemeyebilir. Yani pasif hale gelen linki içeren yollar kaybedilmiş olmaktadır. Aktif hale geçen linkler de yeniden yol seçimine dahil edilmektedir.

Yapılan incelemelere göre AY uydu sistemlerindeki yol kayıplarının, JY uydu sistemlerine göre daha az olduğu belirlenmiştir. Bu durum, özellikle avuç-içi bilgisayarlara servis sağlama konusunda AY uydu sistemlerine önemli bir avantaj sağlamaktadır. Çünkü iletişim sırasında kesinti olmadığı zaman verimli ve zaman açısından da etkin bir haberleşme olmaktadır. Aksi halde o andaki aktif linkleri içeren yeni bir yolun bulunması için yeni işlemlere ve ek zamana ihtiyaç duyulmaktadır. Bu da anında iletişim sağlanması gereken durumlarda dezavantaj olmaktadır.

Yol kayıplarının olmama veya azaltılma gereksinimi, yönlendirme algoritmalarının kapsamına da girmektedir. Başka bir deyişle, oluşturulan yönlendirme algoritmalarında

en az gecikmeye sahip yolu bulmanın yanı sıra elde edilen yolların dayanıklı olmasına da çalışılmaktadır. Dayanıklı yollar üzerinden iletişim sağlanırken, herhangi bir linkin kopma durumu olmadığı için hızlı bir haberleşme sağlanabilmektedir. Gelişen teknolojinin etkisiyle farklı haberleşme yollarının birbirleriyle karşılaştırılmaları söz konusu olduğu için zamanla en uygun yöntem ayakta durabilmektedir. Bu uygunluk kavramının içine hız ve kesintisiz iletişim de girdiği için uydu sistemlerinde bunlara dikkat edilmesi gerekmektedir. Yapılan çalışmalar mevcut sistemlerin iyileştirilmesi ve geliştirilmesine dayandığı gibi, yeni yöntemlerin de elde edilmesi gerekmektedir. Yönlendirme algoritmalarında mevcut yöntemlere ek olarak yeni bir algoritma tasarlarken bütün özellikleri düşünerek daha etkin metodun ortaya çıkarılması amaçlanmıştır.

Önceden yapılan çalışmalar arasında, yönlendirme algoritması olarak farklı yöntemler bulunmaktadır. Bunların kullanılmasıyla bir algoritmanın hem statik hem dinamik şekilleri birbirleriyle karşılaştırılmıştır. Örnek bir çalışmada bir SDO yardımıyla yönlendirme işlemi gerçekleştirilmiştir. İşlemler için zaman aralıkları mevcuttur. Statik yönlendirmede yönlendirme bilgisi tüm zaman aralıklarında sabittir ve değişikliğe uğramaz. Buna karşılık dinamik yönlendirmede yönlendirme bilgisi zaman içinde değişebilmektedir. Bunun nedeni uyduların birbirlerine göre konumunun ve linklerin aktiflik durumlarının değişebilmesidir. Bir uydu ağında linkler aktifken pasif, pasifken de aktif hale gelebilir. UAL'ler sayesinde gecikme süresi düşebilmekte ve atmosferdeki iletim kayıpları yok edilmektedir (Chang ve diğ., 1996b).

Bir başka çalışmada da link yapılarında sürekli değişiklik olan bir uydu ağ topolojisi ele alınmaktadır. Burada önemli olan nokta, hangi linklerin oluşturulup hangilerinin devre dışı bırakılacağını seçmek ve böylece uyduların birbirleriyle iletişimini kurmaktır. Ağ topolojisine periyodik olarak değişiklikler uygulanır, bunlar bir kritere bağlı olarak gerçekleştirilir. Bu kriterler, sabit sayıdaki anten ve alıcılardan oluşan sabit sayıdaki uygun linklere getirilen kısıtlardan oluşmaktadır. Bu durumun oluşması için ağ topolojisindeki her bir düğümün ağ yapısı hakkında bilgiye sahip olması gerekmektedir. Düğümlerin kendi trafiklerine göre değişen gecikme süreleri bulunmaktadır. En kısa yol hesabında gecikme süreleri de düşünülmektedir. Toplam gecikme sürelerinin minimum yapılmasına çalışılarak uygun ağ durumu elde edilmektedir. Ağ yapısı zayıf olduğu

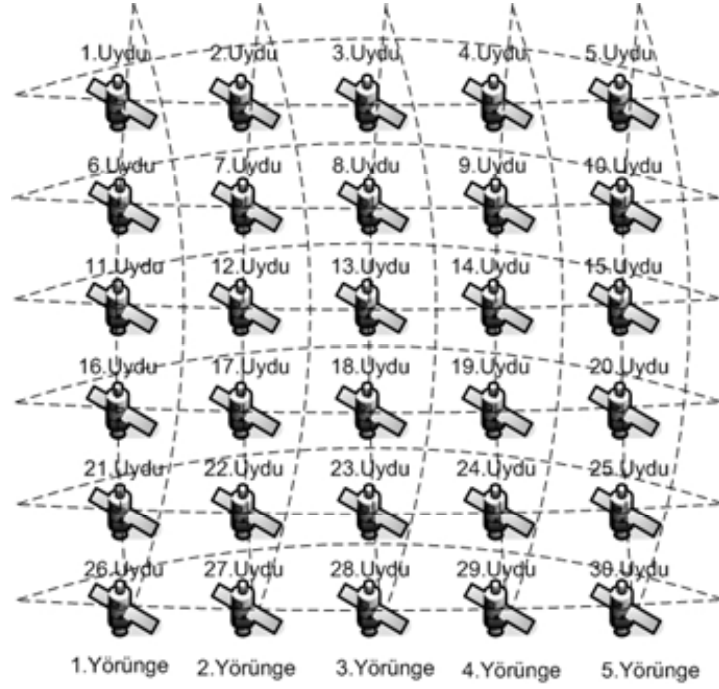
zaman görüş alanında bulunma ve gecikme süreleri üzerinde gerekli değişiklikler yapılır. Görüş alanında bulunma süresi uzun olan yollar seçilir. Bu aşamada birçok yol varsa gecikme süresi kısa olana öncelik verilir. Genel amaç, ağdaki linkler açısından bağ sayısını arttırmaktır. Zaman içinde dinamiklikten kaynaklanan kopmalar ve link kayıplarının yerine yeni linkler devreye konur (Noakes ve diğ., 1993).

Uyarlanır (adaptive) bir yönlendirme algoritmasında da zamanla değişen trafik yapısı kullanılmaktadır. Aktarım halindeki çağrıların bloke olma olasılıkları azaltılmaktadır. Bu amaçla kullanılan ağ yapısı, bağlantı uyumlu ATM teknolojisi üzerinde kurulu yüksek hızlı anahtarlama teknolojisidir. UAL'ler yardımıyla iletişim sağlanmaktadır. UAL'ler kenar, uydular da köşe olacak şekilde ağın matematiksel ifadesi çıkarılır. Herbir kenar için uzaklık ve kalan bant genişliği olmak üzere iki ayrı özellik gözlemlenmektedir. Bir uydu, yol seçimi sırasında herbir kenarın kalan bant genişliğinin belirli bir limitle karşılaştırılmasına göre çağrıyı kabul etme olasılığına sahiptir. Uyarlanır yönlendirmenin yapıp yapılmayacağı bu olasılığa göre çıkarılmaktadır. Bu hesaplamalar, her bölgeye bir uydu denk gelecek şekilde ağın bölgelere ayrılmasıyla yapılmaktadır. Bant genişlikleri belirli limite kadar mevcutsa gelen çağrı kabul edilmektedir. Böyle bir durumda çağrılarının bloke olma olasılıkları azaltılmış olur. Performans analizinde algoritmanın, aktarılan çağrılarının bloke olma olasılıklarını oldukça düşürdüğü görülmektedir. Toplam çağrı bloke olma olasılığı da tüm yönlendirme işlemine göre biraz yüksektir. Bunun nedeni trafik yük dağılımı için daha fazla kaynağa gereksinim duymasıdır (Kim ve Park, 2000).

3.3. DİNAMİK YÖNLENDİRME ALGORİTMASINDA YOL SEÇİMİ

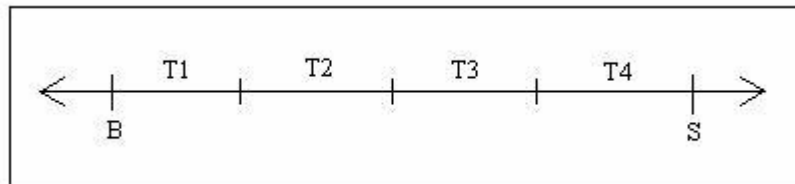
Uygulamada kullanılan uydu ağı topolojilerinin yapısı, ağdaki toplam yörünge sayısına ve herbir yörüngedeki uydu sayısına bağlı olarak değişmektedir. Şekil 3.3'te 30 uydudan oluşan ve tüm linkleri aktif olan bir topoloji örneği görülmektedir. Şekilde görüldüğü gibi uydular 5 yörüngeye dağıtılacak şekilde konumlandırılmıştır. Böylece herbir yörüngeye 6 tane uydu gelecek şekilde dağılım ortaya çıkmıştır. Uygulamada da bu şekilde topoloji yapıları elde edildikten sonra uydular arasındaki linklerin aktivitesi belirlenmektedir. Bir uydunun tüm komşu uydularla arasında linkler bulunabilmektedir. Ancak dinamik yapıdan dolayı bazı linklerin aktif olmadığı zamanlar olmaktadır. Yani

iki komşu uydunun arasında bulunan link işlem görmeyip pasif kalabilmektedir. Bu durumda bu link üzerinden paket iletimi gerçekleşmemektedir. Yol seçiminde de bu özellik göz önünde bulundurulmaktadır. Böylelikle pasif olan link seçilmeden başka bir linke yönelme olmaktadır. Gösterimlerde aradaki link görünmüyorsa o linkin pasif olduğu düşünülmektedir.



Şekil 3.3: Tüm linkleri aktif olan bir uydu topolojisi örneği

Linklerin durumunun zamanla değişmesiyle yeni topolojiler ortaya çıkmaktadır. Böylece bir çağrının başlangıç zamanından itibaren bitiş zamanına gelinceye kadar sistemin farklı topolojileri ortaya çıkar. Bu farklı topolojiler üzerinde yol seçimi yapılarak yeni jenerasyonlar elde edilir. Sonuçta da en uygun yollar seçilir.



Şekil 3.4: Çağrı süresince topoloji değişikliklerinin olduğu aralıklar

Şekil 3.4'te gösterilen örnek zaman doğrusunda bir çağrı ile ilgili verilen zaman aralıkları görülmektedir. B çağrının başlangıç zamanını, S ise bitiş zamanını göstermektedir. Bu zaman noktaları arasında eşit uzunluktaki 4 farklı zaman aralığı çıkarılmıştır. T_1 , T_2 , T_3 ve T_4 zaman aralıklarında uydu ağının yapısında linklerin aktifliğiyle ilgili değişiklikler olmaktadır. Yani sistem bu zaman aralıklarında farklı topolojiler gibi davranmaktadır. Herbir aralık için sırayla ve ayrı ayrı yol seçimleri yapıp jenerasyonlar elde edilmektedir. Jenerasyonlar üzerinde yapılan genetik hesaplamalar da aralıklar arasında birbirinden bağımsız şekilde yapılmaktadır. Sonuçta yine aralık sayısı kadar jenerasyon elde edilerek bunlar birbirleriyle karşılaştırılmaktadır. Bu karşılaştırmadaki temel kriter aktarım sayısıdır. Böylece aktarım sayısı en az olacak şekilde en uygun son yollar seçilmektedir. Böylece uydu sisteminin dinamik değişikliğinden yola çıkılarak gecikme süresi açısından en uygun, en kalıcı ve çağrının kesilmesine engel olan yollar elde edilmektedir.

Topolojilerde yol seçimindeki kriterler için uydu ağındaki linklerin her birine ait bir gecikme süresi (d) ve bir ağırlık değeri (w) vardır. d değişkeni, herbir linke verilen bir tamsayı değeridir. w ise bir linkin aktif olup olmadığını ifade eden değerdir. Link aktif ise bu değer 1, pasif ise 0 değerini almaktadır.

Bir jenerasyonda yollar elde edilirken kaynak uydudan yola çıkılarak aralarındaki linkin w değeri 1 olan uydular takip edilerek hedef düğüme ulaşılır. Böylece elde edilen tüm yollar bir jenerasyonu ortaya çıkarmaktadır. Uydu ağı sayesinde gerçekleştirilen kablosuz iletişimde herbir yol, bir çağrının dolaştığı uyduları göstermektedir. Bir çağrı boyunca elde edilen çeşitli yolların bir araya gelmesiyle bir jenerasyon oluşmaktadır. Denklem (3.1)'de bir jenerasyon boyunca elde edilen yolların gösterimi görülmektedir.

3.3.1. Uygunluk Değerlerinin Hesaplanması

Uygunluk değerlerinin elde edilmesinde önemli olan 3 aşama vardır. Bunlar sırasıyla gecikme sürelerinin (D) hesaplanması, uydu ağına gelen çağrıların bloke olma olasılıklarının devreye girmesi ve sonuçların uygunluk değerlerine dönüştürülmesi şeklindedir. Bu aşamaları aşağıdaki biçimde açıklamak mümkündür:

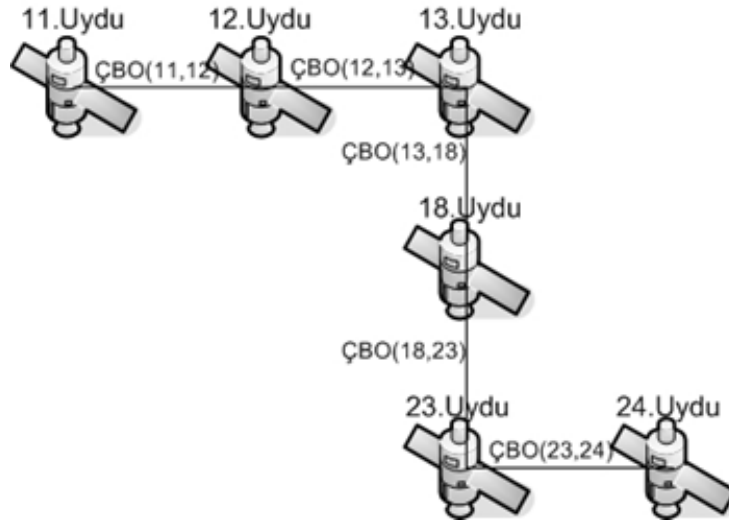
1. D değerleri hesaplanırken ilgili yola ait tüm linklerin gecikme sürelerini gösteren d ifadeleri toplanmaktadır:

$$D_{Y_i}(\Delta t) = \sum_{j=1}^{k_i-1} d_{u_{ij}, u_{i(j+1)}} \quad (i=1,2,\dots,n) \quad (3.4)$$

Burada i değeri ilgili yolun indisini, $d_{u_{ij}, u_{i(j+1)}}$ değeri de u_{ij} ile $u_{i(j+1)}$ uyduları arasındaki linkin gecikme değerini göstermektedir.

Yukarıda bahsedildiği gibi çağrının başlangıcından bitişine kadar geçen sürenin aynı uzunluktaki farklı zaman dilimlerine bölündüğü düşünülmektedir. Herbir zaman diliminde elde edilen yollar, bir topolojiyi oluşturmaktadır. D formülündeki Δt değeri de bu zaman aralıklarını göstermektedir. Böylece ilgili Δt süresinde topolojide genetik işlemlere dayalı hesaplamalar yapıp en son jenerasyonlar elde edilmektedir. Bölüm 3.3.2’de ayrıntılı şekilde anlatıldığı gibi bu durumun ardışık zaman dilimleri boyunca sürdüğü düşünüldüğünde de sistemin dinamikliği görülmektedir.

2. Sırayla bir yoldaki tüm linklerin Çağrı Bloke Olma olasılıkları (ÇBO) hesaplanmaktadır. Bu konudaki önemli kısımlar Bölüm 3.5’te ayrıntılı şekilde açıklanmaktadır. Şekil 3.3’teki ağdan alınan ve 11.uyduyu kaynak, 24.uyduyu da hedef olarak gösteren yol örneği Şekil 3.5’te belirtilmektedir. Burada görüldüğü gibi paket iletiminde 6 uydudan, dolayısıyla da 5 linkten oluşan bir yol takip edilmektedir.



Şekil 3.5: Bir yol üzerindeki tüm linklerin ÇBO gösterimi

Bütün linklerin sonuçları (1-ÇBO) şeklinde tek tek hesaplanarak o linkin işleme girme olasılığı bulunur. Tüm linkler için bu değer bulunup birbirleriyle çarpılarak ilgili yolun kullanılma olasılığı elde edilir. Bu ifadeleri aşağıdaki şekilde göstermek mümkündür:

$$P_{Y_i} = (1 - \text{ÇBO}_{u_{ij}, u_{i(j+1)}}) \cdot (1 - \text{ÇBO}_{u_{i(j+1)}, u_{i(j+2)}}) \cdot \dots \cdot (1 - \text{ÇBO}_{u_{i(k-1)}, u_{ik}}) \quad (3.5)$$

$$F_{Y_i} = \frac{D_{Y_i} * P_{Y_i}}{\sum_{g=1}^n D_{Y_g} * P_{Y_g}} \quad (i=1,2,\dots,n), (j=1,2,\dots,k-1) \quad (3.6)$$

Buradaki $\text{ÇBO}_{u_{ij}, u_{i(j+1)}}$ ifadesi u_{ij} ile $u_{i(j+1)}$ uyduları arasındaki linkin ÇBO sonucunu vermektedir. Bölüm 3.5'te anlatıldığı gibi ÇBO formülündeki üs, ilgili linkin sahip olduğu kanal (sunucu) sayısını ifade etmektedir. F değeri de Y_i yolunun uygunluk fonksiyonunu göstermektedir. Böylece bir yolun bloke durumundan uzak şekilde kullanıma uygun olması için gerekli değer hesaplanmış olmaktadır.

- İkinci aşamada her bir yol için elde edilen P olasılık değeri, o yolun toplam gecikme süresiyle çarpılmaktadır. Böylece bir yolun kullanılma durumunda gecikme süresinin ne kadar olduğu hesaplanmış olmaktadır. Buna göre gecikme

süresi büyük olan yolun da uygunluk değeri büyük olmaktadır. Bu aşamada yeni bir işlem gerekmektedir. Çünkü gecikme süresi düşük olan yolun uygunluk değerinin büyük olması zorunludur. Yeni algoritmada ana hedeflerden biri de budur.

Tablo 3.1: Uygunluk fonksiyon değerlerine karşılık gelen Uygunluk değerleri

Uygunluk fonksiyon değerleri (F)	Uygunluk değerleri
2.960048	0.023810
1.814296	0.052381
2.630889	0.033333
0.329138	0.080952
0.189007	0.095238
0.189007	0.090476
0.495586	0.071429
1.814296	0.047619
0.265001	0.085714
2.630889	0.028571
1.702172	0.066667
11.558361	0.004762
1.702172	0.061905
0.329138	0.076190
4.985862	0.009524
1.702172	0.057143
1.957111	0.042857
2.960048	0.019048
3.348441	0.014286
2.464910	0.038095

Hesaplanacak yeni uygunluk değerleri için tüm yolların sayısından, yani popülasyonun nüfusundan yararlanılmaktadır. Buna göre n tane yola sahip olan bir jenerasyonda (n nüfuslu bir popülasyonda) gecikme süresi minimum olan yoldan başlayarak sırayla $\frac{n}{(n*(n+1))/2}$, $\frac{n-1}{(n*(n+1))/2}$, $\frac{n-2}{(n*(n+1))/2}$, ..., $\frac{1}{(n*(n+1))/2}$ değerleri yeni oranları oluşturmaktadır. Böylece 20 yola sahip bir popülasyonda minimum gecikme süreli yolun oranı 0,095238 olmaktadır. Diğer oranlar da sırayla 0,090476; 0,085714;.....; 0,004762 şeklinde hesaplanmaktadır. Uygulamada 20 yol için elde

edilmiş olan F değerleri ve buna karşılık gelen yeni uygunluk değerleri (oranlar) Tablo 3.1’de gösterilmektedir.

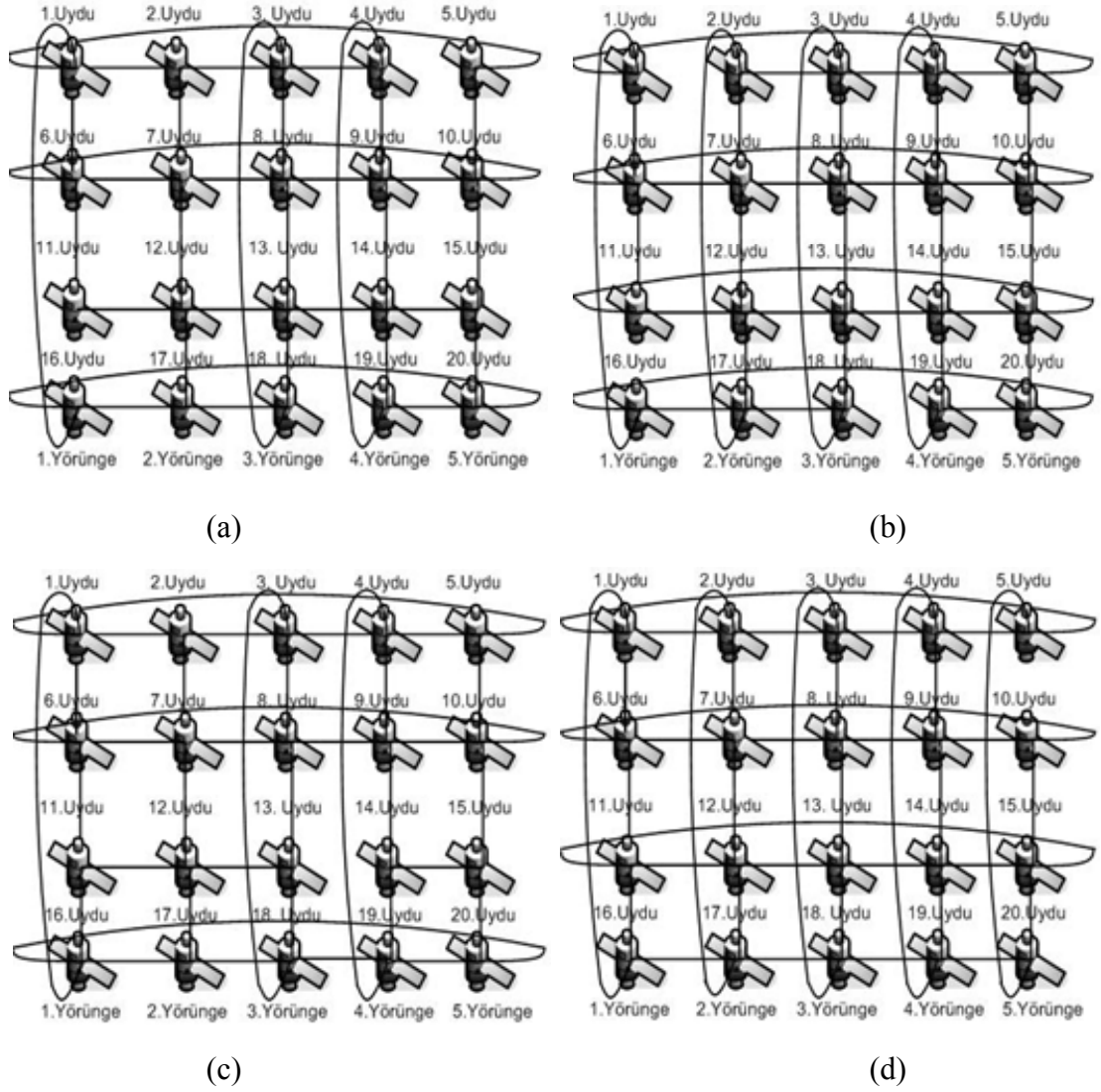
3.3.2. Dinamik Topolojilerin Elde Edilmesi

Yeni algoritmada farklı uydu sayılarına sahip birçok AY uydu sistemi kullanılmıştır. Bir AY uydu sisteminin de değişik zaman aralıklarında birbirinden farklı yapıları oluşturduğu görülmektedir. Yani zaman ilerledikçe sistemdeki uyduların komşu uydularla aralarındaki linkin durumu değişebilmektedir. Böylece yeni bir sistem yapısı ortaya çıkmaktadır. Uydu hareketlerinden kaynaklanan bu nedenden dolayı yeni yolların elde edilmesi gerekmektedir. Bunun yanı sıra önceden de bahsedildiği gibi en kalıcı yolların seçilmesi sağlanarak da bu problem aşılmaktadır. Algoritmada ana hedef, gecikme süresi düşük ve kalıcı olan yolu seçmeyi sağlayan kriterlerin fonksiyona katılmasıdır.

Şekil 3.6’da 20 tane uyduya sahip bir sistemde zamanla oluşan 4 farklı topoloji görülmektedir. Burada olduğu gibi uydu sistemlerinde zamanla bazı linkler aktifleşirken bazıları da işlevlerini kaybetmektedirler. Örneğin Şekil 3.6.a’da 1-2 uyduları arasındaki link bir süre sonra Şekil 3.6.b’deki gibi devre dışı kalabilmektedir. Benzer şekilde Şekil 3.6.c’de 13-14 uyduları arasındaki link bulunmamasına karşın Şekil 3.6.d’de bu link aktif olup işleme geçmiştir. Bu durum genel olarak topolojinin birkaç linkini etkilemektedir. Birçok linkte önceki zaman dilimindeki durum aynen korunmaktadır.

Şekil 3.6’daki uydu ağında dinamik yapı göz önünde bulundurularak algoritma yardımıyla birçok yol elde edilmektedir. Bunun için her bir topoloji diğerinden bağımsız olarak ele alınmaktadır. Bir topolojinin kendine ait farklı yolları elde edilir. Bu yollar üzerinde yeniden üretim ve çaprazlama gibi genetik işlemler yapılır.

Şekil 3.6’daki dört topolojiden yararlanarak yol hesabını elde etmek amacıyla paket iletiminin kaynağı 4. uydu, hedefi de 6. uydu olarak seçilsin. Topolojilerden elde edilen yollar Tablo 3.2’de görülmektedir.



Şekil 3.6: Bir uydu sisteminde zamanla farklı topolojilerin ortaya çıkması

Tablo 3.2: Dört farklı topolojiden elde edilen yollar

1.Topoloji	2.Topoloji	3.Topoloji	4.Topoloji
4,3,8,13,12,17,16,1,6	4,3,2,7,6	4,5,1,6	4,5,1,6
4,19,14,13,12,11,12,7,6	4,3,8,7,6	4,3,2,1,6	4,5,1,6
4,3,8,7,6	4,9,8,7,6	4,9,8,7,6	4,3,2,7,6
4,3,8,7,6	4,5,1,6	4,5,1,6	4,3,2,1,6
4,3,8,13,12,7,6	4,3,8,7,6	4,3,2,1,6	4,5,1,6

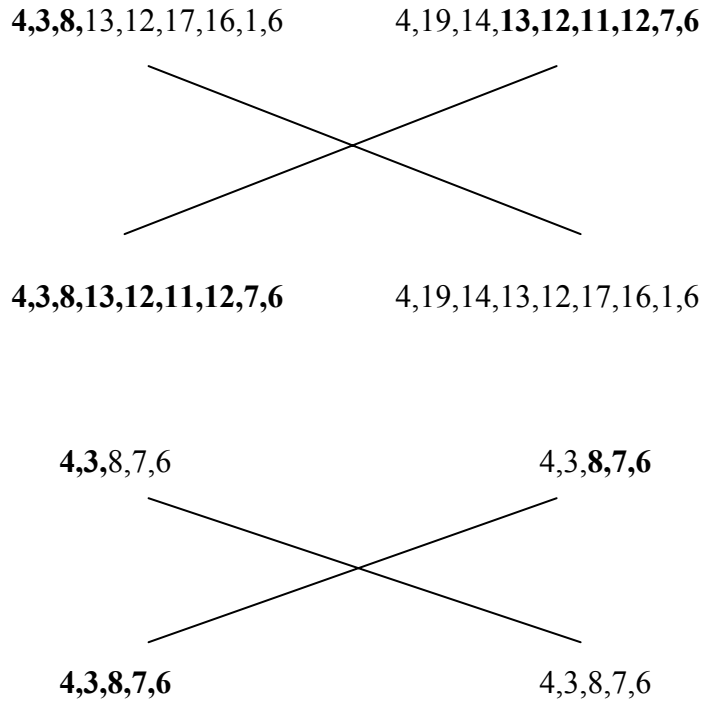
Herbir topolojideki yollar kendi aralarında genetik işlemlere tabi tutularak yeni jenerasyonlar elde edilmektedir. Genetik işlemlerin yapılması için öncelikle herbir

yolun “uygunluk değeri” hesaplanmaktadır. Bu değerin hesaplanması için gerekli adımlar Bölüm 3.3.1’de ayrıntılı şekilde açıklanmaktadır. Açıklanan işlemler dizisinin uygulanması sonucunda Tablo 3.2’deki 1.topolojiden elde edilen yolların uygunluk değerleri şu şekilde sıralanmaktadır:

Tablo 3.3: İlk topolojideki yollar ve bunların uygunluk değerleri

Yollar	Uygunluk değerleri
4,3,8,13,12,17,16,1,6	0,13
4,19,14,13,12,11,12,7,6	0,07
4,3,8,7,6	0,33
4,3,8,7,6	0,27
4,3,8,13,12,7,6	0,2

Uygunluk değerlerinden yola çıkılarak çaprazlama işlemi için birbirine uygun yol çiftleri elde edilir. Burada 1-2 ile 3-4 yol çiftlerinin çaprazlama işlemine girdikleri düşünüldüğü zaman aşağıdaki sonuçlar elde edilmektedir:



Ortaya çıkan dört yeni yol, yeni jenerasyonu oluşturmuştur. Aynı yolların çaprazlamaya girerek yine aynı yolların elde edilmesi genetik algoritma açısından uygun olmayan bir

durum değildir. Aksine çaprazlama işlemi uygunluk değeri yüksek olan yollar arasında olduğu için ilgili yolun korunarak bir sonraki jenerasyona katılması avantajdır. Önemli olan, uygunluk değeri yüksek olan bireyleri bir sonraki jenerasyona geçirebilmektir.

3.3.3. Dayanıklılık Kriterinin Uygunluk Değerine Etkisi

Bu aşamada yeni jenerasyonlardaki yollar aynı topoloji içerisinde önceki jenerasyonda bulunan yollarla karşılaştırılır ve dayanıklılık kriteri ölçülür. Bunun için bir sayaç yardımıyla bir yolun kaç kez tekrarladığı hesaplanmaktadır. Tablo 3.4'te görüldüğü gibi yeni jenerasyondaki yollar bir önceki jenerasyonda da varsa Dayanıklılık kriteri 1 artıyor. Bunun dışında bu değer, 1 olmaktadır. Dayanıklılık kriteri sayesinde, tekrarlanan yolların seçilme olasılığının artması sağlanmaktadır. Böylece etkinliği yüksek olan yolların sonraki jenerasyona etkileri de devam etmektedir. Tablo 3.4'te dayanıklılık kriterinin hesabı görülmektedir:

Tablo 3.4: Dayanıklılık kriterinin hesabı

1.Jenerasyon	DK1	2.Jenerasyon	DK2	3.Jenerasyon	DK3
A	1	A	2	A	3
B	1	B	2	H	1
C	1	E	2	E	3
D	1	F	1	F	2
E	1	G	1	I	1

Tablo 3.3'te bulunan yollar üzerinde genetik işlemlerin yapılması sonucu yeni jenerasyonlar elde edilmiştir. Bu jenerasyonlardaki yollar üzerinde dayanıklılık kriterlerinin hesaplanması ve bu sayıların uygunluk değerlerine olan etkisi Tablo 3.5'te özetlenmektedir.

Tablo 3.5'te belirtilen UD'ler tüm jenerasyon üzerinde toplamı 1 olacak şekilde hesaplanmaktadır. Bu değerlerin amacı, ilgili yolun tüm jenerasyon üzerindeki etkinlik değerini ortaya çıkarmaktır. Dayanıklılık kriteriyle çarpım işleminden sonra toplamı 1 olmadığı için bir kez de normalizasyon işleminden geçirilmektedirler. Tablo 3.5'teki son sütunda bu işlemin sonucu verilmektedir.

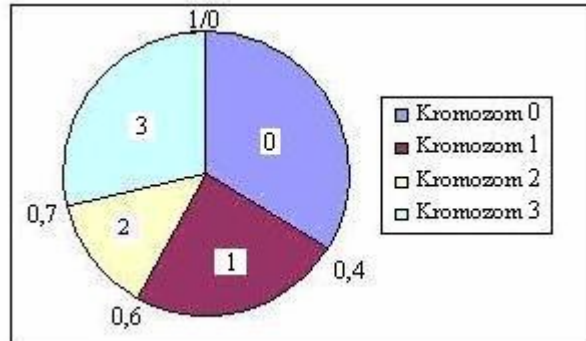
Tablo 3.5: Dayanıklılık kriteriyle birlikte hesaplanan yeni uygunluk değerleri

Başlangıç jenerasyonundaki yollar	Bir sonraki jenerasyondaki yollar	İkinci jenerasyondaki yolların Uygunluk Değerleri (UD)	Dayanıklılık kriteri (DK)	Yeni UD'ler (UD*DK)	Normalizasyon sonucu UD'ler
4,3,8,13,12,17,16,1,6	4,3,8,13,12,11,12,7,6	0,13	1	0,13	0,072
4,19,14,13,12,11,12,7,6	4,19,14,13,12,17,16,1,6	0,07	1	0,07	0,038
4,3,8,7,6	4,3,8,7,6	0,33	2	0,66	0,37
4,3,8,7,6	4,3,8,7,6	0,27	2	0,54	0,3
4,3,8,13,12,7,6	4,3,8,13,12,7,6	0,2	2	0,4	0,22

3.4. RULET ÇARKI YÖNTEMİNİN YOL SEÇİMİNDE KULLANIMI

Genetik algoritmada çaprazlama ve mutasyon gibi genetik işlemler yapılmadan önce başlangıç jenerasyonu, Yeniden Üretim işleminden geçirilmektedir. Kullanılan bu işlemin uygulama aşamasında Seçim yöntemleri bulunmaktadır. Bölüm 2.5.1.1'de açıklandığı gibi bu yöntemlerden biri olan Rulet Çarkında herbir bireyin yani kromozomun seçilme olasılığı $\frac{f(i)}{\sum_i f(i)}$ ifadesiyle hesaplanmaktadır. Rulet çarkının

değerleri bu şekildeki oranlardan oluştuğu için [0,1] aralığında çıkmaktadır. 0'dan başlayarak herbir kromozomun oranı eklenerek yeni sınır elde edilmektedir. Böylece son sınır 1 değerine karşılık gelmektedir. Uygunluk değerleri sırasıyla 0.4; 0.2; 0.1 ve 0.3 olan 4 kromozomun oluşturduğu Rulet Çarkı Şekil 3.7'de görülmektedir:



Şekil 3.7: Dört kromozomdan oluşan Rulet Çarkı

Uydu ağı yapısında elde edilen yolların her birinin bir uygunluk değeri bulunmaktadır. Bu değerlere göre oluşturulan Rulet Çarkından yararlanarak yeni bir jenerasyonun üretimi için kromozomlar arasında seçim yapılmaktadır. Oluşturulan uygulamada yeniden üretim işlemini yapmak için hazırlanan alt fonksiyon Şekil 3.8’de yer almaktadır.

```
void regeneration(double y_oran[pop],int yollar[pop][krom_size]){

    //y_oran değerlerine göre Rulet Çarkı oluşturulur. Rastgele sayılar atanıp
    //buna denk gelen Rulet dilimindeki kromozom Regeneration edilir.
    //Point'ler Rulet çarkındaki değerlerdir.
    int i,j,k,t=0,sayac=0;
    double reg,point[pop]={0};

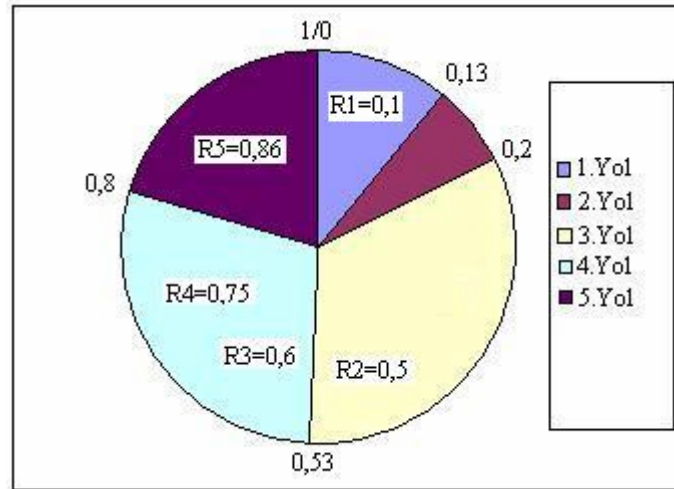
    printf("Point[0]=0\n");
    for(i=1;i<pop;i++){
        point[i]=point[i-1]+y_oran[i-1];
        printf("Point[%d]=%f\n",i,point[i]);
    }
    for(i=0;i<pop;i++){
        do{
            reg=((unsigned)rand())/32767.0;
        } while((reg<0.0)|| (reg>1.0));

        printf("Reg=%f\n",reg);
        for(j=0;j<pop;j++){
            if((reg>point[j])&&(reg<=point[j+1])){
                printf("Reg.edilen kromozomun indisi:%d\n",j);
                for(k=0;k<krom_size;k++)
                    yollar2[reg_ind][k]=yollar[j][k];
                reg_ind++;
                break;
            }
        }
    }
    printf("Reg_indis=%d\n",reg_ind);
}
```

Şekil 3.8: Yeniden Üretim alt fonksiyonu

Şekil 3.8’den anlaşıldığı gibi uydu ağındaki elde edilen yolların seçim oranlarından oluşan bir dizi ile bu yolları tutan bir başka dizi fonksiyona parametre olarak verilmektedir. Sonraki aşamada oranlar birbirine eklenecek şekilde Rulet Çarkının sınır değerleri bulunmaktadır. Bu değerler sırayla *point* dizisini oluşturmaktadır. Rastgele olarak bir sayı belirlenerek bu sayının hangi aralığa girdiği gözlenmektedir. Şekil 3.7’ye bakılarak bu kısım yorumlanırsa örneğin rastgele belirlenen sayı 0,63 ise, bu sayı *point*[2] ve *point*[3] aralığında yer alacağı için seçilen kromozomun indisi 2 olmaktadır. Böylece *Kromozom 2* yeni jenerasyonda tekrar ortaya çıkmaktadır.

Uygulamada normalde Tablo 3.5’te görülen başlangıç jenerasyonundan hemen sonra yeniden üretim işlemi gerçekleştirilmektedir. Bu üretim işleminde ortaya çıkan jenerasyonla birlikte çaprazlama işlemlerine başlanmaktadır. Böylece Tablo 3.3’teki ilk jenerasyon üzerinde oluşturulan Rulet çarkı Şekil 3.9’da görülmektedir:



Şekil 3.9: Uygulamadaki 5 yol için elde edilen Rulet Çarkı ve yol seçimi için belirlenen rastgele sayılar

Şekil 3.9’daki Rulet çarkında görüldüğü gibi rastgele belirlenen 5 sayı hangi kromozomun alanına giriyorsa, ilgili kromozom yeniden üretilip sonraki jenerasyonda tekrar ortaya çıkmaktadır. Bu işlem sonucunda yeni üretilen jenerasyon Tablo 3.6’da gösterilmektedir:

Tablo 3.6: Yeniden Üretim işlemi sonucu elde edilen jenerasyon

Başlangıç jenerasyonu	Yeni üretilen jenerasyon
4,3,8,13,12,17,16,1,6	4,3,8,13,12,17,16,1,6
4,19,14,13,12,11,10,7,6	4,3,8,7,6
4,3,8,7,6	4,3,8,7,6
4,3,8,7,6	4,3,8,7,6
4,3,8,13,12,7,6	4,3,8,13,12,7,6

3.5. ÇAĞRILARIN BLOKE OLMA OLASILIKLARI

Statik uydu ağında tek 1 paketin iletimi söz konusu idi. Bu nedenle herhangi bir trafik yapısına göre paket gelişi olmamıştır. Dinamik AY uydu topolojisinde ise paketler uydulara bazı trafik değerlerine göre gelirler. Bu aşamada trafik ve konuma bağlı olan Çağrı Bloke Olma Olasılığı (ÇBO-Call Blocking Probability) değerleri devreye girmektedir. Burada bir uyduya gelen çağrı paketinin, bir linkteki trafik yoğunluğu nedeniyle hedefe ulaştırılamadığı görülmektedir. Böylece ortaya çıkan bloke olma olasılıkları için herbir uyduya gelen paketlerle uydudan çıkan paketler hesaba katılır. Bu işlemlerde kuyruk sistemlerinden yararlanılmaktadır. Bu sistemler bazı akış işlemlerinin gerçekleştiği dinamik yapılarda önemli rol oynamaktadır. Günlük olayların birçoğunda bu sistemler için örnekler bulunmaktadır. Örneğin telefon ve telgrafların iletimi, bir yoldaki otomobillerin hareketleri, marketteki müşterilerin alışverişleri, zaman paylaşımı bilgisayar sistemlerinde programların akışı gibi birçok aktivitede kuyruk sistemlerine rastlanmaktadır. Kuyruk sistemlerinde yapı olarak bir ağ ve birtakım trafik oranlarıyla bu ağa gelen paketler bulunmaktadır. Trafik oranları, olasılık hesaplamalarıyla ilgili bazı metotlardan faydalanılarak gözlemlenebilmektedir. Bölüm 3.5.1’de bu konuyla ilgili örnek yöntemlerden bahsedilmektedir.

3.5.1. Olasılık Süreçleri ve Dağılımlarıyla İlgili Bazı Önemli Kavramlar

3.5.1.1. Poisson Dağılımı

Bir olasılık dağılım kuralıdır. Belirli bir zaman aralığında bir olayın olma olasılığını vermektedir. (0,t) zaman aralığında k tane olayın olma olasılığı $P_k(t)$ şu şekilde hesaplanmaktadır (Kleinrock, 1975):

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad (k \geq 0, t \geq 0) \quad (3.7)$$

Bu formülden anlaşılacağı gibi olayların bir saniyedeki ortalama meydana geliş oranları λ ise t aralığında ortalama meydana geliş oranları da λt olmak zorundadır.

Bir sisteme elemanların girişi sabit bir oranda olduğunda homojen dağılım olmaktadır. Sistemlerde normalde geliş aralıkları ve oranlar belirli bir değerde olmadığı için homojen olmayan yapı söz konusudur. Homojen olmayan Poisson dağılımında ($N(t)$) geliş oranını gösteren $\lambda(t)$ değeri, zamana bağlı negatif olmayan ve integrali alınabilen bir fonksiyondur. Sistemin ortalama değer fonksiyonu olarak tanımlanan servis oranı şu şekilde hesaplanmaktadır (Johnson ve diğ., 1991):

$$\mu(t) \equiv E[N(t)] = \int_0^t \lambda(z).dz \quad (t \geq 0) \quad (3.8)$$

Homojen olmayan Poisson dağılımı konusunda yapılan bir çalışmada homojen yapıdan heterojen yapıya geçişin mümkün olduğu belirtilmektedir (Massey ve diğ., 1996). Bu fikre göre tüm zaman aralığı, küçük zaman dilimlerine bölünmektedir. Herbir zaman dilimindeki geliş oranının lineer olduğu düşünülmektedir. Örneğin bir gün içerisindeki telefon çağrılarının geliş oran fonksiyonunun lineer olmadığı görülmektedir. Buna karşılık belirli saat aralıklarında bu fonksiyonun lineer olduğu görülmektedir. Yapılan simülasyon yardımıyla çağrılarının belirli saat aralıklarındaki geliş oranlarının sabit değerde olmasa da lineere yakın olduğu görülmektedir. Bu şekilde lineer sistem elde

edildikten sonra trafik dağılımıyla ilgili fonksiyonda parametre tahminini yapmak daha da kolaylaşmaktadır.

Homojen olmayan Poisson dağılımının kullanıldığı başka bir çalışmada da daha önceden tasarlanan Poisson-Arrival-Location Model (PALM) yapısından yararlanılmıştır (Massey ve Whitt, 1994). PALM yapısı kullanılarak bir karayolundaki kablosuz aygıtların iletişimi incelenmektedir. Burada tek yöllü trafik düşünülerek daha karmaşık sistemler için temel oluşturabilecek bir yapı tasarlanmaktadır. Standart PALM yapısına göre çağrılar, homojen olmayan Poisson dağılımına göre sisteme gelmektedir. Çağrıların iletimi sırasında ortaya çıkan gelişim süreci ve aktarım durumları gözlemlenmektedir. Bir çağrının aktarım durumu, zamana ve mekana bağlı bir fonksiyon şeklinde hesaplanmaktadır. Aktarılan veya yeni başlayan çağrılardan bloke olanların oranı hesaplanmaktadır. Ayrıca tüm araçların sabit hızda olduğu düşünülerek geliş fonksiyonundan yola çıkarak çağrı yoğunluğuyla ilgili fonksiyon elde edilmektedir. Araçların düşünme, çağrı iletme ve tamamlama zamanları kullanılarak bulundukları konum elde edilmektedir. Belirli konumlardaki çağrı başlangıç sayısı, çağrı bitiş sayısı ve aktif çağrı sayısı gibi değerler de Poisson integrallerinden yararlanılarak çözülmektedir. Bu temel sistem kullanılarak daha karmaşık yapılar da analiz edilebilir. Bunun için büyük sistemlerin küçük alt sistemlere bölünmesi düşünülmektedir. Örneğin aynı yönde ilerleyen ama farklı başlangıç noktalarına sahip araçlar için ya da zıt yönde giden araçlar için ayrı ayrı analiz yapılarak bu durum gerçekleştirilebilmektedir. Kablosuz haberleşme sistemlerine karayolu PALM yapısını geçirmek için de sistem hücrelere bölünmektedir. Herbir hücrenin taşıdığı çağrı sayısının ve izin verdiği bloke oranının limitleri vardır. Bunlardan yola çıkılarak bloke olma olasılıkları hesaplanmaktadır.

3.5.1.2. Markov Modellemesi

Olasılık süreçlerinden biridir. 1907’de A.A.Markov tarafından ortaya çıkarılmıştır. Tahmini bir süreç oluşturan rastgele sayılar arasındaki bağımlılığı ortaya koymaktadır. Bu modelde durumlar, zaman çizelgesine uygun olarak tek tek işleme tabi tutulmaktadır. Bu şekilde ayırık durum uzayındaki bir Markov süreci, Markov zinciri adını almaktadır. Markov zinciri, ayırık-zaman ve sürekli-zaman olmak üzere iki türdür. Ayırık-zaman zincirinde herbir durum kendinden bir önceki duruma bağlıdır, bu nedenle

daha önceki diğer durumlarla bir bağlantısı yoktur. Durumları oluşturan elemanlar da rastgele değerlerdir. Böylece rastgele X_{n+1} değeri kendinden önceki X_n değerine bağımlıdır.

Ayrık-zaman Markov zincirindeki durum değişikliklerinin olduğu yerlerde sınırlar $0,1,2,\dots,n,\dots$ tamsayılarına karşılık gelecek şekilde düzenlenince sürekli-zaman Markov zinciri elde edilmektedir. Bu zincirde durum değişiklikleri herhangi bir anda olabilmektedir. Bir durumun kendinden önceki duruma bağımlılığı olduğu için durumların zaman süreçleri net olarak bilinmelidir. Bu süre üssel olarak hesaplanır (Kleinrock, 1975).

Analitik olarak Markov özelliği şu şekilde formüllere dönüştürülmektedir:

$$\begin{aligned} P[X(t_{n+1})=x_{n+1} | X(t_n)=x_n, X(t_{n-1})=x_{n-1}, \dots, X(t_1)=x_1] \\ = P[X(t_{n+1})=x_{n+1} | X(t_n)=x_n] \end{aligned} \quad (3.9)$$

Burada $t_1 < t_2 < \dots < t_n < t_{n+1}$ şeklindedir ve x_i ayrık konum uzayında yer almaktadır. Markov süreçleri için elde edilen ifadeler kuyruk teorisi konusuna temel olmaktadır.

3.5.1.3. Yarı-Markov Süreci

Ayrık-zaman Markov zincirinde zaman doğrusundaki herbir aralıkta bir durumdan başka bir duruma geçiş söz konusudur. Bir durum için harcanan zaman geometrik dağılıma uygundur. Durumlar arasındaki geçiş olasılıkları da isteğe bağlı olarak değişmektedir. Bir durumda bulunma zamanının dağılımı keyfi olarak ayarlanırsa, ayrık-zaman yarı-Markov süreci ortaya çıkmış olur. Böylece durum geçişleri sırasındaki zaman, keyfi olasılık dağılımına göre bulunmaktadır. Burada zaman doğrusundaki herbir noktada işlemler Markov zincirine göre oluşmaktadır. Bu durum, gömülü Markov zinciri olarak da düşünülebilir.

Sürekli-zaman yarı-Markov sürecinde de zamanın herhangi bir anında durum geçişi olabilmektedir. Markov sürecinde bir durum sırasında geçen zamanın üssel dağılıma uygun olduğu bilinmektedir. Yarı-Markov sürecinde ise keyfi dağılım uygulanmaktadır.

Olasılık değerlerini veren formülleri elde etmek amacıyla m tane sunucuya sahip bir sistem (M/M/m/m) düşünülmektedir. M/M/m/m ifadesindeki ilk M/M değerleri, Markov modeline göre sisteme geliş ve sistemden çıkış oranlarını vermektedir. Üçüncü değer olan m , sistemin sunucu sayısını göstermektedir. Bu değer aynı zamanda servis verilebilecek eleman sayısının maksimum değerini de belirtmektedir. Son değişken olan m de sisteme bu sayıdaki elemandan fazlası gelirse onların reddedileceğini yani sisteme gelen elemanlar için bekleme kuyruğunun olmadığını ifade etmektedir. Böylece m sunucuya sahip böyle bir sistemde sunucuların tümü meşgulse yeni gelen eleman sistemden çıkarılacaktır. Böyle bir model için Şekil 3.10'daki sisteme dayanarak aşağıdaki doğum ve ölüm katsayıları seçilmektedir:

$$\lambda_k = \begin{cases} \lambda & k < m \\ 0 & k \geq m \end{cases} \quad (3.10)$$

$$\mu_k = k\mu \quad k = 1, 2, \dots, m \quad (3.11)$$

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \quad k=0, 1, 2, \dots \quad (3.12)$$

Şekil 3.10'daki sistem için (3.12) denkleminde yararlanarak k tane sunucunun meşgul olma olasılığı şu şekilde hesaplanmaktadır:

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} \quad k \leq m \quad (3.13)$$

Bu denklem aşağıdaki koşullu yapıyla da ifade edilebilir:

$$p_k = \begin{cases} p_0 \left(\frac{\lambda}{\mu} \right)^k \frac{1}{k!} & k \leq m \\ 0 & k > m \end{cases} \quad (3.14)$$

(3.14) formülü p_0 değerini elde etmek için kullanıldığında aşağıdaki denklem elde edilir:

$$p_0 = \left[\sum_{k=0}^m \left(\frac{\lambda}{\mu} \right)^k \frac{1}{k!} \right]^{-1} \quad (3.15)$$

Sonuçta m tane sunucunun meşgul olma olasılığı şu şekilde hesaplanmaktadır (Kleinrock, 1975):

$$p_m = \frac{(\lambda / \mu)^m / m!}{\sum_{k=0}^m (\lambda / \mu)^k / k!} \quad (3.16)$$

Erlang-B formülü olarak adlandırılan bu ifadenin uydu ağlarında j . link için hesabı şu şekilde olmaktadır:

$$p_j = \frac{(\lambda_j / \mu)^{m_j} / m_j!}{\sum_{k=0}^{m_j} (\lambda_j / \mu)^k / k!} \quad (j \in Z^+) \quad (3.17)$$

Yeni yönlendirme algoritmasında kullanılan uydu topolojilerinde olasılık dağılımının Erlang-B formülüne uygun yapıda olduğu görülmektedir. Erlang-B'nin kullanılma nedeni, topolojide uyduların içerisinde paketleri bekleterek uygun vakitte gönderim işlemini gerçekleştiren kuyruk yapılarının olmadığı düşünülmesidir. Böyle bir kuyruk yapısı olduğu zaman Erlang-C formülleri kullanılmaktadır.

Erlang-B formülünün kullanıldığı yerlerde problemin amacına göre sonuçlar elde edilmektedir. Örneğin WATM kablosuz uydu ağlarındaki kanal yönetimi konusunda yapılan çalışmada bu formül yardımıyla sistemde belirli sayıda müşterinin bulunma

olasılığı hesaplanmaktadır. Sisteme geliş ve sistemden çıkış oranlarının hesaba katılmasıyla bulunan bu olasılık yardımıyla çağrının bloke olma durumu bulunmaktadır. Kanal tahsisi yapılırken sistemdeki belirli sayıdaki müşteriye göre yeni çağrıların kabul edilip edilmeyeceği konusu kararlaştırılır. Eğer yeterli miktarda bant genişliği yoksa bu çağrı reddedilir. Aktarılan çağrılar için ise geçit kanallarında yer ayrılmaktadır. Kanallarda belirli bir limite ulaşıldıktan sonra gelen yeni çağrılar da reddedilir. Çağrıların ortalama bekleme süreleri ve geliş oranları da önemli değişkenler arasındadır. Böylece bu probleme getirilen çözümde, Erlang-B formülü yardımıyla olasılık hesaplarının yapılması sonucu katkıda bulunmaktadır (Chen ve Jamalipour, 2001a).

Sistem modeline Erlang formülünün katıldığı başka bir çalışmada bu sayede sistemde belirlenen sabit bir sayıda müşterinin olma olasılığı hesaplanmaktadır. Burada λ değeri yeni çağrı geliş oranını, μ değeri de ortalama çağrı bekleme süresini göstermektedir. Tasarlanan LEO/MEO SATATM ağlarında herbir uydunun bir ATM gibi davrandığı düşünülerek trafik uyumlu bir yönlendirme algoritması ortaya çıkarılmıştır. Olasılık hesaplamaları yardımıyla yeni bir çağrının veya aktarılan bir çağrının bloke olma olasılıkları bulunmaktadır. Haberleşmeye uygun kanal sayısı, trafik ağırlık birimi gibi değerler de hesaplara katılarak yol seçim aşamasına gidilmektedir. Burada da “kayma genişliği” ismiyle belirlenen bir kritere göre tolerans değeri verilerek o tolerans aralığındaki yollar seçilmektedir (Chen ve Jamalipour, 2001b).

3.5.2. Uydu Ağındaki Herbir Linkin ÇBO Değerinin Hesaplanması

Uydu ağ yapısında Markov modellemesinden yararlanarak yola çıkılmaktadır. Markov modelindeki bir olayın kendinden hemen önceki olaya bağımlı olması gibi, uydu ağındaki yol belirlemesinde de bir linkin seçimi, bir önceki linkin seçilmesine bağlıdır. Ayrıca belirli bir zaman süresince bir olayın olma olasılığı hesaplanmadığı için Poisson dağılımından yararlanılmamaktadır. Poisson dağılımında bir zaman süresince ortalama olay sayıları kullanıldığı için belirli bir andaki veya tek bir olaya bağlı ölçümler yapılamamaktadır.

(3.17) formülü, uydu ağındaki yollara uygulanabilmektedir. Böylece herhangi bir yola ait herbir linkin trafik açısından dolu olması, yani yolun paketleri iletememesi durumunun olasılığı elde edilmiş olur. Formüldeki m_j sayısının, bir yol üzerindeki

linklerin her birinin sahip olduğu sunucu sayısını verdiği düşünülmektedir. Bu sayı uygulama içerisinde herbir link için ayrı ayrı üretilmektedir. Linkin aktif olma olasılığı da $(1-p_j)$ şeklinde hesaplanmaktadır. Ardışık linklerin birbirinden bağımsız olduğu düşünülerek olasılıklar birbirleriyle çarpılarak ilgili yolun seçilme olasılığı elde edilmiş olur. Böylece l adet linke sahip olan bir yolun P olasılığı aşağıdaki şekilde elde edilmektedir:

$$P = (1 - p_1).(1 - p_2).....(1 - p_l) \quad (3.18)$$

Bu denklemin sadeleştirilmesiyle aşağıdaki formül elde edilmektedir:

$$P = \prod_{t=1}^l (1 - p_t) \quad (3.19)$$

Denklemdaki p değeri Çağrı bloke olma olasılığı olduğu için P değerinin yeni gösterimi şu şekilde de olabilmektedir:

$$P = \prod_{t=1}^l (1 - CBO_t) \quad (3.20)$$

Böylece elde edilen olasılık değeri (P), herbir yolun iletim yapabilme olasılığını yani o yolun aktiflik durumunu vermektedir. Herbir linkin sunucu sayısını gösteren m değeri birbirinden farklıdır. Bu değer, ilgili linkin iletim işini yapan sunucu sayısını verdiği için linkler arasında farklılık olmaktadır. Dinamik bir yapıda düşünüldüğü zaman, linklere gelen paket sayısı ile linklerden çıkan paket sayısının da birbirinden farklı olduğu görülür. Uyduların kapsama alanlarında sisteme çağrı paketlerinin gelişi, bölgeden bölgeye değişiklik göstermektedir. Coğrafik ve sosyal yapılara göre bazı bölgelerde yoğun iletişim olurken bazı yerlerde de seyrek iletişim söz konusudur. Böylece doğum katsayısı olarak adlandırılan λ değeri bölgelere bağlı olarak değişiklik göstermektedir. Yani uydu ayak izinin kapsadığı bölgeye göre sisteme giren paket sayısı değişmektedir. Sürekli hareket halindeki bir sistemde trafik gibi ayırt edici özelliklerin de bulunması, yol seçimi sırasında işlemlere katılmaktadır. Öte yandan ölüm katsayısı

denen μ değeri de uyduların servis kapasitesine bağlı olduğu için sistemden çıkan paket sayısı özdeş uydular için eşit olmaktadır.

Uygulamada ÇBO olasılığını bulmak için öncelikle her bir linkin m değeri belirlenmektedir. Bir kaynak uydudan yola çıkılarak hedef uyduya doğru elde edilen yolların her birinin link sunucu sayıları bir diziye atanır. Sonra o yola ait ÇBO olasılığı için alt fonksiyona bu dizi yollanarak Markov modelinin hesaplamaları yapılır. 20 uydulu bir AY uydu sisteminde kaynak uydu olarak 7, hedef uydu olarak da 19 indisli uydular ele alındığı zaman Şekil 3.11'deki 5 yol elde edilen yollar arasında bulunmaktadır:

```

1.Yol: 8,3,4,9,4,9,4,5,20.
2.Yol: 8,3,4,9,4,5,20.
3.Yol: 8,7,8,7,12,11,16,20.
4.Yol: 8,3,2,17,16,20.
5.Yol: 8,13,18,19,14,15,10,15,20.

1.Yol için linklerin sunucu sayıları:
Sunucu_say[8][3]=3
Sunucu_say[3][4]=5
Sunucu_say[4][9]=4
Sunucu_say[9][4]=4
Sunucu_say[4][9]=4
Sunucu_say[9][4]=4
Sunucu_say[4][5]=1
Sunucu_say[5][20]=5

1.Yolun Sunucu sayılarından oluşan yeni dizi:
yeni_sunucu_say[1][10]={3,5,4,4,4,4,1,5,-1,-1}

```

Şekil 3.11: Linklerin ve yolların sunucu sayıları

Şekil 3.11'deki gösterime göre tüm yolların sunucu sayıları şu şekilde elde edilmiştir:

```

yeni_sunucu_say[1][10]={3,5,4,4,4,4,1,5,-1,-1}
yeni_sunucu_say[2][10]={3,5,4,4,1,5,-1,-1,-1,-1}
yeni_sunucu_say[3][10]={5,5,5,5,5,4,4,-1,-1,-1}
yeni_sunucu_say[4][10]={3,3,1,5,4,-1,-1,-1,-1,-1}
yeni_sunucu_say[5][10]={2,5,5,4,1,2,2,3,-1,-1}

```

Dizilerin boyutu birbirine eşittir ve bu boyut sabit bir tamsayıyla sınırlıdır. Örnekte bu limit 10 eleman uzunluğundadır. İlgili dizinin o boyuta ulaşacak sayıda elemanı yoksa bu kısımlara boş eleman anlamına gelen “-1” terimi konulmuştur.

Bundan sonraki aşamada herbir yola ait olarak elde edilen dizi ÇBO hesabının yapıldığı fonksiyona gönderilmektedir. Böylece herbir yola ait dizinin herbir elemanına karşılık gelen (1-ÇBO) değeri hesaplanarak birbiriyle çarpılır. Sonuçta o yolun P olasılık değeri elde edilir. $\lambda_j=7$, $\mu=9$ olacak şekilde bu ifadelerin hesap işlemleri 1. yol örnek alınırsa şu şekilde olmaktadır:

$$ÇBO_1 = \frac{(\lambda_1 / \mu)^3 / 3!}{\sum_{k=0}^3 (\lambda_1 / \mu)^k / k!} = \frac{(7/9)^3 / 3!}{\sum_{k=0}^3 (7/9)^k / k!} = 0,036327$$

8 linke sahip olan 1.yolun tüm linkleri için aynı işlem yapıldıktan sonra bu yolun kullanılma olasılığı aşağıdaki şekilde hesaplanmaktadır:

$$P_1 = \prod_{i=1}^8 (1 - ÇBO_i) = (1 - ÇBO_1) \cdot (1 - ÇBO_2) \cdot (1 - ÇBO_3) \cdot (1 - ÇBO_4) \cdot (1 - ÇBO_5) \cdot (1 - ÇBO_6) \cdot (1 - ÇBO_7) \cdot (1 - ÇBO_8)$$

$$P_1 = (1 - 0,036327) \cdot (1 - 0,001090) \cdot (1 - 0,007014)^4 \cdot (1 - 0,4375) \cdot (1 - 0,001090)$$

Sonuçta $P_1 = 0,52587$ olarak elde edilmektedir. Daha sonra bu olasılık değeri, yolun toplam gecikme süresiyle işleme girmektedir. Böylece bir yolun kullanılma durumunda gecikme süresinin ne kadar olduğu hesaplanmış olmaktadır.

3.6. ÇBO DEĞERLERİNİN ZAMAN VE KONUMA BAĞLI HESABI

Bölüm 3.5’te paketlerin trafik dağılımının tek biçimli olduğu düşünülerek ÇBO hesaplamaları yapılmıştır. Başka bir deyişle dünya üzerinde tüm bölgelerde eşit oranda trafiğin sisteme girdiği ve yine eşit oranda trafiğin sistemden çıktığı düşünülmektedir. ÇBO hesaplamalarından önce zaman kavramı olarak sadece çağrı süresi düşünülmüştür.

ÇBO formüllerinde de konum kriteri olarak kullanıcının bulunduğu yere göre değişen λ_j değeri mevcuttur.

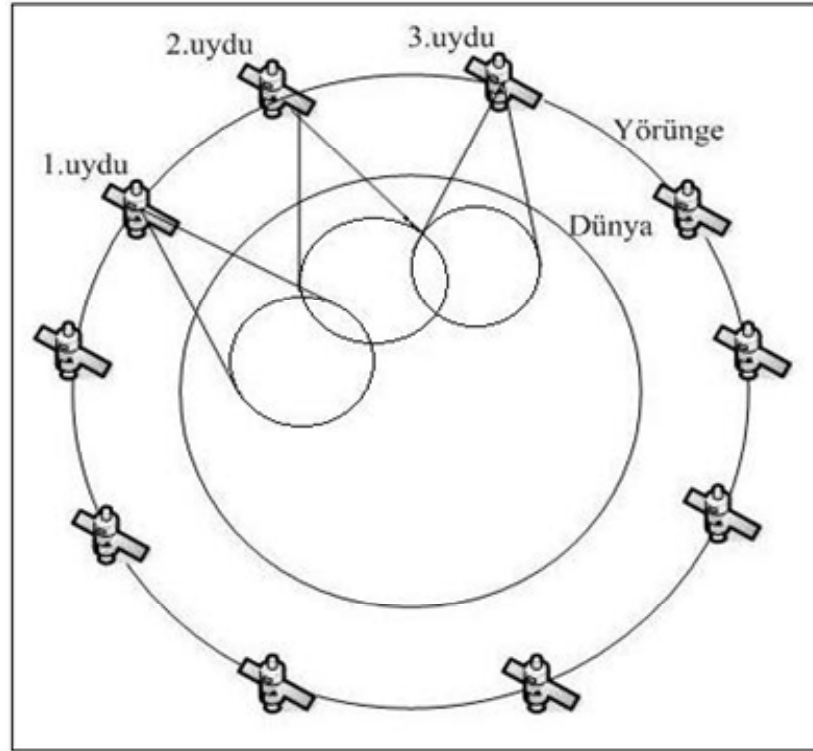
Bu bölümde ise algoritmaya içinde bulunulan zaman dilimi ve yeryüzündeki konumun alan olarak değeri eklenmektedir. Bu yeni değerler ÇBO fonksiyonuna katılmaktadır. Böylece gerçek hayatta gündüz ve gece saatlerindeki haberleşme yoğunluğundaki farklılıklar formüllerde kullanılan zaman değerine göre gözlenebilmektedir. İletişim sırasında paketlerin sisteme gelişinin zamana göre modellenmesiyle ilgili örnek bir çalışma mevcuttur (Alnowibet, 2004). Bu çalışmada sabit olmayan Erlang kayıp modeli (nonstationary Erlang loss model) kullanılmaktadır. Sinüzoidal fonksiyonlardan yararlanılarak zamana bağlı geliş oranları elde edilmektedir. AY uydu sistemleri üzerinde algoritmanın uygulanmasıyla uyduların senkronize hareketlerine ve kendi kapsama alanlarını geçme sürelerine göre geliş oranları elde edilmektedir. Uydular arasındaki aktif çağrı sayısına ve linklerin kapasitesine göre analiz yapılarak ÇBO değerleri hesaplanmaktadır.

Zamanın yanı sıra uydunun kapsadığı alana göre ortaya çıkan paket trafiği de ÇBO değerine etki eden bir başka faktör olarak gözlemlenmektedir. Örneğin yoğun şehir hayatının olduğu bölgelerde kablosuz iletişimin çok olması beklenirken, okyanus gibi alanlarda iletişim seyrek olmaktadır. Böylece iletişim yoluyla sisteme giren ve sistemden çıkan paket sayıları, konuma bağlı olarak değiştiği için trafik dağılımında gerekli değişiklikler yapılmaktadır. Bu amaçla λ_j değeri konumla bağlantılı şekilde trafik yoğunluğuna göre değişmektedir. Kapsama alanını bulurken farklı bir geometrik yaklaşım uygulayarak matematiksel denklemler elde edilmektedir. Elde edilen yeni formüllerde kullanılan değişkenler sayesinde uyduların yörünge uzunlukları, dünyanın çevresi, sistem periyodu gibi farklı parametreler devreye girebilmektedir. Bundan dolayı gerçek hayattaki uydu sistemleri için kullanılabilecek yeni formüller elde edilmiş olmaktadır.

3.6.1. Konum ve Kapsama Alanıyla İlgili Kavramlar

Kablosuz haberleşme sistemlerinde, dolayısıyla da uydu haberleşmesinde önemli bir konu kapsama alanının kontrolünün sağlanmasıdır. Sürekli hareket halinde olan uydu sistemlerinin servis verdikleri alanların konum olarak ifade edilmeleri karmaşık

olmaktadır. Bununla birlikte dinamiklikten kaynaklanan sürekli zamanlı konum güncellemeleri olmaktadır. Bu güncellemeleri de içine alan önemli bir kavram konum yönetimidir. Konum yönetimiyle ilgili birçok yaklaşım mevcuttur. Şekil 3.12’de bir yörünge üzerinde bulunan uyduların ayak izlerinin görünümü yer almaktadır.

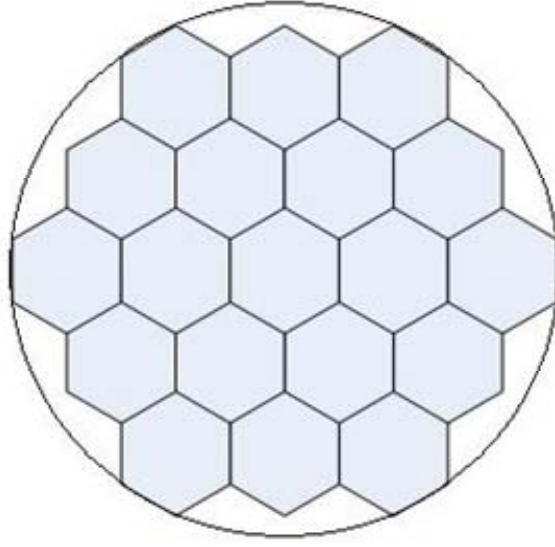


Şekil 3.12: Bir yörüngedeki uyduların ayak izleri

Şekil 3.12’de görüldüğü gibi aynı yörüngedeki uyduların ayak izleri birbirleriyle kesişebilmektedir. Benzer şekilde birbirinden farklı yörüngelerde bulunan uyduların da ayak izlerinin kesişmesi mümkündür. Dinamiklikten kaynaklanan bu durum, uydulara gönderilen sinyallerin bir zaman sonra karışmasına neden olur. Böylece yer istasyonunun istediği servis gerçekleşmemiş olmaktadır. Dolayısıyla çakışmanın olduğu uydular arasındaki linkin pasif hale gelmesi, başka bir deyişle oradaki çağrının bloke durumu söz konusu olur. Dinamik yapıdan dolayı elde edilen farklı topolojilerin temel nedeni de budur.

Kapsama alanının hücre yapılarıyla gösterilmesi sayesinde konum bilgisinin analitik olarak ifade edilmesi kolaylaşmıştır. Böylece önceden bahsedilen konum yönetimi için yapılan çalışmalara kolaylık kazandırılmıştır. Şekil 3.13’te tek bir uyduya ait ayak izinin

hücre yapısı görülmektedir. Bu yapıyı ortaya çıkaran etmen, uydularda bulunan çok spot huzmeli (multi-spot-beam) antenlerin fonksiyonudur. Böylece bir uydu birçok hücreyi kapsamaktadır. Herbir hücrenin maksimum görünürlük süresi yaklaşık 1~2 dakikadır. AY uydu sistemlerinin yaygın kullanılmasıyla birlikte bu konuda geliştirmeler yapılmaya başlanmıştır. Bu sistemler sayesinde dünya çapında bir kapsama oluşturmak amacıyla mobil terminalin konumunun sürekli olarak güncellenmesi gerekmekte ve bu bilginin o andaki sinyal iletimi için kullanılması sağlanmaktadır. Bu işlemler yapılırken hücrelerin hareketli olduğu göz önünde bulundurulmaktadır. Uydu sistemlerinin karasal sistemlerden bir farkı da burada ortaya çıkar. Uydu sistemlerinde konum güncellemesini ve çağrı akışını uydunun hareketi belirlerken karasal sistemlerde kullanıcının hareketi belirler. Bu bilgilerden yararlanarak AY uydu yapılarında dinamik harekete dayalı konum yönetimi için yapılan bir çalışmada hücrelerin yapıları farklı bir şekilde tasarlanarak konum güncelleme maliyeti düşürülmektedir (Liang ve Nai-Thong, 2002). Bu çalışmada belirtildiği gibi konum yönetimi temel olarak iki faktörden oluşur. Bunların birincisi konum güncellemeleri, diğeri de yer istasyonunun çağrısıdır. Karasal yapılarda tüm ağ konum alanlarına ayrılır ve herbir alanın birkaç hücresi bulunur. Kablosuz bir terminal yeni bir alana girdiğinde bu terminalin kendi konumunu önceki veritabanında güncellemesi gerekmektedir. Çağrı zamanında ise kullanıcının konumunun tespit edilmesi için tüm hücreler ağ tarafından sayfanır. Bu adımlar, bant genişliği ve terminallerin işlem gücü açısından çok maliyetli olmaktadır. Bu maliyeti düşürmek için konum alanını oluşturan hücreler, iki spot huzmesini içine alan meta-hücre isimli yapılardan meydana getirilmiştir. Kablosuz terminalin hücre sınırlarında geçiş aralığı genişletilerek herbir değişimde daha fazla mesafe kat edilmesi sağlanmaktadır. Böylece iki normal hücreyi kapsayan bir meta-hücre yapısıyla terminalin hücreden hücreye geçiş sayısı azalmış olmaktadır. Algoritmanın işleyişi sırasında iki farklı uydudan gelen iki hücreyi kapsayan meta-hücrelerde güç sarfıyatı fazla olmaktadır. Bu problemi çözmek için de zorlanmış konum güncelleme yapısı devreye girerek iki farklı uydunun arasındaki sınır geçişi sırasında konum güncelleme işlemini yapmaktadır.



Şekil 3.13: Bir uyduya ait ayak izinin hücresel gösterimi

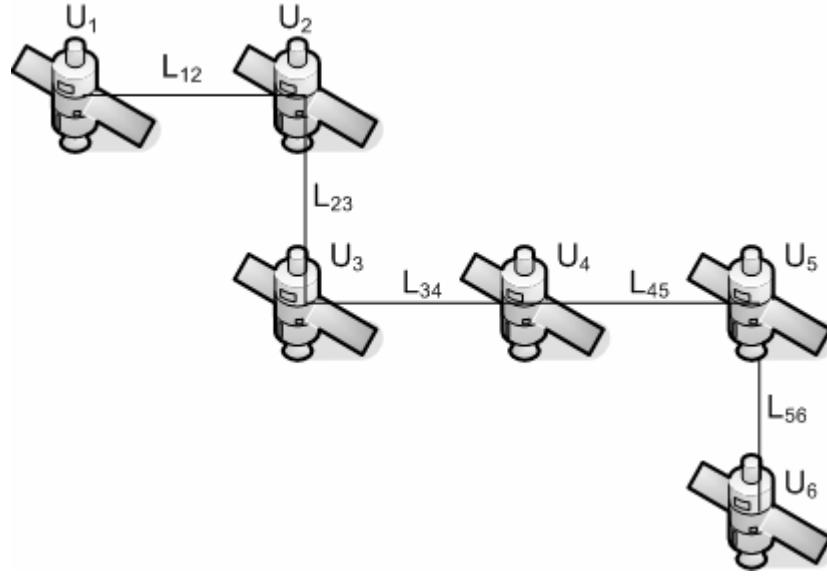
Kablosuz sistemlerde terminallerin konumlarıyla ilgili bir başka çalışmada ise bir karayolu modeli üzerinde kablosuz aygıtların bağlantısı gerçekleştirilmektedir (Massey ve Whitt, 1994). Kablolu bir sistemde servisler çağrı süreleriyle ifade edilirken kablosuz sistemlerde konum ve zaman birlikte devreye girmektedir. Konum uzayını gösteren alan sonlu hücre parçalarına ayrılır ve bir zaman fonksiyonu yardımıyla da kablosuz iletişimde verilen servis ifade edilir. Tek yönlü bir karayolu modelinde homojen olmayan Poisson özelliğine uygun şekilde araç girişi gözlemlenmektedir. Belirli bir zamanda karayoluna giriş yapan araçların hareketleri, kararlı bir fonksiyon yardımıyla hesaplanır ve belirli bir süredeki konumları tespit edilir. Araçların hızlarından ve geçen zamandan yola çıkılarak konum belirlenir. Belirli bir zamanda belirli bir mesafedeki aktif çağrılar, çağrı başlangıçları, çağrı bitişleri ve aktarım sayıları gibi değerler de hesaplanmaktadır. Örneğin aktarım sayısı hesabıyla bir hücreye servis veren baz istasyonundan başka hücredeki baz istasyonunun alanına geçiş sayısı bulunmuş olur. Bir uzay-zaman grafiği sayesinde araçların düşünme çağrı süreleri konuma bağlı şekilde gösterilmektedir. Karayolunun hücrelere bölündüğü düşünülerek her bir hücreye bir baz istasyonunun servis verdiği görülmektedir. Böylece bir hücre üzerinde bulunan araç, o hücrenin baz istasyonu tarafından servis alacaktır. Hücrelerin sınırları konum olarak bilindiği için dolayısıyla her bir baz istasyonunun servis alanı da bilinmektedir. Çağrının hücrede oluşması iki şekilde mümkündür: Birincisi hücre içerisinde bulunan bir aracın

çağrıyı başlatmasıyla, diğeri de hücreye yeni giren bir araca önceden yapılan bir çağrının aktarılmasıyla olmaktadır. Aynı şekilde çağrı bitirme işlemi de hücre içerisinde bulunan aracın bir çağrıyı bitirmesiyle ya da hücreden yeni ayrılan bir araca önceden mevcut bir çağrının aktarılmasıyla oluşmaktadır. Bu çalışmada ayrıca herbir hücrenin kanal kapasitesi ve çağrının bloke olma olasılığı da düşünülmektedir. Kapasite için bir hücrenin sahip olacağı maksimum çağrı sayısı sınırlandırılmaktadır. Bloke olma olasılıkları için de yeni başlayan ve aktarılan çağrıların ayrı ayrı bloke durumları ele alınmaktadır. Bu alanlardaki çağrı sayıları ve yoğunlukları hesaplanarak son olasılık değerleri elde edilmektedir.

3.6.2. ÇBO Değerlerinin Analitik Çözümü

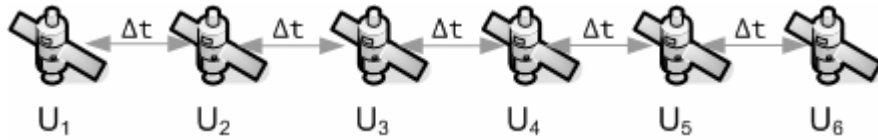
Bu bölümde ÇBO hesabı için trafik değerlerine etki eden yeni parametreler bulunmaktadır. Paketlerin iletimi sırasında trafik yoğunluğunu değiştirebilen faktörlerden etkilenilmektedir. Aynı uydu sisteminin birçok farklı coğrafik alandaki iletimleri aynı şekilde olmamaktadır. Başka bir deyişle coğrafik alanların paket yoğunluğuna etkisi, o uydu sisteminin ilgili paketleri iletimine etki etmektedir. İletişimin yoğun olduğu şehir bölgelerinde paket yoğunluğu fazla, tenha bölgelerde paket yoğunluğu azdır. Benzer şekilde farklı saatlerde iletim yoğunluğunda farklılıklar oluşmaktadır. Örneğin gece paket iletiminde düşüş gözlemlenebilmektedir. Bütün bu etkenler düşünülerek algoritmanın ÇBO hesaplamaları genişletilmektedir.

Bir topoloji üzerinde elde edilen yol yapısının genel olarak gösterimi Şekil 3.14'te görülmektedir. Burada çağrının U_1 uydusundan U_6 uydusuna doğru gittiği düşünülmektedir. Herbir uydu çiftinin arasında bir link olduğu bilinmektedir. Örneğin U_1 uydusu U_2 uydusuna L_{12} linki ile bağlıdır.



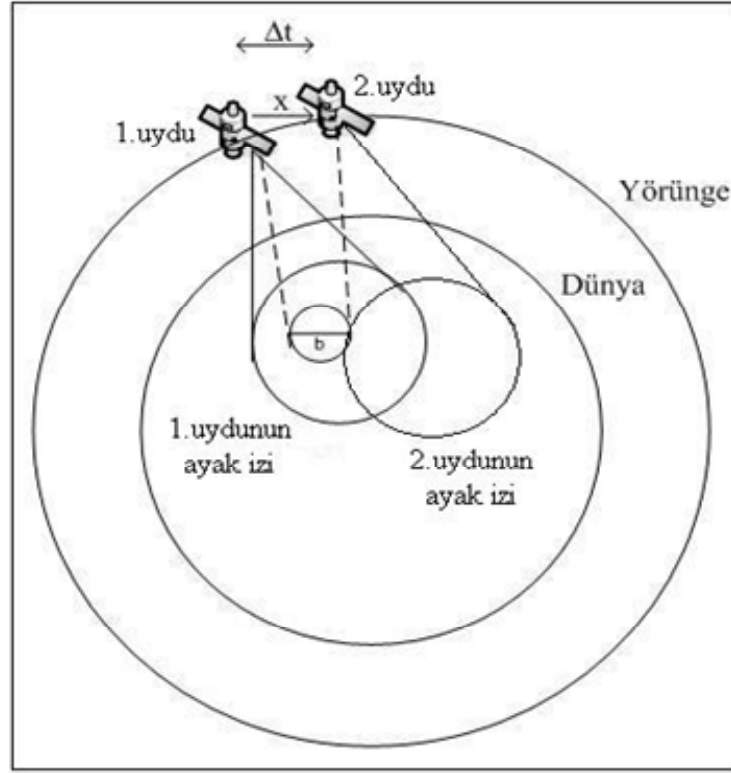
Şekil 3.14: Çağrının iletildiği örnek bir yol

Önceden belirtildiği gibi çağrı süresinin eşit zaman aralıklarına bölündüğü ve herbir aralıkta sabit bir topolojinin olduğu düşünülmektedir. Bu sabit topolojilerin her birinde çok sayıda yol belirleme işlemi yapılmaktadır. Bunun için de sabit topoloji boyunca geçen süre eşit Δt parçalarına bölündüğünde Şekil 3.15'teki yapı ortaya çıkmaktadır:



Şekil 3.15: Bir yol boyunca uydular arası geçiş

Şekil 3.15'e göre U_1 uydusunun Δt süresi boyunca dünya yüzeyinde belirli bir alanı kapsadığı düşünülmektedir. Bu süreden sonra iletimin U_2 uydusuna geçtiği varsayılır. Şekil 3.14'teki L_{12} linkinin çağrı yoğunluğunu hesaplamak için kapsanan alanın hesaplanması gerekmektedir. Aynı hesaplamalar diğer linkler için de yapılır. Şekil 3.16'da görüldüğü gibi 1. ve 2. uydular arasındaki çağrı yoğunluğunu belirleyen alan, 1. uydunun Δt süresince dünya yüzeyinde servis verdiği alandır. Şekil 3.16'da gösterilen bu alan, b çapında dairesel bir bölgedir.



Şekil 3.16: Bir uydunun belirli bir zamanda dünya yüzeyinde kapladığı alan

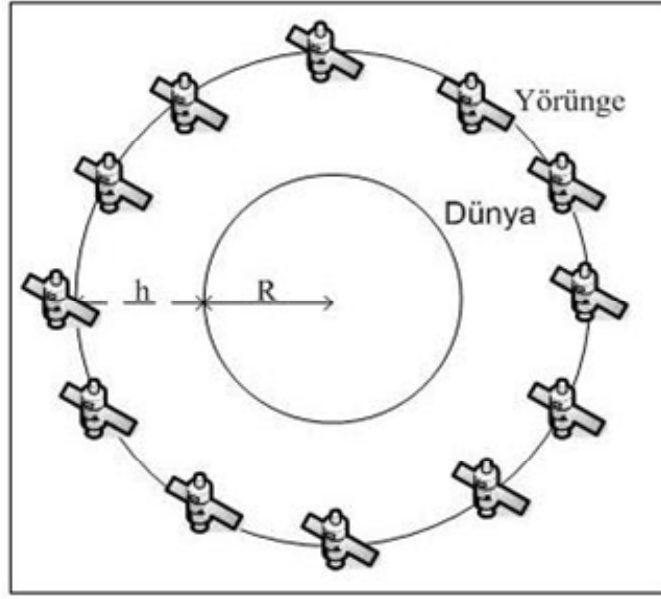
Kepler'in 3. kanununa göre bir uydunun yörünge periyodu olan T , Şekil 3.17'den yararlanılarak aşağıdaki formüle göre hesaplanabilmektedir (Richharia, 1999):

$$T^2 = \frac{4\pi^2}{g}(R + h)^3 \quad (3.21)$$

g : Yerçekimi ivmesi

R :Dünyanın yarıçapı

h :Uydunun dünya yüzeyinden yüksekliği



Şekil 3.17: Dünyanın ve herhangi bir yörüngeye göre konumu

x_y yörünge çevre uzunluğu, V uydunun yörünge üzerindeki hızı ve önceden belirtildiği gibi T yörünge periyodu olacak şekilde Denklem (3.22)'den yola çıkılmaktadır:

$$x_y = V.T \quad (3.22)$$

Δt zamanda bir uydunun yörünge üzerinde gittiği düzlemsel mesafe aşağıdaki ifadede görülmektedir:

$$x = V.\Delta t \quad (3.23)$$

Gelen çağrının N tane uydudan oluşan bir yol üzerinde iletildiği düşünölsün. Bu çağrının süresi $T.\varepsilon_t$ ($0 < \varepsilon_t < 1$) olacağından yol üzerindeki herbir uydu için geçen Δt süresi ve bu sürede kat edilen mesafe şu şekildedir:

$$\Delta t \triangleq \frac{T.\varepsilon_t}{N-1} \Rightarrow x = V \cdot \frac{T.\varepsilon_t}{N-1} \quad (3.24)$$

Bir sonraki aşamada bu mesafenin dünya üzerindeki izdüşümünü bulmak gerekir. Bunun için dünyanın çevresi x_d ile gösterilirse aşağıdaki eşitlik sağlanır:

$$b = \frac{x_d}{x_y} . x \quad (3.25)$$

Denklem (3.25)'te $\frac{x_d}{x_y}$ ifadesinin yerine ε_x şeklinde ondalık bir sayı getirilerek yeni ifade şu şekilde elde edilir:

$$b = x.\varepsilon_x \quad (0 < \varepsilon_x < 1) \quad (3.26)$$

Böylece b değeri aşağıdaki şekilde elde edilir:

$$b = \frac{V.T.\varepsilon_t.\varepsilon_x}{N-1} \quad (3.27)$$

Çağrı yoğunluğunu etkileyen bir başka faktör de içinde bulunulan zamandır. Gün içerisinde zaman ilerledikçe çağrı yoğunluğu artar. Bu özellikten dolayı t geçerli zaman dilimine ait sabit sayıyı gösterirse aşağıdaki sonuç varsayım olarak ortaya çıkar:

$$\varepsilon_t \triangleq t.c \quad (0 < c < 1), (1 \leq t \leq 6) \quad (3.28)$$

Bu varsayım, Denklem (3.27)'de yerine konduğu zaman Denklem (3.29) elde edilmektedir:

$$b = \frac{V.T.t.c.\varepsilon_x}{N-1} \quad (3.29)$$

$$\varepsilon \triangleq c.\varepsilon_x \Rightarrow 0 < \varepsilon < 1 \quad (3.30)$$

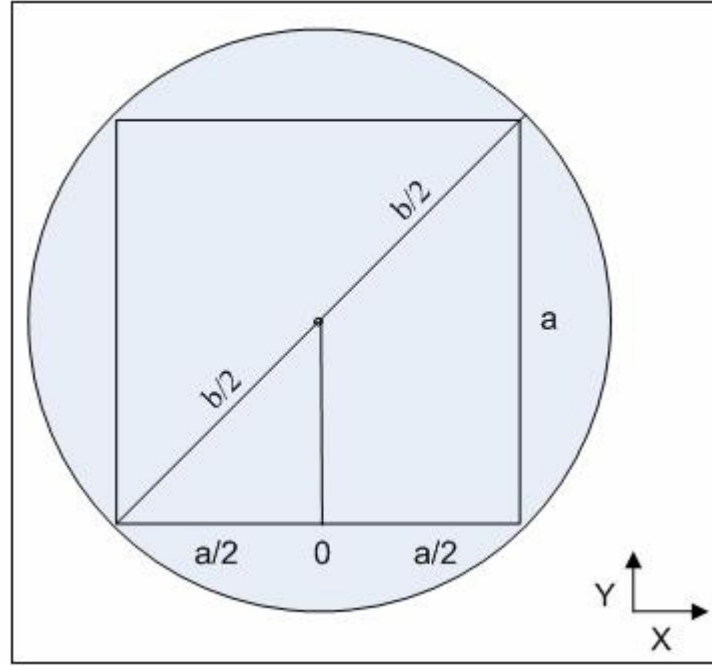
Denklem (3.30)'daki varsayımdan yola çıkılarak Denklem (3.29) şu ifadeye denk olur:

$$b = \frac{V.T.t.\varepsilon}{N-1} \quad (3.31)$$

Denklem (3.22)'den yararlanarak şu sonuç elde edilir:

$$b = \frac{x_y.t.\varepsilon}{N-1} \quad (3.32)$$

Dünya yüzeyinde kapsanan alandan gelecek çağrı yoğunluğunun hesaplanması için bu bölgenin Şekil 3.18'deki gibi karesel yapıya indirgendiği düşünülmektedir. Bu işlemin temel nedeni, çağrıların trafik yoğunluklarının, kapsama alanlarının merkezinde yoğunlaşmasıdır. Bu nedenle kapsama alanını, merkezi içine alacak şekilde yoğunluğu hesaplanabilecek geometrik düzlemsel bir yapıya indirgemek avantaj sağlamaktadır.



Şekil 3.18: Kapsama alanının hesaplanması

Pisagor teoremine göre a kenarı hesaplanır (Pisagor teoremiyle ilgili ayrıntıları Bogomolny (2007) vermektedir.):

$$\begin{aligned}
 b^2 &= a^2 + a^2 \\
 b^2 &= 2.a^2 \\
 a &= \frac{b\sqrt{2}}{2}
 \end{aligned} \tag{3.33}$$

Çağrı yoğunluğu, $f(x)$ fonksiyonu şeklinde gösterilsin:

$$f(x) = \alpha - \frac{k}{x^3} \quad (\alpha \geq 1, \forall k) \tag{3.34}$$

Farklı $f(x)$ fonksiyonlarıyla ilgili ayrıntılı açıklamalar, Bölüm 4.5'te yer almaktadır. Şekil 3.18'deki karenin alanına eşit olan ve bir linke coğrafik alana bağlı olarak gelen yeni paketlerin Trafik yoğunluğu (YTY) $f(x)$ fonksiyonuna bağlı olarak şu şekilde hesaplanmaktadır:

$$\begin{aligned}
YTY &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} f(x) dx dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} \left(\alpha - \frac{k}{x^3} \right) dx dy = \int_0^{\frac{b\sqrt{2}}{2}} \left(\alpha \cdot x + \frac{k}{2x^2} \right) \Big|_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \left(\alpha \cdot \frac{b\sqrt{2}}{4} + \frac{k}{2 \cdot \frac{2b^2}{16}} - \left(\alpha \cdot \frac{-b\sqrt{2}}{4} + \frac{k}{2 \cdot \frac{2b^2}{16}} \right) \right) dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \alpha \cdot \frac{b\sqrt{2}}{2} dy = \left(\alpha \cdot \frac{b\sqrt{2}}{2} \cdot y \right) \Big|_0^{\frac{b\sqrt{2}}{2}} \\
&= \alpha \cdot \frac{b^2}{2}
\end{aligned} \tag{3.35}$$

Denklem (3.35)'teki α değeri, (3.34)'teki fonksiyonda şarta bağlı olarak seçilebilir. Böylece تنها yerlerde düşük, yoğun şehir bölgelerinde yüksek α değerleri seçilerek coğrafik farklılıkların da devreye girmesi sağlanabilir. Bunun dışında Şekil 3.18'deki karenin x eksen koordinatlarının $(0, a)$ yerine $\left(\frac{-a}{2}, \frac{a}{2} \right)$ şeklinde alınmasının nedeni, $x=0$ olduğu zaman bazı $f(x)$ fonksiyonlarında paydanın 0 olma durumunun ortaya çıkmasıdır. Bunu engellemek için karenin sol veya sağ köşelerindeki x değerlerinin 0 olmaması gerekmektedir. Bir diğer önemli nokta da tüm linklerin YTY değerlerinin aynı olmasıdır. Bunun nedeni, düşünülen yörüngenin çembersel olmasıdır. Çembersel yörüngede tüm uydular her yerde eşit hızda ilerlerler ve zamana-konuma bağlı fizik kurallarındaki tüm hesaplamalar aynı olmaktadır.

Belirli bir yol üzerindeki herhangi bir linke gelen yeni paketlerin dışında bir önceki linkten geçerek ulaşan eski paketler de vardır. j . linke ait olan ve eski paketlerin trafik yoğunluğunu gösteren ifade ETY_j olarak gösterilsin. Bir linkin sahip olduğu sunucu sayısı m_j olarak belirlenmişti. Denklem (3.17)'deki formülden yararlanarak YTY ve ETY_j değerleri yerine konulduğu zaman aşağıdaki eşitlik elde edilmektedir:

$$\zeta_{BO_j} = \frac{(\lambda_j / \mu)^{m_j} / m_j!}{\sum_{k=0}^{m_j} (\lambda_j / \mu)^k / k!} = \frac{(YTY + ETY_j)^{m_j} / m_j!}{\sum_{i=0}^{m_j} (YTY + ETY_j)^i / i!} \quad (3.36)$$

Denklem (3.35), Denklem (3.36)'da yerine konur. Bundan sonra b ifadesinin yerine Denklem (3.32) yer aldığı zaman aşağıdaki işlemler ortaya çıkmaktadır:

$$\zeta_{BO_j} = \frac{\left(\alpha \cdot \frac{b^2}{2} + ETY_j \right)^{m_j} / m_j!}{\sum_{i=0}^{m_j} \left(\alpha \cdot \frac{b^2}{2} + ETY_j \right)^i / i!} = \frac{\left(\alpha \cdot \frac{\left(\frac{x_y \cdot t \cdot \varepsilon}{N-1} \right)^2}{2} + ETY_j \right)^{m_j} / m_j!}{\sum_{i=0}^{m_j} \left(\alpha \cdot \frac{\left(\frac{x_y \cdot t \cdot \varepsilon}{N-1} \right)^2}{2} + ETY_j \right)^i / i!} \quad (3.37)$$

ETY_j değerinin hesabına geçmeden önce Denklem (3.37) analiz edildiğinde ortaya çıkan başlıca sonuçlar şöyle özetlenebilmektedir:

Konuma bağlılık:

1. Sabit çağrı süresi ve sabit uydu hızında ilgili yolu oluşturan uydu sayısı (N) arttıkça b değeri, dolayısıyla ÇBO değeri azalır (Denklem (3.27) ve Denklem (3.37)) (ETY_j, ε_t ve ε_x sabit).
2. İçinde bulunulan zaman dilimi, ilgili yolu oluşturan uydu sayısı ve ETY_j değeri sabit tutulmak şartıyla yörünge uzunluğu arttıkça b değeri, dolayısıyla ÇBO değeri artar (Denklem (3.32) ve Denklem (3.37)).

Zamana bağlılık:

1. Sabit c değerinde t arttıkça ε_t değeri, dolayısıyla da sabit ETY_j değerinde de ÇBO değeri artar (Denklem (3.28)).

ETY_j hesabına geçildiğinde karşımıza çıkan iki farklı durum vardır. İlk durum, j .linkin yola ait olan ilk link olma durumudur. Böyle olduğu zaman bir önceki linkten gelen herhangi bir paket olmadığı için $ETY_1=0$ olmaktadır. Birinci link için Denklem (3.37)'den elde edilen sonuç aşağıdaki gibidir:

$$\zeta BO_1 = \frac{\left(\alpha \cdot \frac{\left(\frac{x_y \cdot t \cdot \mathcal{E}}{N-1} \right)^2}{2} \right)^{m_1}}{\sum_{i=0}^{m_1} \left(\alpha \cdot \frac{\left(\frac{x_y \cdot t \cdot \mathcal{E}}{N-1} \right)^2}{2} \right)^i / i!} \quad (3.38)$$

Birinci link dışında herhangi bir linkin ETY_j hesabı, bir önceki linkin ζBO değerine bağlı olmaktadır. Bunun nedeni, bloke olmadan o linke ulaşan paketlerin ETY_j değerini oluşturmalarıdır. $(j+1)$. linke ait olan $ETY_{(j+1)}$ hesabı aşağıdaki şekilde yapılmaktadır ($j \in \mathbb{Z}^+$):

$$ETY_{(j+1)} = YTY(1 - \zeta BO_j) \quad (3.39)$$

Sonuçta Denklem (3.39), Denklem (3.36) ile düşünüldüğü zaman $(j+1)$. linkin ζBO değeri şu şekilde ortaya çıkmaktadır:

$$\begin{aligned}
\zeta BO_{(j+1)} &= \frac{(YTY + ETY_{(j+1)})^{m_{(j+1)}} / m_{(j+1)}!}{\sum_{i=0}^{m_{(j+1)}} (YTY + ETY_{(j+1)})^i / i!} \\
&= \frac{(YTY + (YTY(1 - \zeta BO_j)))^{m_{(j+1)}} / m_{(j+1)}!}{\sum_{i=0}^{m_{(j+1)}} (YTY + (YTY(1 - \zeta BO_j)))^i / i!} \\
&= \frac{((2 - \zeta BO_j)YTY)^{m_{(j+1)}} / m_{(j+1)}!}{\sum_{i=0}^{m_{(j+1)}} ((2 - \zeta BO_j)YTY)^i / i!} \\
&= \frac{\left((2 - \zeta BO_j) \left(\alpha \cdot \frac{b^2}{2} \right) \right)^{m_{(j+1)}} / m_{(j+1)}!}{\sum_{i=0}^{m_{(j+1)}} \left((2 - \zeta BO_j) \left(\alpha \cdot \frac{b^2}{2} \right) \right)^i / i!} \\
&= \frac{\left((2 - \zeta BO_j) \left(\alpha \cdot \frac{\left(\frac{x_y \cdot t \cdot \mathcal{E}}{N-1} \right)^2}{2} \right) \right)^{m_{(j+1)}} / m_{(j+1)}!}{\sum_{i=0}^{m_{(j+1)}} \left((2 - \zeta BO_j) \left(\alpha \cdot \frac{\left(\frac{x_y \cdot t \cdot \mathcal{E}}{N-1} \right)^2}{2} \right) \right)^i / i!}
\end{aligned} \tag{3.40}$$

4. BULGULAR

4.1. STATİK ALGORİTMANIN PERFORMANS ÖLÇÜMÜ

4.1.1. Gecikme Sürelerinin Hesabı

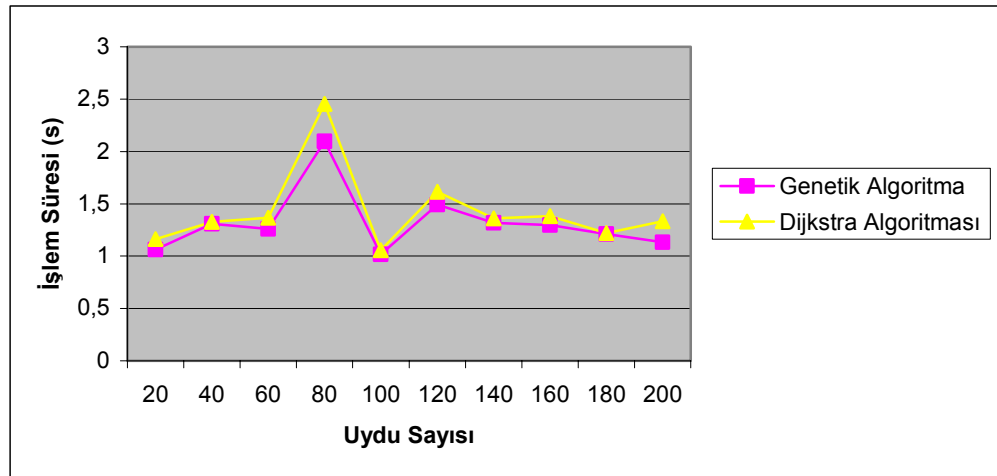
Bir algoritmanın gecikme süresini hesaplamak için, fonksiyonlar işleme geçmeden önce zaman başlatılır ve en son işlemle birlikte zaman sonlandırılır. Uygulamada böylece tüm işlemler yapılırken ölçülen süre tutulur. Bu da gecikme süresini vermektedir. Genetik algoritmanın özellikleri kullanılarak oluşturulan yeni yönlendirme algoritmasının statik yapısında, birçok uydu sayısına sahip topolojiler ele alınmaktadır. Bu topolojilerde uygun yolun bulunması için geçen süre hesaplanmaktadır. Aynı işlemler, en çok bilinen Link Durum Algoritmalarından biri olan Dijkstra Algoritması için gerçekleştirilerek bu iki algoritmanın gecikme süreleri açısından performansları karşılaştırılmaktadır. Burada önemli bir nokta, link durum algoritmalarında ağ yapısındaki her bir düğümün, diğer tüm düğümlerin bilgisine sahip olması gerektiğidir. Yeni algoritmada ise böyle bir zorunluluk yoktur. Herbir uydu, paket kendisine geldiği zaman iletim yapabilecek (aktif) komşu linklerden birini seçerek transfer işlemini gerçekleştirir.

Yapılan uygulamada, uydu sayıları üzerinde değişiklikler yapılarak elde edilen çalışma süreleri Tablo 4.1’de gösterilmiştir. 20 uyduya sahip topolojiden başlayarak 200 uydulu topolojiye kadar işlem süreleri hesaplanmaktadır. Her iki algoritma da, herbir topoloji için birkaç kez çalıştırılarak bu sürelerin ortalaması tabloya konulmuştur. Dijkstra algoritmasının çalıştırılmasıyla ortaya çıkan sonuçlar da tabloya eklenerek karşılaştırma işlemi yapılabilmektedir.

Tablo 4.1: Genetik Algoritma ile Dijkstra Algoritmasının ortalama işlem süreleri

Uydu Sayısı	Genetik Algoritma (s)	Dijkstra Algoritması (s)
20	1,063	1,163
40	1,309	1,328
60	1,26	1,366
80	2,097	2,453
100	1,015	1,056
120	1,491	1,613
140	1,319	1,362
160	1,297	1,381
180	1,213	1,222
200	1,131	1,335

Tablo 4.1’de görüldüğü gibi Genetik algoritma aracılığıyla uygun yolların bulunması için geçen süre, Dijkstra algoritmasında elde edilen süreden daha kısadır. Örneğin 100 uyduya sahip bir topolojiyle uygulama yapıldığı zaman, Genetik Algoritma 1,015 saniyede tüm işlemleri yapmaktadır. Bu süre, uygun yol bulunana kadar gerçekleştirilen tüm adımlar için geçen süredir. Aynı uydu sayısı için Dijkstra Algoritması 1,056 saniyeye gereksinim duymaktadır. Bu durumun tüm uydu sayıları düşünülerek ortalama çalışma süresi olarak hesaplanması halinde Genetik algoritmanın daha iyi sonuç verdiği söylenebilir. Şekil 4.1’de bu işlem sürelerinin grafiksel gösterimine yer verilmektedir. Böylece uygunluk fonksiyonunun bir bileşeni olan gecikme süresi hesaplanmış olmaktadır.



Şekil 4.1: İşlem sürelerinin grafiği

4.1.2. Aktarım Sayısının Bulunması

Aktarım sayısını bulmak için uydu ağının farklı zaman aralıklarında farklı topolojilere sahip olduğu düşünülmektedir. Karşılaştırma yapmak amacıyla Greedy Optimizasyon yönteminden yararlanılmaktadır (Zaim, 2001). Greedy Optimizasyon yönteminde bütün elemanların belirli bir oranı vardır. Elemanlar, buna göre büyükten küçüğe doğru dizilir. Oranların toplamı belirli bir limiti aşmayacak şekilde elemanlar sırayla seçilir. Limite ulaşıncaya kadar kalan diğer elemanlar atılır.

Başlangıç düğüm 21, hedef düğüm de 41 alınarak her bir topolojide optimizasyon sonucu seçilen yollar elde edilmektedir. Greedy yönteminde elde edilen yollar aşağıdaki şekilde bulunmuştur:

Aktarım Sayısı=5:

Aralık=0: 21,32,42,41

Aralık=1: 21,32,42,41

Aralık=2: 21,20,31,41

Aralık=3: 21,20,31,41

Aralık=4: 21,22,33,43,42,41

Aralık=5: 21,22,33,43,42,41

Aralık=6: 21,22,33,43,42,41

Aralık=7: 21,22,33,43,42,41

Aralık=8: 21,20,19,31,30,41

Aralık=9: 21,20,19,31,42,41

Aralık=10: 21,32,31,30,41.

Burada görüldüğü gibi 1-2, 3-4, 7-8, 8-9, 9-10 topolojiler arasında aktarım işlemleri gerçekleşmiştir. Yeni algoritmanın uygulamasında ise 60, 80, 100 ve 200 düğüme sahip farklı topolojiler üzerinde denemeler yapılmıştır. Önceki örnekte olduğu gibi kaynak uydu olarak 21. uydu, hedef uydu olarak da 41. uydu düşünülmüştür. Aşağıda görüldüğü gibi aktarım sayısı, 6-7, 8-9, 10-11 topolojileri arasında olmak üzere 3 tane bulunmaktadır:

Aktarım sayısı=3:

1. Topoloji: 21, 27, 33, 39, 40, 41
2. Topoloji: 21, 27, 33, 39, 40, 41
3. Topoloji: 21, 27, 33, 39, 40, 41
4. Topoloji: 21, 27, 33, 39, 40, 41
5. Topoloji: 21, 27, 33, 39, 40, 41
6. Topoloji: 21, 27, 33, 39, 40, 41
7. Topoloji: 21, 22, 28, 34, 40, 41
8. Topoloji: 21, 22, 28, 34, 40, 41
9. Topoloji: 21, 27, 33, 39, 40, 41
10. Topoloji: 21, 27, 33, 39, 40, 41
11. Topoloji: 21, 22, 28, 34, 35, 41.

Şekil 4.2’de uygulamadaki bu örneğin ekran çıktısı görülmektedir.

```

1.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
2.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
3.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
4.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
5.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
6.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
7.topolojiden secilen yol=21,22,28,34,40,41,-1,-
-1,-1.
8.topolojiden secilen yol=21,22,28,34,40,41,-1,-
-1,-1.
9.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
10.topolojiden secilen yol=21,27,33,39,40,41,-1,-
-1,-1.
11.topolojiden secilen yol=21,22,28,34,35,41,-1,-
-1,-1.

```

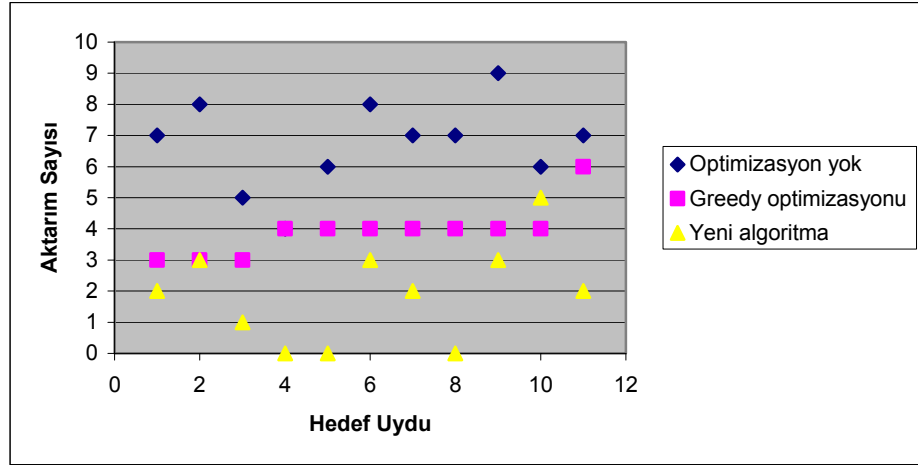
Şekil 4.2: Birbirinden farklı 11 topoloji arasından seçilen yollar

Bu çalışmada oluşturulan algoritmanın, minimum aktarım sayısına sahip yolları seçtiğini göstermek amacıyla 21. uydu kaynak düğüm olsun. Hedef düğümler de sırasıyla 0, 1, 10, 22, 23, 31, 32, 33, 41, 42 ve 43 şeklinde seçilsin. Tablo 4.2’de görüldüğü gibi bu hedef düğümlere karşılık gelen x değerleri (0. uyduya 1 değeri verilerek) işaretlenmektedir. Bu değerler, Şekil 4.3’teki x eksenine karşılık gelmektedir.

Tablo 4.2: Hedef düğümlere karşılık gelen x değerleri

Hedef Düğüm	0	1	10	22	23	31	32	33	41	42	43
x değeri	1	2	3	4	5	6	7	8	9	10	11

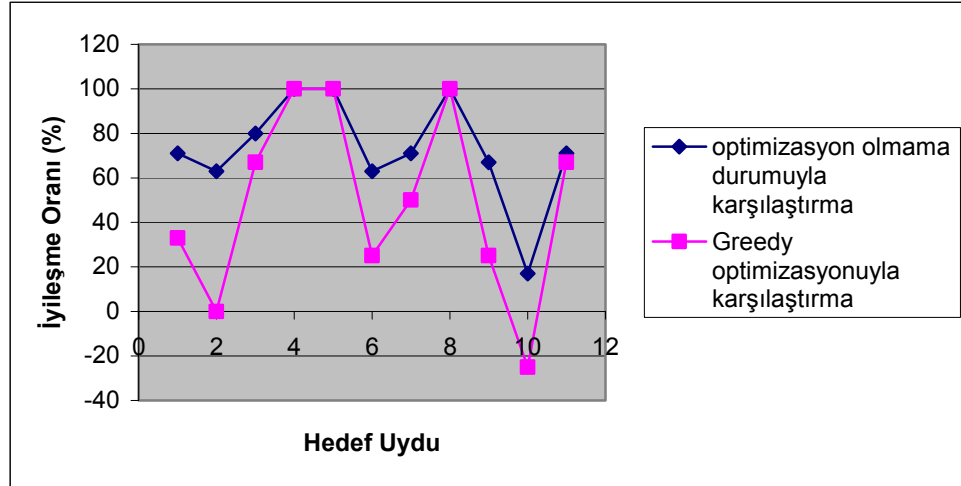
y ekseninde ise uygulamaların çalıştırılmasıyla elde edilen aktarım sayıları görülmektedir. Burada görüldüğü gibi simülasyonda optimizasyon işlemi kullanılmadığı zaman aktarım sayısı en yüksek değerde olmaktadır. Buna karşılık optimizasyon yapıldığı durumlarda bu değer azalmaktadır. Örneğin Greedy optimizasyonu ile bunu görmek mümkündür. Öte yandan yeni algoritmadaki yöntem de aktarım sayısını azaltmaktadır. Grafikte x ekseninde bulunan (y=0) değerler için hiç aktarım olmadığı anlaşılmaktadır. Aktarımın olmaması, yollar arasında herhangi bir düğüm değişikliğinin olmaması durumudur. Şekil 4.3'te görüldüğü gibi hedef düğüm olarak 22, 23 veya 33 (x=4,x=5,x=8) seçildiği zaman; yeni algoritma, aktarım sayısını 0 olarak bulmaktadır.



Şekil 4.3: Aktarım sayılarının karşılaştırılması

Şekil 4.4'te yeni optimizasyon metodunun başarımları % olarak görülmektedir. Bu grafikte optimizasyon uygulanmama durumu ve Greedy optimizasyonu ile oluşan durum yeni algoritma ile karşılaştırılarak aktarım sayısındaki iyileşme % olarak hesaplanmaktadır. Kaynak düğüm önceki grafikte olduğu gibi 21'dir. Hedef düğümler de aynı şekilde kalmaktadır. Aktarım sayısının 0 olarak indirildiği durumlarda başarı oranı %100 olmaktadır. Örneğin Şekil 4.4'teki grafikte x=4 olduğu zaman bu durum karşımıza çıkmıştır. Yeni metotla elde edilen aktarım sayısı diğer durumlarla aynı

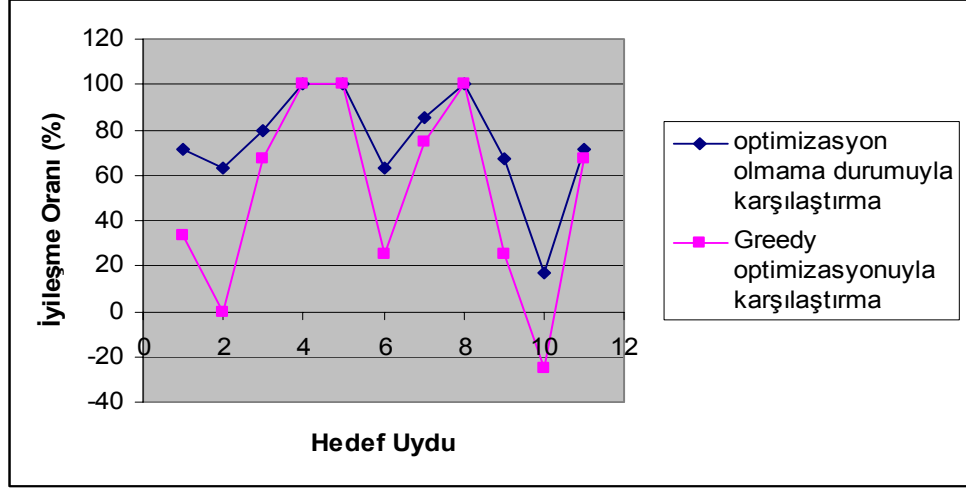
olduğunda bir gelişme söz konusu değildir, bu durumda ise iyileşme yüzdesi 0'dır. Çünkü böyle bir durumda herhangi bir iyileşme söz konusu olmamaktadır. Grafikten anlaşıldığı gibi gelişmenin en çok olduğu hedef uydular 22, 23 veya 33'tür. Bunun nedeni, önceden bahsedildiği gibi yeni metot aracılığıyla bu uydular için hesaplanan aktarım sayılarının 0 olmasıdır.



Şekil 4.4: Aktarım sayısındaki iyileşmenin % hesabı

Bu aşamaya kadar aktarım sayısındaki iyileşme oranları, gecikme süresinde herhangi bir kısıtlama olmadan yapılmıştır. Biraz değişiklik yapılarak grafiğe kısıtlama eklendiğinde işlemler artmaktadır. Şekil 4.5'teki grafiği elde etmek için öncelikle gecikme süresini %50'den daha fazla arttıran yollar seçilerek ayrılır. Başka bir deyişle bir yolun gecikme süresi, kendinden önce gelen yolun gecikme süresinden %50'den fazla oranda büyükse bu yol yeniden hesaplama işlemi için ayrılır. Yeniden hesaplamalar sonucunda gecikme süreleri arasında en fazla %50 fark olan yollar bulunur. Bunların aktarım sayıları göz önüne alınır. Şekil 4.4 ile Şekil 4.5 arasında farklılık, hedef uydu 32 olduğu zaman ($x=7$) ortaya çıkmıştır. Bu adımda elde edilen yolların gecikme sürelerindeki fark %50 oranını aşınca yollar yeniden oluşturulmaya başlanmıştır. Bu kısıtı geçmeyecek şekilde elde edilen yeni yollar arasındaki aktarım sayısı böylece değişmektedir. Şekil 4.5'te görüldüğü gibi aktarım sayısındaki iyileşme bu işlemler sonucu artmıştır. Optimizasyon olmama durumuyla karşılaştırmada %71 olan iyileşme oranı yeni grafikte %85 olmuştur. Benzer şekilde Greedy optimizasyonuyla karşılaştırmada da %50 olan iyileşme oranı %75 olmuştur. Diğer hedef düğümlerle ilgili gecikme sürelerinde kısıtı aşan bir durum söz konusu olmadığı için herhangi bir değişiklik olmamıştır. Bu

işlemlerle farkedilen en önemli durum, gecikme süresinin optimizasyon işlemine katılmasıyla birlikte iyileşme oranında artışın olmasıdır. Ayrıca yeni metodun iyileştirme oranının, y ekseninin pozitif yönünde olması da olumlu gelişmeyi göstermektedir.



Şekil 4.5: Aktarım sayısındaki iyileşmenin % hesabı (gecikme süresi kısıtı % 50 iken)

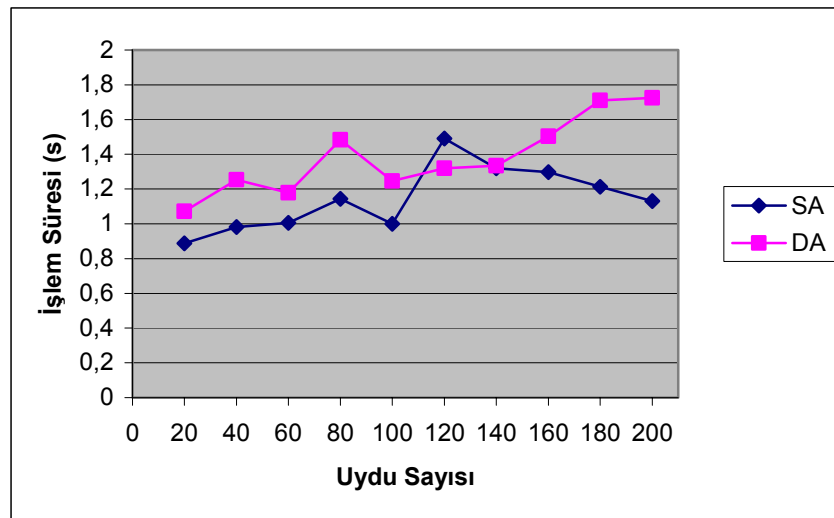
4.2. STATİK VE DİNAMİK ALGORİTMALARIN KARŞILAŞTIRILMASI

Bölüm 4.1’de statik yapı üzerinde işlemler yapılmaktaydı. Bu kısımda uydu sistemlerinin dinamik olduğu düşünülmektedir. Böylece algoritmanın iki farklı durumu ortaya çıkmaktadır. Sistemdeki uydu sayısı 20-200 arasında değişecek şekilde bu iki durumun karşılaştırması yapılmıştır. Buna göre Tablo 4.3’te görülen değerler elde edilmektedir. İlk sütunda sistemdeki uydu sayısı, ikinci sütunda Statik Algoritmanın (SA) tüm uygulama boyunca geçirdiği ortalama çalışma süreleri, üçüncü sütunda da Dinamik Algoritmanın (DA) ortalama çalışma süreleri bulunmaktadır. Bu süreler, girilen kaynak ve hedef düğümlerin aralarındaki mesafeye ve kullanılan bilgisayarın donanım özelliklerine göre de değişebilmektedir. Önemli olan nokta, hem statik, hem de dinamik durum için aynı koşulların ortamda sağlanmasıdır. Karşılaştırma kriterleri eşit olarak sağlandıktan sonra Tablo 4.3’teki değerler ortaya çıkmıştır. Farklı uydu sayılarına sahip sistemlerde uygulamanın birçok kez çalıştırılmasıyla bu değerler bulunmaktadır. Buna göre örneğin 100 uydulu bir sistemde SA uygulaması birçok kez çalıştırıldıktan sonra ortalama çalışma süresi 1,000 sn olarak bulunmuşken, DA uygulaması için ortalama çalışma süresi 1,247 sn şeklinde hesaplanmıştır.

Tablo 4.3: Algoritmanın statik ve dinamik durumlarında elde edilen ortalama çalışma süreleri

Uydu Sayısı	SA (s)	DA (s)
20	0,887	1,072
40	0,982	1,253
60	1,006	1,178
80	1,144	1,484
100	1,000	1,247
120	1,491	1,319
140	1,319	1,334
160	1,297	1,503
180	1,213	1,709
200	1,131	1,725

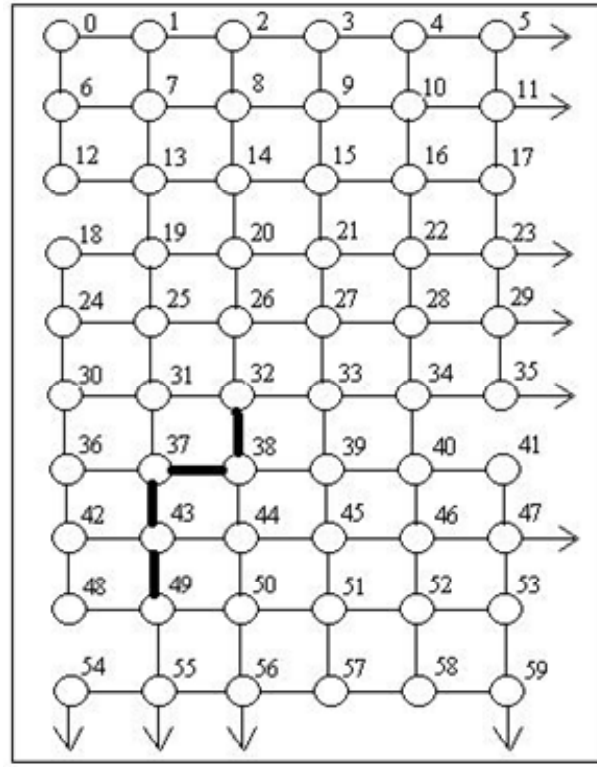
Tablo 4.3'teki işlem sürelerinin karşılaştırması, Şekil 4.6'da görülmektedir. Burada olduğu gibi statik algoritmanın çalışma süresinin daha kısa olduğu sonucu ortaya çıkmaktadır. Bunun nedeni de önceden anlatıldığı gibi dinamik yapıda yönlendirme işleminin daha karmaşık olmasıdır. Linklerin sürekli değişmesinden dolayı, yol üzerindeki bir sonraki linkler seçilirken zaman kaybı yaşanmaktadır. Bu geçen süre içerisinde, dinamik yapıdan kaynaklanan linkin aktif olup olmama durumu kontrol edilmektedir.



Şekil 4.6: SA ve DA işlem sürelerinin karşılaştırması

Dinamik yapıdaki çalışma süresinin daha uzun olma nedenlerinden bir diğeri, topolojilerin değişimi sırasında geçen sürenin de çalışma süresine dahil olmasıdır. Bu değişim sırasında linklere ait son d ve w değerleri devreye girmektedir. Çünkü sonradan aktif olan linklerin bu değerlerinin de işlemlere girmesi gerekmektedir. SA'da uydu sistemi yapısını aynen koruduğu için bu değerler sabittir, bu nedenle değerlerin yeniden hesabı için uygulamada ek süre gerekmez.

SA ve DA uygulamalarını karşılaştırmak amacıyla aktarım sayısına da bakmak mümkündür. Aktarım işlemi, bir uydunun ayak izinin değişmesinden dolayı, servis verdiği çağrı işlemini o ayak izini kapsayan yeni uyduya devretmesidir. Aktarım sayısını bulmak için Şekil 4.7'de görülen 60 uyduya sahip bir AY sisteminin kullanıldığı düşünölsün. Şekilde göröldüğü gibi sistemdeki 60 uydunun düzenli bir dağılım oluşturması için 6 yörünge'nin bulunduğu düşünölmektedir. Uydular bu yörüngelere konumlanmaktadır ve kendi aralarındaki linklerle iletişim sağlamaktadırlar. Burada DA algoritması sırasında linklerin durumunda değişiklikler oluşurken SA uygulaması boyunca aynı topoloji kullanılmakta, fakat bu topolojiden elde edilen yollar zamanla değişmektedir.



Şekil 4.7: Uydu sayısı 60 olan bir topoloji

Şekil 4.7’deki uydu sistemi kullanılarak 32. uydu kaynağı, 49. uydu da hedef uydu olarak ele alındığında birçok farklı yol elde edilebilmektedir. Şekil 4.8’de SA sonucunda seçilen yollar görülürken, Şekil 4.9’da DA uygulaması aracılığıyla seçilmiş olan yollar görülmektedir. Bu şekillerdeki gibi, yol gösterimleri uydu numaraları aracılığıyla belirtilmektedir. Buradaki örneklerde topoloji sayısı 5 olarak alınmıştır, ancak bu değer uygulamada istenildiği kadar büyütülebilmektedir. Herbir topolojide birçok yol bulunmakta ve bunlar arasından birer yol seçilmektedir. Topoloji içerisinde yollar, uygunluk değerlerinin büyükten küçüğe doğru dizilmesine göre sıralandıktan sonra seçim işlemine başlanır. Statik yapıdaki uygunluk değerleri için sadece gecikme süreleri devreye girmektedir. Dinamik yapı için uygunluk değerlerinin hesabı da Bölüm 3.3.1’de ayrıntılı olarak açıklanmıştır. Böylece uygunluk değerlerine göre dizilmiş olan bir topolojideki herbir yol, bir sonraki topolojideki tüm yollarla karşılaştırılır. Aynı olan yol varsa o seçilir. Böylece aktarım sayısının küçük olması hedeflenmektedir. Uygunluk değerlerinin diziliminden dolayı da gecikme süresi küçük olan yolların seçilme olasılığı artmış olmaktadır.

=32,38,37,43,49,-1,-1,-1,-1,-1,
 =32,38,37,43,49,-1,-1,-1,-1,-1,
 =32,38,37,43,49,-1,-1,-1,-1,-1,
 =32,31,37,43,49,-1,-1,-1,-1,-1,
 =32,31,37,43,49,-1,-1,-1,-1,-1

Şekil 4.8: SA uygulaması sonucu elde edilen yollar

1.topolojiden	secilen	yo1=32,31,37,43,49
2.topolojiden	secilen	yo1=32,31,37,43,49
3.topolojiden	secilen	yo1=32,38,44,50,49
4.topolojiden	secilen	yo1=32,38,44,50,49
5.topolojiden	secilen	yo1=32,38,37,43,49

AKTARIM SAYISI=2

Şekil 4.9: DA uygulaması sonucu elde edilen yollar

Şekil 4.8 ve 4.9’da elde edilen yollar arasında aktarım sayıları incelenebilir. Şekil 4.8’de sadece 3.-4. topolojiden seçilmiş olan yollar arasında, uydu sayılarında farklılık görülmektedir. Uydu sayılarının farklılığı, bu yollar arasındaki süre içinde iletim

işleminin başka uydulara verildiğini göstermektedir. Örneğin 3. yol sırasında iletim 2. yoldaki gibi aynı uydular arasında yapıldığı halde, 4. yol sırasında farklı uydulara yönelme olmuştur. Bu durum, uydular arası aktarımın bu sırada gerçekleştiğini göstermektedir. Aynı şekilde Şekil 4.9'da da 2.-3. ve 4.-5. topolojiler arasında aktarım durumunun ortaya çıktığı görülmektedir. Sonuçta bu örnek üzerinde SA uygulamasının aktarım sayısı 1, DA uygulamasının aktarım sayısı da 2 olarak hesaplanmaktadır.

DA aktarım sayısının fazla olma nedeni de yine dinamik yapıdan kaynaklanmaktadır. Sürekli değişen bir sistemde gelen çağrılar aynı uydudan servis alma ihtimali azalmaktadır. Bir uydunun ayak izi değişince servis verdiği bölge de değişmektedir. Böylece önceki zaman diliminde servis verdiği bölgeyi başka uydulara bırakmaktadır. Hem uydu sisteminin, hem de dünyanın sürekli değişen ve kendi yörüngelerinde dönmelerini sağlayan bir hareket içinde bulunmaları bu durumu oluşturmaktadır.

Aktarım durumuna göre yönlendirme algoritmasını oluşturmak amacıyla birçok farklı çalışma yapılmıştır. Örneğin en kısa yol algoritması gibi bir algoritma yardımıyla bir sistemde başlangıçtaki optimal yol durumu tayin edilip daha sonra aktarım durumunun incelendiği bir yöntem de vardır. Buna göre aktarım yapıldığı anda yeniden yönlendirme algoritması tekrar işlemektedir. Bu aşamadaki kaynak uydu, aktarımı devralan uydu olmaktadır. İşlemler yapılırken sıradaki uydunun kapasite açısından iletişimi devralmaya uygun olup olmadığı da gözlemlenmektedir. Ayrıca yeni uydunun, önceden ele alınan yolun oluşturduğu kümede yer alıp almadığı da incelenmektedir. Kümede yer almadığı görüldüğünde ise, bu kümede hedefe en yakın uydulardan başlanarak direk linke sahip olduğu bir başka uydu aranmaktadır. Böyle bir uydu varsa yönlendirme işleminin sıradaki yeni uydusu olarak seçilmektedir. Bu duruma uygun uydu yoksa başa dönülüp yönlendirme işlemine yeniden başlanır. Sonraki kısımda da bir zaman sonra birçok işlemten kaynaklanan optimalden uzaklaşma durumu olabileceği için belirli zaman aralıklarında yol güncellemeleri yapılır. Bunun için başlangıçtaki yolun son terminaller tarafından bilinmesi önemlidir. Bu çalışmada ise kaynak terminalin bu konuda kaynak uyduya istekte bulunması ile zamana ve trafiğe göre belirlenen zaman aralıklarında yeniden yönlendirme işlemine karar verilir. Böylece yol oluşumu devam ederek hedef uyduya kadar işlemler sürmektedir (Uzunalioglu ve Yen, 1997).

4.3. STATİK VE DİNAMİK ALGORİTMALARIN DEĞİŞTİRİLMİŞ DIJKSTRA ALGORİTMASIYLA KARŞILAŞTIRILMASI

Bölüm 4.2’de algoritmanın Dinamik ve Statik yapılarının birbirleriyle karşılaştırılmasına yer verilmektedir. Bu bölümde ise bu karşılaştırmalara değiştirilmiş (modified) Dijkstra algoritması dahil edilmiştir. Birçok farklı çalışmada “modified Dijkstra” terimine yer verilmektedir. Örneğin uydu sistemleri için geliştirilen ATM tabanlı bir yönlendirme algoritmasında UAL’lerden yararlanılarak paket gönderimi yapılmıştır. Burada sanal uçtan uca yollar tayin edilerek bir konuşmanın başlangıç anında bu yollardan birinin seçimi yapılmıştır. En kısa yol algoritması olarak “modified Dijkstra” (kısaca D_Dijkstra olarak gösterelim) en kısa yol algoritması kullanılmıştır (Werner ve diğ., 1997). Bilgisayar ağlarındaki bir başka çalışmada da belirli bir kaynak ve hedef düğüm arasında birçok yol tayin edilmektedir. Daha sonra servis kalitesi ölçümüne göre belirlenmiş bazı kriterleri sağlayan en uygun yol elde edilmeye çalışılmaktadır. Bu algorithmada minimum normalleştirilmiş kenar denilen yeni bir özellik katılarak D_Dijkstra algoritması elde edilmiştir. Bu özellik, uygun olmayan yolları devre dışı bırakarak zaman tasarrufu sağlamaktadır (Shin ve diğ., 2002).

Optik ağlarda yönlendirme işlemiyle ilgili yapılan bir başka çalışmada da D_Dijkstra yapısı elde edilerek bit düzeyinde lojik işlemler için kullanılmıştır. Bu optik ağda fiberlerin içerisindeki dalga boyları kullanılarak bir kaynaktan bir hedef noktasına doğru yönlendirme işlemi yapılmaktadır. Fiberler içerisindeki serbest dalga boylarının bit haritalarıyla gösterildiği düşünülmektedir. Bu haritalarda kaynaktan hedefe doğru olan 1 bitlik alanlar, yönlendirme işlemi için seçilen fonksiyonu oluşturmaktadır. Bit haritası gösterimindeki dinamik yönlendirme işlemi için D_Dijkstra tasarlanmıştır. Çok etkin olan bu algoritmanın bit düzeyindeki lojikselsel işlemleri yaptığı belirtilmiştir (Zhou ve diğ., 2005).

D_Dijkstra algoritmasının bahsedildiği başka bir yönlendirme algoritması da gecikme toleranslı ağlar için geliştirilmiştir. Düşünülen yapıda uydu ağlarının içine dahil edilebildiği bir sistem mevcuttur. Burada mesajlar, bağlantıların gösterildiği bir grafiğe göre bir uçtan diğer uca iletilmektedir. Dinamik olan bu sistemde her bir düğümün sınırlı tampon kapasitesi vardır. Böylece bir uçtan bir uca giden sabit yol bulunmayabilir. Bu

durum genel olarak ağılarda en kısa yolu bulan geleneksel yönlendirme algoritmalarına ters bir durumdur. Bu probleme çözüm olarak düşünülen algoritmada her linkin maliyet değeri zamanla değişmediğinde Dijkstra En Kısa Yol Algoritması kullanılabilir. Ancak maliyetlerin zamanla değiştiği durumlar için D_Dijkstra algoritması oluşturulmuştur. Öncelikle bir mesajın bir düğüme geliş süresi hesaplanmaktadır. Daha sonra bu süre kullanılarak alt linklerin maliyeti hesaplanmaktadır. Bu durum mesajın komşu düğümlere ulaşma süresini etkilemektedir (Jain ve diğ., 2004).

Kendi çalışmamızda Dijkstra Algoritmasındaki topoloji yapısına dinamiklik kazandırılarak D_Dijkstra algoritması elde edilmiştir. Yeni tasarlanan algoritmadaki dinamiklik yapısı burada kullanılmıştır. Bu amaçla çağrı süresi boyunca sistemin belirli zaman dilimlerinde topoloji değişikliklerine uğradığı düşünülmüştür. Herbir zaman aralığında bu değişiklikler linklerin aktif/pasif olma durumuna göre ayarlanmıştır. Böylece bir çağrı süresince sistemin, belirlenen sayıda farklı topolojiye sahip olmasıyla dinamik yapıda olduğu belirlenmiştir. Oluşturulan yeni yapı 20-200 uydu sayısına sahip birçok sistem üzerinde simüle edilmiştir. Herbir sistem için çok sayıda kaynak-hedef uydu arasında farklı yollar elde edilmiştir ve bunların çalışma süreleri hesaplanmıştır. Daha sonra ortalama çalışma süreleri elde edilmiştir. Buradaki çalışma süresi, program aracılığıyla en uygun yolun bulunması için geçen toplam süreyi vermektedir. Tablo 4.4'te yeni tasarlanan genetik algoritmanın dinamik ve statik halleriyle D_Dijkstra algoritmasının ortalama çalışma süreleri görülmektedir.

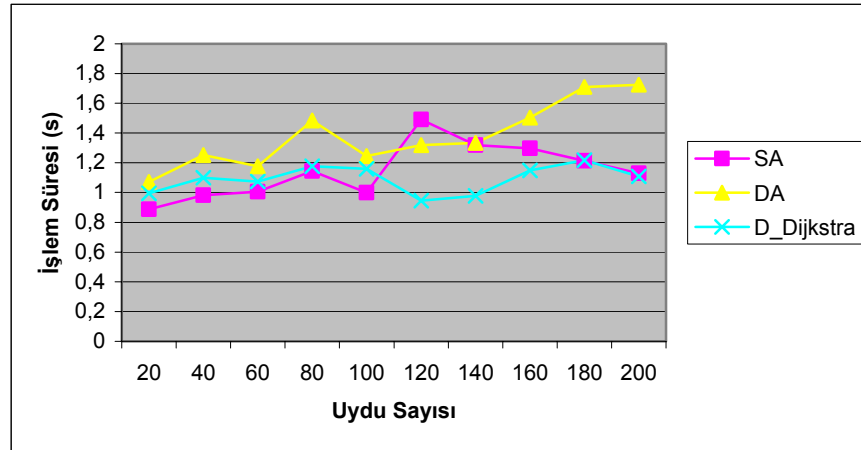
Şekil 4.10'da ise ortalama çalışma sürelerinin grafik üzerindeki dağılımı yer almaktadır. Bu şekilde görüldüğü gibi yeni algoritmanın statik hali dinamik haline göre daha kısa sürede gerçekleşmektedir. Bunun yanı sıra D_Dijkstra algoritmasının çalışma süresi ortalama bir seyir izlemektedir. Yeni algoritmanın dinamik yapısında simülasyon sırasında birçok işlem yapılmaktadır. Bu işlemler dinamik bir sistem üzerinde olduğu için döngüsel olarak tekrarlar, çaprazlama, yeniden üretim gibi işlemler yer almaktadır. Böylece genetik algoritmaya dair işlemler, dinamik bir yapıda çok fazla matematiksel kavramı içerdiği için de zaman olarak daha uzun sürmektedir. Ayrıca bu dinamik algoritmada işlemler sırasında göz önünde bulundurulacak kriterler daha fazladır. ÇBO'larla ilgili ek hesaplar da mevcuttur. Öte yandan D_Dijkstra algoritmasında yol seçim işlemi için en çok tekrar eden yolun belirlenmesi, yolların çaprazlama

işlemleriyle iyileştirilmesi ve paketlerin yoğunluklarından dolayı ortaya çıkan bloke olma olasılıkları gibi hesaplamalar bulunmamaktadır. Buna rağmen algoritmalar arasındaki saniye farklarının çok fazla olmaması performans açısından iyi bir sonuç vermektedir. Birçok topolojide D_Dijkstra algoritmasına oldukça yakın sonuçlar elde edilmektedir.

Tablo 4.4: SA, DA ve D_Dijkstra Algoritması sonucunda elde edilen ortalama çalışma süreleri

Uydu Sayısı	SA (s)	DA (s)	D_Dijkstra (s)
20	0,887	1,072	0,994
40	0,982	1,253	1,1
60	1,006	1,178	1,075
80	1,144	1,484	1,178
100	1	1,247	1,159
120	1,491	1,319	0,946
140	1,319	1,334	0,978
160	1,297	1,503	1,15
180	1,213	1,709	1,219
200	1,131	1,725	1,109

Yeni algoritmanın ÇBO hesaplamaları olmadan ortaya çıkan SA yapısında görüldüğü gibi genelde D_Dijkstra algoritmasından daha iyi sonuçlar elde edilmiştir.



Şekil 4.10: SA, DA ve D_Dijkstra işlem sürelerinin karşılaştırması

4.4. ANALİTİK ÇBO ÇÖZÜMÜNÜN KARŞILAŞTIRMA SONUÇLARI

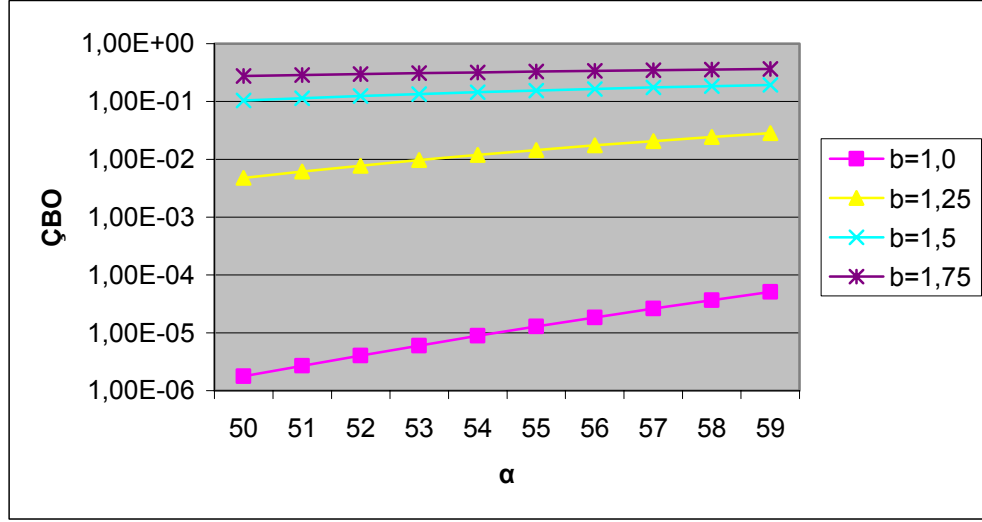
Analitik çözümün elde edilmesinden sonra simülasyon yardımıyla birçok farklı parametreye göre sonuçlar elde edilmiştir. $ETY_j = 10$, $m_j = 65$ olarak ele alındığı zaman Tablo 4.5'te görüldüğü gibi b değeri büyüdükçe, yani kapsanan alan genişledikçe Denklem (3.37)'deki ÇBO değeri de yükselir:

Tablo 4.5: $ETY_j=10$ ve $m_j=65$ durumunda b değerlerine göre ÇBO karşılaştırması

α	ÇBO($b=1,0$ km)	ÇBO($b=1,25$ km)	ÇBO($b=1,5$ km)	ÇBO($b=1,75$ km)
50	1,77E-06	4,79E-03	1,04E-01	2,76E-01
51	2,70E-06	6,15E-03	1,14E-01	2,88E-01
52	4,06E-06	7,78E-03	1,24E-01	2,99E-01
53	6,04E-06	9,70E-03	1,34E-01	3,09E-01
54	8,87E-06	1,19E-02	1,44E-01	3,20E-01
55	1,29E-05	1,45E-02	1,54E-01	3,30E-01
56	1,85E-05	1,74E-02	1,64E-01	3,40E-01
57	2,62E-05	2,07E-02	1,74E-01	3,50E-01
58	3,68E-05	2,43E-02	1,84E-01	3,59E-01
59	5,10E-05	2,82E-02	1,94E-01	3,68E-01

ÇBO değerlerinin, simülasyon dışında pratikte de ölçülebilir olması gerekmektedir. Bu konuda yapılan bazı çalışmalarda ÇBO değerlerinin, $[10^{-6}, 10^{-2}]$ aralığındaki limitleri alt sınır aldıkları görülebilir (Zaim ve diğ., 2003; Obradovic ve Kanth, 2000; Boukhatem ve diğ., 2003). Bu nedenle analitik çözümde kullanılan fonksiyonlardaki sabit sayılara atanan değerler önem taşır. Tablo 4.5'te görüldüğü gibi α değişkeni minimum 50 değerini aldığı anda istenen ÇBO aralıkları elde edilmiş olmaktadır.

Şekil 4.11’de, Tablo 4.5’teki değerlerin grafiksel gösterimi yer almaktadır:

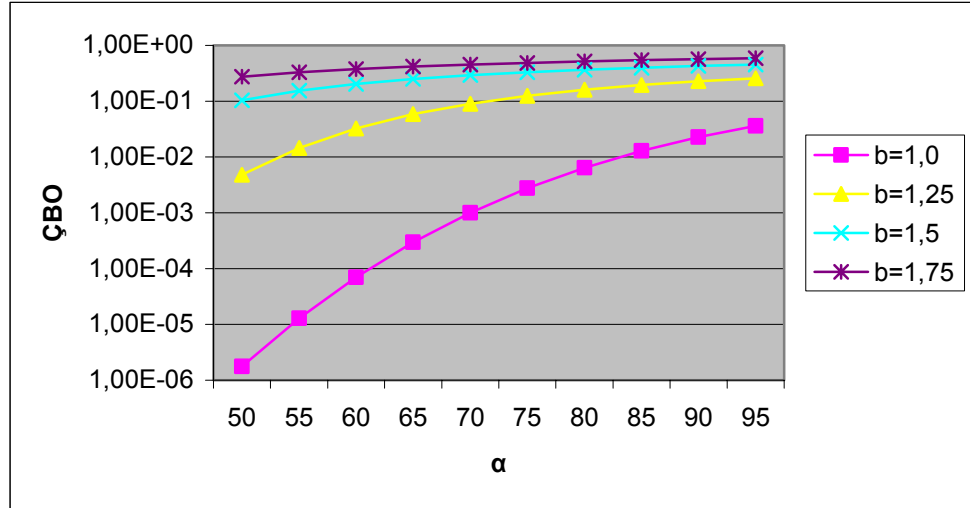


Şekil 4.11: Kapsama alanına (konuma) göre ÇBO karşılaştırmasının grafiksel dağılımı

α değerindeki farklı artış oranlarına göre elde edilen ÇBO sonuçlarıyla ilgili değerler Tablo 4.6 ve Şekil 4.12’de gösterilmektedir:

Tablo 4.6: ETY_j=10 ve m_j=65 durumunda b değerlerine göre ÇBO karşılaştırması (aralıklı α artırımlarıyla)

α	ÇBO(b=1,0 km)	ÇBO(b=1,25 km)	ÇBO(b=1,5 km)	ÇBO(b=1,75 km)
50	1,77E-06	4,79E-03	1,04E-01	2,76E-01
55	1,29E-05	1,45E-02	1,54E-01	3,30E-01
60	7,01E-05	3,25E-02	2,03E-01	3,77E-01
65	2,96E-04	5,85E-02	2,50E-01	4,19E-01
70	1,00E-03	9,01E-02	2,92E-01	4,55E-01
75	2,77E-03	1,24E-01	3,31E-01	4,88E-01
80	6,45E-03	1,60E-01	3,66E-01	5,16E-01
85	1,29E-02	1,94E-01	3,98E-01	5,42E-01
90	2,28E-02	2,27E-01	4,27E-01	5,65E-01
95	3,62E-02	2,58E-01	4,54E-01	5,86E-01



Şekil 4.12: Kapsama alanına (konuma) göre ÇBO karşılaştırmasının grafiksel dağılımı (aralıklı α artımlarıyla)

Tablo 4.6'daki değerler hesaplanırken örnek olarak $b=1,25$ km ve $\alpha=75$ değeri için elde edilen simülasyon çıktısı aşağıdaki gibidir:

```
b=1.25 ve alfa=75 için CBO degeri:
0.12442523389454810000000000
```

Şekil 4.13: $b=1.25$ ve $\alpha=75$ değerleri için ÇBO sonucu ($ETY_j=10$ ve $m_j=65$)

İçinde bulunulan zaman dilimine göre karşılaştırma sonuçlarını elde etmek için

$ETY_j = 10$ ve $m_j = 65$ olmak şartıyla Denklem (3.32)'yi oluşturan $b = \frac{x_y \cdot t \cdot \varepsilon}{N-1}$

formülünden yararlanılmaktadır. $\varepsilon = 10^{-4}$ ve $N = 7$ olarak ele alınsın. x_y yörünge uzunluğu için dünyanın yarıçapından (6378 km) ve ekvator uzunluğundan (40076 km) yararlanarak bir bağıntı kurulmaktadır. AY uyduların konum olarak dünyadan 700-2000 km yükseklikte oldukları bilindiği için bu iki değere karşılık gelen x_y değerleri bulunup bunların ortalamaları hesaplanmaktadır. Şekil 4.14'te bu hesaplamalar için gereken parametreler gösterilmektedir. Dünyanın yarıçapı $R=6378$ km, ekvator uzunluğu (E)=40076 km, dünyaya en yakın AY'nin yüksekliği $h_1=700$ km, dünyadan en uzak AY'nin yüksekliği $h_1+h_2=2000$ km olarak bilinmektedir. Bu değerlerden yola çıkarak aşağıdaki hesaplamalar sonucunda ortalama x_y değeri bulunmaktadır:

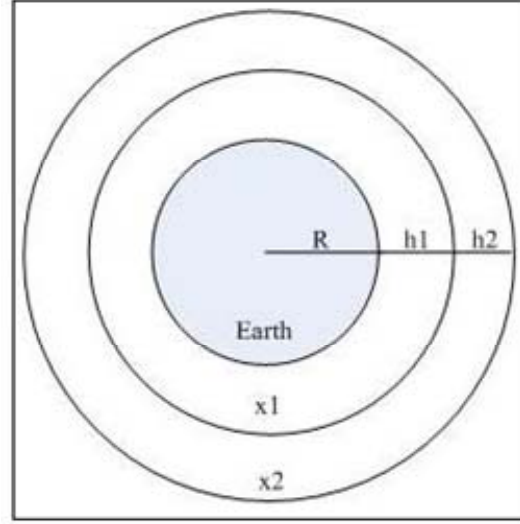
$$\frac{R}{E} = \frac{R + h_1}{x_1} \quad (4.1)$$

$$\frac{6378}{40076} = \frac{6378 + 700}{x_1} \Rightarrow x_1 = 44474 \text{ km}$$

$$\frac{R}{E} = \frac{R + h_1 + h_2}{x_2} \quad (4.2)$$

$$\frac{6378}{40076} = \frac{6378 + 2000}{x_2} \Rightarrow x_2 = 52642 \text{ km}$$

$$x_y = x_{ort} = \frac{x_1 + x_2}{2} \Rightarrow x_y = \frac{44474 + 52642}{2} = 48558 \text{ km} \quad (4.3)$$



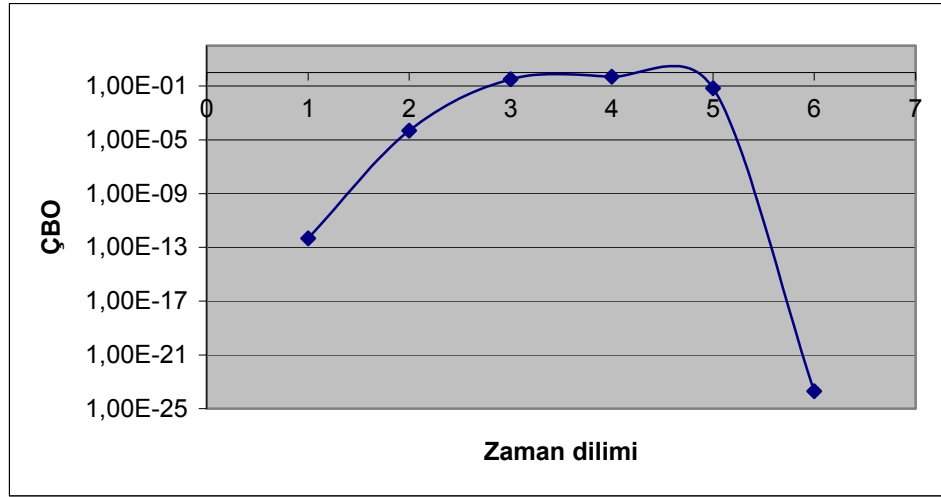
Şekil 4.14: AY uzunlukları için gerekli değişkenler

Elde edilen x_y değeri yaklaşık olarak 48500 km olarak düşünülerek belirlenen t değerlerine göre Denklem (3.37)'deki ÇBO sonuçları bulunmaktadır. Uygulama için gün içerisinde mevcut olan zaman dilimleri 6 parçaya ayrılmaktadır. Zaman dilimlerine karşılık gelen sabit sayıyı gösteren t değerleri ve x eksen değeri belirlenmektedir. $\alpha = 10$ varsayımıyla elde edilen ifadeler ve ÇBO değerleri Tablo 4.7'de gösterilmektedir.

Tablo 4.7: $ET\gamma_j=10$ ve $m_j=65$ durumunda t değerlerine göre ÇBO karşılaştırması

x	Zaman dilimi	t	ÇBO
1	Sabah	2	4,58E-13
2	Kuşluk	3	4,79E-05
3	Akşam	5	3,13E-01
4	İkinci	6	4,98E-01
5	Öğle	4	7,01E-02
6	Gece	1	2,01E-24

Tablo 4.7’den yola çıkılarak linklerde t değerine bağlı ÇBO değişimini gösteren grafik Şekil 4.15’te yer almaktadır.



Şekil 4.15: Zamana göre ÇBO karşılaştırmasının grafiksel dağılımı

Şekil 4.15’te görüldüğü gibi iletişimin yoğun olduğu zaman dilimlerine gidildikçe paket sayısındaki yoğunluktan dolayı ÇBO değerlerinde artış belirmektedir. Örneğin gece vakitlerinde kablosuz iletişim öğle vakitlerine göre daha az olduğu için ÇBO değeri de az olmaktadır. Gece saatlerinde paketlerin yoğunluktan dolayı bloke olma durumları çok nadir rastlanabilen bir durum olmaktadır.

Tablo 4.7’deki değerler hesaplanırken örnek olarak $t=5$ değeri için elde edilen simülasyon çıktısı aşağıdaki gibidir:

t=5 için CBO değeri:
0.3129449356713359700000000

Şekil 4.16: t=5 için ÇBO sonucu ($\alpha=10$, $ET_Y=10$ ve $m_j=65$)

4.5. ANALİTİK ÇBO ÇÖZÜMÜNÜN FARKLI FONKSİYONLARA GÖRE SONUÇLARI

Denklem (3.35)'i elde etmek için birçok $f(x)$ fonksiyonundan yararlanılmıştır. Önemli olan nokta, b veya t değerlerinin sonucu nasıl etkilediğidir. Elde edilen ÇBO formülünde diğer parametrelerin sabit kalması durumunda bu değerlerin yükselmesi ÇBO değerini de yükseltmektedir. Başka bir deyişle uydu sistemindeki linkin o anda yeryüzünde içerdiği alan büyüdükçe çağrılarının bloke olma olasılığı da artar, çünkü birçok paketin trafiği sıkıştırması söz konusudur. Aynı şekilde iletişim yoğunluğuna göre aynı gün içinde zaman dilimlerine göre de çağrılarının bloke olma olasılıklarının değiştiği gözlemlenmektedir.

İşlemler sırasında b veya t değerlerinin ÇBO değeriyle doğru orantılı olduğu örnek fonksiyonlar ve bu fonksiyonlara bağlı olarak ortaya çıkan YTY çözümleri şu şekildedir:

$$1) f(x) = k |x| \begin{cases} kx, & x \geq 0 \\ -kx, & x < 0 \end{cases}, \quad k \geq 1 \quad (4.4)$$

$$\begin{aligned} YTY &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} f(x) dx dy \\ &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^0 -kx dx dy + \int_0^{\frac{b\sqrt{2}}{2}} \int_0^{\frac{b\sqrt{2}}{4}} kx dx dy \\ &= \int_0^{\frac{b\sqrt{2}}{2}} \left. \frac{-kx^2}{2} \right|_{-\frac{b\sqrt{2}}{4}}^0 dy + \int_0^{\frac{b\sqrt{2}}{2}} \left. \frac{kx^2}{2} \right|_0^{\frac{b\sqrt{2}}{4}} dy \end{aligned}$$

$$\begin{aligned}
&= \int_0^{\frac{b\sqrt{2}}{2}} \frac{kb^2}{16} dy + \int_0^{\frac{b\sqrt{2}}{2}} \frac{kb^2}{16} dy \\
&= \frac{b\sqrt{2}}{2} \cdot \frac{2kb^2}{16} = \frac{b^3 k \sqrt{2}}{16}
\end{aligned} \tag{4.5}$$

$$2) \ f(x) = \alpha \cdot x^2 \ (\alpha \geq 1) \tag{4.6}$$

$$\begin{aligned}
YTY &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} f(x) dx dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} \alpha \cdot x^2 dx dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \left(\alpha \cdot \frac{x^3}{3} \right) \Big|_{-b\sqrt{2}/4}^{b\sqrt{2}/4} dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \left(\alpha \cdot \frac{(b\sqrt{2}/4)^3}{3} - \alpha \cdot \frac{(-b\sqrt{2}/4)^3}{3} \right) dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} 2\alpha \cdot \frac{b^3 \cdot 2\sqrt{2}}{192} dy = \left(2\alpha \cdot \frac{b^3 \cdot 2\sqrt{2}}{192} \cdot y \right) \Big|_0^{b\sqrt{2}/2} \\
&= \alpha \cdot \frac{b^4}{48}
\end{aligned} \tag{4.7}$$

Benzer şekilde b veya t değerlerinin ÇBO değeriyle ters orantılı olduğu, yani sistem için yanlış sonuçlar veren fonksiyonlar da vardır. Bunların çözümünden çıkan sonuçlar negatif olduğu için veya 1'den küçük bir katsayıyı bulundurarak ters orantıya neden olduğu için uygun olmamaktadır. Bu fonksiyonlar ve YTY çözümleri aşağıda sıralanmaktadır:

$$1) \quad f(x) = -e^{k|x|} \quad (k \geq 1) \quad (4.8)$$

$$f(x) = -e^{k|x|} \begin{cases} -e^{kx}, & x \geq 0 \\ -e^{-kx}, & x < 0 \end{cases}, \quad k \geq 1 \quad (4.9)$$

$$\begin{aligned} YTY &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} f(x) \, dx \, dy \\ &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^0 -e^{-kx} \, dx \, dy + \int_0^{\frac{b\sqrt{2}}{2}} \int_0^{\frac{b\sqrt{2}}{4}} -e^{kx} \, dx \, dy \\ &= \int_0^{\frac{b\sqrt{2}}{2}} \frac{e^{-kx}}{k} \Big|_{-b\sqrt{2}/4}^0 \, dy + \int_0^{\frac{b\sqrt{2}}{2}} \frac{-e^{kx}}{k} \Big|_0^{b\sqrt{2}/4} \, dy \\ &= \int_0^{\frac{b\sqrt{2}}{2}} \left(\frac{1}{k} - \frac{e^{kb\sqrt{2}/4}}{k} \right) dy + \int_0^{\frac{b\sqrt{2}}{2}} \left(\frac{-e^{kb\sqrt{2}/4}}{k} - \frac{-1}{k} \right) dy \\ &= \frac{b\sqrt{2}}{k} \cdot (1 - e^{kb\sqrt{2}/4}) \end{aligned} \quad (4.10)$$

Bu eşitlik $(1 - e^{kb\sqrt{2}/4})$ çarpanından dolayı her durumda negatif sonuç vermektedir (e=2,718).

$$2) \quad f(x) = -k|x| \quad (k \geq 1) \quad (4.11)$$

$$f(x) = -k|x| \begin{cases} -kx, & x \geq 0 \\ kx, & x < 0 \end{cases}, \quad k \geq 1 \quad (4.12)$$

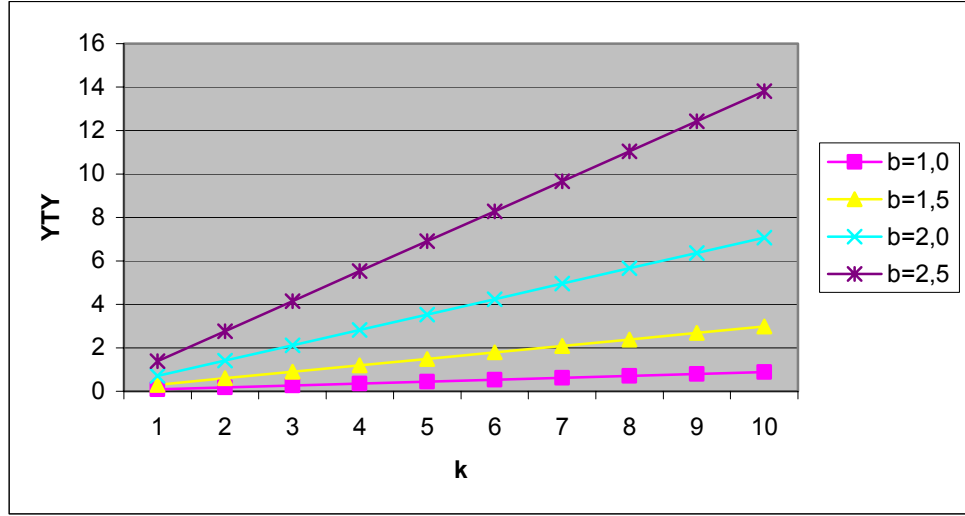
$$\begin{aligned}
YTY &= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^{\frac{b\sqrt{2}}{4}} f(x) dx dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \int_{-\frac{b\sqrt{2}}{4}}^0 kx dx dy + \int_0^{\frac{b\sqrt{2}}{2}} \int_0^{\frac{b\sqrt{2}}{4}} -kx dx dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \frac{kx^2}{2} \Big|_{-b\sqrt{2}/4}^0 dy + \int_0^{\frac{b\sqrt{2}}{2}} \frac{-kx^2}{2} \Big|_0^{b\sqrt{2}/4} dy \\
&= \int_0^{\frac{b\sqrt{2}}{2}} \frac{-kb^2}{16} dy + \int_0^{\frac{b\sqrt{2}}{2}} \frac{-kb^2}{16} dy \\
&= \frac{b\sqrt{2}}{2} \cdot \frac{-2kb^2}{16} \\
&= \frac{-b^3 k \sqrt{2}}{16}
\end{aligned} \tag{4.13}$$

Fonksiyonun 0'dan küçük bir değer verdiği açıkça görülmektedir. Analizlerde istenen doğru artışları veren (4.4) ile yanlış sonuç veren (4.8) fonksiyonları karşılaştırıldığında sonuçların farklı değişimi kolaylıkla gözlenebilmektedir. Bunun için bu iki fonksiyonun *YTY* sonuçlarının analizi yapılmıştır. (4.4) için *YTY* sonuçları:

Tablo 4.8: $f(x) = k|x|$ için b değerlerine göre *YTY* karşılaştırması

k	YTY(b=1,0 km)	YTY(b=1,5 km)	YTY(b=2,0 km)	YTY(b=2,5 km)
1	0,088388	0,298311	0,707107	1,381068
2	0,176777	0,596621	1,414214	2,762136
3	0,265165	0,894932	2,12132	4,143204
4	0,353553	1,193243	2,828427	5,524272
5	0,441942	1,491553	3,535534	6,90534
6	0,53033	1,789864	4,242641	8,286408
7	0,618718	2,088175	4,949747	9,667476
8	0,707107	2,386485	5,656854	11,048543
9	0,795495	2,684796	6,363961	12,429611
10	0,883883	2,983107	7,071068	13,810679

Tablo 4.8'deki sonuçların grafiksel gösterimi Şekil 4.17'dedir:

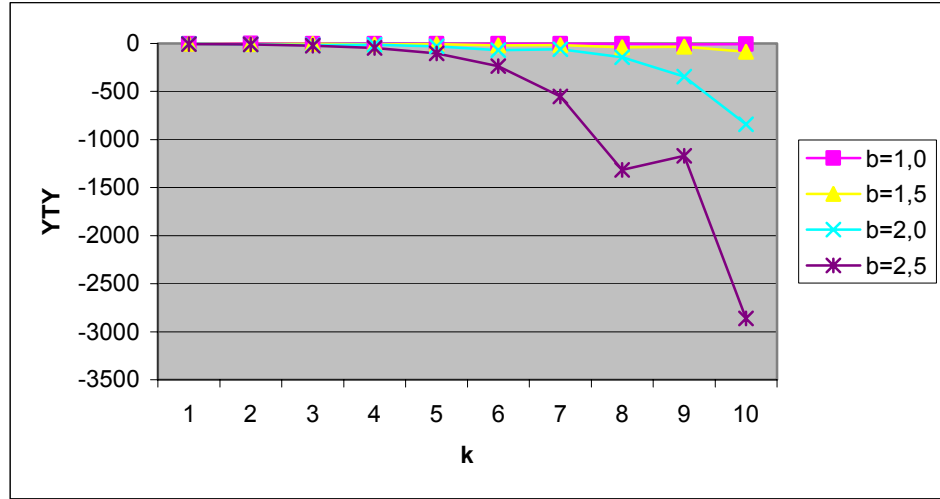


Şekil 4.17: $f(x) = k | x |$ için konuma göre YTY karşılaştırmasının grafiksel dağılımı

Şekil 4.17'de görüldüğü gibi büyük kapsama alanına sahip yerlerdeki YTY'ler, küçük alanlardakinden daha fazladır. Olması gereken sonuç budur, çünkü büyük alanlardan gelecek paket yoğunluğu daha fazladır. Ayrıca alanın büyüklüğüne göre artan YTY değeri, paketlerin bloke olma ihtimallerini de artırmaktadır. Aynı şekilde (4.8) için YTY sonuçları Tablo 4.9'da, grafiksel gösterim Şekil 4.18'de yer almaktadır:

Tablo 4.9: $f(x) = -e^{k|x|}$ için b değerlerine göre YTY karşılaştırması

k	YTY(b=1,0 km)	YTY(b=1,5 km)	YTY(b=2,0 km)	YTY(b=2,5 km)
1	-2,42962	-3,6444	-4,859238	-6,074047
2	-1,21481	-6,77499	-9,03332	-11,291654
3	-3,01111	-4,51666	-17,988127	-22,485159
4	-2,25833	-10,118322	-13,491096	-47,354509
5	-1,80666	-8,094657	-30,30689	-104,18245
6	-4,49703	-18,9418	-69,45497	-236,9856
7	-3,8546	-16,235832	-59,53283	-552,97645
8	-3,37277	-39,06842	-142,19136	-1315,8755
9	-8,418579	-34,727484	-344,07424	-1169,6671
10	-7,576721	-85,314815	-842,16033	-2861,8471



Şekil 4.18: $f(x) = -e^{k|x|}$ için konuma göre YTY karşılaştırmasının grafiksel dağılımı

Şekil 4.18’de gözlenebildiği gibi YTY’ler negatif değerde çıkmaktadır. Normalde pozitif olması gereken bu değeri, negatif veren fonksiyonlar algoritmada kullanılamaz. Grafikte görüldüğü gibi kapsama alanındaki büyüme, YTY değerinde olması gereken artışı ortaya çıkarmamaktadır. Aksine aynı kapsama alanı içerisinde bile YTY değerlerinde bazen büyüme, bazen küçülme olduğu görülmektedir.

Önemli bir nokta, önceden yapılan çalışmalarda trafik yoğunluğu birimlerinin “çağrı sayısı/dakika” (Chen ve Jamalipour, 2001b; Septiawan, 2004), “terabyte/gün” (Korçak ve diğ., 2007) gibi örnekler şeklinde ele alınmış olmasıdır. Bu çalışmada ise özel olarak böyle bir sınıflandırma yapılmadan fonksiyona göre şekillenen bir yapı mevcuttur. YTY değeri bulunduktan sonra ÇBO değeri $[0,1]$ aralığında elde edilmiştir.

5. TARTIŞMA VE SONUÇ

Zaman ilerledikçe uydu sistemlerinin önemi artmaktadır. Teknolojik ihtiyaçların, dünyada uzak bölgeler arasındaki iletişimi kapsamıyla birlikte uydu iletişimi gün geçtikçe yayılmaktadır. İletişim sırasında uyduların hem kendi aralarındaki, hem de karasal sistemlerle bağlantılarındaki paket iletiminin en hızlı ve güvenilir şekilde yapılması bir zorunluluktur. Yönlendirme algoritmalarının devreye girdiği bu işlemler için gerekirse optimizasyon yöntemleri de ele alınmalıdır. Bu çalışmada paketlerin iletildiği yollar, gecikme süresi, dayanıklılık kriteri ve çağrıların bloke olma olasılıkları açısından incelenmektedir. İletişimin yapıldığı alanın büyüklüğüne ve zaman dilimine göre çağrıların bloke olma olasılıkları hesaplanmaktadır. Önceden yapılan çalışmalarda bütün bu kriterler bir arada bulunmamaktadır. Bu açıdan değerlendirildiğinde yeni algoritmanın içerdiği önem anlaşılmaktadır. Ayrıca hesaplamalarda kullanılan ölçüler, AY yörüngelerin ve ekvatorun gerçek uzunluklarını kapsamaktadır. Bu nedenle algoritmanın mevcut uydu sistemlerinde kullanılması durumunda kolaylık sağlanmış olmaktadır. Simülasyonda kullanılan sistemlerin uydu sayıları, AY sistemlerinde bulunan uydu sayılarından da fazla olabilmektedir. Bu durum, algoritmanın büyük uydu sayılı sistemlerde rahatlıkla kullanılabileceğini göstermektedir. Elde edilen sonuçlarda görüldüğü gibi uydu sayılarının artması, algoritmanın çalışma düzenini değiştirmemektedir. Sonuçta gecikme süresi düşük, aktarım sayısı az ve çağrı bloke olma olasılık değeri düşük olan yollar sırayla karşımıza çıkmaktadır.

Gecikme süresinin düşük olması, uydunun çalışma enerjisinin kısa zaman boyunca kullanılmasını sağladığı için böyle bir durumda uyduların güç tüketimi çok fazla olmamaktadır. Bu da kullanılan uyduların çok maliyet gerektirmeyen ve düşük güç tüketimini sağlayan sistemlerden oluşabilmesine olanak vermektedir. Aktarım sayısı düşük olan yolun dayanıklılık kriteri, dolayısıyla kalıcılık durumu fazladır. Böylece çağrıların iletiminde başlangıç zamanından bitiş zamanına kadar kalıcı olan yolun seçilmesi tercih edilmektedir. Çağrıyı herhangi bir anda reddeden veya bloke eden

yolun seçilmesi ise haberleşmenin kalitesini düşürmektedir. Özellikle interaktif haberleşme çeşitlerinde anında iletişim gerçekleşeceği için çağrıyı kesintiye uğratmadan iletim boyunca taşıyabilecek yolun seçilmesi zorunlu olmaktadır. Çağrı bloke olma olasılık değerinden elde edilen analitik çözümle de çağrıların iletimleri için güvenli, kesintiye neden olmayan yollar ortaya çıkmaktadır.

Bu çalışmanın ileriki aşamalarında karasal terminal sistemleriyle bağlantı oluşturulabilir. Böylece UAL bağlantılarına ek olarak YAL bağlantıları da devreye girmiş olmaktadır. Karasal linklerdeki trafik yoğunlukları da hesaba katılarak uydu sistemine ulaşmadan önceki gecikme değerleri de hesap edilebilir. Böylece kara-uydu birleşik sistemindeki tüm paketlerin iletimleri boyunca geçen süre ve trafik yoğunluğuna neden olan şartlar ortaya çıkar. Ayrıca bu iki ortamın ayrı ayrı durumları incelenip karşılaştırılabilir. Bunun dışında herbir uyduya gelen paketlerin kuyrukta beklediği düşünülerek işlemler yeniden hesaplanabilir. Kuyruk analizleri çeşitli olasılık modellerine dayandırılabilir. Bu amaçla olasılık dağılımları ayrıntılı şekilde incelenerek paketlerin iletiminde yeni modeller ortaya çıkarılabilir. Kuyruktaki bekleme süreleri en aza indirgenmeye çalışılarak geçen süreye dayanan yeni analitik çözümler elde edilebilir. Böylelikle bu çalışmaya farklı trafik senaryoları kolaylıkla adapte edilebilir.

KAYNAKLAR

- ALNOWIBET, K.A., 2004, *Nonstationary Erlang Loss Queues and Networks*, PhD Thesis, Graduate Faculty of North Carolina State University, Raleigh.
- BEASLEY, D., BULL, D.R. and MARTIN, R.R., 1993, An Overview of Genetic Algorithms : Part 2, Research Topics. *University Computing*. Vol.15, No.4, 170-181.
- BOGOMOLNY, A., 2007, *Pythagorean Theorem* [online], <http://www.cut-the-knot.org/pythagoras/index.shtml> [Ziyaret Tarihi: 21 Ağustos 2007].
- BOUKHATEM, L., GAITI, D. and PUJOLLE, G., 2003, A time-based reservation scheme for managing handovers in satellite systems. *International Journal of Network Management*. Vol.13, No.2, 139-145.
- BURGESS, C.J., 1999, *Tournament Selection* [online], University of Bristol, <http://www.cs.bris.ac.uk/~colin/evolcollect1/evolcollect0/sld050.htm> [Ziyaret Tarihi: 20 Şubat 2007].
- CARNAHAN, J. and SIMHA, R., 2001, Nature's Algorithms: Natural and Social Metaphors in Algorithm Design, *IEEE Potentials*, Vol.20, Issue 2 (April-May 2001).
- CHANG, H.S., KIM, B.W., LEE, C.G., MIN, S.L., CHOI, Y., YANG, H.S., KIM, D.N. and KIM, C.S., 1996a, FSA-Based link assignment and routing in low-earth orbit satellite networks. *IEEE Transactions on Vehicular Technology (IEEE VTC'96)*, April 1996, Atlanta, GA.
- CHANG, H.S., CHOI, Y., KIM, B.W., YANG, H.S., LEE, C.G., KIM, D.N., MIN, S.L. and KIM, C.S., 1996b, Performance Comparison of Static Routing and Dynamic Routing in Low-Earth Orbit Satellite Networks. *International Journal of Software Engineering and Knowledge Engineering*. Vol.6, No.4, December 1996, 585-608.
- CHEN, C., 2003, A QoS-based routing algorithm in multimedia satellite networks. *IEEE 58th Vehicular Technology Conference (VTC2003-Fall Conference)*, 6-9 October 2003, Orlando, Florida. Vol.4, 0-7803-7954-3, 2703-2707.
- CHEN, J. and JAMALIPOUR, A., 2001a, Adaptive Channel Management for Routing and Handoff in Broadband WATM Mobile Satellite Networks. *IEEE International Conference on Communications (ICC2001)*, 11-14 June 2001, Helsinki, Finland. 2928-2932.

- CHEN, J. and JAMALIPOUR, A., 2001b, A Smooth Routing Algorithm Based on Traffic Oriented Path Preserved Scheme for Mobile Satellite Networks. *IEEE Global Telecommunications Conference (GLOBECOM '01)*, 25-29 November 2001, San Antonio, TX, USA. Vol.4, 0-7803-7206-9, 2725-2729.
- CHOU, H., PREMKUMAR, G. and CHU, C.-H., 2001, Genetic Algorithms for Communications Network Design-An Empirical Study of the Factors that Influence Performance. *IEEE Transactions on Evolutionary Computation*. Vol.5, No.3 (June 2001).
- COMPASSROSE, 1999, *Highly elliptical orbit* [online], CompassRose International Publications, http://www.compassroseintl.com/pubs/Intro_to_sats.html [Ziyaret Tarihi: 14 Şubat 2007].
- CORREA, R.C., FERREIRA, A. and REBREYEND, P., 1999, Scheduling Multiprocessor Tasks with Genetic Algorithms. *IEEE Transactions on Parallel and Distributed Systems*. Vol.10, No.8 (August 1999).
- EKICI, E. and AKYILDIZ, I.F., 2001, Datagram routing algorithms for LEO satellite networks. *IEEE/ACM Transactions on Networking (TON)*. Vol. 9, Issue 2 (April 2001), ISSN: 1063-6692, 137-147.
- ERCETIN, O., KRISHNAMURTHY, S., DAO, S.K. and TASSIULAS, L., 2000, A Predictive QoS Routing Protocol for Broadband LEO Satellite Networks. *Personal, Indoor, Mobile, Radio Communications Conference 2000, September 2000, London, UK*.
- FISSGUS, U., 2000, Scheduling Using Genetic Algorithms. *The 20th International Conference on Distributed Computing Systems (ICDCS 2000)*, 10-13 April 2000. 662-669.
- FRANCK, L. and MARAL, G., 2000, Candidate algorithms for routing in a network of Inter-satellite links. *18th AIAA International Communications Satellite Systems Conference and Exhibit, 10-14 April 2000 Oakland, CA*. Collection of Technical Papers, Vol. 2 (A00-25001 06-32).
- HAUPT, S.E. and HAUPT, R.L., 1998, Optimizing Complex Systems. *Aerospace Conference, 21-28 March 1998 Snowmass at Aspen, CO, USA*. Proceedings, Vol.4, 0-7803-4311-5, 241-247.
- HENDERSON, T.R. and KATZ, R.H., 2000, On Distributed, Geographic-Based Packet Routing for LEO Satellite Networks. *Proceedings of IEEE Globecom 2000, November 2000, San Francisco, CA*.
- HU, Y. and LI, V.O.K., 2001, Logical topology-based routing in LEO constellations. *IEEE International Conference on Communications (ICC 2001)*, 11-14 June 2001, Helsinki, Finland. Vol.10, 0-7803-7097-1, 3172-3176.

- IEEE, 1995, *Chapter 4, Introduction to Evolutionary Computing Techniques* [online], IEEE Xplore, 0-8186-7085-1, <http://ieeexplore.ieee.org/iel2/3190/9073/00403482.pdf?arnumber=403482> [Ziyaret Tarihi: 15 Şubat 2007].
- INETDAEMON.COM, 2005, *Satellite Orbits* [online], InetDaemon Enterprises, <http://www.inetdaemon.com/tutorials/satellite/orbits.shtml> [Ziyaret Tarihi: 14 Şubat 2007].
- JAIN, S., FALL, K. and PATRA, R., 2004, Routing in a Delay Tolerant Network. *SIGCOMM 2004, 30 August-3 September 2004, Portland, Oregon, USA*. Association for Computing Machinery (ACM) 1-58113-862-8/04/008.
- JOHNSON, M.A., LEE, S. and WILSON, J.R., 1991, Experimental Evaluation of a Procedure for Estimating Nonhomogeneous Poisson Processes Having Cyclic Behavior. *Proceedings of the 1991 Winter Simulation Conference*, 958-967.
- JUKAN, A., NGUYEN, H.N. and VAN AS, H.R., 2000, An Approach to QoS-based Routing for LEO Satellite Networks. *International Conference on Communication Technology (ICCT 2000), 21-25 August 2000, Beijing, China*. Vol. I (2000), 0-7803-6394-9, 922 – 929.
- KIM, Y.S. and PARK, W.J., 2000, Adaptive Routing in LEO Satellite Networks. *Vehicular Technology Conference Proceedings, VTC 2000-Spring, Tokyo, Japan*. Vol.3, 1983-1987.
- KLEINROCK, L., 1975, *Queueing Systems, Volume 1: Theory*, A Wiley-Interscience Publication, John Wiley&Sons, USA, 0-471-49110-1.
- KO, K.-T., TANG, K.-S., CHAN, C.-Y., MAN, K.-F. and KWONG, S., 1997, Using Genetic Algorithms to Design Mesh Networks. *Computer*. Vol.30, Issue 8 (August 1997), 0018-9162, 56-61.
- KOJIMA, K., KAWAMATA, W. and MATSUO, H., 2000, Network based Genetic Algorithm. *7th International Conference on Parallel and Distributed Systems: Workshops, 4-7 July 2000*. 0-7695-0571-6, 15-20.
- KORÇAK, Ö., ALAGÖZ, F. and JAMALIPOUR, A., 2007, Priority-based adaptive routing in N GEO satellite networks. *International Journal of Communication Systems*. Vol.20, March 2007, 313-333.
- KOZA, J.R., 1995, Survey of Genetic Algorithms and Genetic Programming. *Proceedings of 1995 WESCON Conference*. 0-7803-2636-9, 589-594.
- LIANG, W. and NAI-THONG, Z., 2002, Dynamic Location Management in LEO Networks. *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 15-18 September 2002*. Vol.3, 0-7803-7589-0, 1397-1401.

- LUI, J.C.S., TAM, P.T.S. and CHAN, H.W., 2000, Routing and Channel Reservation Strategies for a Low Earth Orbit Satellite System. *Eurocomm 2000, 17-20 May 2000, Munich, Germany*.
- MAREK, O., 1998, *Rank Selection* [online], <http://cs.felk.cvut.cz/~xobitko/ga/selection.html> [Ziyaret Tarihi: 20 Şubat 2007].
- MASSEY, W.A. and WHITT, W., 1994, A Stochastic Model to Capture Space and Time Dynamics in Wireless Communication Systems. *Probability in the Engineering and Informational Sciences*. Vol.8, 541-569.
- MASSEY, W.A., PARKER, G.A. and WHITT, W., 1996, Estimating the Parameters of a Nonhomogeneous Poisson Process With Linear Rate. *Telecommunication Systems*. Vol.5, J.C.Baltzer AG. Science Publishers, 361-388.
- MIKAC, B., INKRET, R. and PODNAR, I., 1998, An Application of Heuristic Search Techniques in Telecommunication System Design. *2nd International Conference on Knowledge-Based Intelligent Electronic Systems, 21-23 April 1998, Adelaide, Australia*. 0-7803-4316-6, 311-317.
- MORGAN, W.L. and GORDON, G.D., 1989, *Communications Satellite Handbook*, A Wiley-Interscience Publication, John Wiley&Sons, USA, 0-471-31603-2.
- NOAKES, M.D., CAIN, J.B., NIETO, J.W. and ALTHOUSE, E.L., 1993, An Adaptive Link Assignment Algorithm for Dynamically Changing Topologies. *IEEE Transactions on Communications*. Vol.41, No.5, May 1993, 694-706.
- OBRADOVIC, V. and KANTH, M.L., 2000, Improvement of the Capacity of the LEO Mobile Satellite Systems Using Channel Reshuffling. *IEEE Global Telecommunications Conference (GLOBECOM'00), 27 November-1 December 2000, San Francisco, CA, USA*. Vol.2, 0-7803-6451-1, 1166-1170.
- POHLHEIM, H., 2005, *Stochastic Universal Sampling* [online], http://www.geatbx.com/docu/algindex-02.html#P452_27492 [Ziyaret Tarihi: 20 Şubat 2007].
- RAD, R.F., 2004, *Satellite Communications: A potential to meet the explosive Internet demand* [online], Teksia Inc., http://www.siliconiran.com/magazine/science_technology/issue2.shtml [Ziyaret Tarihi: 14 Şubat 2007].
- RICHHARIA, M., 1999, *Satellite Communication Systems Design Principles, Second Edition*, McGraw-Hill, 0-07-134208-7.
- SEPTIAWAN, R., 2004, *Multiservice Traffic Allocation in LEO Satellite Communications*, Thesis (PhD). Faculty of Information Technology, Bond University.

- SHIN, D.-W., CHONG, E.K.P. and SIEGEL, H.J., 2002, A Multiconstraint QoS Routing Scheme Using a Modified Dijkstra's Algorithm. *Networks 2002, B.Bing and P.Lorenz, Eds., 26-29 August 2002, Atlanta, Georgia.* 65-76.
- TALEB, T., KATO, N. and NEMOTO, Y., 2005, Recent Trends In IP/NGEO Satellite Communication Systems: Transport, Routing, and Mobility Management Concerns. *IEEE Wireless Communications, October 2005.* Vol.12, Issue 5, ISSN: 1536-1284, 63-69.
- THE TECH MUSEUM OF INNOVATION, 2006, *The Satellite Site (What is a Satellite, anyway?, What do Satellites do?, Satellite Orbits)* [online], Lockheed Martin Co., <http://www.thetech.org/exhibits/online/satellite/> [Ziyaret Tarihi: 14 Şubat 2007].
- UZUNALIOGLU, H. and YEN, W., 1997, Managing Connection Handover in Satellite Networks. *IEEE Global Telecommunications Conference (GLOBECOM'97), 3-8 November 1997, Phoenix, AZ, USA.* Vol.3, 0-7803-4198-8, 1606-1610.
- WERNER, M., DELUCCHI, C., MARAL, G. and DE RIDDER, J.-J., 1996, ATM Virtual Path Routing for LEO/MEO Satellite Networks With Intersatellite Links. *Satellite Systems for Mobile Communications and Navigation, 13-15 May 1996.* Conference Publication No. 424 © IEE, 143-146.
- WERNER, M., DELUCCHI, C., VÖGEL, H.-J., MARAL, G. and RIDDER, J.-J.D., 1997, ATM-Based Routing in LEO/MEO Satellite Networks with Intersatellite Links. *IEEE Journal on Selected Areas in Communications.* Vol.15, No.1, January 1997, 69-82.
- WERNER, M., MAYER, C., MARAL, G. and HOLZBOCK, M., 1998, A Neural Network Approach To Distributed Adaptive Routing of LEO Intersatellite Link Traffic. *48th Vehicular Technology Conference (VTC'98), 18-21 May 1998, Ottawa, Canada.* Vol.2, 0-7803-4320-4, 1498-1502.
- WHITLEY, D., 1993, *A Genetic Algorithm Tutorial* [online], Colorado State University, <http://citeseer.ist.psu.edu/correct/177719> [Ziyaret Tarihi: 15 Şubat 2007].
- WHITLEY, D., 2001, An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls. *Information and Software Technology.* Vol.43, No.14, 817-831.
- WHITLEY, D., 2002, *Genetic Algorithms and Evolutionary Computing* [online], Van Nostrand's Scientific Encyclopedia, <http://citeseer.ist.psu.edu/correct/570959> [Ziyaret Tarihi: 15 Şubat 2007].
- WIKIMEDIA FOUNDATION, INC., 2006, *Birth-death process* [online], http://en.wikipedia.org/wiki/Birth-death_process [Ziyaret Tarihi: 8 Mart 2007].
- WILLIS, M.J., HIDEN, H.G., MARENBACH, P., McKAY, B. and MONTAGUE, G.A., 1997, Genetic Programming: An Introduction and Survey of Applications.

Genetic Algorithms in Engineering Systems: Innovations and Applications, 2-4 September 1997. Conference Publication No.446,© IEE.

- ZAIM, A.H., 2001, Routing in LEO Satellite Networks: An Implementation. *Istanbul University-Journal of Electrical & Electronics (IU-JEEE)*. Vol.1, ISSN: 1303-0914, 179-191.
- ZAIM, A.H., PERROS, H.G. and ROUSKAS, G.N., 2003, Computing Call-Blocking Probabilities in LEO Satellite Constellations. *IEEE Transactions on Vehicular Technology*. Vol.52, No.3, May 2003, 622-636.
- ZHOU, B., BASSIOUNI, M.A. and LI, G., 2005, Routing and Wavelength Assignment in Optical Networks Using Logical Link Representation and Efficient Bitwise Computation. *Photonic Network Communications*. Springer Netherlands, vol.10, No.3, November 2005, 333-346.

EKLER

EK-A ALGORİTMADAKİ ADIMLARIN ÖZETİ

Algoritmada gerçekleşen tüm işlemleri şu şekilde sıralamak mümkündür:

1. Sistemin tasarımı yapılır. Komşu uydular arasına UAL'ler konur. Gerekli UAL'lerin aktifleştirilmesi sağlanır.
2. Herbir topoloji için (herbir Δt aralığı için):

Herbir yol için:

- i. Gecikme süreleri hesaplanır.
- ii. ÇBO değeri hesaplanır. Bunun için yolun tüm linklerine ait sunucu sayısı ve yol üzerindeki uydu sayısı bilinmelidir.
- iii. ÇBO değerlerinin katılmasıyla elde edilen yeni gecikme süreleri bulunur.
- iv. Uygunluk fonksiyonu için oranlar hesaplanır. Bunun için n tane yola sahip olan bir jenerasyonda (n nüfuslu bir popülasyonda) gecikme süresi minimum olan yoldan başlayarak sırayla $\frac{n}{(n * (n + 1)) / 2}$, $\frac{n - 1}{(n * (n + 1)) / 2}$, $\frac{n - 2}{(n * (n + 1)) / 2}$, ..., $\frac{1}{(n * (n + 1)) / 2}$ değerleri yeni oranları oluşturmaktadır.
- v. Yeniden üretim işlemi gerçekleştirilir.
- vi. Oluşan yeni jenerasyonda i.-iv. adımları tekrar gerçekleştirilir.

Çaprazlama işlemleri yapılır. Ortaya çıkan jenerasyonlar arasında ölçülebilen Dayanıklılık Kriteri hesaplanır. Bundan sonra i.-iv. adımları tekrar gerçekleştirilir.

EK-B SİMÜLASYON ÇIKTILARI

Bu bölümde, Şekil 4.7'deki gibi 60 uyduluk AY sistemi üzerinde simülasyonun çalıştırılmasıyla elde edilen sonuçlar yer almaktadır. Bu sistemde yer alan 26, 27, 32, 33, 38 ve 39 numaralı uydular en yoğun paket trafiğinin olduğu bölgelerde bulunsun. Böyle bir durumda paketlerin bloke olma durumu artacağı için belirtilen uyduların içinde bulunduğu yolların seçilme olasılığının düşük olması gerekmektedir. Birçok kaynak-hedef uydu çifti belirlenerek simülasyon sonucunda paketlerin izlediği yollar elde edilmiştir. Kaynak-hedef uyduların, özellikle yoğun uydu bölgelerinin civarında seçilmesine dikkat edilmiştir. Kullanılan sistem dinamik olduğu için topolojiler arasında linklerin aktiflik durumları farklı olabilmektedir. Yani Şekil 4.7'de görülen link durumları zamanla değişerek sistem farklılaşmaktadır.

Kaynak uydu olarak 19, hedef uydu olarak da 34 seçildiğinde, haberleşme süresince ardışık 5 zaman dilimindeki topolojilerden elde edilen yollar sırayla aşağıda görülmektedir:

1. TOPOLOJIDE YOLLARIN DIZILMIS SON HALİ:

```
1.Yol=19,18,23,22,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
2.Yol=19,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
3.Yol=19,18,24,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
4.Yol=19,18,24,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
5.Yol=19,18,12,17,23,18,24,29,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
6.Yol=19,18,23,17,23,22,23,29,28,29,35,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
7.Yol=19,18,24,18,19,18,24,25,19,25,19,18,19,18,24,25,24,29,28,34.
8.Yol=19,25,26,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
9.Yol=19,18,24,30,35,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
10.Yol=19,25,26,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
11.Yol=19,18,12,17,23,18,24,29,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
12.Yol=19,25,26,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
13.Yol=19,25,26,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
14.Yol=19,25,26,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
15.Yol=19,25,26,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
16.Yol=19,25,26,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
17.Yol=19,20,26,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
18.Yol=19,25,24,30,31,25,26,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
19.Yol=19,25,24,30,31,25,26,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
20.Yol=19,25,26,20,21,15,21,27,33,27,21,20,21,15,16,22,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
```

2. TOPOLOJIDE YOLLARIN DIZILMIS SON HALİ:

```
1.Yol=19,18,23,22,16,22,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
2.Yol=19,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
3.Yol=19,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
4.Yol=19,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
5.Yol=19,13,14,20,14,15,21,22,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
6.Yol=19,13,14,20,14,15,21,22,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
7.Yol=19,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
8.Yol=19,13,14,20,14,15,21,22,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
9.Yol=19,18,23,22,21,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
10.Yol=19,18,23,22,21,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
11.Yol=19,20,26,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
12.Yol=19,25,24,18,24,30,24,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
13.Yol=19,20,26,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
14.Yol=19,25,31,32,26,25,31,37,36,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
15.Yol=19,25,31,32,26,25,31,37,36,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
16.Yol=19,25,26,25,26,25,31,30,35,41,40,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
17.Yol=19,18,12,6,11,17,23,17,23,18,24,18,23,22,28,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
18.Yol=19,18,12,6,11,17,23,17,23,18,24,18,23,22,28,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
19.Yol=19,18,12,6,11,17,23,17,23,18,24,18,23,22,28,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
20.Yol=19,25,31,37,38,39,38,39,40,41,47,42,47,46,40,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
```

3. TOPOLOJİDE YOLLARIN DİZİLMİŞ SON HALİ:

1.Yol=19,25,31,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 2.Yol=19,25,24,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 3.Yol=19,18,23,17,23,18,12,18,24,29,28,29,35,34,-1,-1,-1,-1,-1,-1.
 4.Yol=19,18,23,17,23,18,12,18,24,29,28,29,35,34,-1,-1,-1,-1,-1,-1.
 5.Yol=19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 6.Yol=19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 7.Yol=19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 8.Yol=19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 9.Yol=19,20,26,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 10.Yol=19,25,31,25,19,25,26,20,21,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1.
 11.Yol=19,25,26,20,19,25,24,18,24,29,35,41,40,46,40,34,-1,-1,-1,-1.
 12.Yol=19,20,26,27,33,32,38,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 13.Yol=19,20,26,27,33,32,38,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 14.Yol=19,13,12,6,0,54,48,42,36,37,38,39,40,34,-1,-1,-1,-1,-1,-1.
 15.Yol=19,25,31,32,38,32,33,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 16.Yol=19,20,26,27,33,32,38,32,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 17.Yol=19,25,31,37,38,39,38,44,43,37,38,44,45,46,40,34,-1,-1,-1,-1.
 18.Yol=19,20,21,27,21,20,26,32,31,32,26,25,31,37,31,30,36,30,35,34.
 19.Yol=19,18,12,17,23,17,23,18,19,20,26,27,26,20,21,27,28,34,-1,-1.
 20.Yol=19,20,21,27,21,27,26,20,26,32,26,25,31,37,38,44,38,39,33,34.

4. TOPOLOJİDE YOLLARIN DİZİLMİŞ SON HALİ:

1.Yol=19,18,23,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 2.Yol=19,18,23,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 3.Yol=19,18,23,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 4.Yol=19,18,23,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 5.Yol=19,20,19,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 6.Yol=19,25,24,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 7.Yol=19,25,24,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 8.Yol=19,25,24,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 9.Yol=19,13,14,13,12,17,16,17,23,29,28,34,-1,-1,-1,-1,-1,-1,-1,-1.
 10.Yol=19,13,14,13,12,17,16,17,23,29,28,34,-1,-1,-1,-1,-1,-1,-1,-1.
 11.Yol=19,13,12,17,12,13,14,8,14,15,16,10,16,17,12,18,23,22,28,34.
 12.Yol=19,20,21,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 13.Yol=19,20,21,22,21,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 14.Yol=19,13,19,20,19,20,26,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 15.Yol=19,25,26,32,31,30,24,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1.
 16.Yol=19,18,19,18,24,18,24,18,24,25,31,32,33,32,33,34,-1,-1,-1,-1.
 17.Yol=19,18,19,18,24,18,24,18,24,25,31,32,33,32,33,34,-1,-1,-1,-1.
 18.Yol=19,20,21,15,16,15,9,8,9,15,21,27,28,27,33,34,-1,-1,-1,-1.
 19.Yol=19,25,26,32,38,32,38,37,38,39,40,46,52,53,47,41,35,34,-1,-1.
 20.Yol=19,25,26,32,38,32,38,37,38,39,40,46,52,53,47,41,35,34,-1,-1.

5. TOPOLOJİDE YOLLARIN DİZİLMİŞ SON HALİ:

1.Yol=19,25,31,25,31,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 2.Yol=19,25,24,29,35,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 3.Yol=19,18,24,18,24,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 4.Yol=19,18,24,18,24,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 5.Yol=19,18,24,18,24,30,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 6.Yol=19,18,23,29,24,18,23,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 7.Yol=19,13,12,18,24,29,35,29,24,29,35,34,-1,-1,-1,-1,-1,-1,-1,-1.
 8.Yol=19,18,19,18,24,29,28,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 9.Yol=19,18,24,18,23,29,35,41,40,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 10.Yol=19,18,19,18,24,29,28,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 11.Yol=19,18,19,18,24,29,28,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 12.Yol=19,18,19,18,24,29,28,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 13.Yol=19,18,23,29,24,30,36,41,40,41,40,34,-1,-1,-1,-1,-1,-1,-1,-1.
 14.Yol=19,20,21,27,28,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 15.Yol=19,18,19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 16.Yol=19,18,19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 17.Yol=19,18,19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 18.Yol=19,18,19,20,21,27,33,34,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1.
 19.Yol=19,25,31,30,31,30,24,25,26,32,31,32,38,32,33,34,-1,-1,-1,-1.
 20.Yol=19,25,31,30,31,30,24,25,26,32,31,32,38,32,33,34,-1,-1,-1,-1.

Topolojilerde dikkat edildiği gibi yoğun paket trafiğinin olduğu uyduları içeren yollar seçim sırasına göre daha geride yer almaktadır. Bir topolojideki her bir yol bir sonraki topolojideki tüm yollarla karşılaştırılarak aktarım sayısı en düşük olacak şekilde seçim işlemi yapılmaktadır. Bu örnekte sonuç olarak paketlerin iletimi boyunca seçilebilecek temel yollar aşağıdaki şekilde elde edilmektedir:

```
1.topolojiden secilen yol=19,18,23,29,35,34,-1,-1
-1,-1.
2.topolojiden secilen yol=19,18,23,29,35,34,-1,-1
-1,-1.
3.topolojiden secilen yol=19,25,31,25,31,30,35,34
-1,-1.
4.topolojiden secilen yol=19,20,21,27,28,34,-1,-1
-1,-1.
5.topolojiden secilen yol=19,20,21,27,28,34,-1,-1
-1,-1.
HANDOVER SAYISI=2
Tum program boyunca gecen sure=1.484000
```

Yukarıdaki ekran çıktısında görüldüğü gibi, 2.-3. ve 3.-4. topolojiler arasında düğüm farklılaşması olduğu için 2 tane aktarım işlemi gerçekleşmektedir.

Benzer şekilde 60 uyduluk sistemde, kaynak-hedef uydu çifti olarak 25-41 uyduları ele alındığında, paketlerin iletimi boyunca 10 ayrı topolojinin ortaya çıktığı düşünülmektedir elde edilen sonuç aşağıda görülmektedir:

```
1.topolojiden secilen yol=25,31,37,36,30,36,41,-1,-1
-1,-1.
2.topolojiden secilen yol=25,31,25,31,37,36,41,-1,-1
-1,-1.
3.topolojiden secilen yol=25,24,29,24,30,36,30,36,41
-1,-1.
4.topolojiden secilen yol=25,24,30,35,41,-1,-1,-1,-1
-1,-1.
5.topolojiden secilen yol=25,24,30,35,41,-1,-1,-1,-1
-1,-1.
6.topolojiden secilen yol=25,31,30,36,41,-1,-1,-1,-1
-1,-1.
7.topolojiden secilen yol=25,24,30,36,41,-1,-1,-1,-1
-1,-1.
8.topolojiden secilen yol=25,24,30,36,41,-1,-1,-1,-1
-1,-1.
9.topolojiden secilen yol=25,24,30,36,41,-1,-1,-1,-1
-1,-1.
10.topolojiden secilen yol=25,24,30,36,41,-1,-1,-1,-1
-1,-1.
HANDOVER SAYISI=5
Tum program boyunca gecen sure=2.218000
```

Kaynak-hedef uydu çifti olarak 24-35 uyduları ele alındığında, paketlerin iletimi boyunca 5 ayrı topolojinin ortaya çıktığı düşünülmektedir elde edilen sonuç şu şekildedir:

```

1.topolojiden secilen yol=24,29,35,-1,-1
-1,-1.
2.topolojiden secilen yol=24,29,35,-1,-1
-1,-1.
3.topolojiden secilen yol=24,29,35,-1,-1
-1,-1.
4.topolojiden secilen yol=24,29,35,-1,-1
-1,-1.
5.topolojiden secilen yol=24,30,35,-1,-1
-1,-1.
HANDOVER SAYISI=1
Tum program boyunca gecen sure=1.250000

```

Sonuçlarda görüldüğü gibi, öncelikle trafiği yoğun bölgede olmayan uyduların seçimine öncelik verilmektedir. Buna rağmen bazı durumlarda aktarım sayısını minimize edecek yollar içerisinde yoğun bölgelerdeki uydular yer alabilmektedir. Bu durum, aktarım sayısı göz önünde bulundurulduğu için kaçınılmaz bir durumdur. Örneğin aşağıdaki yol seçiminde yoğun paket trafiğinin olduğu 33, 38 ve 39. uyduların bulunduğu yollar, seçim işlemi gerilerdedir. Buna rağmen, önlerindeki yollar arasında aktarım sayısını düşürecek uygun yol bulunmadığı için bu yollar seçilmiştir. Kaynak-hedef uydu çifti olarak 44-34 uyduları ele alındığında bu durum şu şekilde görülmektedir:

```

1.topolojiden secilen yol=44,45,39,33,34,
-1,-1.
2.topolojiden secilen yol=44,45,39,33,34,
-1,-1.
3.topolojiden secilen yol=44,38,39,40,34,
-1,-1.
4.topolojiden secilen yol=44,38,39,40,34,
-1,-1.
5.topolojiden secilen yol=44,38,39,40,34,
-1,-1.
6.topolojiden secilen yol=44,38,39,40,34,
-1,-1.
HANDOVER SAYISI=1
Tum program boyunca gecen sure=1.625000

```

Yukarıdaki ekran çıktısında görüldüğü gibi sadece 2.-3. topolojiler arasında aktarım işlemi gerçekleşmiştir. Simülasyon sonuçlarında, gecikme süresi, yolun dayanıklılık durumu, aktarım sayısı, paketlerin yoğunluk derecesi ve bloke olma olasılıkları gibi birçok kriter devreye katılarak bulunan en uygun yollar elde edilmektedir.

ÖZGEÇMİŞ

Derya Yılmaz, 03.03.1979 tarihinde Midyat'ta doğdu. İlkokulu, Ömerli Atatürk İlkokulu'nda bitirdi. Ortaokulu Mardin Anadolu Lisesi, liseyi Aksaray Hazım Kulak Anadolu Lisesi'nde tamamladı. 1997 yılında İstanbul Üniversitesi Bilgisayar Bilimleri Mühendisliği Bölümüne girdi ve lisans derecesini 2001 yılında aldı. Aynı sene İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans eğitime ve araştırma görevliliğine başladı. 2003 yılında Yüksek Lisansını tamamladı. Ekim 2003'te İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda başladığı doktora çalışmaları ve araştırma görevliliği halen sürmektedir.