

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**TELEKOMÜNİKASYON ŞEBEKELERİNİN TRAFİK
YÖNLENDİRMESİNDE YAPAY ZEKA UYGULAMALARI**

**T.C. YÜKSEKÖĞRETİM KURULU
BOKÜMANTASYON MERKEZİ**

**YÜKSEK LİSANS TEZİ
Müh. Ersan AKSOY
504981057**

10130 -

**Tezin Enstitüye Verildiği Tarih : 03 Mayıs 2001
Tezin Savunulduğu Tarih : 06 Haziran 2001**

**Tez Danışmanı : Prof. Dr. Günsel DURUSOY
Diğer Jüri Üyeleri Doç. Dr. Hakan DELİÇ (B.Ü.)
Doç. Dr. Tamer ÖLMEZ**

Günsel Duruso
Delic
Olmez

HAZİRAN 2001

ÖNSÖZ

Bu tezin konusunun belirlenmesinde ve yönlendirilmesinde sürekli destek olan yüksek lisans danışmanım öğretim üyesi Sayın Prof. Dr. Günsel Durusoy'a, tez içeriği konusunda bana fikir veren Sayın Doç. Dr. Şakir Kocabaş'a teşekkür ederim.

Bu çalışmayı, telekomünikasyon alanı trafik yönlendirme konusunda bu doğrultuda yapılan çalışmalara yardımcı olacağını temenni ederek merhum babam Kasım Aksoy'a ithaf ederim.

Haziran, 2001

Ersan Aksoy

İÇİNDEKİLER

KISALTMALAR.....	vii
ÖZET	viii
SUMMARY	x
1. GİRİŞ.....	1
2. BAĞLAŞMA VE TELETRAFİK	2
2.1. Bağlaşma	2
2.2. Teletrafik.....	3
2.2.1. Erlang	4
2.2.1.1. Karşılaşılan trafik	5
2.2.1.2. Geçirilen trafik	6
2.2.1.3. Tıkanan trafik	6
2.2.2. Kuyruklama modeli.....	7
3. TRAFİK YÖNLENDİRME UYGULAMALARI.....	8
3.1. Trafik Yönlendirme	8
3.1.1. Yönlendirme algoritması.....	8
3.1.1.1. Stokastik nitelikli yönlendirme	9
3.1.1.2. Deterministik nitelikli yönlendirme	9
3.1.1.3. Zaman farkına bağlı yönlendirme	10

3.1.1.4. Ekonomik nitelikli yönlendirme	10
3.1.1.5. Şebeke meşguliyetine göre yönlendirme	11
3.1.1.6. Seçim kararı belirleme niteliğine göre yönlendirme	11
3.2. Yönlendirme ve İşaretleşme	12
3.2.1. Ortak kanal işaretleşmesi	12
3.3. Yönlendirme Sistemlerinin Sağlaması Gereken Koşullar	14
4. GRAF TEORİSİ VE YÖNLENDİRME ALGORİTMALARI	19
4.1. Graf Teorisi.....	19
4.1.1. Yönsüz graf	19
4.1.1.1. Adım (Walk)	20
4.1.1.2. İz (Path)	20
4.1.1.3. Çevrim (Loop).....	20
4.1.1.4. Birleşik ve ayrık graflar	21
4.1.1.5. Altgraf (Subgraph)	22
4.1.1.6. Ağaç (Tree)	22
4.1.2. Yönlü graf	23
4.2. Yönlendirme Algoritmaları.....	24
4.2.1. Minimum ağırlıklı kapsayan ağaç	24
4.2.1.1. Fragment	25
4.2.1.2. MST oluşturma algoritmaları.....	25
4.2.2. En kısa iz algoritmaları	26
4.2.2.1. Dijkstra algoritması.....	26
4.2.2.2. Bellman-Ford algoritması	27
4.2.2.3. Floyd-Warshall algoritması.....	28

4.2.2.4. Genetik algoritma.....	28
4.2.3. Yönlendirme algoritmalarının devre bağlaşmalı şebekelere uygulanması	29
4.2.3.1. Dinamik, hiyerarşik olmayan yönlendirme.....	30
4.2.3.2. Dinamik olarak kontrol edilen yönlendirme	31
4.2.3.3. Bilgi iletme.....	35
4.2.3.4. Trafik veritabanı.....	35
4.2.3.5. Taşma algoritması	36
4.2.3.6. Diğer algoritmalar	36
4.2.4. Yönlendirme algoritmalarının paket bağlaşmalı şebekelere uygulanması.....	36
4.2.4.1. En kısa iz yönlendirme algoritması.....	38
4.2.4.2. Çok izli yönlendirme.....	38
4.2.4.3. Öncelikle en kısa iz yönlendirme algoritması	39
4.2.4.4. Merkezi yönlendirme	42
4.2.4.5. Yalıtılmış yönlendirme.....	43
4.2.4.6. Delta yönlendirme	45
4.2.4.7. Dağıtılmış yönlendirme.....	46
4.2.4.8. En iyi yönlendirme	46
4.2.4.9. Taşma algoritması	48
4.2.4.10. Hiyerarşik yönlendirme.....	49
4.2.4.11. Akış-tabanlı yönlendirme.....	50
4.2.4.12. ATM tabanlı yönlendirme.....	51
4.2.4.13. Yayın yönlendirme.....	53
5. YÖNLENDİRME VE YAPAY ZEKA	56
5.1. Yapay Zeka	56
5.2. Trafik Yönlendirmede Yapay Zeka.....	57
5.2.1. Yönlendirmede indüktif öğrenme uygulamaları	57
5.2.2. Yönlendirmede yapay sinir ağları uygulamaları	58
5.2.3. Yönlendirmede genetik algoritma uygulamaları.....	60

5.2.3.1. Basit genetik algoritma	66
5.2.3.2. Genetik algoritmalar teorisi.....	72
5.2.3.3. GA'ların diğer yönlendirme algoritmalarıyla karşılaştırılması.....	75
6. GA İLE YÖNLENDİRMENİN BİLGİSAYAR SİMÜLASYONU.....	77
6.1. Sistemin Tanımlanması.....	77
6.2. Katsayıların Belirlenmesi	79
6.3. Programın Akış Diyagramı	79
7. SONUÇLAR VE TARTIŞMA.....	84
KAYNAKLAR.....	85
EK	88
ÖZGEÇMİŞ	106

KISALTMALAR

ACC	: Automatic Congestion Control
ATM	: Asynchronous Transfer Mode
CAC	: Connection Access Control
CCS	: Common Canal Signaling
DCR	: Dynamically Controlled Routing
DCS	: Digital Cross Connect
DNHR	: Dynamic, non-Hierarchical Routing
GA	: Genetic Algorithm
ISDN	: Integrated Services Digital Network
NNs	: Neural Networks
NPC	: Network Parameter Control
OSPF	: Open Shortest Path First
OTS	: Originating Tool Switch
QOS	: Quality of Service
RCC	: Routing Control Center
RNN	: Random Neural Network
SA	: Simulated Annealing
SBL	: Similarty Based Learning
SPC	: Stored Program Controlled
TCP/IP	: Transfer Control Protocol/Internet Protocol
TOS	: Type of Service
TSM	: Trunk Status Map
TTS	: Terminating Tool Switch
UPC	: Usage Parameter Control
VTs	: Via Tool Switch

TELEKOMÜNİKASYON ŞEBEKELERİNİN TRAFİK YÖNLENDİRMESİNDE YAPAY ZEKA UYGULAMALARI

ÖZET

Bu tezde, mühendislik optimizasyon problemlerinin çözümünde yeni bir yaklaşım olan Genetik Algoritma ile telekomünikasyon şebekelerinde trafik yönlendirme için en iyi çözümün bulunması amaçlanmış, ayrıca indüktif öğrenme ve yapay sinir ağları gibi diğer yapay zeka uygulamalarından kısaca bahsedilmiştir.

İletişim sistemlerinde yönlendirme (routing) fonksiyonu, hizmet kalitesi, dayanıklılık (survivability), maliyet ve güvenilirlik gibi önemli sistem özelliklerini etkiler. Yönlendirmenin hedefi, şebekedeki herhangi iki düğüm arasında başarılı bir bağlaşma kurmaktır. Bunun için, belli bir bağlaşma girişimi için, bir düğümde belirli bir devre grubu seçilir. Bu seçim, şebekedeki hata veya yığılma durumunda etkilenir. Bundan dolayı yönlendirme seçimi, şebekenin o andaki durumuna göre belirlenebilmelidir. Yönlendirme algoritması, düğüm çiftleri arasındaki bağlantıların hangi yollar üzerinden kurulacağını belirler. Eğer bu yollar her zaman aynıysa “sabit yönlendirme şeması”, zamanla değişiyorsa “dinamik yönlendirme şeması”, denir. Tezde kullanılan yönlendirme şeması, şebekenin değişen trafik akış şartlarına göre zamanla değiştiğinden dinamiktir.

Genetik Algoritmalar, doğal seçim ve doğal genetik temellerine dayalı bir arama algoritmasıdır. Bu yöntem ile her yeni nesilde meydana getirilen suni yaratıkların (0 ve 1’lerden oluşan diziler) dayanıklılığı hesaplanmaktadır. Bu diziler uygunluk fonksiyonu nisbetince bir sonraki nesilde temsil edilebilir. Her nesilde yürütülen çaprazlama (crossover) ve mutasyon işlevleri yeni yaratıklar meydana getirirler. Bu yaratıklar elemine edilerek gelen nesillerin çeşitlemelerinden daha iyi nesiller elde edilir.

Yönlendirme algoritmalarının kullanıldığı şebekelerde en kısa izi bulmaya yönelik tanımlanmış üç standart algoritma vardır. Deterministik hesaplama yöntemlerine dayanan bu algoritmalara alternatif olarak rastgele yönlendirilmiş bir algoritma olan GA (Genetik Algoritma), diğer algoritmalara oranla daha basittir ve kısa süreli hesaplama avantajına sahiptir. Tezde tasarlanan yüz düğümlü, merkezi yönlendirme esasına göre işleyen geniş alan şebekesinde birinci düğümden yüzüncü düğüme, değişen trafik şartlarına bağlı olarak, en kısa izin bulunması amacıyla GA yöntemi uygulanmıştır. Sonuç olarak bu yöntemin, her nesilde daha iyi sonuçlar vermek üzere, istenilen amacı sağladığı görülmüştür. Dinamik bir yöntem olarak tasarlanan GA yönlendirmesi seçilen nesil sayısına bağlı olarak daha iyi sonuçlar verebilir. Bu yöntemde düğümler arası hatlar belirli sırada numaralanır ve istenilen düğüm çiftleri arasındaki yollar ikili kod halinde yazılır. Bu diziler daha sonra GA operasyonlarıyla değiştirilerek en yüksek uygunluk fonksiyonu değeri elde edilmeye çalışılır. En son nesildeki en yüksek uygunluktaki dizi seçilerek deşifre edilir ve en uygun yol bulunur. Uygunluk fonksiyonu; hatlardaki gecikme, hat uzunluğu, iletim kalitesinin kombinasyonundan oluşan bir fonksiyon olup içeriği bilgisayar programının başlangıcında belirlenir. Örnek bir tasarım ve sonuçları, yorumlarıyla beraber sunulmuştur.

ARTIFICIAL INTELLIGENCE APPLICATIONS OF ROUTING FOR TELECOMMUNICATION NETWORKS

SUMMARY

In this thesis, a new approach to the engineering, optimization problem, optimization Via Genetic Algorithms, has been objected to find optimum solution of routing for telecommunication networks, in addition has been discussed, in brief the other applications of artifical intelligence like inductive learning and neural networks.

The routing function is one of the function in a communication system with most influence on important system characteristic like; grade of service, survivability, cost and reliability. The objective of the routing is to establish a successful connection between any two nodes in the network. The function of the traffic routing is the selection of a special circuit group, for a given connection attempt, at a node in the network. The choice of a circuit group may be affected by information of the availibity of failure or overload conditions of the network. So, it should be determined the selection of routing for the state of the network at that time. The routing scheme defines how a set of routes is made available for calls between a pair of nodes. If the set of routes in the routing scheme is called “fixed”, if the set of routes in the routing pattern varies, this is called “dynamic” or “adaptive”. The routing scheme that is used in the thesis is dynamic because of changing traffic conditions within time.

Genetic Algorithms are search algorithms based on the procedure of natural selection and natural genetics. By this method, it is calculated the fitness of every artifical creature (string made of 0s or 1s) in every generation. These strings are represented to the next generation for their fitness function values. The operators of crossover and mutation which are running in every generation, form new creatures. By elemination of these creatures it has been gotten better generation from previous generation.

These has been defined three standart algorithm to find minimum weight path in the networks that are used with routing schemes. It is a random directed algorithm, GA (Genetic Algorithm) is easy to understand and has got the advantage of short calculation duration proper to the other algorithms which are based on deterministic calculation methods. The planned network defined in this thesis is a wide area network running on the ultimate of central routing, has got a hundred nodes and GA method is implemented to find the shortest path between first and last nodes related to the conditions of traffic variety. As a conclusion, it has been seen that this method provides the desired conditions. GA routing may gives better results depend on selected generation number. In this way, the links between nodes are numbered and the desired ways are descripted as binary strings. Then, these strings are changed by the ways of GA operations to find and get the maximum fitness function value. The fitness function is produced from delay on links, length of links, quality of communication and it has been determined at the beginning of computer programme. A network pattern and its result are given at the appendix with its comments.

1. GİRİŞ

Yapay zeka alanında 80'li yıllardan sonra artan çalışmalar birçok endüstri alanında kullanılmaya başlanmıştır. Telekomünikasyon ise bu gelişmelerle orantılı olarak oldukça etkin biçimde bilgisayar yazılım ve donanım destekli uygulamaları kullanır hale gelmiştir. Genişleyen haberleşme şebekeleri, bunları istenilen kalitede yönetecek akıllı programlara ihtiyaç duymaktadır. Bu tezde, iki alan uygun bir şekilde birleştirilerek araştırmaya ve geliştirmeye olanak tanıyan bir çalışma amaç edinilmiştir.

Bölüm 2'de bağlaşma ve haberleşme trafiği konusunda temel bilgiler verilmiştir. Bölüm 3'te trafik yönlendirme kavramı incelenmiş ve çeşitli uygulamaları örneklerle gösterilmiştir. Bölüm 4'te trafik yönlendirmede kullanılan graf teorisine dayalı en kısa izi bulmaya yönelik standart algoritmalar yanında tezde çalışma konusu edilen yapay zeka uyarlamalı alternatif bir algoritma önerilmiştir. Bu bölümde ayrıca çeşitli şebeke yapıları üzerinden yönlendirme türleri tanıtılmıştır. Bölüm 5'te yapay zeka kısaca tanıtılmış, trafik yönlendirmeyeyle ilgili teorik bağlantılar sunulmuş ve son olarak genetik algoritmalar etrafıca incelenmiştir. Bölüm 6'da ise tezde verilen bilgisayar simülasyonu anlatılmış ve diğer algoritmalara karşı avantaj ve dezavantajları belirtilmiştir.

2. BAĞLAŞMA VE TELETRAFİK

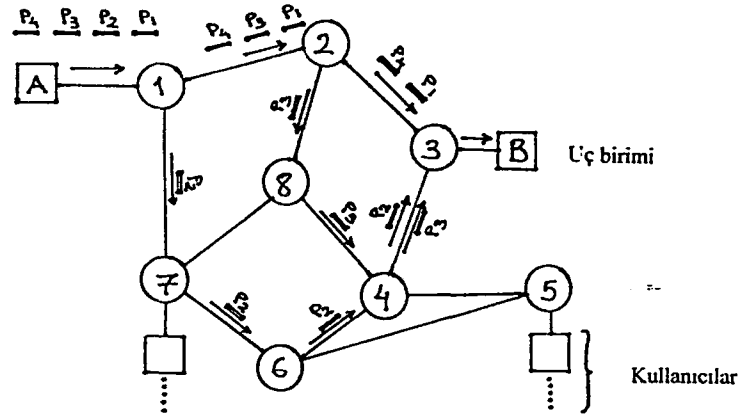
2.1. Bağlaşma

Haberleşme alanında bağlaşma; birbiriyle haberleşmek isteyen iki noktanın, kullanıcılara tahsis edilen iki uç cihazının bir yolla ve belirli bir süre bağlanmasıyla gerçekleştirilir. Telekomünikasyonda bağlaşma sistemleri çeşitli sınıflara ayrılır. Haberleşme yöntemleri açısından telekomünikasyon şebekeleri iki kısma ayrılır:

1. Devre bağlaşmalı şebekeler
2. Paket bağlaşmalı şebekeler

Devre bağlaşmalı şebekeler, telefon haberleşmesini sağlamak amacıyla geliştirilmiş bir türdür. Bu haberleşme türünde haberin depolanması söz konusu olmayıp gerçek zamanda haberleşilmektedir. Çağıran ve çağrılan uç noktaları (abone) arasında konuşma boyunca kurulan yol hizmette kalır[1].

Bilgisayarlar ve akıllı makinalarda üretilen verinin hedef noktasına gönderilmesinde özellikle 1970'lerde geliştirilmiş paket bağlaşmalı şebekelerde iletilecek mesaj paketlere bölünerek numaralandırılır ve farklı zamanlarda farklı düğümler üzerinden aynı hedefe varacak şekilde aktarılır. Bu sistemlerin devre bağlaşmalı sistemlere birçok açıdan üstünlüğü vardır. Bunlar; patlamalı tipten veri trafiğine uygun olması ve hafızalı iletim özelliğinden kaynaklanan, düğümleri birbirine bağlayan yolların gerektiği durumda kullanılmasının sağladığı ekonomik olması durumlarıdır. Şekil 2.1'de bir paket bağlaşmalı şebeke örneği gösterilmiştir.



Şekil 2.1. Paket bağlaşmalı bir şebeke

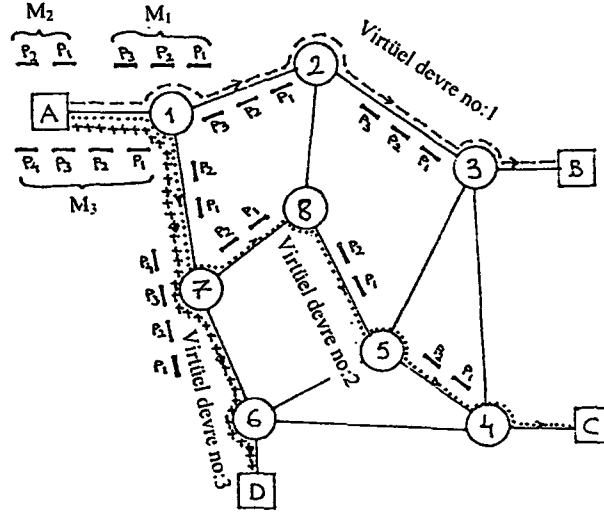
Paket bağlaşma tekniğinin iki çalışma şekli vardır:

1. Datagram modu
2. Virtüel devre modu

Datagram modunda her bir paketin gideceği bir sonraki düğüm kararı, paketin o anda bulunduğu düğümde diğer düğümlerden bağımsız şekilde verilir. Virtüel devre modunda ise bağlaşacak iki uç noktası arasındaki iletim yolu, haberleşme başlamadan önce belirlenir. Tezde öne sürülen paket bağlaşmalı şebeke modelinde de virtüel devre modunda çalışma doğrultusunda simülasyon programı geliştirilmiştir. Şekil 2.2’de virtüel devre modunda çalışan bir şebeke örneği gösterilmiştir [2].

2.2. Teletrafik

Trafik için çeşitli birimler tanımlanmıştır. Bunlardan uluslararası alanda genel kabul gören birim “Erlang”tır.



Şekil 2.2. Virtüel devre modunda çalışan şebeke

2.2.1. Erlang

Erlang birimi ITU-T (1946) tarafından şöyle tanımlanmıştır:

“Bir devre veya devreler grubu içerisinde geçen trafiğin karşılanması:

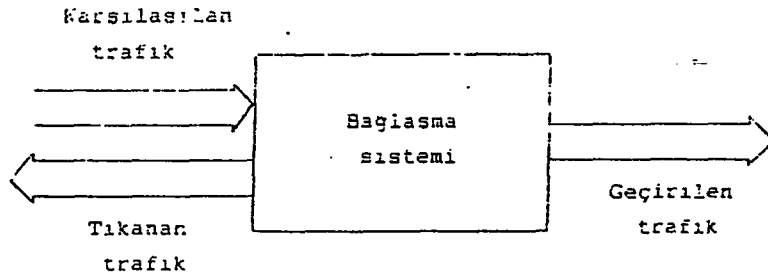
Bir devreler veya bağlantı organları grubunda bir T periyodu süresince trafiğin şiddetinin ortalaması, toplam meşguliyetin T'ye oranına eşittir. Bu şekilde tanımlanan trafik şiddeti birimine Erlang denir.”

Bir grupta taşınan trafik gruptaki meşgul alet sayısından daha fazla değildir. Grupta alet başına taşınan trafiğe aletin meşguliyeti ya da meşguliyet oranı denir.

Genel olarak üç tür trafik tanımlanır:

- 1- Karşılaşılan trafik
- 2- Taşınan trafik
- 3- Tıkanan trafik

Şekil 2.3'te trafik türleri gösterilmiştir.



Şekil 2.3. Sistemdeki trafik türleri

2.2.1.1. Karşılaşılan trafik

Yüklü saat içinde ortalama iletim süreleri Δt_1 kadar olan n_1 adet ve ortalama süreleri Δt_2 olan n_2 adet ve ortalama süreleri Δt_m olan n_m adet mesaj karşılanmış olsun. Karşılaşılan trafiğin hacmi (2.1) bağıntısı ile ifade edilir.

$$A = \sum_{i=1}^m \Delta t_i \cdot n_i \quad (2.1)$$

Bir gün içinde birbirini izleyen 60 dakika içindeki en büyük A değerine yüklü saat trafiği denir. Saatin bölümleri cinsinden ifade edilen iletim süreleri sonucu trafik şiddetinin birimi Erlang olur ve boyutsuzdur.

A değerinin fiziksel anlamı için aşağıdaki örnek incelenmektedir.

Bir yüklü saat içinde,

Ortalaması	128 ms	süren	1000 mesaj
Ortalaması	100 ms	süren	400 mesaj
Ortalaması	70 ms	süren	300 mesaj
Ortalaması	40 ms	süren	500 mesaj
Ortalaması	10 ms	süren	1200 mesaj

sisteme geldiği zaman denklem (2.1)'in yardımıyla geçirilecek trafik hesaplanır.

$$A = \sum_{i=1}^5 \Delta t_i \cdot n_i = \left(\frac{128}{3.600.000} \right) \cdot 1000 + \left(\frac{100}{3.600.000} \right) \cdot 400 + \left(\frac{70}{3.600.000} \right) \cdot 300 + \left(\frac{40}{3.600.000} \right) \cdot 500 + \left(\frac{10}{3.600.000} \right) \cdot 1200$$
$$A = \frac{128.1000 + 100.400 + 70.300 + 40.500 + 10.1200}{3.600.000} = 0.16 \text{ Erlang}$$

Mesajların hizmet organını meşgul ettiği sürenin yüklü saate oranı 0.16 değerini verir. İdeal durumda mesajların ard arda geldiğini varsaysak herbir mesaj bloke olmadan hizmet alabilir fakat pratikte gelişlerin stokastik dağılımı uyarınca bazı mesajlar bloke olabilir. Eğer sistemde yeterince bekleme pozisyonu varsa bu durumda mesaj hizmeti gecikmeli olarak alabilecektir.

2.2.1.2. Geçirilen trafik

Hizmet organına gelen mesajlar yeterince bekleme yeri yoksa kaybolabilir. Bu durumda taşınan trafik, karşılaşılan trafikten küçük olur. Eğer hizmet veren sistemde karşılaşılan trafiğe göre yeterli sayıda hizmet organı varsa bloke edilen trafik gözardı edilebilir. Bu durumda geçirilen trafik karşılaşılan trafiğe eşit kabul edilebilir. Geçirilen trafik için denklem (2.2) tanımlanabilir.

$$A^1 = A[1 - B(A, N)] \quad (2.2)$$

Burada B (A, N), A Erlang trafik ve N hizmet organı bulunması durumunda blokaj olasılığıdır.

2.2.1.3. Tıkanan trafik

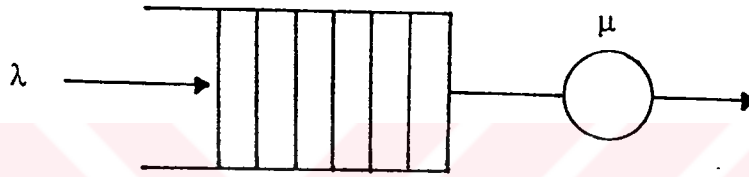
Tanım olarak tıkanan trafik bloke edilen mesajların oluşturduğu trafiktir.

$$T = A - A^1 \quad (2.3)$$

Denklem (2.3)'te tıkanan trafik formülü vermiştir.

2.2.2. Kuyruklama modeli

Paket bağlaşmalı sistemlerde veri paketlerini karşılayan hizmet organları gelen paketleri belirli bir disiplin içinde sıralayarak hizmet verirler. Mesaj paketlerinin hizmet alıncaya kadar beklediği hizmet sistemi modeline kuyruklama modeli denir. Şekil 2.4'te basit bir kuyruklama modeli gösterilmiştir. Burada λ ortalama paket geliş hızı, μ ortalama hizmet süresidir.



Şekil 2.4. Bir hizmet organlı kuyruklama modeli

Uygulamada bekleme yeri sayısı oldukça büyükse, hesaplamalarda bekleme yeri sayısı sonsuz kabul edilebilir, bu durumda blokaj olmadığı kabul edilir. Genel olarak bir kuyruklama sisteminde çok sayıda talep varsa, bir talebin hizmet alma süresi uzar. Diğer bir ifadeyle talep sayısı ile hizmet alma süresi arasında doğrusal bir ilişki vardır. Little teoremi de denilen denklem (2.4) aşağıdaki şekilde tanımlanır.

$$E_n = \lambda E_T \quad (2.4)$$

Burada E_n ortalama talep (paket) sayısı, E_T ise ortalama hizmet alma süresidir.

3. TRAFİK YÖNLENDİRME UYGULAMALARI

3.1. Trafik Yönlendirme

Haberleşme şebekelerinin temel özelliklerinden biri bünyesinde bulundurduğu aboneleri (uç noktaları, terminaller) bağlı oldukları bağlaşma merkezleri vasıtasıyla birbirlerine bağlamaktır. Herhangi iki bağlaşma merkezi arasında sözkonusu olan trafik akışının, yolların o andaki durumuna göre, en uygun olan yoldan gönderilmesi işlevi yönlendirme (routing) fonksiyonunun esas amacıdır. Yönlendirme fonksiyonu, telekomünikasyon şebekesinde genel olarak aşırı trafik yükü veya hata durumlarında çağırmanın güvenilir bir şekilde iletilmesi için belirli devre gruplarını seçer. Bu durumda yol atama işlevi geçici olarak uygun algoritmayı uygular.

Birbirlerine devre grupları vasıtasıyla bağlanan bağlaşma merkezlerinin (düğüm) oluşturduğu bir şebeke yapısında düğüm çiftleri arasında tek ya da çift yönlü iletme uygun doğrudan veya dolaylı (indirect route) çeşitli devre grupları bulunabilir. Dolaylı bir yolda kaynak ve hedef düğümleri birbirine bağlayan yolda başka düğümler de bulunur.

3.1.1. Yönlendirme algoritması

Yönlendirme algoritması, düğüm çiftleri arasındaki yol kümelerinin kullanımını belirler. İki çeşit yönlendirme şeması vardır.

1. Sabit (fixed) yönlendirme şeması
2. Dinamik yönlendirme şeması

Sabit yönlendirme algoritmasında yönlendirme şeması değişmez. Dinamik yönlendirme şemasında ise yol kümeleri duruma ya da zamana bağlı olarak değişir.

Bu deęişiklikler periyodik , aperi-yodik, manuel olarak önceden belirlenmiş veya şebekenin o andaki durumuna baęlı olarak gerekleşebilir [4].

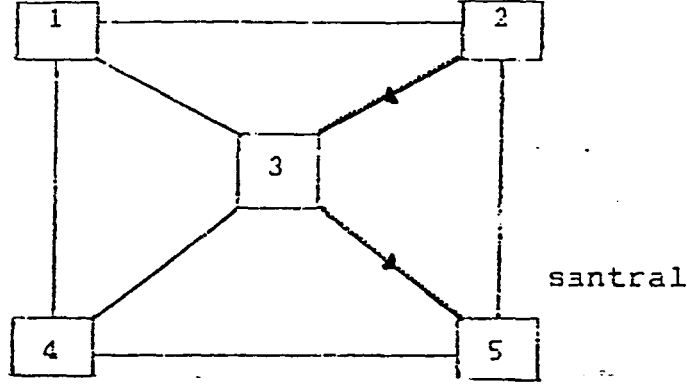
Yönlendirme stratejileri, teknolojinin sağladığı imkanlara ve haberleşmenin türü yani uygulama alanına baęlı olarak çeşitli şekillerde düzenlenebilir. Bunlar iletilecek mesajın türünde (devre ya da paket baęlaşmalı haberleşme), şebekenin tahsis edildiğı amaca göre deęişiklik gösterir (askeri ya da kamu şebekesi).

3.1.1.1. Stokastik nitelikli yönlendirme

Şebekede kullanılan hatlar belirli bir olasılık dağılımına göre kullanılıyorsa bu durumda yönlendirme stratejisinin olasılıksal olduğundan söz edilir. Herhangi bir düğüm çifti arasında birden fazla iz (baęlaşma yolu) bulunan şebekelerde kullanılan “çok izli yönlendirme” de her düğüm gelen trafiğı çıkışlarına belli bir olasılık dağılımına göre yönlendirir. Burada her düğüm mevcut olası hedefler için belirli sayıda (örneğin 5) çıkış hattını gösteren bir tablo üzerinden, üretilen rastgele bir sayının bulunduğu aralığa tekabül eden yolu seçer. Böylece mevcut trafiğı birden fazla hat üzerine aktararak hatlardaki trafik yükünü azaltır ve hizmet kalitesini artırır. Tezde kullanılan algoritma da nisbeten olasılıksal yönlendirme esasına dayanan bir anlayışa göre geliştirilmiştir. Bu algoritmada belirli düğümler arasındaki baęlaşma olasılığı baęlanmış ve böylece yönlendirme şeması belirli düğüm çiftleri arasındaki yolları kullanma durumunu dolayısıyla trafik akışını kısıtlamıştır [3].

3.1.1.2. Deterministik nitelikli yönlendirme

Şebekenin belirli bir düğümünde, belirlenmiş bir yoldan taşan trafik aynı yol kümesi üzerinden olasılığa baęlı olmadan akıyorsa, bu yönlendirme yapısına “sıra düzensel” dir denir ve deterministik nitelikli bir alternatif yönlendirmeden söz edilir. Şekil 3.1’de alternatif yönlendirmeye bir örnek şebeke yapısı verilmiştir [3]



Şekil 3.1. Şebeke yapısında alternatif yönlendirme

3.1.1.3. Zaman farkına bağlı yönlendirme

Ulusal ya da geniş coğrafi alanları kapsayan şebeke yapılarının bir kısmında meşguliyet oranı yüksek olurken diğer kısmında yerel saat uygun olmadığından boş bulunan ya da az kullanılan düğümler olabilir. Yukarıda söz konusu edilen ardışıl yönlendirme metodları bu farkın oluşturduğu avantajları kullanamadığından verimli sonuç vermezler. Oysa zaman farkından kaynaklanan meşguliyet seviyeleri arasındaki farkı; çok yoğun trafik altındaki kısmın yollarını az yoğun kısma kaydırabilen yönlendirme algoritmaları daha verimli sonuçlar verir [3].

Zamana bağlı yönlendirme stratejileri ABD, Rusya gibi geniş coğrafi alan üzerinde bulunan ülkelerde (DNHR metodu) kullanılmakta olan statik olmayan stratejilerdir. Bu metodda coğrafi alanın genişliğine göre (genellikle 1 saat) belirli zaman dilimlerinde yönlendirme stratejisi değiştirilir. ---

3.1.1.4. Ekonomik nitelikli yönlendirme

Bir şebekede mevcut yollar arasında abone ücretlendirme farkının bulunduğu durumlarda (donanım farklılığından kaynaklanan maliyet özelliklerinden dolayı) maliyeti en aza indirecek yönlendirme algoritmaları düzenlenebilir. Ayrıca şebekedeki yolların iletim kalitesi göz önünde bulundurularak belirli bir dengede yol seçimi yapılabilir. İşaret/gürültü oranı, çağırma kurma, bağlaşma, propagasyon hızı gibi etkenler maliyete göre belirli bir oranda göz önünde bulundurularak yönlendirme fonksiyonları düzenlenir [3].

3.1.1.5. Şebeke meşguliyetine göre yönlendirme

Yönlendirme algoritmalarının yapısı, şebeke meşguliyet durumuna bağlı olarak değişiyorsa dinamik yönlendirme söz konusudur. Bundan dolayı bunlara uyarlanabilir (adaptive) şemalar da denir. Şebekedeki diğer bağlaşma merkezleri ya da merkezi bir veritabanında (örneğin Routing Control Center) toplanan şebekenin düğümlerinden çıkan devre gruplarının meşguliyeti ya da blokaj durumları hakkındaki bilgiler güncelleştirilerek yönlendirme kararları her santralda veya hepsinin bağlı olduğu merkezi bir işlemcide verilir ve böylece hizmet güvenilirliği sağlanır. Bu metotta yönlendirme stratejisi meşguliyet durumuyla eş-zamanlı değişir[3].

3.1.1.6. Seçim kararı belirleme niteliğine göre yönlendirme

Önceden bahsedildiği gibi ileriye doğru çağrı kontrolün da olduğu gibi yönlendirme stratejisi mevcut yolları her bir düğümde kararlaştırarak kaynak-hedef uçlar arasında yol kurabilir. “Adım-adım” (Step by step) yönlendirme stratejisi olarak bilinen bu yöntemde kaynak düğüm seçtiği çıkan devre grubu üzerinden ilettiği veriyi bir sonraki düğüm kendi karar kıldığı yol üzerinden iletir. Bu yöntemin uygulandığı düğümlerde bir sonraki düğüme olan tüm hatları tıkalı olan düğüm geriye dönüp verinin olası diğer yollar üzerinden gönderilmesi için veriyi bir önceki düğüme gönderir. Geri-dönüslülük (crank-back) özelliği sayesinde şebekenin verimi artar. Devre bağlaşmalı telefon şebekeleri ve datagram iletim modunda çalışan şebekeler bu yönlendirme stratejilerini kullanır. Burada karar mekanizması merkezi olarak değil dağıtılmış biçimde yapılandırılmıştır [3].

Virtüel devre modunda olduğu gibi iletim yolunun kaynak santral tarafından tek bir şekilde belirlenerek yapılan yönlendirme ise “doğrudan”(right-through) yönlendirme stratejisi olarak adlandırılır. Bu şekilde yolun tamamı başlangıç aşamasında belirlenir.

3.2. Yönlendirme ve İşaretleşme

Bir telekomünikasyon şebekesinde işaretleşme fonksiyonu, sistemin çeşitli elemanları ve terminalleri arasında, bağlaşma yolunun kurulması, iletim bittiğinde çözülmesi ve kontrolünü sağlayan bilgilerin elektriksel işaretler olarak alınıp verilmesini sağlar. Yönlendirme işlevi ve işaretleşme arasında belirli bir çağırma girişimi veya trafik akışı için devre grubunu seçme işleminden, yönlendirme kararının karşı santrale haber verilmesi doğrultusunda ilişki vardır.

Geleneksel anlamda telekomünikasyon sistemlerinden olan telefon sistemlerinin ilk uygulamalarında bağlantılar operatörler yardımıyla manuel olarak gerçekleştirilmiştir, daha sonra elektromekanik otomatik santrallar kurulmuştur. Elektronikteki gelişmelerle bir yandan SPC santrallar gündeme gelmiş, öte yandan işaretleşmede santrallar arası sayısal hatlardan yararlanılmıştır. Aboneler ve bağlaşma merkezleri arasındaki haberleşme analog olmasına karşın santrallar (düğümler) arasındaki haberleşme iletim kalitesi (S/N oranı, bağlaşma gecikmesi vb.) ve ekonomik olması nedeniyle PCM hatları üzerinden sağlanabilmektedir.

3.2.1. Ortak kanal işaretleşmesi

Ortak Kanal İşaretleşmesinde, (CCS No:7 sisteminde olduğu gibi) işaretleşme bilgisi konuşma kanalından ayrılmıştır. Yani işaretleşme bilgisi birçok konuşma kanalı için ayrı tek bir ortak kanaldan taşınır. Bu halde, her işaretleşme kanalı ilgili çağırmanın etiketini de içerir. Oysa işaretleşme bilgisinin çağırma ile aynı yoldan iletimi şebekenin yeteneğini kısıtlar. Örneğin, otomatik yeniden arama, şebeke yönetimi gibi özellikler bağımsız işaretleşme olanağına ihtiyaç duyar.

No:7 Sistemi, yüksek hızda çalışması istenen ISDN şebekelerinde olduğu gibi, işaretleşme bilgisini kendi ayrık sayısal şebekesi üzerinden göndererek veri iletim kanalının üzerinden yapılan band içi işaretleşmede ortaya çıkan sınırlamaları ortadan kaldırır. Ayrık yapı, işlemcilere anında bilgi alışverişi imkanı tanır. Bu bilgi şebeke yönetimi ve müşteri hizmetlerini içerir. Ayrıca diğer işaretleşme sistemleri kaynaktan

hedefe doğru tek yönlü işaretleşme imkanı tanıdığı halde No:7 tamamen iki yönlüdür. Bu üstünlükler nedeniyle No:7 ortak kanal işaretleşmesi band-ıçi işaretleşme sistemlerinin yerini almıştır.

No: 7 uçtan uca işaretleşme yeteneği ile, bağlaşma merkezi ve yönlendirme merkezleri arasında yönlendirme bilgilerinin iletimi için güvenilir ve esnek iletişim sağlayarak merkezi ve dağıtılmış uyarlanabilir yönlendirme algoritmalarında önemli rol oynar.

Günümüzde şebeke yönetim kontrolünün genellikle No:7 ile gerçekleştirilmesi, otomatik tıkanma kontrolünün (ACC) işlerlik kazanmasını sağlamıştır. Burada, tıkanan bağlaşma merkezinin tıkanma seviyesini diğer bağlaşma merkezlerine iletmesini sağlayan ACC vasıtasıyla tıkanan bağlaşma merkezinin trafiği, diğer bağlaşma merkezlerinin önceden belirlenmiş santrallararası hat rezervasyonu kullanılarak azaltılır. Ayrıca, ortak kanal işaretleşmesi, kaynağa, aranan hedefin meşgul/boş durumuna bakabilme olanağı sağlar. Böylece gerçekleştirilemeyecek çağırımlar tarafından şebekenin daha fazla tıkanması, çağırmanın kaynağa engellenmesi ile önlenir.

Yönlendirme kararı vermede, No:7'nin katkısı olan diğer bir uygulamada kaynak, hedefe giden olası tüm yollara arama (search-out) mesajı gönderir. Hedef ise, en iyi yolu tanımlayan geri mesaj (search-back) gönderir. En iyi yol, en az bağlantı (link) ve en az olasılıkla öncelikli çağırma içermesi gibi etkenlere bağlıdır. Daha sonra çağırma, kaynak tarafından seçilen yol üzerinde kurulur.

Yüksek kapasiteli işaretleşme kanalları ve yeni işlemciler, şebekedeki kaynakların daha verimli kullanılmasını sağlar. Yüksek kapasiteli işaretleşme kanallarının olması ile, şebekede yönlendirme kararları verilirken tüm şebeke yapısını kullanma olanakları sağlayan yeni algoritmalar geliştirilmiştir.

3.3. Yönlendirme Sistemlerinin Sağlaması Gereken Koşullar

Günümüzde telekomünikasyon şebekelerinde iletim şekli şebeke yapılarına göre farklılıklar göstermektedir. Kullanılan şebekelerin çoğu telefon haberleşmesi, TV dağıtımı, devre bağlaşmalı veya paketlenmiş veri aktarımı gibi belirli amaçlara hasredilmiştir.

Geleneksel telefon haberleşmesinde sadece ses ortamı kullanılır ve her çağırmada sadece iki telefon 64 kbit/sn hızında iletim yolları ve bu hızı karşılayabilen bağlaşma organları ile birbirine bağlanır. Faks haberleşmesi gibi bazı uygulamalar, yaygın telefon şebekesini kullanır. Bir şebekeyi yeni uygulamalar için bazı düzeltmeler yapmadan kullanmak, belirgin kusurlar ortaya çıkarabilir. Çünkü bu şebekeler, bilinmeyen ve günümüzde hızla değişen hizmet gereksinimlerini karşılayacak yapıda olmayabilir. Bu yüzden telefon şebekesi üzerinden veri iletimi, analog ses iletimi donanımının band genişliğinin, esnekliğinin ve kalitesinin eksikliği ile sınırlanmıştır. Telefon şebekeleri sabit band genişlikli hizmet için tasarlandığından, bunları değişken bit-hızlı veri trafiği için kullanmak, pahalı adaptasyonlar gerektirmiştir. Kamu telefon şebekeleri ses-dışı haberleşme taleplerini abonelerin istediği ölçüde karşılayamadığı için, diğer haberleşmelere hasredilmiş şebekeler doğmuştur. Bir firmanın çeşitli ünitelerini birbirine bağlayan kamu veya özel veri şebekeleri bunlara örnek olarak verilebilir.

ISDN'nin sağladığı imkanlar kullanılarak veri, sayısallaştırılmış ses, sayısal faks, sayısallaştırılmış görüntü, sayısal alarm vb. trafikler karıştırılarak belirli hızlarda kanallar üzerinden iletilmektedir. Buna rağmen ISDN'nin 2 Mbit/sn'lik maksimum erişim hızı, 1990'lardan sonra telekomünikasyon alanında yaşanan gelişmelerin gerektirdiği iletim hızlarının oldukça gerisinde kalmıştır. Günümüzde 100 Mbit/sn'lik hızlarda iletimin gerekli olduğu uygulamalar için geniş bandlı ISDN (BISDN) hizmetleri geliştirilmiştir.

BISDN uygulamalar, çok-ortamlı, çok-noktaya erişme olanağı veren, çok-hızlı iletim ve bağlaşdırma imkanı tanıyan gelişmiş hizmetlerdir. Bu sayede ses, veri, hareketsiz

görüntü veya tam hareketli resim ortamlarının herhangi bir kombinasyonu birlikte iletim kalitesini düşürmeden hedeflere ulaştırılabilir. Halbuki, lokal alan şebekelerinin (LAN'ların) bağlanması ve iyi bir çözünürlükle hareketli resimlerin transmisyonu (video konferans vb.) gibi hallerde, çok daha yüksek bit hızları gereksinimi bu şekilde gerçekleştirilebilir [2].

BISDN'in özelliklerini açıklamak için şu örnek verilebilir: Video konferans iyi oturmuş ancak iş alanında halen yaygın olarak kullanılmayan bir hizmettir; kişiler arasında hızlı bilgi alış-verişini kolaylaştırır. BISDN sayesinde bu hizmet bugünün TV kalitesinde, hatta daha iyi resim kalitesine imkan vererek video konferansın yaygın bir telekomünikasyon aracı olmasını sağlayabilir.

Buradan da anlaşılacağı gibi bu gibi geniş bantlı uygulamalarda şebekenin iletim hızını dolayısıyla düğümlerdeki gecikmeyi en aza indirecek şekilde yönlendirme algoritması tasarlanması kaçınılmazdır. Bunun aksine telefon haberleşmesi gibi uygulamalarda ise gecikme öncelikli sorun olmayıp trafik yükünün tıkanma olasılığı düşük olan devre grupları üzerinden yönlendirilmesi daha önemlidir. Yönlendirme sistemleri, bu gibi şebeke özelliklerini göz önünde bulundurarak, çeşitli koşulları sağlayacak biçimde belirlenebilir. Bu koşullar şu başlıklar altında incelenebilir.

1. Hizmet kalitesi
2. Güvenilirlik
3. Yoğun trafikte yüksek verim yeteneği
4. Trafik seviyesindeki değişikliğe duyarlılık
5. Trafik ve donanım değişikliğine uyarlanma

Hizmet kalitesi çeşitli şekillerde tanımlanabilir. Genel olarak bir yığılma (ya da tıkanma) fonksiyonunun yorumu olarak tanımlanabileceği gibi çağırmanın hizmet sistemi tarafından bloke edilme durumu olan tıkanmanın bir fonksiyonu olarak değerlendirilebilir. Kayıp olasılığı her trafik akışı için verildiğinde hizmet kalitesinin, en iyi yol atama ile birlikte gerçekleşmesi kesin değildir. Bu durumda her trafik akış olasılığı önceden verilmiş bir değerden küçük olabileceği gibi belirli bir aralıkta olması da istenebilir. Ayrıca trafik akışının değişen değerlerinde farklı kayıp

olasılıklarının dalgalanmasının (varyans) mümkün olduğunca düşük olması sağlanmalıdır. Bu belirlemeler yönlendirme sistemi tasarımında önceden göz önünde bulundurulmuş koşullardır [5].

Yönlendirme stratejisini belirlemede önemli etkenlerden biri de yönlendirme şemalarının doğruluğu ve tutarlılığı açısından tanımlanan güvenilirlik özelliğidir. Şebekenin çok düğümlü olması gibi durumlarda yönlendirme kararları, çıkmaz noktalara (deadlock), çevrim yinelenmelerine (loop), çağırma ya da paketlerin mevcut olmayan yollara yönlendirilmesi sonucu kaybolması gibi istenmeyen durumlara sebep olabilir. Alternatif yönlendirmede aynı düğüm çiftleri arasında birden fazla yolun olması çevrim olasılığını artırır. Bu durumda yönlendirme stratejileri belirli bir hata olasılığı altında bu durumları en aza indirecek şekilde tasarlanmalıdır. Merkezi bir düğüm tarafından yapılabilecek bu hesaplamalar bu düğümün bozulması sonucu bütün şebekeyi etkileyeceğinden bu özellik şebekenin düğümleri arasında paylaştırılarak giderilmeye çalışılır. Böylece şebekenin bir kısmında meydana gelen hata, diğer düğümler tarafından giderilir.

Belirli bir trafik akışı ve öngörülen hizmet kalitesi altında bir düğümde gerekli aletlerin ya da hizmet organlarının sayısının belirlenmesine “boyutlandırma” denir. Telefon haberleşmesinde kullanılan Erlang tabloları bu işlevi sağlaması için geliştirilmişlerdir. Yönlendirme işlevi tıkanma ile yakından ilişkilidir. Yol atama fonksiyonunun doğrudan ya da dolaylı yollardaki devre sayısı trafiği geçirme kalitesini, ayrıca geri dönüşlülük özelliği sayesinde trafiğin bloke edilmesi sonucu çağırma ya da paketin kaybolması olasılığı düşeceğinden trafiği geçirme özelliğini olumlu yönde etkiler. Bu açıdan yönlendirme şemaları, en yüksek trafiği geçirecek şekilde belirlenir.

Trafik seviyesinde meydana gelen değişiklikler dolayısıyla daha düşük seviyedeki trafik için uygun olan yönlendirme algoritmaları yüksek seviyedeki trafik için verimli olmayabilir. Bu durumda belirli trafik seviyesi aralıkları için uygun olan yol atama şemaları tanımlanmalıdır.

Dinamik (uyarlanabilir) yönlendirme şemaları, son yıllarda gelişen yazılım uygulamalarıyla geniş ölçüde kullanılmaktadır. Özellikle SPC santralleri gibi işlemci kontrollü sistemlerin geliştirilmesiyle adaptif yapıların sağladığı avantajlardan yararlanılmaktadır. Uyarlamalı yönlendirme algoritmaları şebekenin donanımına ve trafik seviyesindeki değişimlere gerektiği zaman cevap verebildiğinden diğer algoritmalara göre daha çok tercih edilir. Özellikle askeri haberleşmenin söz konusu olduğu telekomünikasyon şebekelerinde şebeke topolojisinin mobil yapısı ve ortaya çıkan yeni ihtiyaçlar doğrultusunda donanım değişiklikleri göz önüne alan yönlendirme stratejileri çok önemlidir. Ortalamasının varyansının altında olduğu oldukça düzensiz (kaba, dalgalı) yapıdaki bir trafik akışı (chaotic) altında şebekede uygulanacak yol atama işlevi, hizmet kalitesini doğrudan etkileyeceği için çok önemlidir. Trafik seviyesindeki değişikliğin zamanla gösterdiği özelliklere göre yol atama stratejileri farklılıklar arzederler. Trafik önceden az çok kestirilebiliyor ve herhangi bir zaman aralığındaki ortalaması çok az değişiyorsa (durağan) deterministik (belli) bir yönlendirme algoritması daha ekonomik olacağından dinamik yönlendirme algoritmalarına tercih edilir. Buna karşın, önceden kestirilemeyen ve zamanla gelişimi bilinmeyen trafik durumlarında uyarlamalı algoritmalarla daha verimli sonuçlar alınmaktadır [6].

Şebeke yapısının herhangi bir yerinde meydana gelen bir bozulma, hata veya karışıklık sebebi ile meydana çıkan yeni durumda korunan hizmet seviyesine şebeke dayanıklılığı (survivability) denir. Bu ölçüt yüklü saatteki trafiği taşımak için gerekli olan devre grubu veya olanak sayısını veren şebeke verimliliği (network efficiency) performansını doğrudan etkiler. Günümüzde ISDN uygulamalarında olduğu gibi yüksek veri hızlarında haberleşme yapan şebekelerin donanımında kullanılan fiber-optik kablolar sayesinde bağlantı esnasında propagasyon ve bağlaşma elemanlarının sebep olduğu gecikmeden kaynaklanan “şebeke verimliliği düşmesi” giderilmiştir. Bunun yanında dinamik yönlendirme ile bu bağlantı kablolarının masraflı olmasından kaynaklanan tasarruflu kullanım politikası giderilerek şebeke dayanıklılığı artırılabilir.

Yukarıda bahsedilen özellikler şebeke tipine bağlı olarak işletici tarafından belirli bir sırada veya bazıları ihmal edilebileceği göz önünde tutularak yönlendirme algoritmalarının oluşturulmasında kullanılır. Bunun yanında şebekenin verdiği hizmete bağlı olarak farklı bazı koşullarda (ücret farklarından kaynaklanan belirli yolların öncelikle tercih edilmesi, iletilen verinin anlaşılır olmasını sağlayan işaret gürültü oranı gibi faktörler) göz önünde tutularak yol atama işlevlerinde dikkate alınabilir.



4. GRAF TEORİSİ VE YÖNLENDİRME ALGORİTMALARI

4.1. Graf Teorisi

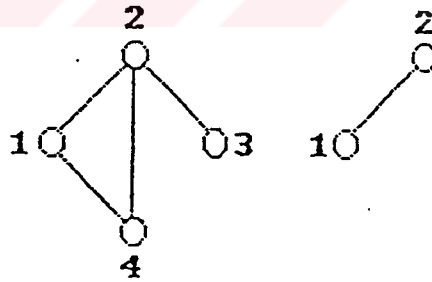
Graf teorisi, şebeke yapılarının sembolik gösterim üzerinden incelenmesine ve hesaplamalarının basitliği nedeniyle özelliklerinin belirlenmesine imkan verdiğiinden, sık kullanılan bir teodir. Bu yaklaşım iki şekilde incelenebilir.

4.1.1. Yönsüz graf

Sonlu sayıda düğümde meydana gelen D kümesi ve bu kümenin elemanları arasında seçilen birbirinden farklı çiftlerin oluşturduğu Ç kümesinin meydana getirdiği sistem graf (G) olarak tanımlanır. Şekil 4.1'de graf örnekleri verilmiştir. Graf matematiksel olarak,

$$G = (D, \mathcal{C}) \quad (4.1)$$

şeklinde gösterilir [3].



Şekil 4.1. Graf örnekleri

Grafın tanımlanmasının asıl amacı, düğümlerin yerlerinin şekilsel gösterilimine imkan vererek, düğüm çiftlerinin oluşturduğu dalların (kenar, bağ) belirlenmesini sağlamaktır. n_1 ve n_2 düğümler olmak üzere, (n_1, n_2) dalı n_1 ve n_2 düğümlerine aittir.

4.1.1.1. Adım (Walk)

Adım n_1 başlangıç ve n_l bitiş düğümü olmak üzere bu iki düğüm arasında tanımlanır.

Bu graf aşağıdaki şekilde tanımlanır.

- 1) n_1 düğümüne yalnız bir dal bağlıdır.
- 2) n_l düğümüne yalnız bir dal bağlıdır.
- 3) Adımı oluşturan diğer düğümlere en az iki dal bağlıdır.

Adımı oluşturan dalların birer düğümleri sırasıyla ortak olmak üzere $(n_1, n_2), (n_2, n_3), \dots, (n_{l-1}, n_l)$

$[(n_i, n_j) \in G; 1 \leq i \leq l, j=i+1]$ ya da

$(n_1, n_2, n_3, \dots, n_l)$ şeklindeki gösterilime adım (walk) denir.

4.1.1.2. İz (Path)

İz n_1 başlangıç ve n_l bitiş düğümleri olmak üzere iki düğüm arasında tanımlanır ve aşağıdaki şartları sağlar.

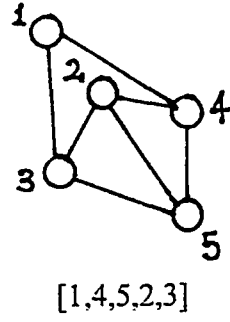
- 1) n_1 düğümüne yalnız bir dal bağlıdır.
- 2) n_l düğümüne yalnız bir dal bağlıdır.
- 3) izi oluşturan diğer düğümlere yalnız iki dal bağlıdır.

iz, tekrarlanmış düğüm içermeyen bir adım olarak tanımlanabilir. (Şekil 4.2)

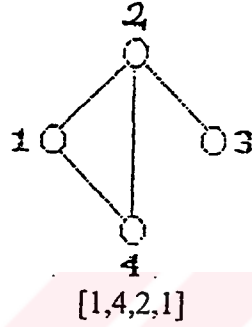
4.1.1.3. Çevrim (Loop)

Çevrim diye her bir düğümüne yalnız iki dalın bağlı olduğu grafa, başka bir deyişle kapalı ize denir.

Burada $n_1=n_l$ ve $l>3$ olmak üzere $(n_1, n_2, n_3, \dots, n_l)$ izine “çevrim” denir. (Şekil 4.3).



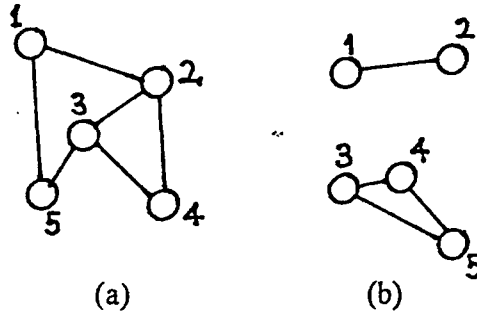
Şekil 4.2. İz gösterilimi



Şekil 4.3. Çevrim gösterilimi

4.1.1.4. Birleşik ve ayrık graflar

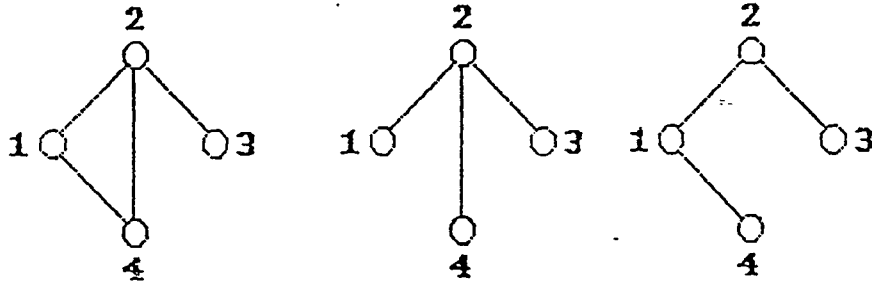
Bir grafta, dalları izleyerek bir düğümden diğer her bir düğüme geçilebiliyorsa o grafa “birleşik (connected) graf” denir. (Şekil 4.4. (a)). Bir graf birleşik değilse o grafa “ayrık (disconnected) graf” denir. (Şekil 4.4. (b)).



Şekil 4.4. (a) Birleşik graf örneği
(b) Ayrık graf örneği

4.1.1.5. Altgraf (Subgraph)

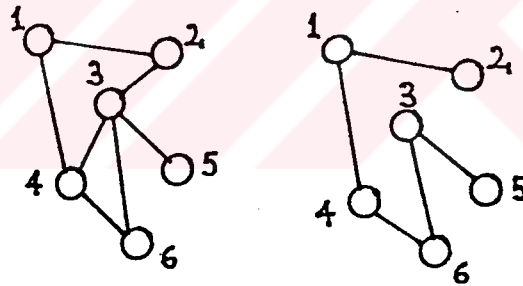
Bir G grafindaki bazı düğüm ve dalların oluşturduğu grafa “altgraf” denir. $D^1 \subset D$ ve $\mathcal{C}^1 \subset \mathcal{C}$ olmak üzere $G^1 \subset G$ iken G^1 grafi G ’nin altgrafıdır.



Şekil 4.5. Graf ve iki altgrafı

4.1.1.6. Ağaç (Tree)

Herhangi bir çevrim içermeyen birleşik grafa “ağaç” denir. G grafin tüm düğümlerini içeren ve bir ağaç olan grafa da “kapsayan ağaç” denir. (şekil 4.6).



Şekil 4.6. Graf ve kapsayan ağaç

Kapsayan ağacı oluşturmak için çeşitli yöntemler vardır. Basit bir algoritma aşağıdaki şekilde tanımlanır.

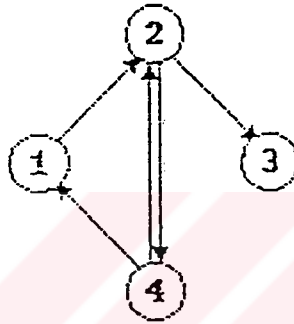
- 1- n , D 'de herhangi bir düğüm olmak üzere $D^1 = [n]$, $\mathcal{C}^1 = []$ olsun.
- 2- $(i, j) \in \mathcal{C}$, $i \in D^1$, $j \in D - D^1$ olmak üzere D^1 ve $\mathcal{C}^1$ 'yi $D^1 := D^1 \cup [j]$ ve $\mathcal{C}^1 := \mathcal{C}^1 \cup [(i, j)]$ şeklinde güncelle ve

3. adıma geç.

3- $D^1=D$ ise $G^1 = (D^1, \mathcal{C}^1)$ kapsayan ağaçtır yoksa 2.adıma git.

4.1.2. Yönlü graf

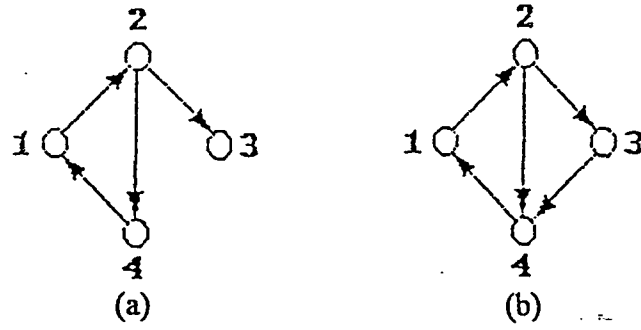
Sonlu elemanlı ve boş olmayan düğüm kümesi D ve D kümesinden farklı düğümlerin oluşturduğu sıralı çiftlerin kümesi $\mathcal{C}$ ile meydana getirilen $G = (D; \mathcal{C})$ grafına “yönlü (directed) graf” denir. Şekil olarak gösterilimde yönsüz grafın dalları ok işareti ile yönlendirilerek gösterilir (Şekil 4.7).



Şekil 4.7. Yönlü (Directed) Graf

Herhangi bir $G = (D, \mathcal{C})$ için, $D^1=D$ ve $(i,j) \in \mathcal{C}$, ya da $(j, i) \in \mathcal{C}$ veya her ikisi iken $(i,j) \in \mathcal{C}^1$ ise, $G^1=(D^1, \mathcal{C}^1)$ şeklinde bir yönsüz graf mevcuttur. $(n_1, n_2, n_3, \dots, n_l)$ yönsüz grafta adım, iz veya çevrim ise, yönlü grafta da adım, iz veya çevrimdir. (n_i, n_{i+1}) , $1 < i < l-1$ için G 'de yönlü dal ise $(n_1, n_2, n_3, \dots, n_l)$ yönlü adımdır. Yönlü bir iz (directed path), tekrarlanmış düğüm içermeyen yönlü adımdır. $l > 2$ ve $n_1=n_l$ olan ve başka tekrarlanmış düğüm içermeyen yönlü adım $(n_1, n_2, n_3, \dots, n_l)$ 'a yönlü çevrim adı verilir.

Yönsüz bir graf birleşikse, yönlü graf da birleşiktir. Herhangi iki düğümü arasında yönlü bir iz bulunan yönlü grafa “tam birleşik (strongly connected) graf” denir. (Şekil 4.8).



Şekil 4.8. (a) Birleşik graf
(b) Tam birleşik graf

4.2. Yönlendirme Algoritmaları

Yönlendirme metodları, graf teorisinin özelliklerini kullanarak devre ve paket bağlaşmalı şebekeler için kullanılan algoritmaları içermektedir. Herhangi iki düğüm arasında mevcut izlerden en uygun olanı, en kısa iz (shortest path) metoduyla belirlenir. En kısa iz özelliği her bir dala şebekenin hatlarını temsilen belirli bir karakteristik değer vererek (örneğin tıkanma olasılığı) mevcut izlerden dal değeri toplamı en küçük olanı seçilir ve en iyi iletim şekli yönlendirme şeması vasıtasıyla çözmeye çalışılır.

4.2.1. Minimum ağırlıklı kapsayan ağaç

G birleşik grafında D düğümleri, $\mathcal{C}$ dalları ve w_{ij} her $(i, j) \in \mathcal{C}$ dalının ağırlığını gösterebilir. Minimum ağırlıklı kapsayan ağaç (minimum weight spanning tree) dal ağırlıklarının toplamı minimum olacak şekildeki ağaçtır. Dal ağırlıkları haberleşme maliyeti, gecikme, tıkanma gibi fiziksel büyüklükleri temsil ettiğinde MST ağırlığı da iletinin kapsayan ağaç boyunca tüm düğümlere iletilmesinin maliyeti olur.

4.2.1.1. Fragment

MST'nin herhangi bir alt ağacı olan bir graf fragment olarak isimlendirilir. Herhangi bir dalın en az bir düğümünün fragmente ait olmaması durumunda çıkan dal olarak tanımlanır.

4.2.1.2. MST oluşturma algoritmaları

MST'nin meydana getirilmesi için çeşitli algoritmalar mevcuttur. Bunlar şu şekilde sıralanabilir:

1. Prim-Dijkstra Algoritması
2. Kruskal Algoritması
3. Dağıtılmış Algoritma

Prim-Dijkstra algoritmasında seçilen herhangi bir düğüm, fragment olarak düşünülür. Minimum ağırlıklı çıkan dallar peşisıra fragmente eklenerek genişler.

Kruskal'in algoritmasında ise, her düğüm bağımsız birer fragment olarak ele alınır. Herhangi iki fragmenti çevrim içermeyecek şekilde birleştirerek genişleyen fragmentler kümesi bir fragment olana kadar algoritma devam ettirilir. İlk iki algoritma D düğüm sayısının bir eksiği sayıda iterasyonla sonuçlanır.

Dağıtılmış algoritmada ise tek bir düğüme kadar inebilen fragment kümeleri her iterasyonda eş zamanlı olarak minimum ağırlıklı çıkan dallar eklenip tek bir graf elde edinceye kadar devam eder.

Dal ağırlıklarının aynı olması durumunda dalların ulaştığı düğümlerden minimum çıkan dala sahip olan düğüme bağlı dal seçilerek çevrim oluşması önlenir.

4.2.2. En kısa iz algoritmaları

G;D düğümlerinden ve Ç dallarından oluşan bir graf olsun. Grafın her bir dalının ağırlığını gösteren sembol, (i, j) dal olmak üzere w_{ij} olsun. w_{ij} reel sayı olmak üzere $(i, j, k, \dots, l, m)$ yönlü izinin uzunluğu $(w_{ij} + w_{jk} + \dots + w_{lm})$ olarak hesaplanır. Yönlü adım ve çevrim uzunlukları da aynı şekilde hesaplanır. Bu durumda iki düğüm arasında minimum ağırlıklı iz bulunması, bu düğümler arasındaki en kısa izi belirtir.

Dal ağırlıkları şebekede karşı düşen hatların maliyeti, santrallerarası hatlardaki gecikme vb. fiziksel büyüklüklere karşı düşebilir. i düğümünden j düğümüne mesaj iletiminin gerçekleşeceği minimum maliyetli yol bu iki düğüm arasındaki en kısa izin ağırlığına eşittir. Dal ağırlıklarının ileti gecikmelerini temsil ettiği durumda bu aynı zamanda en az gecikme olan yol demektir. Ancak, en az gecikmeli yolun tercih edildiği yönlendirme algoritmaları, taşınan trafik yükünün düşmesi problemini çözecek biçimde geliştirilmelidir. Bu ise algoritmanın yapısını daha karmaşık bir hale getirir. Eğer dal ağırlıkları o dalın temsil ettiği linkin şebekede kullanım olasılığı olan p_{ij} 'yi temsil ediyorsa bu düğümler arasındaki en kısa uzunluklu iz (dal ağırlıkları - $\ln p_{ij}$ olmak üzere) aynı zamanda en güvenilir yani mesajın yerine ulaşma garantisi en belirli yolu temsil eder. Bunlar yanında dal ağırlıkları iletim kalitesi gibi donanımsal özellikleri temsil eden büyüklüklerin negatif logaritmik değerleri olarak tanımlanırsa en kısa iz aynı zamanda işaret-gürültü oranı en yüksek ya da bağlaşma gecikmesi en az yolu belirtir.

Yukarıda uygulama alanları belirtilen en kısa iz problemini çözüme kavuşturmak için standart algoritmalar yanında tezde ele alınan yeni bir algoritma aşağıda verilmiştir.

4.2.2.1. Dijkstra algoritması

Bu algoritmanın esası belirlenen bir düğümden (1) itibaren bu düğüme bağlı en kısa uzunluklu dalın eklenerek oluşan yeni grafın katılan düğümüne bağlı dalların en kısa uzunluklu olanı ile önceden ihmal edilen dallardan yeni katılan düğüme bağlı dal

varsa bununla karşılaştırılarak en kısa iz belirlenir ve bu iterasyon düğümler bitene kadar devam eder. Karşılaştırma sonucu yeni katılan i düğümü 1 düğümünden en kısa izin kestirimi ile etiketlenerek kesinleştirilir. W_i : Algoritmanın gelişimi aşağıdaki şekilde ifade edilir (Şekil 4.9).

Başlangıçta düğüm kümesi $P = [1]$, $W_j = w_{ij}, j \neq 1$ için ve $W_1 = 0$

1) $i \notin P$ ise en yakın düğümü bul.

$W_i = \min W_j$

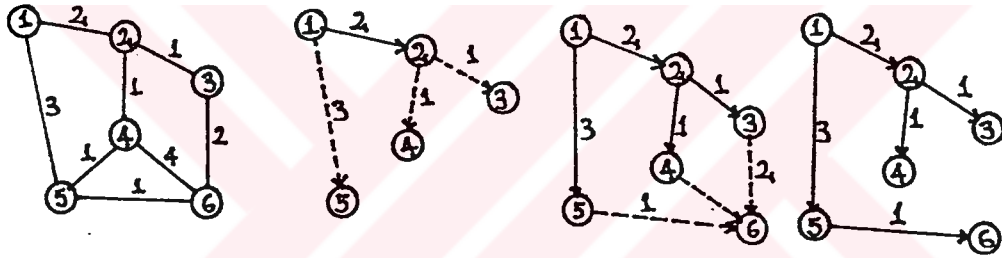
$j \notin P$

$P = P \cup [i]$, P tüm düğümleri içermiyorsa 2.adıma geç aksi halde algoritmayı bitir.

2) Tüm $j \notin P$ için etiketleri kesinleştir.

$W_j := \min [W_j, W_i + w_{ij}]$

1. adıma dön.



Şekil 4.9. Dijkstra en kısa iz algoritması

4.2.2.2. Bellman-Ford algoritması

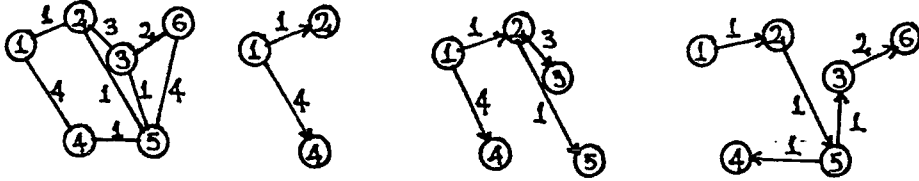
Algoritmanın esası, en kısa izin sırasıyla bir dal, iki dal ve devamı sürecinde içerdiği izlerin en fazla h dal içermesi durumunda en kısa uzunluklu iz seçiminin bulunmasıdır. Grafta (i, j) dalı mevcut değilse $w_{ij} = \infty$ alınır. Kaynak düğümünün 1 olduğunu kabul edersek $W_1^{(h)}$, 1'den i düğümüne ($\leq h$) için en kısa iz uzunluğu olsun. Böylece tüm h 'lar için $W_1^{(h)} = 0$ olur. Algoritmanın gelişimi aşağıdaki gibidir (Şekil 4.10).

1) Tüm $i \neq 1$ 'ler için, $W_i^{(0)} = \infty$

2) Ardışıl $h \geq 0$ ve tüm $i \neq 1$ 'ler için,

$W_i^{(h+1)} = \min_j [W_j^{(h)} + w_{ji}]$

Bu algoritma en uzun N-1 iterasyonda biter ve en kısa iz N-1 dal içerir. Her halukarda Dijkstra algoritmasından daha karmaşık bir yapıya sahiptir.



Şekil 4.10. Bellman-Ford en kısa iz algoritması

4.2.2.3. Floyd-Warshall algoritması

Bu algoritmanın diğerlerinden üstünlüğü sadece belli bir düğüm için değil tüm düğüm çiftleri için en kısa izlerin eş zamanlı bulunabilmesidir. Algoritma öncelikle bir düğüm için en kısa izi hesaplar sonra önceki bulduyuyla yeni eklenen düğümün oluşturduğu grafları karşılaştırarak bir fazla düğüm içeren graf yapısı için en kısa izi bular. N-1 iterasyonda sonuca ulaşır. Algoritmanın gelişimi şu şekildedir.

$W_{ij}^{(n)}$: i'den j'ye 1,2,.....,n ara düğümler olmak üzere en kısa iz uzunluğu olsun.

1) $i \neq j$ olmak üzere tüm i,j 'ler için

$$W_{ij}^{(0)} = w_{ij}$$

2) tüm $i \neq j$ 'ler için, $n=0,1,2,.....,N-1$ olmak üzere,

$$W_{ij}^{(n+1)} = \min [W_{ij}^{(n)}, W_{i(n+1)}^{(n)} + W_{(n+1)j}^{(n)}]$$

4.2.2.4. Genetik algoritma

Genetik algoritma yukarıda bahsedilen algoritmalarla oranla belirli bir kaynak düğümünden itibaren en kısa izin hesaplanmasında daha az güvenilir olduğu halde, ileriki bölümlerde etraflıca inceleneceği gibi bu metodun kurgu ve hesaplama kolaylıkları, yazılım haline getirmedeki esneklik ve sonuçları elde etmede sarf ettiği zaman göz önüne alındığında oldukça avantajlı görülmektedir. Bu metod bağımsız bir şekilde inceleneceğinden burada kısaca bahsedilmekle yetinilmiştir.

4.2.3. Yönlendirme algoritmalarının devre bağlaşmalı şebekelere uygulanması

Devre bağlaşmalı trafiğin söz konusu olduğu klasik telefon şebekeleri otomatik yönlendirme türünde adım-adım karar mekanizmalarına sahip santrallar üzerinden işlem yaparlar. Burada çağırın tarafından girilen dijital abonenin bağlı olduğu santraldan itibaren belirli gruplar halinde analiz edilerek hedef uca doğru yönlendirilir. Önceleri doğrudan yollar üzerinden yönlendirme yapıldığından alternatif imkanların, bu doğrudan yollar üzerindeki trafik yükünü arttırmış ve verimliliği düşürmüştür. Sonraları, doğrudan yollarda boş hat olmadığı hallerde seçenekli yönlendirme (alternate routing) yapılmıştır ve sorun nisbeten çözümlenmiştir. Bu durumda taşın trafik, tandem santrallar üzerinden taşınmaktadır.

Çevrim oluşması gibi sorunlar yüzünden şebeke yapıları hiyerarşik bir düzene göre tasarlanmaktadır. Bu şekilde kaynak santraldan çıkan çağırma hedef santralin bulunduğu seviyeye yönelecek şekilde hiyerarşik yönlendirmeye tabi tutulur. Bu çeşit yönlendirmede ara santrallar her hedef için çevrimi önleyecek belirli seçenek gruplarına sahiptir. Bu şekildeki yönlendirme gereksiz iletim sorununu hallederek verimliliği artırır.

Dinamik yönlendirme ise, yukarıda söz konusu edilen sabit seçenekli yönlendirmeye göre değişen trafik yüküne daha iyi adapte olabilen bir yöntemdir. Sabit seçenekli yönlendirmede geri dönüşlülük (crank back), belirli hedeflere belirli trafiğin tahsis edilmesi gibi önlemlerle hizmet kalitesi artırılabilir. SPC santralları ve ISDN'deki ayrı işaretleşme kanallarının dinamik yönlendirme algoritmaları için uygun altyapıyı hazırlamıştır.

Dinamik yönlendirmenin şebekedeki trafik yükü değişikliklerine ve topoloji düzenlemelerine eş zamanlı cevap verebilmek için santralların yönlendirme algoritmalarının programlanabilir ve uyarlanabilir olması, bir ya da daha fazla yönlendirme hesaplama merkezi olması (RCC gibi) ayrıca işaretleşme kanalları yönlendirme bilgilerini santrallara taşıyacak kadar geniş kapasiteli olması gerekmektedir.

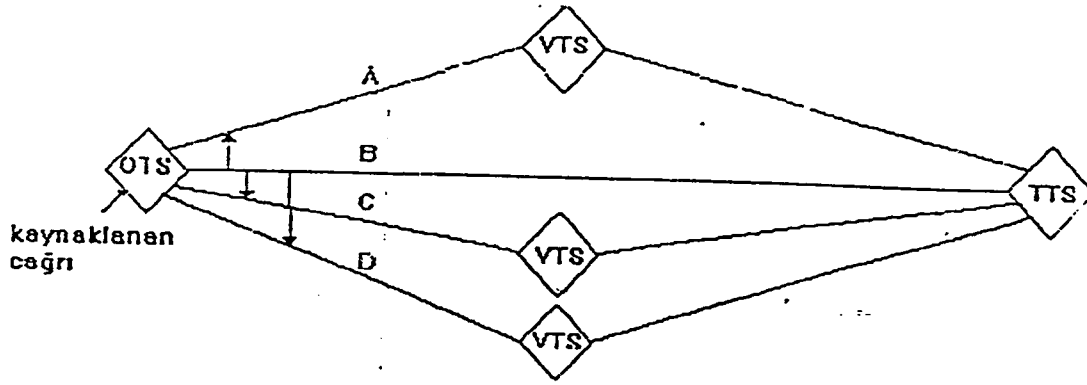
Devre bağlaşmalı yönlendirme algoritmaları iki şekilde ele alınabilir.

4.2.3.1. Dinamik, hiyerarşik olmayan yönlendirme

Hiyerarşik yönlendirmede her düğüm, hedef düğüme göre belirli çıkan devre gruplarını seçen seviye yükselmesi esasına dayanan bir algoritma uygulanır. DNHR ise zamana bağlı dinamik bir yönlendirme şemasıdır. Geniş coğrafi alanlar üzerine kurulu olan şebekelerde belirli bölgelerin meşgul saat farklılıklarına dayanan ve bu farklılığı trafik yükünü yayarak verimliliğe dönüştürmeye çalışan bir yöntemdir. Bu sayede mevcut hiyerarşik şebekelerdekiyle aynı hizmet kalitesi şartı altında daha az çıkan devre grubu ve imkan kullandığından verimlilik önemli miktarda artar. DNHR şebekesinde 3 çeşit santral bulunur. Kaynak santrali (originating tool switch-OTS) çağırmanın tıkanana ya da hedefine varana kadar kontrolü sağlayan santraldır. Hedef santral (terminating tool switch-TTS) çağırmanın ulaşacağı santraldır. İz üzerindeki geçiş santrali (VIA tool switch-VTS) kaynak ile hedef arasındaki santraldır. İz üzerindeki santraller arasındaki işaretleşme için CCS kullanılır. İşaretleşme kanalı bunun yanında çağırmanın hedefe varamadığı durumlarda kontrolün tekrar OTS'ye geçmesi için VTS'den işaretleşme bilgisinin (geri dönüşlülük) iletilmesini sağlar. Böylece şebekenin hizmet kalitesi artırılmış olur (Şekil 4.11) [3].

Bu yöntemde gün 10 zaman dilimine ayrılmıştır. Böylece her periyoda bir yönlendirme şeması karşılık düşürülmüştür. Her santraldan, merkezdeki veri toplama ve işleme sistemi tarafından gözden geçirilmek üzere işaretleşme kanalından trafik bilgileri gönderilir. Merkezi düğüm tarafından bu bilgiler ışığında en iyi yönlendirme şeması belirlenir. Bunun yanında DNHR santralleri belli bir trafik tipi için otomatik kapasite ayırma özelliğine sahiptir. Böylece bu çağırımlar tıkanma olasılığına karşı korunur ama şebekenin çağırma şemalarına ve hatalarına gerçek zamanda uyum sağlaması zorlaşır.

Şekil 4.11'deki DNHR şebekesinde yönlendirme örüntüsü kaynak ile hedef arasındaki dört olası yolu seçme sırasını zaman dilimlerine göre ayarlar. Mesela, öğleden önce BDCA sırasında iken öğleden sonra BCDA sırası daha uygun olabilir.



Şekil 4.11. Dinamik, hiyerarşik olmayan yönlendirme

Mevcut bütün yolların tıkalı olması durumunda CCS vasıtasıyla VTS, OTS'ye geri dönüşlülük mekanizmasıyla çağırmaı santrallararası başka boş hatlara yönlendirmesini sağlayabilir (gerçek zaman yolları). DNHR'nin gerçek zaman düzenlemesi için hatların durum haritası (TSM) kullanır. Böylece eş zamanlı ölçümler ile çağırmanın boş yollardan en iyi seçeneği kullanılması sağlanır.

4.2.3.2. Dinamik olarak kontrol edilen yönlendirme

Merkezi uyarlamalı yönlendirme algoritmasının kullanıldığı Northern Telecom tarafından geliştirilen dinamik olarak kontrol edilen yönlendirme (Dynamically Controlled Routing-DCR-) sistemi geniş alan şebekeleri için tasarlanmıştır. Santrallardan merkezi işlemciye her 10 sn'de bir gelen santrallararası hatların durumlarını belirten bilgiler ışığında, veri bağlantı kanalları vasıtasıyla santrallara belirli hedefler için ara santral bağlantılarını içeren yönlendirme tabloları gönderilir. Santrallararası hat durumları, ayırık zamanda alınan ölçümler vasıtasıyla belirlenen periyotta hatlardaki meşguliyet sayısı kaydedilir. Böylece elde edilen zaman serisi yardımıyla bir önceki meşguliyet sayısı ya da taşınan trafik bir sonraki periyottaki meşguliyet ya da taşınan trafik tahmin edilebilir [7]. Eğer trafik değişimi çok hızlı ise 10 sn. gibi kısa periyotlar boyunca değişimin sınırlı olmasından yararlanılarak bir sonraki periyot için örnek alınabilir. Bunun yanında trafik akışındaki değişim zamana göre çok keskinse bir sonraki periyot için çeşitli hesaplama modelleri önerilmiştir. Bob ve Jenkins şebeke yönetimi için gerekli olan, örneklenmiş sürecin gelecek

değerleri için bir sonraki peryottaki meşguliyet sayısı (x_{n+1}) ya da taşınan trafik (y_{n+1}) için t_n anında tahminler yapmayı sağlayan iyi bilinen bir metodoloji belirtmişlerdir. (4.2) (4.3) [1].

$$x_{n+1}=G_1(x_j, y_j, u_j); j=0,1,\dots,n \quad (4.2)$$

$$y_{n+1}=G_1(x_j, y_j, u_j); j=0,1,\dots,n \quad (4.3)$$

u_j : t_{j-1} - t_j anında gelen çağırımlar

Bu şekilde modellenen meşguliyet durumları için bir sonraki peryotta esas alınarak yönlendirme algoritmaları dinamik olarak uyarlanabilir. Santrallar arasındaki bağlantı sayılarının kısıtlı olduğu durumlarda DCR verimli bir yöntemdir. Bununla birlikte kaynak ile hedef arasında bir ara santral içeren iki bağ olduğundan, tıkanma olasılıklarının büyük olması yüzünden yedekte sabit hiyerarşik yönlendirme bulundurulur. Zira hiyerarşik bir alt-şebeke DCR şebekesine bağlı olabilir. DCR yöntemi trafikteki kısa süreli değişimlere ve donanım arızalarına cevap verme durumuna daha duyarlı olduğundan sabit seçenekli yönlendirmeye göre daha kullanışlıdır.

Bir DCR şebekesinde N, I, X ve Z diye tabir edilen dört çeşit santral sınıflandırması vardır. N, I, X santralları DCR'nın elemanları olarak çalışmasına rağmen Z santrali dış santral olarak belirlenir. N tipi santrallar yönlendirme işlemcisi tarafından diğer I/N santralları için ara santral konumunda değerlendirilir. I tipi santrallarda yönlendirme işlemcisi tarafından kaydedilmiş programlar vasıtasıyla çalıştırılmasına rağmen ara santral olarak önerilmezler. I/N santrallar, X.25 haberleşme protokolu uyarınca her 10 sn'de bir çıkan devre gruplarının trafik durumunu yönlendirme işlemcisine göndererek şebeke yönlendirme stratejisinin güncelleşmesini sağlar.

DCR şebekelerinde kaynak ile hedef santral arasında kurulan link hatları, doğrudan birbirine bağladığı ara santralların haberleşmelerine imkan tanımak için bu tipte trafiğe belirli bir kapasite ayırır. Koruma parametresi olarak tanımlanan değer bu yönde tahsis edilmiş santrallararası hat sayısının değerini verir. Bu parametre doğrudan trafiğin durumuna bağlı olarak birkaç sn.de bir yönlendirme işlemcisi

tarafından genel trafik dağılımını da göz önüne alınarak güncelleştirilir. Şebekenin aşırı yükte çalışması durumunda diğer doğrudan trafik akışlarını tıkamamak için kaynak-hedef arasında geçiş santralleri kullanılmadan doğrudan bağlantı sağlanır.

Herhangi bir ara santral için kullanılabilir santrallerarası hat sayısı, boş imkan sayısından koruma faktörü çıkartılarak bulunan değerdir. Yönlendirme işlemcisi bu ara santral üzerinden kaynak ile hedef santraller arasındaki iz çeşidini birbiriyle çakışmayacak şekilde seçer. Bundan dolayı kaynaktan bu ara santrale olan bağlar ile ara santraldan hedefe doğru bulunan kullanılabilir hat sayısı arasından minimum sayıda olanı kadar iz belirlenir. Eğer ara santral aşırı yük altında çalışıyorsa, başka düğüm çiftleri için geçiş düğümü olamayacağını merkezi işlemciye bildirir.

DCR şebekesinde kaynak ile hedef santraller arasında doğrudan bağlantı bulunmadığı durumlarda bağlaşma sırasında ara santrallerdeki koruma faktörü göz önüne alınmaz. Bu çeşit trafiğe “engelli (handicapped) trafik” denir. Engelli trafik ara santraller üzerinden daha fazla seçeneğe sahip olduğundan doğrudan bağlaşmaya göre daha üstündür.

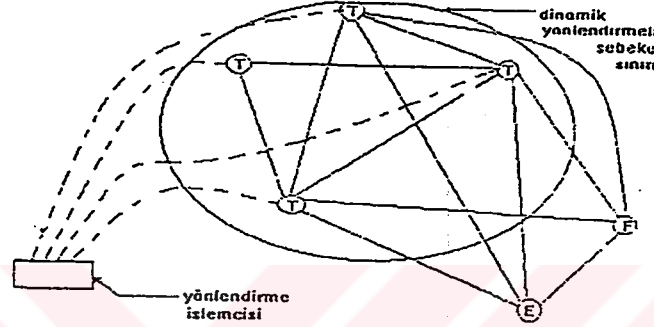
Bu yönlendirme sistemi ikiden fazla seri bağ içeren izler için hesaplama zorluğuna sahip olduğundan tercih edilmez. Buna karşın kısa süreli güncelleştirmeler nedeniyle en az yüklü ara santralin seçimi her ara santralde en uygun santrallerarası hatta karar verilmesi gibi algılanabileceğinden merkezi olmayan yönlendirmeye benzer.

Günümüzde şehirlerarası trafik için kullanılan yönlendirme metodu doğrudan, iki-hat seçenek yol ya da yedek taşıma yolu üzerinden kullanılır. Böylece trafiğin büyük kısmı doğrudan ve iki-bağ seçenekli yollardan taşınırken çok az bir kısmı gerektiğinde yedek yolu kullanır (%2 civarında).

Telefon haberleşmesinde son 20 yıldaki gelişmeleri göz önüne alındığında şehirlerarası telefon şebekelerinde kaynak ile hedef arasındaki doğrudan hat gruplarını kullanma sıklığının arttığı, bunun yanında doğrudan hat bulunmaması halinde sırasıyla uygulanan iki-bağ bağlaşma ya da hiyerarşik yönlendirme

sıklığının yıllara göre düştüğü görülür. Burada amaç doğrudan bağlantıların tıkanması durumunda dinamik yönlendirme metodu sayesinde mümkün olan en fazla imkanın kullanılmasıdır.

DCR şebekesine girmeyen sabit hiyerarşik şebekeye ait bir santral DCR şebekesiyle bağlantı kurduğunda ilk bağlandığı DCR şebeke santrali üzerinden dinamik şekilde çağırma yönlendirilir (Şekil 4.12).



Şekil 4.12. DCR şebekesine bağlanan sabit hiyerarşik yönlendirmeli santral uygulaması

Çağrılar, sabit hiyerarşik şebekeye ait santraldan DCR şebekesinin hedef santralına yönlendirilmişse, kaynak santraldan itibaren izlenen yoldaki ilk DCR santrali çağırma kendi şebeke yönlendirme esasına göre iletir. Ters durumda hedef DCR şebekesinin dışındaysa DCR şebekesi hiyerarşik şebeke kurallarını uygulayarak çağırma alt seviyedeki santrale iletir.

DCR şebeke yönlendirmesi ile DNHR arasında verimlilik tabanında bazı farklar vardır. Öncelikle DNHR uzun zaman periyotları için önceden belirlenmiş yönlendirme algoritmaları kullandığı için kısa süreli trafik akış değişimleri karşısında DCR yöntemindeki gibi esnek değildir. Buna karşın DCR yönteminde yönlendirme işlemcisinin konumu çok önemli olduğundan burada meydana gelen bir hata ya da aşırı yük dolayısıyla oluşan tıkanma bütün bir şebekenin verimini düşürebilir. DNHR'de yanlış yönlendirilmiş (misrouted) ya da hedefine ulaşamamış bir çağırma geri dönüşlülük özelliği sayesinde tekrar yönlendirme sürecine sokulabilir. Buna karşın, DCR'de uygun yol mevcuttan tıkalı ya da hatalı bir yola yönlendirilen

çağırmanın tekrar yönlendirilme ihtimali yoktur, DNHR'de ise VTS'ye verilen çağırma kontrol yetkisi ile bu işlem gerçekleştirilir. DCR'de ise bu sorun daha sık güncelleştirme ile tıkalı ya da hatalı yolları anında ayıklamaya çalışarak çağırma kaybolmasını önlemeye çalışarak çözülür. Ayrıca DCR'de, geri dönen çağırımların boş yere şebeke trafiğini meşgul ederek verimliliği düşürme durumu söz konusu değildir.

4.2.3.3. Bilgi iletme

Tıkanma bilgilerinin, işaretleşme kanalı aracılığıyla, komşu santrallarda birbirlerine gönderilmesi sonucu kaynaktan çıkan çağırma ara santrallarda hedefe doğru doğrudan bağlantılı yol öncelikli olmak üzere en az tıkanan trunklar üzerinden sırasıyla sınanarak belirlenen yola verilir. Bilgi iletmede (Information passing) tüm şebeke durumu anında göz önüne alınmaz; periyodik olarak kısmi bölgeler üzerinden en iyi yol hesaplaması yapılmasına rağmen şebekede çağırmanın tekrar yönlendirilmek üzere geri gönderilmesi hizmet verimini attırır.

4.2.3.4. Trafik veritabanı

Çağırmanın kaynaktan kontrol edilmesine imkan vererek şebekenin trafik durumunu ayarlamasına olarak tanıyan bu algoritma, ortak kanal ile belirli bölgeler arasında bilgi iletiminin kullanıldığı bir metottur.

Bu yöntemde, dağıtılmış bölgesel veritabanları CCS vasıtasıyla periyodik olarak ya da gerektiğinde şebekenin trafik yükü açısından genel hat durumunu ve kendi bölgesinin abonelerinin meşguliyet durumlarını diğer veritabanlarıyla değiş tokuş eder. Bölgesel veri tabanı ilk olarak hedef ucun boş olup olmadığını kontrol eder. Boşsa çağırmayı en iyi yoldan iletme için seçenekleri kontrol eder, boş değilse çağırma tıkanır. Bu metodun önemli dezavantajı sürekli ve fazla miktarda veri alışverişi ve veri işlemedir.

4.2.3.5. Taşma algoritması

Taşma (Flooding) yönteminde, kaynak belirli bir hedef için olası tüm çıkış hatlarına araştır mesajını CCS yoluyla gönderir ve geri gelen araştır dönüşü mesajındaki en az bağ ve en az olasılıklı öncelikli çağırma içeren yol olma gibi özelliklere göre en iyi yolu seçer. Merkezi olmayan bu yönlendirme trafik yükünü çok arttırdığından kamu şebekelerinde tercih edilmemesine rağmen belirli çağırımlar için askeri şebekelerde kullanılabilir. Esas amacı düğümler arasındaki doğrudan bağlantıları kullanmak ve şebekede hata olması durumunda yeni öncelikli yolları belirlemektir.

4.2.3.6. Diğer algoritmalar

Devre bağlaşmalı yönlendirmede yukarıda belirtilenlere ek olarak ideal trafik durumları göz önünde bulundurularak oluşturulan Markov yönlendirme, minimum yolları içerecek şekilde değişen şebeke durum şartlarına uyan ağaçları kullanarak uyarlamalı bir yönlendirme tanımlayan kapsayan ağaçla adaptif yönlendirme ve mevcut yönlendirme algoritmalarının ara santralleri kullanarak yaptığı işlemde serbest kalan çağırımlar için ara santraller üzerinde doğrudan bağlantılar arayarak trafik yükünü azaltmaya yarayan yeniden düzenleme (Rearrangement) sayılabilir.

4.2.4. Yönlendirme algoritmalarının paket bağlaşmalı şebekelere uygulanması

Paket bağlaşmalı şebekelerde yönlendirme devre bağlaşmalı şebekelerde olduğu gibi bilgileri birbirine aktararak şebekenin trafik durumu hakkında genel bir algoritma oluşturmayı amaçlayan bağımsız karar verebilen düğümler ya da merkezi yönlendirmeli şebekeler söz konusudur. Paket bağlaşmalı şebekeler katmanlı haberleşme mimarisinin bir çeşidi olan 1977'de ISO tarafından OSI (Açık Sistem Bağlantısı Referans Modeli) modeli önerilmiştir. Yedi katmandan oluşan bu yapıda üçüncü katman olan şebeke katmanı mesajın, şebeke veya şebekeler üzerinden amaçlanan uca yönlendirilmesinden sorumludur. Datagram modunda çalışan şebekelerde bu katman her düğümden karar verilmesini sağlarken, sanal devre modunda iletimin başlangıcında yolun belirlenmesini ve bunun diğer düğümler

üzerinden izlenmesini sağlar. Şebeke katmanı ayrıca yığılma kontrolü, farklı şebekeler arası bağlantı işlemlerini de tanımlar.

Paket bağlaşımlı şebekelerde yönlendirme uç düğümler arasında veri katmanı ve taşıma katmanı uyumu yanında tüm aracı düğümlerle uyum gerektirir. Şebekede meydana gelen bir hatadan kaynaklanan çağırmanın hedefe varamaması olasılığı yeniden yönlendirme özelliği ile önlenmeli ve veri tabanı sürekli yenilenerek düğümlerdeki trafik yüküne uyumlu yönlendirme algoritmaları üretilmelidir.

Bu şebekelerde yönlendirme şeması iki temel performansa göre değerlendirilir: Bunlar, giren trafiği bloke etme durumunu gösteren iş çıkarma yeteneği (throughput), diğeri ise düğümlerdeki kuyruklama sisteminde bekleyen paketlerin ortalama paket gecikmesidir. İş çıkarma yeteneği ile ortalama paket gecikmesi doğru orantılı olduğundan, ilkinin artırırken ikincisini azaltmak zıt amaçlar olduğundan hizmet kalitesinin istediği ölçüde iyi bir denge noktasında tutulmaları gerekmektedir. (4.4) ve (4.5)'teki eşitlikler göz önüne alınırsa sistemin taşıyabileceği maksimum yükte yaklaşıldıkça ortalama paket gecikmesi sonsuza gider ve kuyruğun dolmasına sebep olur. Burada C, hizmet organının iletim kapasitesi, ρ ise hizmet organının trafik yükünü gösterir.

Ortalama paket gecikmesi: T

$$\text{Little formülü } T = \frac{E[n]}{\lambda} \Rightarrow T = \frac{1}{\mu C - \lambda} \quad (4.4)$$

$$\rho = \frac{\lambda}{\mu C} \quad (4.5)$$

Böylece, şebekede reddedilen trafiği ayarlayan akış kontrolünün belirlediği verim, ortalama kuyruk gecikmesini ayarlayan yönlendirme algoritmasının oluşturduğu geri beslemeyle dengeye çekilmeye çalışılır. Şebekedeki trafik düşük olduğunda akış kontrolü bütün trafiği sistemden geçirerek verimi artırır, eğer trafik artarsa akış kontrolü bir kısım trafiği engelleyerek gecikmeyi düşürmeye çalışarak yönlendirme

algoritmasının belirli sınırlar içinde çalışmasına imkan verir. Yönlendirme algoritması ise daha az gecikme sağlayan yolları belirleyerek akış kontrolünün şebekeye daha fazla trafik yükü sokmasını sağlamasına imkan verir. Bu şekilde etkileşimli çalışmayla şebekeden maksimum verim alınmaya çalışılır.

4.2.4.1. En kısa iz yönlendirme algoritması

Paket bağlaşmalı şebekede her santrale bir düğüm, her hatta da bir dal karşılık düşürülerek oluşturulan graf üzerinden belirli dal etiketleme yöntemine göre oluşturulmuş dal uzunlukları üzerinden hesaplanan en kısa iz yönlendirme algoritmasının bahsedilen düğüm çifti arasından seçtiği yoldur.

Dal etiketleri, düğümler arası uzaklık, iletilecek mesajı taşıyan kanalın band genişliği, iletim maliyeti, ortalama kuyruk uzunluğu, ortalama paket gecikmesi faktörlerinin kombinasyonundan oluşur. Ayrıca gönderilecek test paketleri ortalama kuyruk uzunluğunu elde etmede kullanılır. Bu şekilde 4.2.2.'de Bahsedilen algoritmalar yardımıyla en kısa uzunluklu iz bulunabilir [3].

Tezde çalışma konusu edilen trafik yönlendirme simülasyonu da kısmen en kısa iz algoritmasının mantığına dayanmaktadır.

4.2.4.2. Çok izli yönlendirme

Paket bağlaşmalı şebekelerin bir çoğunda herhangi iki farklı düğüm arasında birden fazla yolun mevcut olduğu yönlendirme algoritmaları kullanılır. Belirli bir düğüm çifti arasında sadece en iyi izi kullanan en kısa iz algoritmasının aksine çok izli yönlendirme (Multipath Routing) trafiği birden fazla yola yayarak her iletişim hattındaki yükü azaltabilir ve şebekeyi ekonomik kullanabilir. Çok izli yönlendirme hem datagram modunda hem de virtüel devre modunda kullanılabilir.

Bu yönlendirme algoritmasında şebeke operatörlerince manuel olarak hazırlanan sabit yönlendirme tabloları gereğince her düğüm olası hedefler için 3 veya daha fazla

seenek ieren satırlardan oluřur. Santral iinde oluřturulan rastgele bir sayının bu seeneklerden hangisinin ierdiėi aralıėa dūřuyorsa bir sonraki dūėume o hat yada yol uzerinden iletim yapılır. Bu durumda her satırdaki aralık uzunluklarının toplamı, olasılık teorisi gereėince 1 olacaktır.

ok izli ynlendirme, trafiėi farklı izlere ynlendirdiėinden farklı trafik tipleri iin farklı seenekler sunar ve bylece paket uzunluėuna gre uyumlu seilen yollar vasıtasıyla gecikme nisbeten dūřurlmř olur. rneėin, kısa paketler yer hatlarından, uzun paketler uydu hatlarından gnderilebilir. Bu ynlendirme algoritması, birbirinden baėımsız yol seenekleri vasıtasıyla birinin arızalanması ya da tıkanması durumunda iletimi gerekleřtirebildiėinden gvenilirdir. Birbirinden baėımsız izler, her defasında kaynak ve hedef dūėumler arasında ardışık olarak bulunan en kısa izlerin ıkarılmasıyla geriye kalan dal ve graflar uzerinden en kısa iz bulunması yntemiyle elde edilebilir [3].

4.2.3.4. ncelikle en kısa iz ynlendirme algoritması

Paket baėlařmalı řebekeler iin son yıllarda dizayn edilen ncelikle en kısa iz ynlendirme algoritması (Open Shortest Path First-OSPF-), TCP/IP internet ortamı iin tasarlanmıřtır. OSPF; ortak bir ynlendirme algoritması ile ynlendirme bilgilerini deėiř-tokuř eden ynelticilerden oluřan tek bir otonom sisteme ait bilgilerin ynelticiler arasında daėıtılmasını saėlar. Ayrıca IP subnetlemesini (alt-řebeke ayırma), TOS (Type of service=servis tipi) tabanlı ynlendirmeyi ve harici ynlendirme bilgilerini destekler. OSPF ayrıca řebeke ynlendirme gncelleřtirmeleri iin gizlilik saėlar.

OSPF dinamik bir ynlendirme algoritması olarak IP paketlerin, IP paket bařlıėındaki gideceėi yerin IP adresine ve IP servis tipine gre ynlendirir. Sistemdeki donanım deėiřikliklerini kısa srede algılayıp yeni ynlendirme algoritmalarını az bir trafik yk altında oluřturur. Burada ynelticinin arabaėdařım nitesinin bozulması buna rnek verilebilir. Bir link ynlendirme ynteminde, herbir yneltici ortak bir veri tabanına sahiptir. Bu veritabanının herbir baėımsız parası

belli bir yöneltici kullanılabilir arabağdaşım ve ulaşılabilir komşuları hakkındaki bilgileri verir. Yöneltici, flooding denen yöntemle yerel durumunu otonom sistem boyunca dağıtır [2].

Bütün yönelticiler, aynı algoritmayı çalıştırır. Her bir yöneltici, topolojik veri tabanından, kendisi ilk nokta olmak üzere en kısa yollardan oluşan bir ağaç yapısı oluşturur. Bu en kısa yol ağacı, otonom sistemdeki her bir yere giden yönü verir. OSPF, herbir TOS için ayrı bir yön hesaplar. Belli bir yer için eşit değerli yönler varsa trafik yükü bunlar arasında dengeli dağıtılır. OSPF, belli bir grup şebekenin kümelenmesini sağlar. Böyle bir gruplamaya bir alan (bölge) denir. Bir bölgenin topolojisi otonom sistemdeki diğer bölgelerden saklıdır. Bu ise birbirinden habersiz bölgeler arasındaki yönlendirme trafiğinin azalmasına sebep olur. OSPF algoritması otonom sistemdeki yönlendirme işlemine sadece güvenilir (sağlam) routerları katar.

OSPF algoritmasının kullanıldığı paket bağlaşmalı şebekenin topolojik veritabanı, yönlü bir graf tanımlar. OSPF algoritması aşağıdaki tipte fiziksel şebekeleri destekler.

1. Uçtan-uca (Point-to-point) şebekeler
2. Yayın şebekeleri
3. Yayın türü olmayan şebekeler

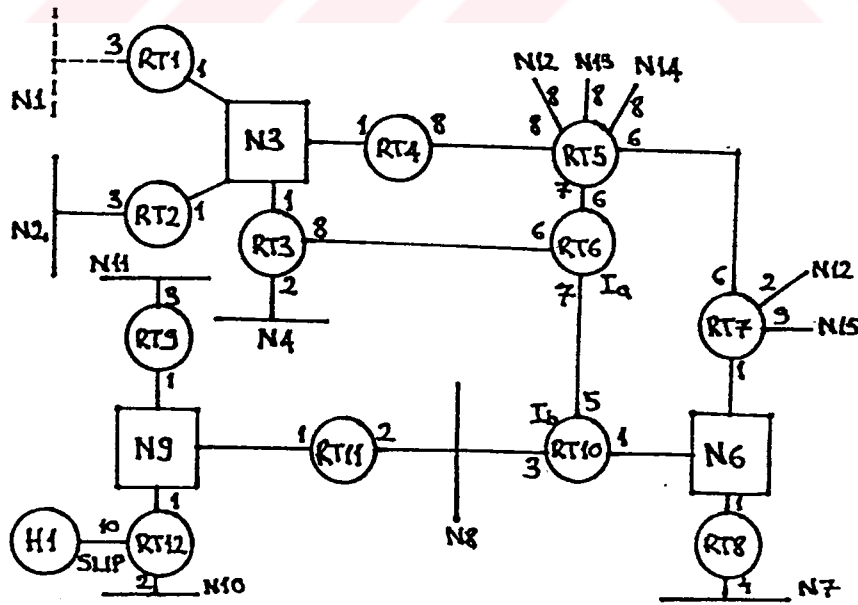
Uçtan-uca şebekeler, tek bir çift yönelticiden oluşan şebekelerdir.

Yayın şebekeleri, birden fazla bağlı yönelticileri destekleyen şebekelerdir. Bu tip şebekelerde tek bir mesaj bağlı olan bütün yönelticilerde aynı anda görülebilir. Bu şebekelerde komşu yönelticiler, OSPF'nin Hello uyarlaması ile dinamik olarak keşfedilir. Hello protokolü çok erişimli şebekelerde komşuluk ilişkilerinin kurulması ve sürdürülmesi için dinamik olarak kullanılır. Ethernet şebekeleri, yayın şebekelerine bir örnektir.

Yayın türü olmayan şebekeler birden fazla yönelticiyi destekleyen fakat yayın yapamayan şebekelerdir. Bu şebekelerde de komşu yönelticiler OSPF'nin Hello protokolü ile bulunur. Yayın yapabilme yeteneği olmadığından, Hello protokolünün doğru işlemesi için ek konfigürasyonlar gereklidir. Bu şebekelerde, OSPF algoritma paketleri her bir komşu yönelticilere çok hedefli (belli bir gruba) olarak gönderilir.

OSPF, herhangi bir varış yeri için farklı servis tiplerine göre farklı yönler hesaplayabilir. IP TOS değeri, IP paket başlığında yer alır. Bütün OSPF uygulamaları TOS'a göre yönleri hesaplayabilmelidir. Buna karşı, bütün yönelticiler TOS 0'a göre bütün paketleri yönlendirebilir. Bazı yönelticiler sadece TOS 0'a göre bazı, yönelticilerse her tür TOS değerine göre yön hesaplayabilirler. Bazen TOS 0 dışındaki değerler için herhangi bir yol bulunamaz. Bu gibi durumlarda hangi TOS istenirse istensin TOS 0'a göre hesaplama yapılır.

Harici yönlendirme bilgisi, EGP gibi başka bir yönlendirme algoritması tarafından veya statik şekilde konfigüre edilerek oluşturulur. Default yönler otonom sistemin harici yönlendirme bilgisi olarak dahil edilebilir. Harici yönlendirme bilgisi bütün otonom sistem boyunca yayınlanır. Şekil 4.13'te bütün yönelticiler RT7'nin 2 ve 9 numaralı metriklerinin harici yönü olduğunu bilirler.



Şekil 4.13. Bir otonom sistem örneği

OSPF iki tip harici metrik destekler. Tip 1 harici metrik, link iletim maliyetinin metriğine eşittir. Şekil 4.13'te RT5 ve RT7'nin tip 1 harici metrik yayınladığı ele alınırsa her bir harici yön için RT6'dan uzaklık, harici yönlerin değerleriyle RT6'dan ilan edilen yönelticiye olan uzaklık toplanarak elde edilir. RT5 ve RT7, N12'ye bir harici yön ilan ediyor. Burada, RT simgesi yöneltici, N simgesi erişim organı, H simgesi kullanıcı uç anlamında gösterilmiştir.

Bu durumda RT7, N12'ye 2'lik bir mesafe, RT5 ise 8'lik bir mesafe ilan ediyor. Bundan dolayı RT7 tercih edilir. Tip 2 harici metrikler daha basittir. En düşük harici metriği ilan eden otonom sistem sınır yönelticisi seçilir. Otonom sistem sınır yönelticisine olan dahili mesafeler göz ardı edilir. Örneğin RT5 ve RT7'nin tip 2 harici yön ilan ettiği varsayalım. O zaman bütün N12 için olan trafik RT7'ye yönelecek, çünkü $2 < 8$ 'dir. Tip 2 yönler için eşit değerler olduğu zaman, bu eşitliği bozmak için dahili mesafelerde hesaba katılır. Gidilecek herhangi bir yer için birçok eşit değerli yön olursa bunların hepsi keşfedilir ve kullanılır.

4.2.4.4. Merkezi yönlendirme

Dinamik yönlendirme algoritması olan merkezi yönlendirmede her santraldan komşu santralların durumu, kuyruk uzunluğu, hatlardaki trafik miktarı hakkında yönlendirme kontrol merkezine (Routing Control Center-RCC.) periyodik olarak bilgi aktarılır. Topolojisinin statik ve trafiğin az değiştiği durumlarda statik yönlendirme algoritmaları tercih edilirken, gün içinde hatların arızalandığı ve trafik akışının dalgalandığı durumlarda merkezi yönlendirmenin sağladığı çözümler daha verimlidir.

RCC'de toplanan bilgiler şebeke düzeyi kapsamında değerlendirilerek her düğüm çifti için en kısa iz algoritması gibi yöntemlerle en iyi yollar belirlenir ve santrallara gönderilir. Bu yönüyle şebekenin geneline göre değerlendirme yaptığı için diğer algoritmalara göre daha avantajlı görünen bu yöntemin sakıncaları da mevcuttur. Öncelikle istenen verimin sağlanması için güncelleşme periyodu, trafikteki değişimin istenen değişim sınırları dışındaki sapmalarına uyacak ölçüde olmalıdır. Ayrıca

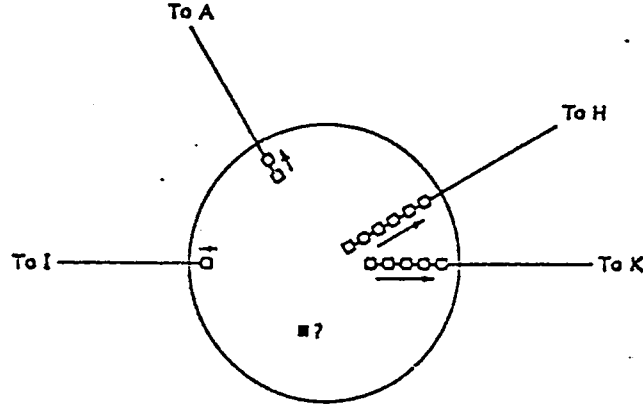
RCC'nin bozulması durumunda şebekenin zor durumda kalması olasılığı vardır. Bunun için RCC'nin yedeklenmesi ise masraflı olur. Oluşturulan yönlendirme tablolarının düğümlere uzaklıkları nisbetince farklı aktarılmaları sonucu yeni karar verme durumunda ihtilaflar ortaya çıkabilir ya da çevrimler oluşabilir. RCC'nin en iyi yol hesabı yanında yedek yol önerileri yoksa bu yol üzerindeki hata, arıza ya da tıkanmalar bu uç düğümlerin bağlaşmasını engelleyebilir. Santraller RCC'ye veri gönderirken RCC'ye yakın olan hatlar üzerinden daha fazla santral verisi gönderileceğinden bu hatlarda yığılma ya da tıkanma olabilir [3].

RCC'de yapılan en iyi yol hesaplamaları geniş şebekeler için uzun sürebilir. Bunun için hesaplama algoritmasının iyi seçilmesi lazımdır. En kısa iz hesaplama yöntemlerinde gösterilen ilk üç algoritmaya göre tezde ele alınan genetik algoritma daha basit ve kısa süreli hesaplama özelliğine sahip olduğundan hesaplama periyodunu daha kısa tutma şansı olur. Bu durumda hızlı değişen trafiğe daha uyumlu cevap verme avantajı ortaya çıkar.

4.2.4.5. Yalıtılmış yönlendirme

Merkezi olmayan yönlendirme algoritmalarından olan yalıtılmış yönlendirmede, düğümler yönlendirme kararlarını komşu düğümlerle bilgi değiş-tokuş etmeden kendi bünyelerinde bağımsız karar verirler. Yalıtılmış uyarlanır (adaptive) algoritmalar olarak da adlandırılan bu metodda kısmi olarak topoloji ve trafik değişimlerine uyularak çalışılarak merkezi yönlendirmede RCC kaynaklı sorunlar aşılmaya çalışılır.

Yalıtılmış algoritmalarından en çok bilineni, Baran'ın "sıcak patates" (hot potato) algoritmasıdır. Sıcak bir patatesi tutan bir kişinin bunu bir an önce başkasına vererek kurtulmaya çalışmasındaki duruma benzetilerek, düğüm gelen paketi çıkış hatlarına bağlı kuyruklardan en kısa olanına yerleştirerek en kısa sürede onu hedefe bakmaksızın diğer düğüme göndermeye çalışır. Şekil 4.14'teki düğümde gelen paket, mevcut 4 çıkış hattında en az paketin sırada beklediği (1 paket) I kuyruğuna yönlendirilir.



Şekil 4.14. Düğümdeki çıkış kuyrukları

Sıcak patates algoritmasının bazı dezavantajları vardır. Bunların başında hedefin göz önünde bulundurulmamasından kaynaklanan şebeke içi gereksiz yollar ve sürekli çevrimler olabilir. Bu sorun algoritmanın statik yönlendirme ile beraber değerlendirilmesiyle çözülmeye çalışılmıştır.

Bu şekilde gelen paket sadece kuyruk uzunluğuna göre değil aynı zamanda belirli hedefler için belirli yönlendirme seçeneklerinin söz konusu olduğu statik yönlendirme tablosuna göre kuyruğa yerleştirilir. Kuyrukların statik ağırlığı tabloya göre yönlendirme olasılıklarıyla belirlenir. Kuyruk belirli bir eşik değeri aşmadıkça, en iyi statik seçim kullanılır. Statik ağırlık belirli bir sınırın altına inmedikçe de en kısa kuyruk tercih edilir. Alternatif bir yol, kuyruk uzunluğu ve statik ağırlığın tersinin toplamı minimum olacak şekilde yönlendirme seçeneğinin kullanılmasıdır. İstenen özellik, kullanılan algoritmadan bağımsız olarak düşük yük altında yüksek statik ağırlığı olan kuyruğun seçilmesidir. Kuyruk eşik değerini aşarsa trafik diğer hatlara yönlendirilir.

Baran tarafından önerilen diğer algoritma geriye doğru öğrenmedir (back-ward learning). Burada işleyiş tarzı mevcut hedeflere doğru yol seçimi için bu hedeflerden daha önce gelen paketlerin bu hedefler hakkındaki bilgilerini kullanarak karar vermektir. Bu yöntemde her paket kaynak kimliğini içerecek şekilde belirlenir. Paket her düğüm geçişinde bunu kaydedecek bir sayaca da sahiptir. Sayaç her düğüm geçişinde bir artırılır. Bu durumda bir düğüm belirli bir hedef için önceden sahip

olduğu en kısa iz hesaplaması gereğince bu hedef düğümden gelen paketlerin sayacı bu izden daha az düğüme sahipse söz konusu hedef için en uygun yolun içerdiği düğüm sayısını bu şekilde kaydeder ve buna uygun olarak algoritmayı güncelleştirir. Düğümler sadece en iyi değişiklikleri kaydettiğinden bir hattın bozulması veya aşırı yüklenmesi durumlarından haberdar olmayabilir. Bu sorunu aşmak için düğümler periyodik olarak yönlendirme tablolarını güncelleştirilecek şekilde düzenlenmelidir. Bu ise yönlendirmeyi belirli bir kalite sınavasından uzak en iyi olmayan seçimlere sürükler. Öğrenme işlemini sık olmayacak biçimde yinelemek bu durumu düzeltebilir.

4.2.4.6. Delta yönlendirme

Merkezi ve yalıtılmış yönlendirmenin avantajlarını birleştirmek için önerilen delta yönlendirme algoritmasında her düğüm çıkış hatlarındaki gecikme, kuyruk uzunluğu, bandgenişliği gibi bilgileri hesaplayarak RCC'ye gönderir. RCC gelen bilgiler ışığında olası düğüm çiftleri arasında en kısa izleri hesapladığı gibi diğer en iyi yolları da sırasına göre belirli sayıda hesaplar. Bu yollar birbirinden bağımsız olmak zorunda değilse de başlangıç hatları farklıdır. En iyi ize göre farkı belirlenen bir diskriminant ($=\Delta$) değerinden küçük olan yollar en iyi yolla eşdeğer kabul edilir, çünkü aralarındaki maliyet farkı dikkate alınmayacak ölçüde azdır. Böylece bu eşdeğer yollar sadece başlangıç hatları itibarıyla bahsedilen düğümlere iletilir. Başlangıç hattının yeterli olması; aynı hedef için bir sonraki düğüm de eşdeğer yol seçeneklerine sahip olduğu durumuyla açıklanır.

Bu yöntemde Δ değeri ayarlanarak algoritmanın merkezi ve yalıtılmış özellikleri arasındaki ağırlık ayarlanabilir. $\Delta \rightarrow 0$ ise diğer tüm yollar en iyi yola eşdeğer olamayacağı için sadece RCC tarafından bildirilen en iyi çıkış hattı dolayısıyla algoritma merkezi olur. Bunun yanında $\Delta \rightarrow \infty$ ise her olası çıkış değeriyle eşdeğer olduğu için karar o anda mevcut kuyruk uzunluğuna göre düğüm tarafından yalıtılmış algoritma doğrultusunda verilir.

4.2.4.7. Dağıtılmış yönlendirme

ARPANET tarafından kullanılan bu yönlendirme algoritmasında her düğüm komşu düğümlerle iletim bilgilerini değiştirir. Bu bilgiler komşu düğümlere olan uzaklık, kuyruk uzunluğu ya da gönderilen yansıma paketleriyle ölçülen gecikme olabileceği gibi, bununla beraber ulaşmak istenen hedefle ilgili komşu düğümlerden alınan kestirimlerdir. Bu iki bilgiyi sentezleyerek kendisi de komşusu olduğu düğümlere olası hedeflerle ilgili kestirimleri gönderir [3].

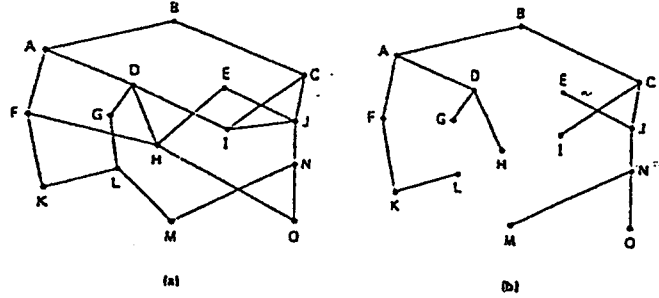
Gecikmenin kullanıldığı durumlarda her düğümün komşularına olan gecikmeyi bildiği kabul edilir. Bu durumda bir düğüm X hedefine ulaşmak için komşu düğümlerinden X hedefine olan kestirimleri alır ve her aldığı kestirimi ilgili komşu düğüme olan gecikmeyle toplayarak X hedefine, komşu düğümler üzerinden toplam gecikmeyi hesaplar. Sonuç olarak en küçük değeri veren komşu düğüme bağlı çıkış hattını seçer.

4.2.4.8. En İyi yönlendirme

En iyi yönlendirme esası iki düğüm çifti arasında en iyi yönün bilinmesi durumunda bu yol içindeki bir düğümün bu iki düğümle olan en iyi yolları başlangıçta belirtilen iki düğüm arasındaki en iyi yolun ara düğümlerle uç düğüm arasındaki parçaları olmasına dayanır. Aksi durumda ilk belirtilen en iyi yolun herhangi bir parçasının daha uygun başka parçayla ve doğal olarak toplam yolun daha uygun bir yolla değişmesi gerekir.

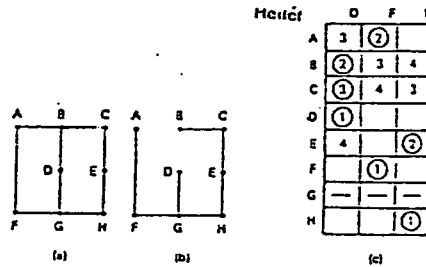
Yukarıda bahsedilen özellik gereğince tüm kaynaklardan belirli bir hedef düğüme olan en iyi izlerin oluşturduğu graf bir ağaçtır. Hedef düğüm ve dışındaki düğümleri (kaynak düğümler) içerdiği için bütün düğümler bu grafa mevcuttur. Bu grafa çevrim olsaydı çevrimin herhangi iki düğümü arasında birbirinden bağımsız iki yol olurdu ve bu yollar en azından bir kaynak düğüm için kullanılmak durumundadır. Aksi takdirde kullanılmayan bu yol en iyi yol parçası olarak bulunmazdı. Bu durumda ise bu kaynak düğüm için hedefe ulaşmada en iyi yolun bu kısmında iki

seçeneği olurdu. Halbuki en iyi yol bir tanedir. Bu çelişki çevrim olmadığı sonucunu çıkarır. Sonuç olarak bu graf “sink ağacı” denilen bir ağaçtır. (Şekil 4.15). Böyle bir ağaçta her paket sonlu ve çevrimsiz şekilde hedefe ulaşır [3].



Şekil 4.15. (a) şebeke
(b) B düğümü için sink ağacı

Belirli bir sink ağacında kaynaktan hedefe ilerleme yönünde önceki düğüme sonraki düğüme göre “up stream”, sonraki düğüme ise önceki düğüme göre “down stream” denir. Sink ağacında herhangi bir hat arızalandığı veya devreden çıktığı zaman bu hat üzerinden hedefe giden düğümlerin, ağaçta çevrim olmadığı için, başka yolları yoktur. Bu durumda sink ağacının bağımsız olarak kendi sink ağacı üzerindeki yolundan ulaşmasına imkan veren bir komşusu aranır ve bunlar arasından en uygunu belirlenir. Bu yöntemi esas alan Chu tarafından geliştirilen bir algoritma mevcuttur. Şekil 4.16’da her kolon, belli bir çıkış hattı üzerinden hedef düğüme olan düğüm sayılarını verir. G düğümü için çıkış hatları D,F ve H’tır. Satırlardaki en küçük değer o hedef için en iyi yol olarak daire içine alınarak gösterilmiştir.



Şekil 4.16. (a) Şebeke
(b) H için Sink ağacı
(c) G’nin kullandığı yönlendirme tablosu

Bu şebeke üzerinden GH hattının hatalı olması durumunda G düğümü H kolonunda bulunan tüm girişleri kullanılmaz olarak siler. Hedef satırlarında bu değerler göze alınmaz. Bu durumda satırlarda kalan varsa diğer yollar arasından en iyi yol seçilir. Örneğin E hedefi için, D yolu üzerinden dört düğümlük bir alternatif yol vardır. Aynı durum H hedefi için söz konusu değildir. Çünkü Şekil 4.16'da görüldüğü gibi GH hattı H'nin sink ağacında bulunur ve G'nin her iki komşusunda H'a G üzerinden ulaştığı için yani G'nin upstream'ı oldukları için tabloda bunlara ait alternatif yollar mevcut değildir. Bundan dolayı G her iki komşusuna "H'a ulaşamıyorum" mesajı içeren kontrol paketleri gönderir. Bu komşuların ağaçları üzerinden G'nin bu düğümler üzerinden H hedefine olan en iyi yolların hangisi uygunsa o yolun ait olduğu ağacın ilgili bulunduğu komşu düğüm seçilir. Bu şebekede D böyle bir seçenek yola sahiptir. F'de böyle bir yola sahip olmasına karşılık D'deki seçeneğe göre daha fazla düğüm içermektedir. Bu durumda D üzerinden seçenekli yol tercih edilir. Bu yöntemde eğer şebeke birbirine tek hatla bağlı iki alt şebekeye ayrılmışsa algoritmada tıkanan hatta karşılık hedefe varan seçenek yol bulunmayabilir.

4.2.4.9. Taşma algoritması

Taşma algoritması, yalıtılmış algoritmanın en geniş yorumlanabilecek durumudur. Bu algoritmada ara düğüme gelen paket çoğullanarak her çıkış hattına verilir. Oluşturulan paket sayılarının sonsuza gitmesini önlemek için sayaç metodu uygulanır. Paketlere eklenen sayaç kısmı başlangıç ile hedef arasındaki düğüm sayısını dikkate alarak belirli bir değerden her düğüm geçişinde bir eksiltir. Sıfıra ulaşan sayaç durumunda paket şebekeden atılır. Kaynak-hedef uzaklığı bilinmiyorsa olsa durumların en kötüsü tercih edilir. Diğer bir yol ise her kaynak başlangıç paketlerine bir numara verir. Her düğüm ise diğer düğümlere ilişkin bir listeye sahip olur. Paket bu düğüme geldiğinde düğüm o paketin belirttiği numaradaki düğümün hafızada işlenip işlenmediğini kontrol eder. Sadece ilk gelen paketler için o kaynak düğümlerle ilgili hafıza bölümüne numaralar işlendiği için, eğer gelen paket için bu numara hafızada mevcutsa önceden gelmiş olduğundan yeni paketin eskisinin bir aşamada yedeklenmiş hali olduğuna karar vererek şebekeden çıkarır.

Taşma kamu şebekeleri için uygun olmasa da askeri uygulamalarda kullanılmaktadır. Askeri alanda bazı haberleşme türleri oldukça önemlidir. Bu nedenle mevcut imkanlar dahilinde en kısa sürede en güvenilir yoldan iletimin sağlanması istenir. Taşma algoritması ise bir düğümde mevcut her çıkışı kullandığından paralel bütün yollar arasından en iyisini de otomatikman seçer. Bu durumda taşmanın yüklü olması istenir. Ancak taşma algoritmasının sebep olduğu aşırı yüklü trafik durumunu hafifletmek için şebekede gerekli olan haberleşme için öncelik sağlanmalıdır. Bir başka yol ise seçici taşmadır (selective flooding). Bu algoritmada, giren paketler sadece yaklaşık olarak hedefe yönelik çıkışlara verilerek yük azaltılır. Belirli bir doğrultuda uzantılı yayılan şebekelerde bu yöntem uygulanabilir. Diğer bir yol rastgele adım algoritmasıdır (random walk). Düğüm gelen paketi rastgele bir hat üzerinden çıkışa verir. Şebeke düğümleri, ağ (mesh) yapılı şebekedeki gibi, birbirine birçok yolla bağlıysa bu yöntemin başarılı olma olasılığı artar.

4.2.4.10. Hiyerarşik yönlendirme

Büyük boyutlu şebekelerde düğümlerdeki yönlendirme tabloları düğüm sayısı nisbetince geniş ölçekli olduğundan hem düğümde hafıza sorunu oluşur hem de algoritma hesaplamada parametre sayısı arttığından süre uzamasına neden olur. Şebekenin genişlemesi ile her bir düğüm için bazı hedeflere yol belirleyemeyecek duruma gelirse hiyerarşik yönlendirme uygulamak gerekli olacaktır.

Telefon şebekelerinde kullanıldığı üzere hiyerarşik yönlendirmede şebeke bölgelere ayrılır ve bölgeler de birbiriyle belli düğümler vasıtasıyla bağlanırlar. Bu düğümler bölgeler arası bağlantı düğümleri olarak tanımlanır. Böylece bir düğüm sadece kendi bölgesinin bilgilerine göre yönlendirme yapar ve diğer bölgelerdeki düğümlere o düğümlerin ait olduğu bölgelerarası hat üzerinden erişmek dışında bir belirlemede bulunmaz. Bu şekilde düğümlerin hesap yükü azalır. Farklı şebekeler birbirine hiyerarşik yönlendirme doğrultusunda bağlanır.

Hiyerarşik yönlendirme bölge seviye sayısına göre ayrılır. Çoğu zaman iki seviyeli hiyerarşik yönlendirme yeterlidir. Seviye sayısı arttığında bölgeler birbirini kapsayacak şekilde alt bölge seviyelerine doğru kümelendirilir.

Hiyerarşik yönlendirmenin bazı dezavantajları vardır. Artan yol uzunluğu bunlardan biridir. Bunun nedeni, bir bölgedeki düğüm, başka bölgedeki düğüme ulaşmak istediği bölgeye bağlı bölgelerarası yoldan gitmek zorundadır; fakat bu arayol hedef bölgedeki düğümlerin kaynak bölgeye göre ortalama uzunluğuna göre belirlenir. Ulaşılmak istenen düğüm ise bölgeye olan uzaklığının üstünde bir uzaklıkta olduğu zaman kaynak düğüme yollar üzerinden daha yakın olabilir.

Şebekeler büyüdükçe hiyerarşi seviyesinin belirlenmesi önemlidir. Hiyerarşi seviyesi arttığında, düğümlerin hedef belirleme yükü azalır fakat yukarıda bahsedilen yol uzunluğunun artması olasılığı artar. Bundan dolayı Kamoun ve Kleinrock en iyi çözümü önermişlerdir. N düğümlü bir şebeke için en iyi seviye sayısı $\ln N$ 'den küçük en büyük tam sayıdır. Her düğümden $\ln N$ giriş vardır. Örneğin, 500 düğümlü bir şebeke iki seviyeli hiyerarşide 20 bölgeye ayrılrsa her bölgede 25 düğüm olur. Bölge içi girişler 24 ve bölge dışı girişler 19 olmak üzere her düğümden 43 giriş olmalıdır. Üç seviyeli hiyerarşide ise 10 bölge, 5 altbölge ve 10 düğüm olur. Her düğüm yerel düğümler için 9, altbölgeler için 4 ve bölgeler için 9 girişe sahip olacaktır.

4.2.4.11. Akış-tabanlı yönlendirme

Akış tabanlı yönlendirme (Flow-based routing) şebekelerdeki düğüm çiftleri arasındaki ortalama gecikmeler üzerinden şebekenin ağırlıklı ortalama gecikmesini minimum yapmaya yarayan bir yöntemdir. Eğer şebeke düğümleri arasındaki trafik düzenli ve zaman dağılımı periyodik ise yönlendirme matematiksel olarak optimize edilebilir.

Algoritmada, bağların kapasite ve ortalama paket akışı biliniyorsa kuyruklama teorisinde kullanılan gecikme formülü yardımıyla gecikme bulunur. Her bağın

gecikmesiyle taşıdığı trafik cinsinden ağırlığı nisbetince şebekenin ağırlaştırılmış akış (flow-weighted) ortalaması bulunur. Böylece yönlendirme işlemi bu ağırlaştırılmış akış ortalamasını minimum yapmaya çalışır.

Bu yöntemde, öncelikle düğümlerin birbirine ne şekilde bağlandığını belirten topolojik yapı bilinmelidir. Bunun yanında hatlarda akan trafiği, saniyede geçiş yapan paket cinsinden veren Fij trafik matrisi (i: kaynak, j: hedef) ve hatların taşıma kapasite matrisi Cij kbit/sn cinsinden verilmelidir. Son olarak kullanılan yönlendirme algoritması seçilmelidir.

Sadece tek yol bulan yönlendirme algoritmaları seçilirse belirli düğüm çiftleri arasındaki trafik sadece bu yol üzerinden akacağı için en küçük ortalama gecikmeyi temsilen en iyi yönlendirme algoritmasının işlemesi kolaydır. Buna karşılık, çok izli yönlendirme algoritması kullanılırsa kaynak ve hedef arasında değişik trafik dağılımlarına göre sonsuz sayıda yol mevcut olduğundan hesap yapmak çok zordur. Bu sorun, trafiğin her olası yola belirli oranlarda dağıtılarak bu oranları temsil eden parametreler cinsinden bağdaki akışlar belirlenerek bu parametreleri en iyi değere çekmeye çalışan basit teknikler vasıtasıyla aşılmaya çalışılır [3].

4.2.4.12. ATM tabanlı yönlendirme

Genellikle patlamalı tipten veri trafiğin söz konusu olduğu ATM tekniği ile gerçekleştirilen trafik akışını düzenlemek ve buna göre trafik sınıflarını belirlemek için tasarlanmıştır.

Kaynak aşamasında trafiğe göre ATM katmanının düzenlenmesi ise trafik yönlendirme esasına dayanan metodda trafik kontrol kısmı şebekeden tahsis edilen bağlantıların sıkışmaya neden olmaması için yapılır. Yığılma kontrolü kısmı ise şebekeden alınan bağlantıların yoğunluğunu minimize etmek, trafiği rahatlatmak ve yaymak, yığılma süresini kontrol etmeyi amaçlar. Bu olaylar, kaynak dışında, yığılma nedeniyle bir veya birçok şebeke düğümünde yapılabilir.

ATM katmanı trafik ve yığılma kontrolü için bütün B-ISDN hizmetlerinde QOS (Hizmet Kalitesi) sınıfları kullanır. Bu QOS sınıfları çalışmakta olan şebeke performansı ile uyuşacak şekilde belirlenir. Ayrıca ATM katmanı CAC (Bağlantı kabul kontrolü) protokolü gereğince trafik kılavuzu nisbetince trafik kalitesini belirleyen bağlantılar tanımlanır. Bu protokol şebekenin o andaki performans kriterleriyle uyumlu olmalıdır. Bunlara ek olarak UPC/NPC (Şebeke parametresi kontrolü) bu uyumluluğu sağlar. Bu protokol, trafik hacmine göre kullanıcı trafiğinin ve hücre yönetiminin geçerliliğinin izlenmesi ve kontrolü için şebeke tarafından yapılan bir grup hareket doğrultusunda belirlenir. Bunlardan sonra geniş bandlı şebekede trafik ve yığılma kontrolü karmaşıklığını çözmek için adım-adım denetleme algoritması uygulanır. UPC/NPC tarafından istenmeyen bir parametre kullanımı için şebeke performansının belirlenen seviyede devamı doğrultusunda ATM katmanları denetlenir ve uygun olmayanları ayıklanır.

Virtüel yol uyarınca yapılan bağlantılarda trafik kontrolü; CAC yapılması, istenen değişik QOS'a göre trafik tiplerini ayırarak bir çeşit öncelik kontrolü yapılması, trafik kontrolü için diğer düğümlere mesaj dağıtımı için UPC protokolünü kullanır. ATM tabanlı bir şebekede CAC'nın temelinde, yapılan bir çağırmanın bağlantı isteği sadece eldeki kaynaklar, bağlantı için yeterli ve çağırmanın istediği QOS şartları sağlanabiliyorsa ve bu süre içinde devam ettirilebiliyorsa kabul edilir. Bu açıdan şebeke dinamik bir yönlendirme esasına göre çalışmaktadır.

Kullanım parametresi kontrolü, trafik hacmine göre kullanıcı trafiğinin ve hücre yönetiminin geçerliliğinin izlenmesi ve kontrolü için şebeke tarafından yapılan bir grup hareket olarak tanımlanır. Asıl amacı, şebeke imkanlarının normal parametrelerinin zedelenmesinin farkedilmesiyle, halen kurulmuş bağlantıların hizmet kalitelerinin etkilenmesi sonucunu doğuracak davranışlardan kaçınmaktır. UPC şebeke içinde santral girişi, konsantratör ve çapraz bağlantı noktalarında gelen hücrelerde bir bozulma bulmuşsa hücreleri atabilir ya da etiketleyerek yığılma durumunda ise bunları egale ederek şebeke yükünü ayarlar [2].

ATM tabanlı yönlendirme, başlangıç aşamasında ve bağlaşma esnasında ATM hücrelerini aktif olarak yönlendiren ve denetleyen bir algoritma türü olarak B-ISDN şebekelerinde kullanılır.

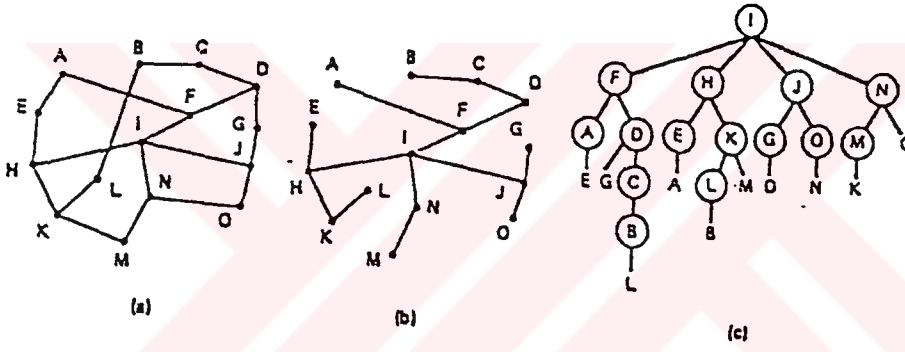
4.2.4.13. Yayın yönlendirme

Yayın yönlendirme, dağıtılmış veri tabanı güncelleştirmede kullanılan yöntem gibi düğümlerin diğer tüm düğümlere mesaj göndermesi esasına dayanır. Bir paketin çoğullanarak tüm hedeflere eşzamanlı gönderilmesine “yayın” (broadcasting) denir. Kaynağın her hedefe bir liste gereğince farklı paket göndermesi en basit yönlendirme algoritmasıdır. Bandgenişliği imkanını gereğince kullanamadığı için elverişsizdir. Taşma (flooding) noktadan noktaya iletişim için, çok fazla bandgenişliği gerektirdiğinden, kullanışlı bir metod değildir. Başka metodlar yayın hızını karşılayamazsa bu yöntem tercih edilir.

Çok hedefli yönlendirmede (multidestination routing) istenen hedefe ulaşmak için ya hedef listesi ya da bit haritası pakete eklenir. Düğüme bir paket vardığında, düğüm hedef listesiyle ilgili çıkış hatlarını tüm hedefleri kontrol ederek belirler. Burada bir çıkış hattı en azından bir hedef listesi için en iyi yolsa kullanır. Bu durumda düğüm her çıkış hattı için aynı paketi kopyalar. Her pakette ise o bağlantıyı kullanan hedef içerilir. Algoritmada hedef kümeleri çıkış hatları arasında paylaştırılmıştır. Sınırlı sayıda geçişten sonra her paket sadece bir hedef taşır ve sonunda aynı paketler istenilen hedeflere ulaşmış olur. Çok hedefli yönlendirme ayrı olarak adreslenmiş paket kopyaları gibi işlemesine rağmen aynı izi kullanan paketlerden sadece biri ücretlendirilir.

Diğer bir yayın algoritması her düğüm şebeke için uygun oluşturulan bir kapsayan ağaca göre gelen yayın paketini geldiği bağ hariç ağaca ait diğer bağlara yönlendirir. Bu durumda çevrim içi gereksiz paket yayınları engellenerek bandgenişliği verimli kullanılır. Dezavantajı ise her düğüm uygun kapsayan ağaçları bilmelidir.

Yukarıdaki algoritmaya benzeyen diğer bir algoritma, düğümlerin kapsayan ağaç bilme zorunluluğunu kaldırır. Bu methodda paketin geldiği bağ düğüm tarafından paketin bildirdiği kaynağa iletim yapmak için kullanılıyorsa, düğüm gelen paketin en iyi iz üzerinden geldiğini anlar ve kopyalayarak diğer çıkış bağlarına yöneltir aksi durumda paket şebekeden çıkarılır. Ters iz izleme (reverse path forwarding) algoritması Şekil 4.17’de gösterilmiştir. Bu metod kolay ve verimlidir. Düğümler kapsayan ağaç hakkında bilgi saklamak zorunda değildir. Ayrıca paketin boş kısmında da hedef bilgisine gerek yoktur. Taşmada olduğu gibi bir aşamada algoritmaya müdahalede gereksizdir.



Şekil 4.17. Ters iz izleme algoritması

Şekil 4.17’de şebeke, I düğümü için sink ağacı ve ters iz izleme algoritmasının işleyişi görülmektedir. İlk düğümde I; F, H, J, N düğümlerinin hepsine paket gönderir. Bu düğümler I’ya önceden belirlenen bağ üzerinden erişir. Bu aşamada çevrim oluşturmayacak şekilde ikinci düğümlerden paketler gönderilir; bu vesileyle bir bağ üzerinden kaynağa paket gönderilemiyor (O düğümü) çünkü bu dallanma önceki belirlenmiş düğümle (J düğümü) çevrim oluşturuyor. Üçüncü düğümde üretilen dokuz paketten sadece ikisi iz üzerinden kaynağa paket gönderir (C ve L).

Beşinci düğümünden sonra yayın durur. Böylece üretilen 24 paketten 14'ü hedefine varmış olur.



5. YÖNLENDİRME VE YAPAY ZEKA

5.1. Yapay Zeka

Yapay zeka, zeki davranışın incelenmesidir. Zeki davranışın en iyi şekilde, kopyalanmaya çalışılarak incelenebileceği yaklaşımından hareket eder. Bu kopyalama bilgisayar simülasyonu yolu ile olur. Dolayısıyla yapay zekanın bilgisayar biliminin bir parçası olduğu düşünülebilir. Bununla beraber, zeki davranışlar gösteren faydalı makinaları geliştirmek, pek çok disiplinin birlikte çalışmasını gerektiren çok boyutlu bir konudur. Bunlar arasında psikoloji, dil bilimi, yöneylem araştırması ve mekanik gibi disiplinler sayılabilir.

Günümüzde yapay zeka programlarının önemi sadece akademisyenler tarafından değil, ticari yazılım üreticileri tarafından da anlaşılmıştır. Ardı ardına zeki programlar piyasaya çıkmaktadır. Tezde ise trafik yönlendirme alanında yapay zeka programı üzerinde inceleme yapılmıştır.

Yapay zekanın temel araştırma sahası makine öğrenmesidir. Makine öğrenmesi; akıllı sistemlerin bilgilerini arttırdığı ve yeteneklerini geliştirdiği süreç olarak tanımlanabilir. Makine öğrenmesinin seviyeleri aşağıda şematik olarak verilmiştir[8].

1. Bilgi Seviyesi: İndüktif Öğrenme Metodları
(SBL, ID3 vs.)
Dedüktif (EBL) Metodlar
2. Sembol Seviyesi: Sınıflayıcılar
Genetik Algoritmalar
3. Aygıt Seviyesi: Algılayıcılar (Perceptrons)
Yapay Sinir Ağları (NNs)

5.2. Trafik Yönlendirmede Yapay Zeka

Haberleşme sistemlerinde trafik yönlendirme genellikle öğrenme metoduna dayanmaktadır. Bu doğrultuda özellikle dinamik yönlendirme tekniklerinde şebeke, değişen şartları algılayıp buna göre önlem alma ve hizmet kalitesini geliştirme şansını bir nevi sürekli bilgi alışverişi sayesinde gerçekleştirmektedir.

Öğrenme, yapay zekanın temel ilgi alanı olduğu gibi onun üzerine şebeke yönetim sistemi gibi akıllı sistemlerin bilgilerini belirli programlarla geliştirme ve yeteneklerini arttırma tekniklerini ortaya koymuştur. Tezde bu anlayış doğrultusunda, yönlendirme semantiğine uygun bağlamda indüktif öğrenme metodu, yapay sinir ağları uygulamalarına değinilmiş ve ağırlıklı olarak genetik algoritmalar üzerinde durulmuştur.

5.2.1. Yönlendirmede indüktif öğrenme uygulamaları

İndüktif öğrenme örnekler, deney ve keşiflerden öğrenmeyi içerir. Örneklerden öğrenmede verilen örnek kümesinden akıllı makine genel bir içerik çıkarır. Bu, örneklerin ortak özelliğine dayalı olabileceği gibi analogik yönlerini de göz önüne alabilir. Benzerlik tabanlı öğrenme (SBL) ve içeriksel kümeleme bu öğrenme çeşidine örnektir. Genetik algoritmalar da kısmen bu alanda incelenebilir.

Yönlendirmede dinamik algoritmaların kullandığı şebekelerde esas amaçlanan değişen şebeke şartlarına kısa sürede uyum sağlamaktır. Bu ise ister merkezi ister dağıtılmış veri tabanına dayansın, düğümlerin şebeke hakkındaki kısmi ya da genel bilgileri alıp kullanmasıyla olur. Mesela, paket bağlaşmalı şebekelerde uygulanan taşıma algoritmasında trafik yükünü azaltmak için uygulanan seçici taşıma gibi modellere ek olarak, indüktif öğrenmeye dayalı modeller sunulabilir. Bu modelde her bir düğüm belirli hedefler için belirli çıkış hatlarını kullanarak yedeklediği paketlerin ne kadarının geri geldiği ve hangi hizmet öncelikli paketlerin hangi çıkış kümelerinden sıklıkla geri geldiği kaydedilerek ortak veriler değerlendirilir ve

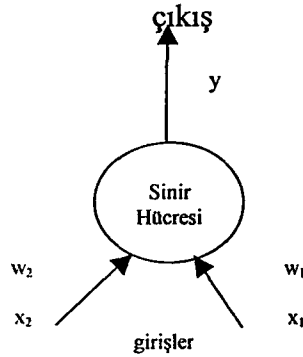
değişen şartlara belirli sürede uyma yolunda hedefler için uygun çıkış hatlarını öğrenir.

İndüktif öğrenme metodu esasen önceki bölümlerde bahsedilen birçok algoritmanın temelinde bulunan genel bir metottur. Yalıtılmış yönlendirme algoritmalarından geriye doğru öğrenme yöntemi, temelde gelen verileri değerlendirip diğerleriyle karşılaştırıp uygun olan bağlantı yollarının öğrenilmesidir. Bunun gibi dağıtılmış yönlendirmede her komşu düğümün sınanarak gecikme sürelerinin öğrenilip belirli hedef için uygun komşu düğümün saptanması da bu öğrenmenin bir çeşididir. Bu örnekler çoğaltılabilir. Bunlar kendi konu başlıkları altında incelendiği için burada ayrıca değinilmeyecektir.

5.2.2. Yönlendirmede yapay sinir ağları uygulamaları

Aygıt seviyesinde öğrenme metodu olan yapay sinir ağları, canlı organizmalarda bulunan sinir sistemin işleyişi esas alınarak tanımlanmıştır. Bu yöntemde akıllı makine, aldığı verilerin çıkışında istenen durumu oluşturup oluşturmadığına göre ağırlıklı değerlendirmelerini değiştirip sistemin işleyişini geliştirmeye çalışır.

McCulloch-Pitts'in modeli NNs'in en yaygın şeklidir (Şekil 5.1). Burada şebekedeki düğüm olarak kabul edebileceğimiz sinirsel yapı birimi "algılayıcı" gereğince çıkışları ağırlıklı olarak (5.1) değerlendirir. Eğer çıkış istenen şekilde değilse, ağırlıklar çıkışı istenen seviyeye ve daha iyisine getirmek için değiştirilir. Buna benzer birkaç algoritma da aynı esasa dayanır [8].



Şekil 5.1. Yapay sinir ağı birimi

$$I = \sum_{i=1}^n w_i \chi_i \quad (5.1)$$

$$Y = \begin{cases} +1 & \text{eğer } I \geq T \\ -1 & \text{eğer } I < T \end{cases}$$

Rosenblatt'ın (1958) çalışma yasası:

$$W_{\text{yeni}} = W_{\text{eski}} + \beta y \chi$$

$$\beta = \begin{cases} +1 & \text{algılayıcının cevabı doğruysa} \\ -1 & \text{algılayıcının cevabı yanlışsa} \end{cases}$$

y = algılayıcının cevabı

w = ağırlık vektörü

χ = giriş vektörü

Telekomünikasyon şebekeleri, düğümleri birer algılayıcı gibi varsayılan sinir sistemleri olarak değerlendirilebilir. Burada girişlerden gelen mesajlar hizmet önceliği, uzunluk vs gibi büyüklüklerle ifade edilebilecek ağırlık vektörü gereğince yönlendirilen çıkışların, şebeke verimi üzerine etkisi gözlemlenip, ona göre ağırlıklar yani mesaj özellikleri belirlenip istenen seviyeye getirilebilir. SPC benzeri santralların son yıllardaki gelişimi ve modern yazılım programları sayesinde bu şekilde öğrenme yöntemlerini kullanan yapılar oluşturmak kolaylaşmıştır.

Yapay sinir ağları ayrıca bazı yönlendirme algoritmalarında kullanılan kapsayan ağaçların belirlenmesine ve düğümlerce bilinmesine gerek bırakmayacak şekilde kullanılabilir. Bu öngöründe yapay sinir ağlarının giriş ağırlıklarını, kullanım olasılığı gibi düşünüldüğünde, uygun çıkışın oluşmasında hangi girişin baskın kalacağı ortaya çıkarılabilir; fakat bu öğrenme süreci geniş alan şebekelerinde uzun süreceği için sakıncalı olabilir.

Geleceğin şebekeleri, değiştirilebilir çok noktalı iletişimi ile donatılacak şekilde hızlı ve ekonomik şekilde oluşturulmaktadır. Bu tip hizmetlere örnek olarak video konferansı, tele-sınıflar, dağıtılmış veritabanı uygulamaları verilebilir. ATM şebekelerde çok-ortamlı bir yayını yönlendirme, kaynak ve bütün hedefler arasında minimum uzunluklu ağaç kullanılarak gerçekleştirilir. Sonuçta, çeşitli yaklaşımlar,

şebekelerdeki Steiner Ağacı probleminin yaklaşık çözümüyle gerçekleştirilir. Rastgele Yapay Sinir Ağı (Random Neural Network-RNN), MST yaklaşımı ve ortalama uzunluk yaklaşımı ile gerçekleştirilen Steiner ağaçlarının kalitesini etkileyici biçimde arttırmıştır [12].

RNN, bağlaşma modeli oluşturmada farklı olmasına rağmen optimizasyon ve öğrenme gibi istenen fonksiyonları gerçekleştirebilir. Bu modelin temel avantajı, adım adım Monte Carlo simülasyonları kullanmadan sayısal olarak çözülebilmesidir. Böylece, diğer modellere göre hesaplama işlemleri açısından oldukça hızlı optimizasyon olanağı tanımaktadır.

5.2.3. Yönlendirmede genetik algoritma uygulamaları

Mühendislik optimizasyon problemlerinin çözümünde genetik algoritmaların kullanıldığı modeller yeni sayılır. Bu bağlamda, doğal seleksiyonun ve genetiğin mekanizmalarına dayanan bir arama yöntemi olan ve mühendislik optimizasyon problemleri açısından daha güvenilir ve kuvvetli olduğu gösterilen Genetik Algoritmalar kuramı, işleyişi ve şebeke trafik yönlendirmede optimize etme uygulamaları tanıtılmıştır [13].

Genetik Algoritmalar kısaca doğal seçim ve doğal genetik temellerine dayanan bir arama algoritmasıdır. Temel fikri doğanın kendi içinde kullandığı ana optimizasyon yaklaşımı olan “Güçlü olan hayatta kalır”dır. Ancak rastgele bilgi alışverişi sayesinde doğadakinin daha da hızlandırılmış bir halidir.

Genetik Algoritma ile her yeni nesilde yeni bir grup yapay oluşumlar meydana getirilmekte (0 ve 1’lerden oluşan diziler) ve her dizinin dayanıklılığı hesaplanmaktadır. Eğer bu diziler bir sonraki nesilde yaşayabilecek kadar dayanıklılırsa, onların genetik bilgisi daha sonraki nesillerde daha dayanıklı bireyler elde edebilmek amacıyla dayanıklı bireylerle bilgi alışverişi yapılarak sonraki nesillere aktarılır.

GA, halen kullanılmakta olan metotlardan temel olarak aşağıda anlatıldığı şekilde farklılıklar göstermektedir [14].

1. GA, parametre kümesinin kendisi ile değil parametrelerin kodlanmış şekilleri ile çalışır.
2. GA, tek bir noktadan arama yapmaz, noktalar topluluğu içerisinde arama yapar.
3. GA, optimize edilecek fonksiyonun kendisi kullanır türevlerini veya yardımcı bilgilerini değil.
4. GA, deterministik kuralları değil, olasılık kurallarını kullanır.

Basit bir genetik algoritmanın genel işleyişi bir el hesaplaması ile gösterilmiştir.

Bu hususta basit bir genetik algoritmanın temel operatörleri de açıklanmıştır. Bunlar:

1. Üreme
2. Çaprazlama
3. Mutasyon

Genetik algoritmalar doğanın ana optimizasyon kuralı olan “Güçlü olan yaşar” kuralından ilham almış olmasına rağmen sadece zekice bir uygulama değildir. Bütün işleyişi GA kuramcıları tarafından derinlemesine araştırılmış ve açıklanmış olan matematiksel olarak ispatlanmış kuramlarla desteklenmektedir. Bu bağlamda GA Temel Kuramı ve matematiksel temelleri de ele alınarak bu yaklaşımın gerçek anlamda her türlü ortamda yanılmadan evrensel en iyiye ulaştığı ispatlanmıştır.

Genetik algoritmalar, John Holland ve meslektaşları tarafından 1960’larda Michigan Üniversitesi’nde geliştirilmiştir, hedefleri şunlardır:

1. Doğal sistemlerin uygulanabilir süreçlerini belirlemek ve basitçe açıklamak.
2. Doğal sistemlerin önemli mekanizmalarını içeren yapay sistem yazılımları tasarlamak [14].

Bu çalışmalar ışığında, yakın zamanda Genetik Algoritmalar telekomünikasyon şebekelerinin topoloji tasarımı ve trafik yönlendirmesi doğrultusunda kullanılmaktadır [9,10]. Sayısal çapraz bağlama şebekesine (Digital Cross Connect Network) yerleştirilenecek ATM şebekesinin tasarımı için önerilen GA, diğer heröstik yaklaşımlara kıyasla şebeke performansında ve hesaplama hızında daha iyi sonuçlar vermiştir [9]. Geniş bandlı ATM şebekelerinde, hücre-tabanlı bağlaşma kapasitesi genellikle DCS şebekelerinin yapısına bağlıdır. DCS, ATM bağlaşmada ve şebeke tasarımında omurga yapıyı teşkil eder. Konfigürasyon, bu omurga yapı üzerinden hizmet kanalları yerleştirilerek gerçekleştirilir. Burada amaç, verilen DCS topolojisi; omurga yapının hat kapasitesi, bağlaşma kapasitesi ve istenen trafik akış şartları altında yerleştirilecek ATM topolojisi, ATM şebekesinin yönlendirilmesi ve link kapasiteleri değiştirilerek ortalama paket gecikmesini en aza indirgenecek algoritmayı gerçekleştirmektir. Genetik Algoritma uyarlaması; hat kapasitesi, ATM bağlaşma kapasitesi ve akış kapasitesi sınırlamaları altında verilen omurga topolojisinde düğümler arası yollardan oluşan yönlendirmelerin hiyerarşik sırasından meydana gelmiş kromozomları kullanır. Bu kromozomlardaki herbir parça (gen), bu parçanın temsil ettiği düğüm çifti arasındaki mevcut yolları düğümler üzerinden geçerse “1” biti, aksi taktirde “0” biti veren dizilerden meydana gelmiştir. İstenen trafik şartları için özel bir kromozom dizisi denenir. Eğer sınırlamalar sağlanmazsa GENOCOP programı yardımıyla çaprazlama ve mutasyon ile kromozomlar işleme tabi tutulur ve uygunluk fonksiyonu yüksek olanlar seçilir. Burada çaprazlama, kromozomların en iyi parçaları yeni bir kromozomda biraraya gelecek şekilde gerçekleştirilir ve 0.05 oranında parçalar mutasyona tabi tutulur. Beş düğümlü omurga topolojisi üzerinden Pentium-100 kullanılarak gerçekleştirilen bir örnekte, klasik yöntemle göre ortalama gecikme daha düşük (1.373 sn →1.371 sn) ve hesaplama süresi daha kısa (30 dak→32.09 sn) çıkmıştır [9].

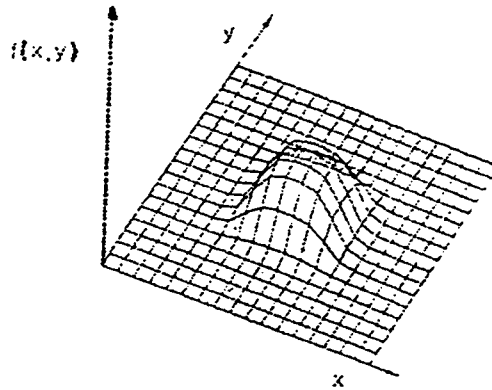
GA'nın uygulanmasındaki anafikir, güçlü ve gerektiğinde çeşitli dış çevre şartlarında verimi ve etkinliği dengeleme, bizim doğada yakın bir şekilde tanık olduğumuz gibi, başarılı bir yaklaşımla veri tabanını taramaktır. Ancak, diğer klasik optimizasyon ve arama tekniklerinde nadiren bulunan kendini yenileme, kendini yöneltme ve yeniden üretim kurallarını GA'ya uyarlamak zorunludur.

Sonuç olarak, doğanın kuvvetli bir performansı yaptığı ve devamlı yapacağı kuralına ulaşırız. Böylece, GA'nın altında yatan biyolojik fikrin doğa çalışmalarından çıkarılabileceği açıktır. Ancak, bütün bunları kabul etmemize rağmen, güvenilirliği ispatlanmamış teknikler yeterli değildir. Bununla beraber, genetik algoritmaların kuramsal ve deneysel olarak kompleks uzaylarda en iyi nokta aramasında başarılı oldukları ispatlanmıştır.

Literatürde üç temel tipte arama metodu vardır: Hesaplama tabanlı arama, listeleme (enumerative) ve rastgele arama.

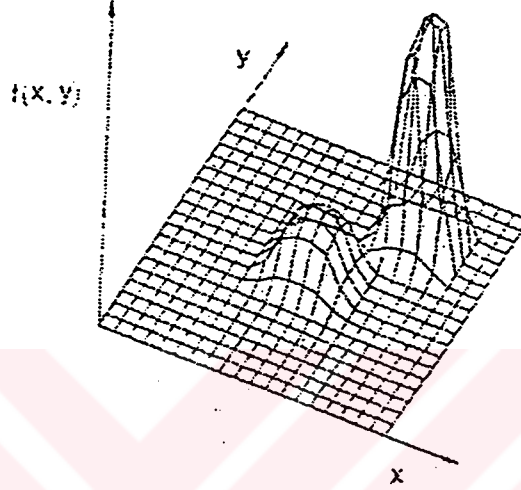
Hesaplama tabanlı metotlar en çok kullanılan metotlardır. Doğrudan ve dolaylı olmak üzere iki ana sınıfa ayrılırlar. Dolaylı metotlar yerel ekstremumu, hedef fonksiyonunun diferansiyelini sıfıra eşitleyerek nonlineer eşitlik kümesinden bulmaya çalışır. Bu, ekstremum noktaları bulan basit hesaplama biçiminin çokboyutlu genelleştirmesidir.

Eğer Şekil 5.2'de verilen sürekli ve yumuşak geçişli fonksiyon optimize edilirse, olası tepeleri bulmak için odaklanma veritabanının en iyileri olduğu söylenen eğimin her doğrultuda sıfır olduğu bu noktalara yöneltilmelidir. Doğrudan metodlarda, yerel en iyi yerel diferansiyelle ilgili doğrultuda hareket etmekle aranır. Bu, fonksiyonda yerel en iyiyi bulmak için izin verilen basamaklı doğrultuda fonksiyonu tırmanmayı tanımlayan “Tepeye Tırmanma”nın (Hill Climbing) ana fikridir. Her ne kadar bunlar en çok bilinen en iyi tanımlama teknikleriyse de aşağıda gösterilen teknikler kadar güçlü değildir.



Şekil 5.2. Tek-tepeli fonksiyonda hesaplama tabanlı arama

Herşeyden önce bahsedilen iki metot da yerel arama metotlarıdır, yani giriş noktalarının civarında en iyi arama için iyi çalışırlar. Bununla beraber Şekil 5.3'teki gibi bir veritabanında arama yapılıyorsa, giriş noktalarını alçak tepenin yakınında vermenin, yüksek tepeyi elemekten kaçırmaya sebep olacağı açıkça görülebilir. Bunun anlamı, genel en iyi için şansa büyük ölçüde bel bağlandığıdır.



Şekil 5.3. Çok-tepeli fonksiyon hatalı arama

Bundan başka, hesaplama tabanlı aramalarda görüldüğü gibi, bunlar türevlerin varlığına bağlıdır; yani süreklilik zorunludur. Yine de, gerçek dünyadaki optimizasyon problemlerinde, süreksizlikler, çok tepelilik durumları ve gürültülü yapı dolayısıyla hesaplama tabanlı metotla çalışılması çok zordur (Şekil 5.4).

Yukarıda bahsedilen ATM şebekesi tasarımında da söz konusu yerleştirilen link ve bağlaşma kapasiteleri, temel veri hızlarının (150 Mbit/sn gibi) tam katları olan ayrık değerlerdir. Bu zorluk GA kullanılarak aşılmıştır [9].

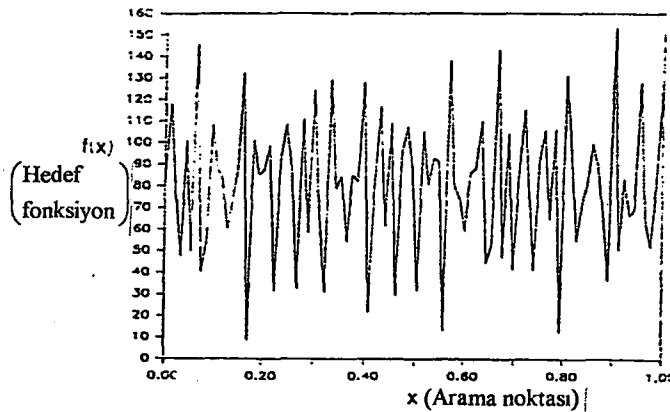
Listeleme metodunda arama algoritması, sınırlı arama uzayında ya da ayrık-sınırlı arama uzayında, uzayın her noktasında hedef fonksiyon değerlerine bir defada bakar. Bu metot olasılıkların sayısı az olduğunda çekicidir, fakat bu basit nedenle genellikle verimlilik açısından elverişsizdir. Çünkü, tasarımcısı Bellman'ın (1961) da belirttiği

gibi, birçok pratik uzay genellikle çok geniştir ve bu metot gerçek problemler için uygun değildir.

Rastgele arama algoritmaları yukarıda bahsedilen metotların etkisizliği ortaya çıkınca çok popüler olmuştur. Fakat, uzun çalışma süreçlerinde adını tekzip edencesine, listeleme metotlarının sağladığı etkinliği aynen sağladığı gözlemlenmiştir. Diğer popüler arama metodu ise, minimal enerji durumlarının aranması için rastgelelik sürecini rehber edinen “Uyarlanmış Tavlama”dır. (Simulated Annealing) [13].

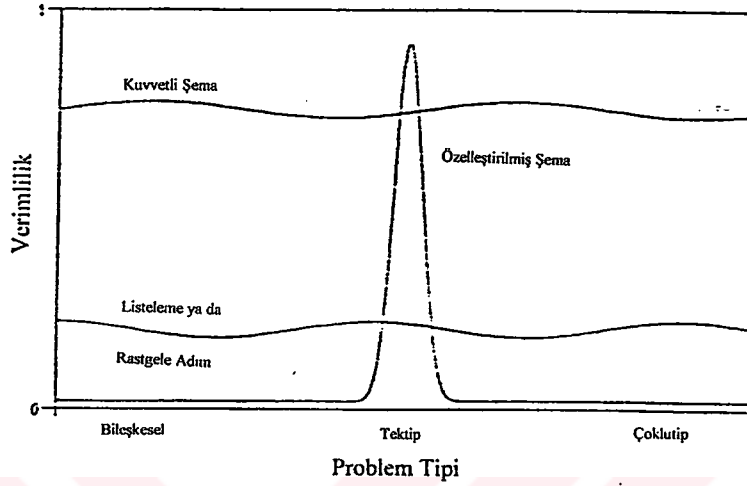
Genetik Algoritma ve Uyarlanmış Tavlama Algoritması şebeke topoloji tasarımında basitçe karşılaştırılabilir. Genelde link yerleştirme optimizasyonu kullanılmasına rağmen [11], gerçek bir bağlaşma organı, link ve trafik kümesi için bağlaşma organlarının sayısı ve yerleştirilmesinin temel alındığı çözümler vardır. Ortalama GA çözümleri, ortalama SA çözümlerinden daha az masraflı olmaktadır. Yapılan bir çalışmada ortalama GA çözümü, ortalama SA çözümünden %5 az masraflı sonuç vermiştir [10]. Burada GA topluluğu “Obur Düşüş” herotiğinden üretilen en az pahalı n çözümden oluşmaktadır.

Bu karşılaştırmada sonuç olarak, bahsedilen metotların güçlü olmadığı açıktır; fakat bu onların kullanışsız olduğu anlamına gelmez. Bu metodların iyi çalıştığı ispatlanan çok sayıda uygulama da mevcuttur. Şekil 5.5, geleneksel metotların verimliliklerini problem tipine göre göstermektedir [15].



Şekil 5.4. Gürültülü, süreksiz ve dengesiz bir fonksiyon örneği

GA bilgisayar destekli çalışmaya uygun, güçlü, temel olarak arama uzayının süreklilik, türevlerinin varlığı, çok-tepelilik vs gibi varsayımlarıyla sınırlanmamış bir algoritmadır [16].



Şekil 5.5. Arama metotlarının verimlilik karşılaştırması

Daha önce de verildiği gibi, genetik algoritmalar bazı temel noktalarla geleneksel tekniklerden farklıdır. Öyle ki, GA'lar bazı sonlu alfabeler üzerinden sonlu uzunlukla diziler halindeki kodlara optimizasyon probleminin doğal değişken kümesi olarak ihtiyaç duyar. En çok kullanılan teknik bu tezde de kullanılan değişkenleri ikili kod dizileriyle temsil etmektir. İkili kodlamada değişken değerleri 0 ve 1'lerden oluşan ikili değerlere çevrilir [15].

5.2.3.1. Basit genetik algoritma

Bu bölümde, basit genetik algoritmaların mekanizması elle hesaplama yoluyla basit bir örnek üzerinde açıklanacaktır. Öncelikle, problemin matematik modelinin tanımı sonradan giriş nesili olarak adlandırılan değerleri veren rastgele sayı üreticini zorunlu kılar [15].

Bu noktada basit genetik algoritmanın ana operatörlerini, tanımlayabiliriz:

1- Yenileme (üreme)

2- Çaprazlama

3- Mutasyon

Yenileme, ayrı dizilerin hedef fonksiyonu f_e göre (biyologlar GA terminolojisindeki gibi bu fonksiyonu uygunluk fonksiyonu olarak tanımlar.) kopyalama sürecidir. Diyebiliriz ki, hedef fonksiyonu maksimize etmek istiyoruz, nesildeki uygunluğu arttırarak dizileri uygunluk değerlerine göre kopyalarız, böylece yüksek uygunluk değerindeki dizilerin, GA’da döller olarak adlandırılan çocukları benzer karakterde olmak üzere diğer nesillerde hayatta kalma olasılığı yüksektir. Başka bir ifadeyle GA’nın ana fikri “En Uygunun Yaşaması”dır.

Giriş topluluğunun aşağıda verildiği gibi dört parçadan oluştuğunu varsayalım:

01101

11000

01000

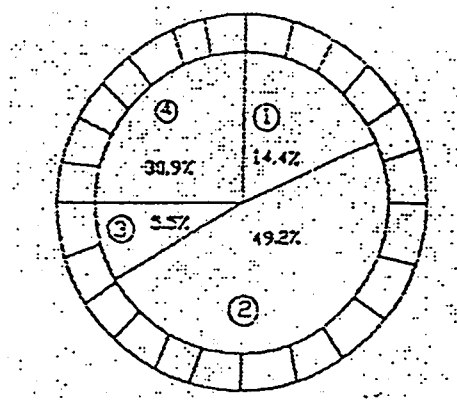
10011

Üreme operatörü birçok şekilde elde edilebilir. Tezde de kullandığımız üzere, en kolay yol, “hilesiz rulet tekeri” yöntemi denilen yöntemdir ve o andaki her dizinin hedef fonksiyonları toplamı bütününde oluşan sayı tekeri üzerinden üreme gerçekleştirilir. Diziler aşağıdaki gibi gösterilebilir. (Tablo 5.1).

Tablo 5.1. Basit diziler ve Uygunluk değerleri [15]

No	Dizi	Uygunluk	% Toplam
1	01101	169	14,4
2	11000	576	49,2
3	01000	64	5,5
4	10011	361	30,9
Toplam		1170	100,0

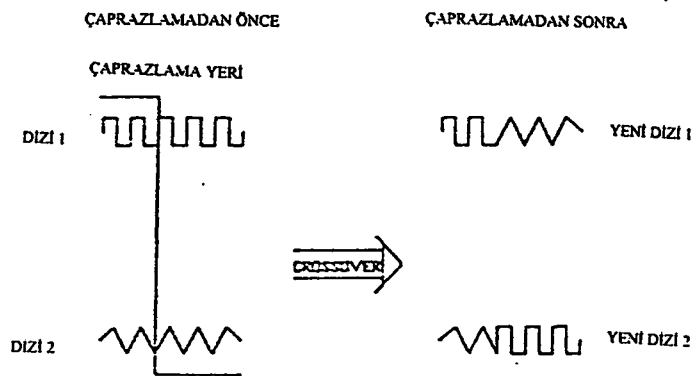
Dört dizi üzerinden uygunluk değerlerini toplarsak, 1170 toplamı elde edilir. Bu nesile karşılık gelen rulet tekerleği Şekil 5.6’da gösterilmiştir.



Şekil 5.6. Basit Rulet tekeri örneği

Diğer nesilleri üretmek üzere rulet tekeri döndürülür ve uygunluk performanslarına göre rulet tekeri üzerindeki alana karşı düşen diziler seçilerek diğer nesil oluşturulur. Genel olarak üretilen veya seçilen dizilerin sayısı toplulukta değişmez. Sonra bu seçilen diziler daha fazla genetik operasyon için havuza konur.

Üretimden sonra, diziler çaprazlama (crossover) operatörüne maruz kalırlar. Şekil 5.7’de basit bir çaprazlama şeması gösterilmiştir.



Şekil 5.7. Basit çaprazlama şeması.

Önce çaprazlamaya tabi tutulacak diziler çaprazlama olasılığına göre seçilir. Sonra aşağıdaki gibi çaprazlanırlar.

1. k gibi bir tamsayı pozisyonu düzgün rastgele biçimde 1 ve $(l-1)$ arasında seçilir.
2. $(k+1)$ ve 1 arasındaki pozisyonlardaki karakterleri değiştirilerek iki yeni dizi oluşturulur [17].

Örneğin; A_1 ve A_2 'nin çaprazlama için seçildiği varsayılın, ayrıca rastgele seçilen çaprazlama noktası $k=4$ olsun (L simgesiyle gösterilmiştir).

$$A_1=0110 L1$$

$$A_2=1100 L0$$

Bir sonraki nesilde A_1^1 ve A_2^1 olarak gösterilen yeni diziler aşağıdaki gibi olur.

$$A_1^1=01100$$

$$A_2^1=11001$$

Basit GA'da mutasyon operatörü bir dizi pozisyonundaki değeri rastgele değiştirmeye yarar. İkili kodlamada bunun anlamı basitçe 1'in 0'a değişimi ya da tersi demektir. Mutasyonun yardımıyla dizi uzayında rastgele erişim yapılır. Mutasyon oldukça gereklidir, çünkü çaprazlama ve üretim her ne kadar arama uzayını ve parçalarını kombine etmekteyse de bazı genetik materyallerin potansiyelini değerlendirmede eksik kalabilir. Yapay genetik sistemlerde mutasyon operatörü geri dönülemez kayıpları engeller. Üretim ve çaprazlama ile birlikte kullanıldığında mutasyon veritabanını etkin olarak arama açısından bir güvenlik politikası olarak rol alır.

Mutasyon GA'da ikincil bir rol oynadığından, için basit GA çalışmalarında iyi sonuçlar elde edebilmek için mutasyon transferinin gerçekleşme frekansı binde bir civarında olmalıdır, aksi halde diğer operatörlerin en iyiye yakın çözümleri diğer nesillerde kolaylıkla deforme olabilir. Mutasyon oranı doğal topluluklarda daha düşük olduğundan GA adaptasyonlarında bu göz önünde bulundurularak ikincil bir önem verilmiştir.

Basit GA'yı canlandırmak için bir kısmi optimizasyon problemi uyarlanabilir. λ değerinin 0 ile 31 arasında değiştiği şartı altında $f(\chi)=\chi^2$ fonksiyonunu maksimize etme problemini ele alalım. Önce problemimizin değişkenlerini bazı sonlu-uzunluklu dizileri kodlayarak oluşturalım. Bu örnek için χ basitçe 5 hanelik ikili dizi şeklinde kodlanır. Değerleri, ikili dizi şeklinde kodlamak bu değeri ikili tabanda aritmetik olarak aşağıdaki örnek gibi oluşturmak anlamına gelir.

$$1.2^4+0.2^3+0.2^2+1.2^1+1.2^0=16+2+1=19$$

Beş tane ikili dijit, bilgisayar değerlendirmesini oluşturmak için oldukça kolaydır, böylece 0 (00000) ve 31 (11111) arasındaki değerleri elde ederiz.

Başlangıçta, rastgele giriş topluluğu seçilir. Daha önceden gösterdiğimiz topluluğu seçtiğimizi varsayalım. Üretim ile nesil başlar. Önceden gösterildiği gibi her parçanın uygunluğu hesaplanır. Her parçanın hayatta kalma olasılığı, bu parçanın uygunluk değerinin o andaki topluluğunun uygunluk değerleri toplamına oranlanarak hesaplanır. Eşleştirme havuzunun bir sonraki nesilini ağırlıklaştırılmış rulet tekerini dört defa çevirerek seçeriz (Toplulukta dört dizi var). Bütün hesaplamalar Tablo 5.2'de verilmiştir.

Elle hesaplanmış GA tablosunun sonuçlarını karşılaştırsak, en iyinin daha fazla kopya, ortalamanın aynı sayıda kopya çıkarmasını, en kötünün ise ölmesini bekleyeceğimizi elde ederiz.

Bütün bunlardan sonra, çaprazlama iki aşamada gerçekleşir:

1. Diziler rastgele çiftleştirilir.
2. Rastgele seçilen noktalardan eşleşmiş dizilerin parçaları değiş tokuş edilir.

Son operatör olan mutasyon bit-bit tabanında gerçekleşir. Mutasyon olasılığının 0,001 olduğunu kabul edelim. 20 tane transfer edilen bit pozisyonundan

$20 \times 0,001 = 0,02$ tane bitin verilen nesilde mutasyona uğramasını bekleriz. Sonuç olarak bu nesil için mutasyon gerçekleşmez.

Üretim, çaprazlama ve mutasyonu takiben nesil ilerlemeye hazırdır. Bu, yeni dizileri deşifre eden yöntemle ve bu değerler üzerinden uygunluk fonksiyonlarını hesaplayarak yapılır. Basit bir nesil sonucu Tablo 5.2’de gösterilmiştir.

Elle hesaplanmış GA tablosunu analiz edersek, maksimum ve ortalama performansların yeni nesilde nasıl oluştuğunun altını çizmek gerekir. Şu var ki, topluluğun ortalama uygunluğu 293’ten 493’e bir nesilde çıkmıştır. Böylece maksimum uygunluk bir nesilde 576’dan 729’a çıkmıştır.

Tablo 5.2. Elle Uyarlanmış GA [15]

Dizi No.	Giriş Populasyonu (Rastgele Üretilmiş)	x Değeri	f(x)	p seçimi	Beklenen Değer	Gerçek Değer (RuletTekerinden)
1	01101	13	169	0,14	0,58	1
2	11000	24	576	0,49	1,97	2
3	01000	8	64	0,06	0,22	0
4	10011	19	361	0,31	1,23	1
Toplama			1170	1,00	4,00	4,0
Ortalama			293	0,25	1,00	1,0
Maks.			576	0,49	1,97	2,0

Tablo 5.3. Elle Uyarlanmış GA [15]

Eşleşen Çiftler	Eşler (Rastgele Seçilmiş)	Çaprazlama Yeri (Rastgele Seçilmiş)	Yeni Topluluk	x değeri	f(x)
01101	2	4	01100	12	144
11000	1	4	11001	25	625
11000	4	2	11011	27	729
10011	3	2	10000	16	256
					1754
					439
					729

Sürekli olarak biyolojiden bazı terminolojiler kullanıldığı için, biyolojik terminoloji GA'daki anlamına uygun düşen karşılıklarıyla gösterilmiştir. Fakat şu belirtilmelidir ki, GA terminolojisi doğal ve yapay terminolojinin bazı karışımlarını kullanır.

Yapay genetik algoritmalarındaki diziler, biyolojik sistemlerdeki kromozomlar benzer. Doğal sistemlerde bir ya da daha fazla kromozom organizmanın toplam genetik dizilimini ve işlemlerini oluşturmak için biraraya gelir. Doğal genetik sistemde toplam genetik birikim genotip olarak adlandırılır. Yapay genetik sistemlerde toplam dizi birikimleri yapı olarak adlandırılır. Doğal sistemlerde organizmanın çevresiyle etkileşimli toplam genetik sistemine fenotip denir. Yapay sistemlerde, kısmi değişken kümesi oluşturan ya da çözüm uzayında bir nokta ortaya çıkaran yapı deşifresi bunu karşı düşer [18].

Yapay genetik sistem tasarımları sayısal ve sayısal olmayan kodlama alternatiflerine sahiptir. Doğal terminolojide, kromozomlar allel adını alan bazı sayısal değerlerden müteşekkil genlerden oluşmuştur. Genetikte, bir genin pozisyonu (locus) genin fonksiyonuyla açıkça tanımlanır. Örneğin, bir hayvanın göz rengi geni vardır, pozisyonu 10 ve alleli (değeri) mavi gözdür. Yapay genetik aramada diziler değişik değerler alan tarayıcı ya da özelliklerden oluşmuştur. Özellikler dizi üzerinde farklı pozisyonlarda bulunabilir. Örneğin 10000 dizisi 10 tabanında 16 olarak deşifre edilir çünkü 1 biti 16 değeri konumundadır.

5.2.3.2. Genetik algoritmalar kuramı

Genetik algoritmaların güvenilirliği ve rastlantusal özelliğin ortaya çıkardığı deterministik metotlarla karşılaştırma zorluğu, yapılan kuramsal çalışmalar sonucunda aşılmıştır. Böylece, tezde ele alınan algoritmanın daha önceki bölümlerde konu edilen diğer üç standart yönlendirme algoritmasıyla karşılaştırma imkanı ortaya çıkmıştır.

Bu bölümde genetik algoritmanın matematiksel ifadesi üzerinde durulmuştur. Bir topluluktaki diziler arasındaki yapısal benzerliklerin tarif etmek için “şema” terimini

tanımlanmıştır[14]. Şemada bir dizi parçasındaki her yerde 1 biti 1 ile, 0 biti 0 ile, 1 ya da 0 biti * ile gösterilir. Örneğin *10*11 şemasının elemanları aşağıdaki gibidir.

010011

010111

110011

110111

Beş uzunluklu bir dizi için pozisyonları 1,0 ya da * olmak üzere $3^5=243$ olası şema vardır. Şemaları kullanarak GA tarafından gerçekte yürütülen işlemleri kontrol yoluyla aramaya yön verilebilir. Üretimin etkisi en yüksek uygunluk değerini seçmesiyle bellidir, fakat kendisi tasarım uzayında yeni noktalar oluşturmadığı için veri uzayının her noktasını elde edemez. Çaprazlama yoluyla diziler arasında bir noktadan itibaren parça değiş tokuşu olur. Eğer * simgesini kullanarak yüksek değerlerde uygunluk sağlayan dizilerin belirli pozisyonlardaki ortak bitlerini göz önünde bulundurulursa, çaprazlamanın bu ortak blok yapıyı bozmayacak noktalardan yapılması bir sonraki nesile daha yüksek uygunluk değerlerinde dizilerin aktarılması sağlanır. Bunun için değiştirilebilir kısımları * ile simgeleyen şema tabloları üzerinden işlem yapılır. Eğer, ortak değerli pozisyonlar arasındaki maksimum bit uzaklığı artarsa bunların çaprazlama ile bozulma olasılığı da artar. Sonuç olarak şu GA teorisi tanımlanır: “Yüksek uygunlukta, kısa tanımlanmış uzunlukta (yapıcı bloklar) şemalar ki ortak değerli pozisyonlar arasındaki maksimum uzunluk değeri düşüktür; nesilden nesile en iyileri içeren örnekleri eksponansiyel artışla yayar.” [19].

Bir şemanın seviyesi ($S(H)$) sabit pozisyonlarının sayısını temsil eder. Mesela $H=011*1^{**}$ şemasının seviyesi 4’tür. Simgesel olarak dizi uzunluğu (l) 7 olsun.

$$S(011*1^{**})=4$$

H şemasının tanımlayıcı uzunluğu ($\delta(H)$), ilk ve son sabit pozisyonlar arasındaki uzaklığı temsil eder. $011*1^{**}$ şemasının tanımlayıcı uzunluğu ($\delta(H)$)=4’tür. Çünkü son sabit pozisyon 5. bittir. 0^{*****} şemasının ilk ve son sabit pozisyonları 1.bit

olduğunda $\delta=0$ 'dır. Şemalar, üretim ve genetik operatörlerin topluluktaki yapıcı bloklar üzerindeki etkilerine göre önem taşırlar.

Bir H şemasının t anında topluluktaki eleman sayısı m olsun.

$$m=m(H,t) \quad (5.2)$$

Bu durumda aşağıdaki eşitlik ispatlanmıştır. (Goldberg, 1989):

$$m(H,t)=m(H,t_0) (1+c)^t \quad (5.3)$$

Burada c, H şemasının tüm elemanlarının ortalama uygunluk değerinin topluluğun ortalama uygunluk değerinin üzerinde olması durumunda pozitif aksi durumda negatif değer alır. Buradan, üretimin şemalar üzerindeki etkisinin, eksponansiyel olarak arttırdığı ya da azalttığı şeklinde olduğu belirlenir.

Yine P_c , çaprazlama olasılığı olmak üzere aşağıdaki eşitsizlik ispatlanmıştır[15]:

$$m(H, t+1) \geq m(H,t) (1+c)^t [1-P_c ((\delta(H)/(1-1)))] \quad (5.4)$$

Bu ifadeden ortalama üstü performans sergileyen ve kısa tanımlanmış uzunluklu şemalar eksponansiyel artış şeklinde örneklenirler.

(5.4)'teki ifadede mutasyonun etkisi göz önüne alınırsa aşağıdaki sonuç çıkar (P_m mutasyon olasılığı):

$$m(H,t+1) \geq m(H,t) (1+c)^t [1-P_c (\delta(H)/(1-1))-s(H).P_m] \quad (5.5)$$

Mutasyonun eklenmesiyle sonuç biraz değişmiştir. En genel biçimde GA uygulamalarında şemalar için şu ifade kullanılır: Kısa, düşük seviyeli, ortalama üstü şemalar sonraki nesillerde eksponansiyel olarak artarlar. Bu genetik algoritmaların temel kuramıdır [15].

Bu sonuçlar ışığında GA hakkında yaklaşımlar ortaya konabilir. Başlangıçta üretilen rastgele seçilmiş toplulukta veya daha sonraki topluluklarda üretilen yukarıdaki özellikleri sağlayan şemaların hayatta kalma olasılıkları artacağı gibi kendinden sonra daha yüksek uygunlukta parçalar oluşturma şansı da artar ve neticede son nesil amaca iyice uyumlaştırılmış hale getirir. Bu ise diğer algoritmalarla göre deterministik kuralların getirdiği zorlukları ve hesap karmaşıklığını aşan GA'nın aynı zamanda sonuca ulaşmada kör atışı bir yaklaşım olmadığı ortaya konularak burada belinen bazı kuramsal yanılgılar aşılmıştır. Pratik uygulamalarda bunun oldukça faydalı sonuçlar verdiği görülmüştür [20,21].

5.2.3.3. GA'ların diğer yönlendirme algoritmalarıyla karşılaştırılması

Şebeke yönlendirme algoritmaları esas olarak en iyi çözümü üretmeyi amaçlar. Bunun için, graf teorisine dayanan en kısa iz algoritmaları geliştirilmiştir. Bu algoritmaların kullanıldığı paket bağlaşmalı şebekeler ister merkezi ister yalıtılmış yönlendirmeli şebekeler olsun en iyi bağlaşma yollarının bulunması bu algoritmaların uygulanmasıyla mümkün olmaktadır. En kısa izin bulunması için üç standart algoritma mevcuttur. Bölüm 4'te bahsedilen bu algoritmalar, deterministik kurallara dayalı, matematiksel hesaplamalarla en iyi çözümü ayrık olasılıklar uzayında bulmak için geliştirilmiştir. Bunun yanında, bu yöntemler, geniş alan şebekelerinde, merkezi yönlendirmede olduğu gibi genel hesaplamaları oldukça karışık olmakta ve uzun bir süreç gerektirmektedir. Ayrık olasılık uzayının büyük sayıda eleman içermesi durumunda bu algoritmalar hiç de pratik olmamaktadır. Yazılım (Software) programlarına uyarlanabilirliği, metotların her olasılığı sınavarak elemesi esasına dayandığından oldukça gelişmiş modeller gerektirir, ayrıca şebekenin değişen şartlarına kısa sürede cevap vermesi mümkün olmayacak hale gelir.

Genetik algoritmanın en kısa iz probleminde diğer algoritmalarla göre avantaj ve dezavantajları vardır. Herşeyden önce genetik algoritmalar, geliştirilmelerinin öncelikli sebeplerinden olan çok fazla sayıda eleman içeren uzaylarda arama

yapmasının diğer deterministik algoritmalara göre daha pratik olması vesilesiyle geniş alan şebekelerinde faydalı olabilir. Ancak, düğüm sayısının az olduğu dar çaplı şebekelerde (LAN gibi) bu algoritma kullanım tasarrufu sağlamadığı gibi, deterministik algoritmaların kolayca ulaşabileceği verimi yakalama şansı da azalır. GA'ların dayandığı esasların diğer algoritmalara göre basit olması sebebiyle yazılım kontrollü şebekelere kolaylıkla uygulanabilir. Deterministik algoritmaların hesaplamalarının uzunluğuna karşı GA'ların hesaplama içeriği sadece uygunluk fonksiyonuyla sınırlıdır.

GA kullanılarak yönetilen şebekelerde deterministik algoritmaların sağlayacağı kesin sonuçlara karşılık, en iyiye yakın sonuçların şebeke verimi açısından karşılaştırılması, kullanıcı hizmet kalitesiyle yakından ilgilidir. Eğer en iyi çözümün sağladığı durumdan belli miktarda sapmalar göz ardı edilirse bu sorun oluşturmaz. Ne var ki, veri iletim hızının çok önemli olduğu uygulamalarda bu sorun ya GA'nın nesil sayısı artırılarak trafik akış değişimine karşı daha az duyarlı hale getirilerek; çok sık dalgalanma gösteren trafik durumlarında ise, değişen şartlara uymakla, en iyi yola nisbeten uzak çözümler elde etmekle kaybedilen verimlilik artırılarak çözülür.

6. GA İLE YÖNLENDİRMENİN BİLGİSAYAR SİMÜLASYONU

6.1. Sistemin Tanımlanması

Belirli iki düğümü arasındaki mevcut yollardan en kısa izi bulmak amacıyla TURBO C 3.0 programlama dili ile yazılan programda, paket bağlaşmalı bir şebeke tanımlanmıştır. 100 düğümden oluşan bu şebeke 10'ar düğümlük bölgelere ayrılmış ve bu bölgeler ve bölgeler içindeki düğümler 1'den 10'a kadar numaralandırılmıştır. Her bölgedeki düğümler 2 tane kendi bölgesindeki düğümlerle ve 4 tane de bir sonraki bölgedeki düğümlerle olmak üzere 6 hatta sahiptir. En son bölgedeki düğümler ise bu bölgedeki son düğüme birer tane hatta sahip olup kendinden önceki bölgeden gelen hatlarla başka bölgeye bağlıdır. Şebeke yapısı program çalıştırıldığında rastgele belirlenir.

Programın amacı birinci bölgedeki birinci düğümden, onuncu bölgedeki onuncu düğüme olan yollar arasında verilen dal etiketi fonksiyonu bağlamında en kısa izi bulmaya çalışmaktır. Bunun için program başında verilen bölge içi hatları kullanma olasılığı nisbetince her bölge kendi içinde bu olasılıkla hat kullanarak bir sonraki bölgeye bağlayan hatlar üzerinden en iyi yolu teşkil etmeye çalışır.

Her bölgenin kendi içindeki hat ve diğer bölgeye bağlandığı hattan müteşekkil bir yolun meydana getirdiği safha dokuz kez tekrarlandığı ve son bölgedeki bölge için, bağlantılar belirli olduğu için her bir olası yol bir bit bölge içi kullanım durumu ve iki bit bölgelerarası kullanım durumu olmak üzere, her aşama, üç bitten oluşan bir kodla temsil edilir. Burada bölge içi kullanımın olmadığı durumlarda bu bit "2" değerini alır ve diğer hallerde bir safhanın her biti 0 ve 1'lerden oluşan ikili dizilerdir. Dokuz safhadan müteşekkil yol, bu durumda 27 ($=3 \times 9$) bitten oluşmaktadır. En son bölgedeki iç hatlar hesaplamaya programın sonunda katılır.

Program başında belirlenen şebeke topolojisi uyarınca bu 27 bitlik dizi, başlangıçta belirlenen iki düğüm arasındaki yolları tanımlamaya yarar.

Programda her bir dalın temsil ettiği link için iletim kapasitesi, iletim kalitesi ve uzaklık rastgele belirlenir. İletim kapasitesi (C) bölge içi hatlar için 192 kbit/sn olup, bölgelerarası hatlar için 1544 kbit/sn ya da 2048 kbit/sn olmaktadır. Hat uzunlukları bölge içi hatlar için 5-50 km arası, diğer durumda 50-100 km arasındadır. İletim kalitesi olarak tanımlanan faktör ise, 1-1000 değerleri arasında değişmektedir. Belirlenen bu değerler üzerinden her bir dalın etiket değeri program başında verilen kombine fonksiyon uyarınca belirlenir. Bu fonksiyon (6.1)'deki gibidir.

$$f(k_1, k_2, k_3) = k_1 * T + k_2 * (1/\text{ilekal}) + k_3 * \text{uzaklık} \quad (6.1)$$

Burada k_1 , k_2 , k_3 katsayılar olup program başında istenir. T, ortalama paket gecikmesi, “ilekal” iletim kalitesidir. T paket gecikmesi (6.2) denklemi uyarınca tanımlanmış olup, değeri her hesaplama periyodu boyunca λ parametresinin (0-0.9 μ C) değerleri arasında rastgele değişmesiyle belirlenir. Böylece T değeri ortalama 4-425 ms arasında değişir. Bilgisayar programı sadece bir hesaplama periyodu boyunca işletilmiştir. Burada μ , paket/bit cinsinden bit başına ortalama paket uzunluğudur. Bir paketin ortalama uzunluğu ise 8192 bit/paket'dir.

$$T = \frac{1}{\mu C - \lambda} \quad (6.2)$$

Program başında verilen çaprazlama ve mutasyon olasılıklarına göre başlangıçta rastgele belirlenen 100 adet 27 bitlik dizi her bir nesilde GA işlemlerine tabi tutularak uygunluk fonksiyonları hesaplanır ve bu fonksiyon değerleri oranınca rulet tekeri yöntemiyle bir sonraki nesil için seçilir. Her bir neslin sonunda, 100 diziden müteşekkil topluluğunun istatistiki sonuçları verilir. Ayrıca erken safhalarda meydana gelen en iyiye yakın çözümlerin ileriki aşamalarda değişime uğramaması için bir önceki neslin en yüksek uygunluktaki dizisi büyükse, bir sonraki

jenerasyonun en küçük uygunluktaki dizisiyle değiştirilerek bu algoritmanın en zayıf noktası bir çeşit ileri GA yöntemi olan “elitizm” ile çözümlenmiştir.

Uygunluk fonksiyonu, dal etiketi fonksiyonunun çarpmaya göre tersi alınarak belirlenmiştir. Böylece en küçük dal etiketleri toplamından müteşekkil yol en yüksek uygunluk değerine sahip olur.

Verilen maksimum nesil sayısı gereğince en son nesildeki en yüksek uygunluktaki dizi seçilerek deşifre edilir ve böylece en iyi yol bulunmuş olur.

6.2. Katsayıların Belirlenmesi

Ekte verilen örnek program için belirlenen katsayılar aşağıdaki gibidir.

$k_1=1, k_2=0, k_3=0$

İçhat kullanma olasılığı = 0,5

Çaprazlama olasılığı = 0,987

Mutasyon olasılığı = 0,0075

Azami nesil sayısı = 50

Bu şekilde belirlenen katsayılar uyarınca en yüksek uygunluktaki dizi, en kısa ortalama paket gecikmesinin olduğu yolu belirler.

Programdaki kısaltmalar aşağıdaki gibidir;

Hn: Hiyerarşi no (Bölge no)

Dn : Derece no (Bölge içi düğüm no)

İh: İçhat

Dh: Dışhat

6.3. Programın Akış Diyagramı

Verilen giriş değerleri uyarınca programın başında şebeke topolojisinin belirlenmesi işlemi gerçekleştirilir. Tablo halinde ekte verilen şebeke topolojisinde ilk sütundaki değerler bölge no.yu, ikinci sütundaki değerler ise o bölgedeki düğüm no.yu belirtir.

Üçüncü ve dördüncü sütundaki değerler o düğümün kendi bölgesinde bağlı olduğu düğüm no.ları, son dört sütundaki değerler ise o düğümün bir sonraki bölgede bağlı olduğu düğümleri göstermektedir. Şebeke topolojisinin kısmi gösterimi Şekil 6.1’de gösterilmiştir.

İkinci aşamada 27 bitlik dizilerden müteşekkil 100 adet giriş dizisi, her bir biti rastgele belirlenecek şekilde üretilir. Burada başlangıçtan itibaren üç atlayarak ilerleyen her sıradaki bit, iç hatları temsil ettiğinden üç değer alabilir. Eğer aldığı değer “2” ise bu dizide “X” simgesi ile gösterilmiştir. Soldan itibaren ilerleyen her üçlü grup, o grup sırasındaki bölgeyi temsil eder.

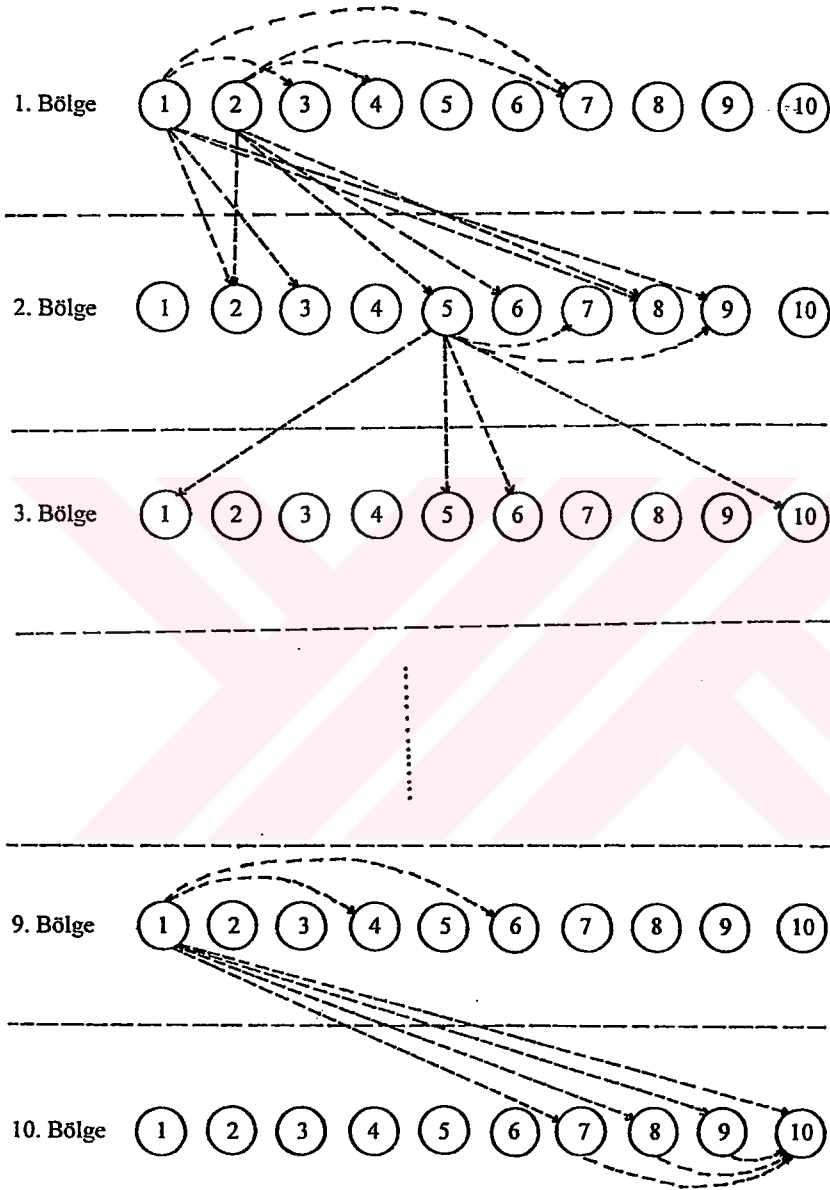
Sonraki aşamada, giriş topluluğundaki dizilerin uygunluk değerleri, bu dizilerin temsil ettiği yolun dallarının verilen fonksiyona göre değerleri toplanarak bulunur. Rulet tekeri yöntemi ile seçilen 100 dizi, çaprazlama ve mutasyon işlemlerine tabi tutulur. Ekte verilen topluluk raporundaki eski diziler sütununda, baştan itibaren, aşağıya doğru iki dizili sıralar, eski nesilden seçilip eşleyen dizileri, çaprazlama noktası ile beraber, karşı tarafındaki yeni diziler kısmında ise yeni nesilde oluşturulan diziler uygunluk değerleri ile beraber verilmiştir. Her bir nesil raporunun sonunda ise, o nesildeki en yüksek, en düşük, ortalama ve toplam uygunluk değerleri, o ana kadar gerçekleştirilen çaprazlama ve bit mutasyonu işlemlerinin sayısı verilmektedir. Ekte ilk 2 ve son 2 nesilin dağılımı ara nesillerin ise sadece istatistiki sonuçlar bulunmaktadır.

Son aşamada ise, en son nesildeki en yüksek uygunluktaki dizi deşifre edilir ve bunun temsil ettiği yol başlangıç (1) düğümünden, hedef (10) düğümüne kadar gösterilir. Burada ikili tireler bölgelerarası linkleri, tekli tireler ise bölge içi linkleri göstermektedir. “X” işaretli yerler, bu noktada düğüm olmadığını, en son belirlenen düğümden itibaren devam edildiğini gösterir. Programın akış diyagramı Şekil 6.2’de gösterilmiştir.

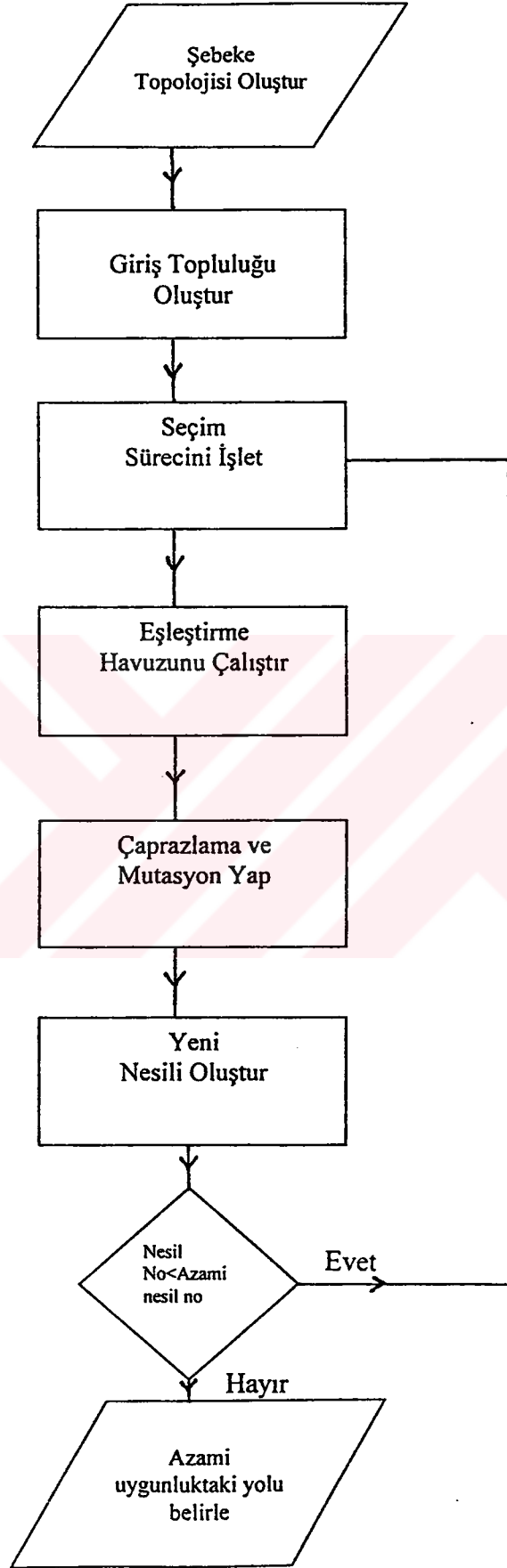
Programın sonunda, GA ile bulunan maksimum uygunluk değeri, aynı topolojik yapı ve trafik şartları altında Bellman-Ford Algoritması ile bulunan gecikme süresinin

tersi ile karşılaştırılır. Ekte ayrıca, azami nesil sayısı 30 olan GA ile BF algoritmaları arasındaki uygunluk değerleri karşılaştırılmıştır. BF Algoritmasının seçilmesinin nedeni, bu algoritmanın diğer deterministik algoritmalara göre daha basit bir akış diyagramına sahip olmasıdır. Burada 20 deneme sonucu elde edilen değerler üzerinden ortalama GA/BF değeri 0.899685 çıkmıştır. GA'nın ortalama hesaplama süresi 4.249084 sn, BF Algoritmasının ise 9.340662 sn çıkmıştır (Celeron 466 MHz). Sonuçta ortalama hesaplama süreleri oranı 0.454901 çıkmıştır. Bu sonuç; gecikme değerinde meydana gelen artmasının, hesaplama süresinde meydana gelen azalmaya oranla oldukça düşük olduğunu göstermektedir.





Şekil 6.1. Şebeke Topolojisi Örneği



Şekil 6.2. Akış Diyagramı

7. SONUÇLAR VE TARTIŞMA

Bu çalışmada, trafik yönlendirme (routing) alanında uygulanan yöntemlerin temelini oluşturan en kısa iz bulma algoritmaları için alternatif bir çözüm olarak Genetik Algoritma önerilmiştir. GA diğer algoritmalara oranla daha esnek ve rastgele bir yapıya sahiptir. Bundan dolayı kesin sonuca ulaşmayı garanti edememesine rağmen kısa süreli hesaplama özelliği ve kolay uyarlanabilir şematik yapısı dolayısıyla zamanın önemli olduğu bağlaşma sistemlerinde verimli olabilir.

Tezde, diğer üç algoritmanın özelliklerinden ve işleyiş biçimlerinden bahsedilmiş olup GA ile karşılaştırmaları yapılmıştır. Bu bağlamda hazırlanan bilgisayar programı vasıtasıyla bir paket bağlaşmalı şebeke üzerinden çalışmada önerilen yöntemin tutarlılığı ve verimi sınanmıştır. Ekte verilen örnek program sonuçları algoritmanın daima daha yüksek uygunlukta diziler ürettiğini ve jenerasyondaki dizilerin toplam uygunluk değerinin ekseriyetle artması bu yöntemin istenilen amacı sağlamada etkili olduğunu göstermiştir. Bu şekilde belirlenen düğüm çiftleri arasındaki en iyi yol, istenen fonksiyon şartlarına göre belirlenir. Burada, şebekedeki her hattaki trafik akış yoğunluğu bilgileri bir merkezi düğümde (RCC) toplanarak, hesaplamalar bu bilgiler üzerinden ayrılıp ilgili düğümlere gönderilir.

GA'ların kullanıldığı yönlendirme algoritmaları hızlı değişen trafik şartlarına kısa sürede uyum sağladığından, oldukça etkilidir. Bu bağlamda yapılacak araştırma ve çalışmalarda yapay zekanın bu uygulama alanı geleceğe dönük ve etkili sonuçlar vermeye adaydır. Ünlü haberleşme kuramcısı J.Hadamard'ın belirttiği gibi, keşif mutlak şansın bir sonucu olmadığı gibi hemen hemen yönlendirilmiş şans ile yol gösterilebilir. GA'lar Hadamard'ın öngördüğü yönlendirme ve şans usulünce birleştirdiği için ayrıca bir öneme haizdir.

KAYNAKLAR

- [1] Çekmece , N., 1992. Zamana Bağlı Trafik Akışının Modellenmesi ve Dinamik Yönelme, *Yüksek Lisans Tezi*, İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul
- [2] Durusoy, G., 1998, İleri Bağlaşma Sistemleri Ders Notları, İ.T.Ü.
- [3] Gün, T., 1992. Devre Bağlaşmalı Telefon Şebekesi İçin Yönlendirme Yazılımı Tasarımı, *Yüksek Lisans Tezi* , İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul.
- [4] ITU-T E. 170 Önerileri, 1988. “International Routing Plan” Blue Book, Fascicle II. 2, IX TH Plenary Assembly, Melbourne, 14-25 November 1988
- [5] ITU-T E.525 Önerileri, 1988. “Service Protection Methods” Blue Book, Fascicle II.3, IX TH Plenary Assembly, Melbourne, 14-25 November 1988
- [6] Botthheim, R., 1987. A Method For Optimization Of Large Mesh-Structured Circuit Switched Networks, Norway, 1987
- [7] Cameron, W.H. Regnier, J., Galloy, P., Savoie, A.M., 1990. Dynamic Routing For Intercity Telephone Networks, *Bell-Northern Research*, Montreal, Canada, 1990.
- [8] Kocabaş, Ş., 1999. Yapay Zeka II Ders Notları, İ.T.Ü.
- [9] Tang, K., Ko, K., Man, K., Kwong, S., 1998. Topology Design and Bandwidth Allocation of Embedded ATM Networks Using Genetic Algorithm, *IEEE Communications Letters*, vol.2, No.6, June 1998.

- [10] Thompson, D.R., Bilbro, G.L., 2000. Comparison of Genetic Algorithm with a Simulated Annealing Algorithm for the Design of an ATM Network, *IEEE Communications Letters*, vol. 4, No.8, August 2000.
- [11] Pierre, S., Legault, G., 1998. A Genetic Algorithm for Designing Distributed Computer Network Topologies, *IEEE Trans. Syst., Man, Cybern*, Vol.28, pp.249-258, April 1998.
- [12] Gelenbe, E., Ghanwani, A., Srinivasan, V., 1997. Improved Neural Heuristics for Multicast Routing, *IEEE Journal on Selected Areas in Communications*, vol.15, pp.147-155, February 1997
- [13] Demirel, L., 1999. Genetic Algorithms in Engineering Optimization, *Msc Thesis*, İ.T.Ü Fen Bilimleri Enstitüsü, İstanbul.
- [14] Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Co., New York.
- [15] Goldberg, D.E., Richardson, J., 1987. Genetic Algorithms with sharing for multi modal function optimization. *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, 41-49.
- [16] Hajela, P., 1999. Nongradient methods in Multi disciplinary Design Optimization-Status and potential, *Journal of Aircraft*, Vol. 36, No:1, January-February 1999.
- [17] Mitchell, M., 1996. An Introduction to Genetic Algorithms, A Bradford Book, MIT Press, Cambridge, MA.
- [18] Bledsoe, W.W., 1961. The use of biological concepts in the analytical study of system. *ORSA-TIMS National Meeting*, November, 1961, San Francisco, CA.

[19] Haftka, R.T., 1993. Optimization of Laminate Stacking Sequence for Buckling Load Maximization By Genetic Algorithm, *AIAA Journal*, Vol, 31, No: 5, May 1993.

[20] Box, G.E.P., 1957. Evolutionary Operation. A. Method For Increasing Industrial Productivity. *Journal of Royal Statistical Society, C*, 6 (2), 81-101.

[21] Bremermann, H.J., 1962. Optimization through evolution and recombination. Self organizing systems, PP.93-106, Spartan Books, Washington, D.C.



EK

-----SEBEKE TOPOLOJYSSY-----							
Hn	Dn	yh1	yh2	Dh1	Dh2	Dh3	Dh4
1	1	3	7	2	3	8	9
1	2	4	7	2	5	6	8
1	3	8	9	3	6	7	10
1	4	6	7	1	2	3	4
1	5	6	8	4	5	7	10
1	6	4	10	5	6	7	10
1	7	4	6	4	6	9	10
1	8	7	9	7	8	9	10
1	9	5	6	1	4	8	9
1	10	1	3	1	2	6	10
2	1	6	7	5	7	8	9
2	2	9	10	2	7	9	10
2	3	4	6	1	4	6	9
2	4	6	9	7	8	9	10
2	5	7	9	1	5	6	10
2	6	1	4	1	3	8	9
2	7	6	10	1	2	6	10
2	8	7	9	5	7	8	9
2	9	7	10	2	6	9	10
2	10	4	6	1	4	6	9
3	1	4	6	7	8	9	10
3	2	7	9	1	5	6	10
3	3	1	4	1	3	8	9
3	4	6	10	1	2	6	10
3	5	7	9	5	7	8	9
3	6	9	10	2	7	9	10
3	7	4	6	1	4	6	9
3	8	4	6	7	8	9	10
3	9	7	10	1	4	5	6
3	10	8	9	1	3	6	10
4	1	2	10	5	6	7	9
4	2	7	9	7	8	9	10
4	3	2	9	2	4	6	10
4	4	1	6	4	6	9	10
4	5	7	9	7	8	9	10
4	6	1	5	1	4	8	9
4	7	1	3	1	2	6	10
4	8	6	7	5	7	8	9
4	9	7	10	2	6	9	10
4	10	4	6	1	4	6	9
5	1	4	6	7	8	9	10
5	2	7	9	1	5	6	10
5	3	1	4	1	3	8	9
5	4	6	10	1	2	6	10
5	5	7	9	5	7	8	9
5	6	9	10	2	7	9	10
5	7	4	6	1	4	6	9
5	8	4	6	7	8	9	10
5	9	7	10	1	4	5	6
5	10	8	9	1	3	6	10
6	1	2	10	5	6	7	9
6	2	7	9	7	8	9	10
6	3	2	9	2	4	6	10

6	4	1	6	4	6	9	10
6	5	7	9	7	8	9	10
6	6	1	5	1	4	8	9
6	7	1	3	1	2	6	10
6	8	6	7	5	7	8	9
6	9	7	10	2	6	9	10
6	10	4	6	1	4	6	9
7	1	4	6	7	8	9	10
7	2	7	9	1	5	6	10
7	3	1	4	1	3	8	9
7	4	6	10	1	2	6	10
7	5	7	9	5	7	8	9
7	6	9	10	2	7	9	10
7	7	4	6	1	4	6	9
7	8	4	6	7	8	9	10
7	9	7	10	1	4	5	6
7	10	8	9	1	3	6	10
8	1	2	10	5	6	7	9
8	2	7	9	7	8	9	10
8	3	2	9	2	4	6	10
8	4	1	6	4	6	9	10
8	5	7	9	7	8	9	10
8	6	1	5	1	4	8	9
8	7	1	3	1	2	6	10
8	8	6	7	5	7	8	9
8	9	7	10	2	6	9	10
8	10	4	6	1	4	6	9
9	1	4	6	7	8	9	10
9	2	7	9	1	5	6	10
9	3	1	4	1	3	8	9
9	4	6	10	1	2	6	10
9	5	7	9	5	7	8	9
9	6	9	10	2	7	9	10
9	7	4	6	1	4	6	9
9	8	4	6	7	8	9	10
9	9	7	10	1	4	5	6
9	10	8	9	1	3	6	10

Hn: Hiyerarşi no (Bölge no)

Dn : Derece no (Bölge içi düğüm no)

İh: İçhat

Dh: Dışhat

Giris populasyonu :

X11010X01000010001X11X10010
X00001X00X01X11X10000000X01
X01X11000X00000X01011010X11
X01001011X00000001X10X00010
000X01X00X11X11X00010110X10
000001X11X10001001001X01100
011100010100001000X11X10X01
011X11X01000X01X10010X10X01
X00001X10X00011111101000X11
111X00X00X00010X00X00X00X01
X00X11000X11X00000X01X00X10
100010X00X10X10110000011000
000X01000X11X00X10001X11X11
X10000X01001011X01011011011
000011011000011011X00X01X11
001X00000X01X11X01X00100110
X01X01000X01100111010X01X11
X10011X01X01X10X10X11101011
001X01000X11X10001X01X11000
X00X01X01011X11X11001X00X10
001X11X00X01X11X01000X11X01
00001011X10110X00X10X10X10
000010011011X10X01X11001111
010000X10X10X01X11X10X0000
X11010X01X00X01000000000X11
011111001011X10010011110X01
X00111001X10010X10001X01001
011101001000X10001110000000
X01111011X11X01001X10X01011
X00001X11001X11X01000010X10
110X00X10011101101X10000X11
011X01X00011001111X01X11X01
X11110X01011X11000011000001
000X01X11X01100010X01X00010
X10X11000000001001000000X11
000X10000X00X10X01X11X10X00
X10001001000X01X01010X11000
011X00011101X10010011010X11
101X01X01X11X11011X01010X11
X11X00X11010X11000001X01X10
X10X11011X11110100011110011
X10000110X10010X01000X10X11
110000X11010X01X10X10X10X00
000001X00011X01X11100100010
011111X01X10011011X11X11X10
X01X10X10000X11X10X01X10101
010000X10X10X00X11X10X01111
X10X00001X10X11011100X00010
000X11000X01X10001101001000
010X00011X11X01X11011X10001
X10X10X01X11X10011111100X00
X10X10011000101X01011010X10
011X01011011001X01X11X10X01
000011101011001X00X01X10011
010X11X11X00X10X01X01010X00
X00X00X01X01X00010X00110011
000X10X11000111X10X00X01001
011X0000001010011001111011

X11010101011X11001X10X00X01
X01X00X11X01X10011X01X01110
X00001001X10X10001X01001X11
101011X01X10X01X01010000000
001000X10X01001X01110X11001
010111010110100000X01X00X11
X01000111X11X10X11001X01X01
X00X11011001X11X00011X10X00
X11X00X01X11000X10X01X01X00
011010X00X01X00010110101X00
011110111X10010X00X10X10011
011X11X11X00X10X10111X00010
011X11X11X00X11000001110X11
X00000X00100X00X01101X11000
X10010X11011101101X11X11000
010X10000X01X00X11000110010
X01X10000X01X00X11000110010
X11X01X11X11011101X01X01X11
001001000011010110X11X10010
000010101X01110010X01X11X01
111X11000011X00X01X10X00X11
111X01X00X01X10X00111X11010
010000X10X11100011011011X10
011X01000X01X01011X01X11000
000000011010010X01X00011001
100010X00001111X01000001100
010010X00X01010100X10100001
X00X11000001X11011X01001010
X00X10X01010X01010X11011X00
X10010011000001X10X00X11010
111X10X00X00011010010000X00
111X01X01X11X01101011000001
10010000000X00X01010X11X01
101X10X00X11X10010X11011101
001100000000000111X01110X11
X10X10X00000100X11000101010
X11X00X10010000000X11X01X00
010001X00001X10001111X11000
011X00X10X01X01001001X00011
101000100001X01X11000100010
X00011X00X01X10010000X11110
101X01X10X10X10X01000011001

Hn1	Hn2	Hn3	Hn4	Hn5	Hn6	Hn7	Hn8	Hn9
X11 ↓ ↓	010 ↓ ↓	X01 ↓ ↓	000 ↓ ↓	010 ↓ ↓	001 ↓ ↓	X11 ↓ ↓	X10 ↓ ↓	X10 ↓ ↓
İh1 Dh1	İh2 Dh2	İh3 Dh3	İh4 Dh4	İh5 Dh5	İh6 Dh6	İh7 Dh7	İh8 Dh8	İh9 Dh9

Yol Gösterilimi: 1B-İh1—Dh1-İh2—Dh2-İh3—Dh3-İh4—Dh4-İh5—Dh5-İh6—Dh6-İh7—Dh7-İh8—Dh8-İh9—Dh9-10S

Yol açılımı : 1B-1—9-7—6-6—7-1—5-7—6-1—6-6—10-10—6-9—5-10S

-----POPULASYON RAPORU-----

Jenerasyon 1

Eski Diziler(Ebeveynler)	xyeri	Uygunluk	Yeni Diziler(Cocuklar)
X10011X01X01X10X10X111101011	11	0.00243167	X10011X01X01X01X11100100010
000001X00011X01X11100100010	11	0.00199347	000001X00011X10X10X11101011
011100010100001000X11X10X01	2	0.00123046	011X00011101X10010011010X11
011X00011101X10010011010X11	2	0.00127569	011100010100001000X11X10X01
000X01X11X01100010X01X00010	7	0.00147117	000X01X11001X1X01000010X10
X00001X11001X11X010000010X10	7	0.00155316	X00001X11X01100011X01X00110
X11X00X11010X11000001X01X10	22	0.0026085	X11X00X11010X11000001X01X11
000011011000011011X00X01X11	22	0.00147387	000011011000011011X00X01X10
000011011000011011X00X01X11	5	0.00203213	000011X01X01X00010X00110011
X00X00X01X01X00010X00110011	5	0.00162927	X00X00011000011011X00X01X11
010000X10X10X01X11X10X0000	16	0.00345579	010000X10X10X01X11000X10X11
X10000110X10010X01000X10X11	16	0.00177769	X10000110X10010X01X10X00000
010001X00001X10001111X11000	16	0.00126334	010001X00001X10001X10000X11
110X00X10011101101X10000X11	16	0.00149521	110X00X10011101101111X11000
111X01X00X01X10X00111X11010	1	0.00123152	110X11000000001001000000X11
X10X11000000001001000000X11	1	0.00243731	X11X01X00X01X10X00111X11010
011111X01X10011011X11X11X10	18	0.00146896	011111X01X10011011X00011001
000000011010010X01X00011001	18	0.00157335	000000011010010X01X11X11X10
011X11X01000X01X10010X10X01	17	0.00219891	011X11X01000X01X10X10X00010
X01001011X00000001X10X00010	17	0.00143111	X01001011X00000001010X10X01
010111010110100000X01X00X11	9	0.000954808	X00001X00X01X11X10X01110X11
000001X11X10001001001X01100	9	0.00250537	100001X11X10100000X01X00X11
010000X10X11100011011011X10	1	0.00219159	010X00X01X01X00010X00110011
X00X00X01X01X00010X00110011	1	0.00214959	X00000X10X11100011011011X10
101011X01X10X01X01010000000	1	0.0023572	100X10X01X11X10011111100X00
X10X10X01X11X10011111100X00	1	0.00181695	X11011X01X10X01X01010000000
010000X10X11100011011011X10	12	0.00136032	010000X10X11110001X11X10010
X11010X01000010001X11X10010	12	0.00121525	X11010X01000000010011011X10
X00001X00X01X11X10000000X01	3	0.00216681	X00011011X11X1001X10X01011
X01111011X11X01001X10X01011	3	0.00234765	X01101X00X01X11X10000000X01
001100000000000111X01110X11	17	0.00113181	001100000000000111000000X01
X00001X00X01X11X10000000X01	17	0.00501647	X00001X00X01X11X10X01110X11
X10000X01001011X01011011011	3	0.00210804	X10001X01X11X11011X01010X11
101X01X01X11X11011X01010X11	3	0.00117099	101X00X01001011X01011011011
X00000X00100X00X01101X11000	1	0.00140251	X01000X10X01001X01110X11001
001000X10X01001X01110X11001	1	0.001743	000000X00100X00X01111X11000
X00X11011001X11X00011X10X00	19	0.00271035	X00X11011001X11X00010X10X11
X10000110X10010X01000X10X11	19	0.0015339	X10000110X10010X01000X10X00
X11010X01X00X01000000000X11	23	0.00174307	X11010X01X00X01000000000000
000X11000X01X10001101001000	23	0.00118832	000X11000X01X10001101001X11
011X00X10X01X01001001X00011	2	0.00236267	011X00X10010000000X11X01X00
X11X00X10010000000X11X01X00	2	0.00233616	X11X00X10X01X01001000X00011
001X11X00X01X11X01000X11X01	6	0.002472	001X11X00X00000001011010X11
X01X11000X00000X01011010X11	6	0.00223031	X01X11000X01X11X01000X11X01
X01001011X00000001X10X00010	5	0.00123021	X010010110000001X10X00X11010
X10010011000001X10X00X11010	5	0.00107011	X10010011X00000001X10X00010
X11X00X11010X11000001X01X10	9	0.00370605	X11X00X11010X10001X01001X11
X00001001X10X10001X01001X11	9	0.00117408	X00001001X10X11000001X01X10
100010X00X10X10110000011000	11	0.00188618	100010X00X10X00010110101X00
011010X00X01X00010110101X00	11	0.00141672	011010X00X00X010110000011000
X00X11000X11X00000X01X00X10	2	0.00192611	X00X00X11010X11000001X01X10
X11X00X11010X11000001X01X10	2	0.00287047	X11X11000X11X00000X01X00X10
X11X01X11X11011101X01X01X11	1	0.00304427	X11X11X11X00X10X10111X00010
011X11X11X00X10X10111X00010	1	0.00345304	011X01X11X11011101X01X01X11

X11X00X11010X11000001X01X10	4	0.00170693	X11X01X01X00X01X00010000000
101011X01X10X01X01010000000	4	0.00147713	101010X11010X11000001X01X10
X10X10X01X11X10011111100X00	5	0.00283351	X10X10X10010000000X11X01X00
X11X00X10010000000X11X01X00	5	0.0027419	X11X00X01X11X1001111100X00
X00X01X01011X11X11001X00X10	22	0.00206648	X00001X01011X11X11001X00000
010000X10X10X01X11X10X00000	22	0.00281506	010000X10X10X01X11X10X00X10
110X00X10011101101X10000X11	23	0.00127466	110X00X10011101101X10000X01
111X00X00X00010X00X00X00X01	23	0.00214685	111X00X00X00010X00X00X00X11
000X10X11000111X10X00X01001	6	0.00441038	000X10X00X00X10X01X11X10X00
000X10000X00X10X01X11X10X00	6	0.00121646	000X10011000111X10X00X01001
X01X11000X00000X01011010X11	13	0.00225551	X01X11000X00000111010X01X11
X01X01000X01100111010X01X11	13	0.00271584	X01X01000X01100X01011010X11
X00001X00X01X11X10000000X01	23	0.00348556	X00001X00X01X11X10000000011
011X00X10X01X01001001X00011	23	0.0027156	011X00X10X01X01001001X00X01
101000100001X01X11000100010	7	0.0013242	101000101001X11X00011X10X00
X00X11011001X11X00011X10X00	7	0.00173456	X00X11010001X01X11000100010
000X01X11X01100010X01X00010	4	0.00205405	000X01000X11X00X10001X11X11
000X01000X11X00X10001X11X11	4	0.00180816	000X01X11X01100010X01X00010
001X01000X11X10001X01X11000	19	0.00143975	001X01000X11X10001X01X01X00
001X11X00X01X11X01000X11X01	16	0.0013192	X11X00X10010000000X11X11000
110000X11010X01X10X10X10X00	9	0.0021675	110000X11011X11X11001X00X10
X00X01X01011X11X11001X00X10	9	0.00229616	X00X01X01010X01X10X10X10X00
001X11X00X01X11X01000X11X01	16	0.00281775	001X11X00X01X11X01X01010X00
010X11X11X00X10X01X01010X00	16	0.00319219	010X11X11X00X10X01000X11X01
X00X01X01011X11X11001X00X10	10	0.00196631	X00X01X01011X01001001X00011
011X00X10X01X01001001X00011	10	0.00365259	011X00X10X01X11X11001X00X10
X00X11011001X11X00011X10X00	2	0.00285635	X00X00X10X01001001001X00011
011X00X10X01X01001001X00011	2	0.0017926	011X11011001X11X00011X10X00
X10001001000X01X01010X11000	21	0.0012991	X10001001000X01X01010X11000
100010X00X10X10110000011000	21	0.0012211	100010X00X10X10110000011000
X01X10000X01X00X11000110010	4	0.00253954	X01X11000X11X00000X01X00X10
X00X11000X11X00000X01X00X10	4	0.00135552	X00X10000X01X00X11000110010
X00X01X01011X11X11001X00X10	1	0.00190526	X00X00001X10X11011100X00010
X10X00001X10X11011100X00010	1	0.00205657	X10X01X01011X11X11001X00X10
X11X00X01X11000X10X01X01X00	8	0.00336633	X11X00X01110X10001X01001X11
X00001001X10X10001X01001X11	8	0.00155709	X00001001X11000X10X01X01X00
X00X00X01X01X00010X00110011	3	0.00134938	X00X11010110100000X01X00X11
01011010110100000X01X00X11	3	0.00187341	010100X01X01X00010X00110011
01110111X10010X00X10X10011	17	0.00160518	01110111X10010X00X10X10X10
00001011X10110X00X10X10X10	17	0.00119134	00001011X10110X00X10X10001
X00X11000X11X00000X01X00X10	1	0.0030423	X01X10X10000X11X10X01X10101
X01X10X10000X11X10X01X10101	1	0.00305804	X00X11000X11X00000X01X00X10
010X00011X11X01X11011X10001	4	0.00121317	010X01000X01100111010X01X11
X01X01000X01100111010X01X11	4	0.00242484	X01X00011X11X01X11011X10001
001X11X00X01X11X01000X11X01	11	0.00287184	001X11X00X01X10011X01X01110
X01X00X11X01X10011X01X01110	11	0.00328359	X01X00X11X01X11X01000X11X01

Not: Jenerasyon 1 & Toplu istatistikler:
max=0.00501647 min=0.000954808 avj=0.00210347
top=0.210347 nmutasyon=15 ncross=50

Jenerasyon 2

Eski Diziler(Ebeveynler)	xyeri	Uygunluk	Yeni Diziler(Cocuklar)
001X01000X11X10001X01X01X00	1	0.00146711	000X01X01011X010001001X00011
X00X01X01011X010001001X00011	1	0.00295066	X01X01000X11X10001X01X01X00
X01X0X10000X11X10X01X10101	12	0.0025174	X01X0X10000X010001001X00011
X00X01X01011X010001001X00011	12	0.00225563	X00X01X01011X11X10X01X10101
X01X11000X01X11X01000X11X01	20	0.00161146	X01X11000X01X11X01000100X00
100X10X01X11X10011111100X00	20	0.00206125	100X10X01X11X100111111X11X01
X11X00X11010X11000001X01X11	3	0.00183572	X11X10X01X00X01000000000000
X11010X01X00X01000000000000	3	0.00204851	X11000X11010X11000001X01X11
000011X01X01X00010X00110011	18	0.00303173	000011X01X01X00010X01X00X10
X01X11000X11X00000X01X00X10	18	0.00181728	X01X11000X11X00000X00110011
01111X01X10011011X00011001	2	0.0017708	011010011X00000001X10X00010
X10010011X0000001X10X00010	2	0.0013773	X10111X01X10011011X00011001
000011X01X01X00010X00110011	22	0.00276146	000011X01X01X00010X00110X11
100001X11X10100000X01X00X11	22	0.00142601	000001X11X10100000X01X00011
X01X10X10000X11X10X01X10101	19	0.0069805	X01X10X10000X11X10X01X10X11
X00X11011001X11X00010X10X11	19	0.00175917	X00X11011001X11X00011X10101
X11X00X11010X11000001X01X11	11	0.0025174	X11X00X11010X11X01000X11X01
X01X00X11X01X11X01000X11X01	11	0.00286432	X01X00X11X01X11000001X01X11
100010X00X10X10110000011000	13	0.00195655	100010X00X10X10010110101X00
100010X00X10X00010110101X00	13	0.00124561	100010X00X10X00110000011000
111X00X00X00010X00X00X00X11	12	0.00112336	111X00X00X00001X01110X11001
X01000X10X01001X01110X11001	12	0.00234006	X01000X10X01010X00X00X00X11
X00011011X11X01001X10X01011	10	0.00183787	X00011011X10000001010X10X01
X01001011X0000001010X10X01	10	0.00275782	X01001011X01X01001X10X01011
010000X10X11110001X11X10010	16	0.0020258	010000X10X11110001X01010X00
001X11X00X01X11X01010X00	16	0.00211533	001X11X00X01X11X01X11X10010
X01X11000X01X11X01000X11X01	25	0.00206772	X01X11000X01X11X01000X11X00
X10X10X10010000000X11X01X00	25	0.0029021	X10X10X10010000000X11X01X01
110X00X10011101101111X11000	1	0.00312387	111X00X00X01X11X11001X00X10
011X00X10X01X11X11001X00X10	1	0.00127029	010X00X10011101101111X11000
011X01X11X11011101X01X01X11	12	0.00222181	011X01X11X11001001001X00111
X00X00X10X01001001001X00011	12	0.00448913	X00X00X10X01011101X01X01X11
X01X10X1000X11X10X01X10101	4	0.00538573	X01X11X10X10100000X01X00X11
100001X11X10100000X01X00X11	4	0.00172662	100000X10100X11X10X01X10101
X00X00X10X01001001001X00011	9	0.00262064	X00X00X10X00010X00X00X00X11
111X00X00X00010X00X00X00X11	9	0.00227035	111X00X00X01001001001X00011
X01101X00X01X11X10000000X01	10	0.00223628	X01101X00X01X11X01000X11X01
X01X00X11X01X11X01000X11X01	10	0.00352942	X01X00X11X01X11X10000000X01
X01X00X11X01X11X01000X11X01	4	0.00194048	X01X01X11X10100000X01X00X11
100001X11X10100000X01X00X11	4	0.00212961	100000X11X01X11X01000X11X01
01111X01X10011011X00011001	13	0.00208713	01111X01X10010001X10X00010
X10010011X0000001X10X00010	13	0.000809509	X01X10X10000X11X10X01X10X11
X10011X01X01X01X11100100010	1	0.00233616	X11X00X10X01X010001000X00011
X11X00X10X01X01001000X00011	1	0.00243167	X10011X01X01X01X11100100010
011X00X10X01X01001001X00X01	5	0.00135229	011X00010100001001X11X10X01
011100010100001000X11X10X01	5	0.00245011	011100X10X01X01001001X00X01
X00X01X01011X01001001X00011	2	0.00254099	X00X00X10X01X01001000X00011
X11X00X10X01X01001000X00011	2	0.00244893	X11X01X01011X01001001X00011
100001X11X10100000X01X00X11	14	0.00179123	100001X11X10100X01000X11X01
010X11X11X00X10X01000X11X01	14	0.00255547	010X11X11X00X10000X01X00X11
001100000000000111000000X01	8	0.00252251	001100000X01X11X10000000011
X00001X00X01X11X10000000011	8	0.000922019	X00001X00000000111000000X01
100001X11X10100000X01X00X11	13	0.00230688	100001X11X10101X11000X10X11
010000X10X10X01X11000X10X11	13	0.00347432	010000X10X10X00000X01X00X11
X00X11011001X11X00010X10X11	19	0.00161073	X01X11011001X11X00011010X01
001X11X00X01X11X01X01010X00	19	0.00370316	001X11X00X01X11X01X00X10X11
100X10X01X11X10011111100X00	23	0.00261167	100X10X01X11X10011111100X11

X00001X00X01X11X10X01110X11	23	0.00385865	X00001X00X01X11X10X01110X00
X11X00X10X01X01001000X00011	8	0.00152787	X11X00X10X11110001X11X10010
010000X10X11110001X11X10010	8	0.00197922	010000X10X01X01001000X00011
X11X00X11010X11000001X01X11	19	0.00193369	X11X00X11110X11000000X11X01
X01X11000X01X11X01000X11X01	19	0.00256934	X01X11000X01X11X01001X01X11
X11X11X11X00X10X10111X00010	21	0.00268726	X11X11X11X00X10X10111X00X10
X10X01X01011X11X11001X00X10	21	0.0015231	X10X01X01011X11X11001X00010
X11X00X01X11X1001111100X00	4	0.00370605	X11X00X11010X10001X01001X11
X11X00X11010X10001X01001X11	4	0.00272228	X11X00X01X11X10001111100X00
011X11011001X11X00011X10X00	23	0.0017926	011X11011001X11X00011X10X00
001X11X00X01X11X01X01010X00	23	0.00281775	001X11X00X01X11X01X01010X00
X11X00X01110X10001X01001X11	14	0.00259689	X11X00X01110X10000001X01X11
X11X00X11010X11000001X01X11	14	0.00162251	X11X00X11010X11001X01001X11
X01000X10X01001X01110X11001	10	0.00419023	X01000X10X00X10001X01001X11
X11X00X11010X10001X01001X11	10	0.00140585	X11X00X11011001X01110X11001
X01X10X10000X11X10X01X10101	3	0.00295514	X01X01X11010X11000001X01X11
X11X00X11010X11000001X01X11	3	0.00150207	X11X10X10000X11X10X01X10101
X00011011X11X01001X10X01011	5	0.00147114	X00011X11001X11X01000010X10
000X01X11001X11X01000010X10	5	0.00255492	000X01011X11X01001X10X01011
011110111X10010X00X10X10X10	17	0.00131679	011110111X10010X00010X10X11
X00X11011001X11X00010X10X11	17	0.002039	X00X11011001X11X00X10X10X10
010X01000X01100111010X01X11	18	0.000968127	010X01000101100111001001X11
000X11000X01X10001101001X11	18	0.00177608	000X11000X01X10101110X01X11
101010X11010X11000001X01X10	12	0.0025437	101010X11010X10001X01001X11
X11X00X11010X10001X01001X11	12	0.00180607	X11X00X11010X11000001X01X10
X00X11011001X11X00010X10X11	21	0.002036	X00X11011001X11X00010X11X01
X01X00X11X01X11X01000X11X01	21	0.00312664	X01X00X11X01X11X01000X10X11
000011X01X01X00010X00110011	22	0.00276146	000011X01X01X00010X00110X11
100001X11X10100000X01X00X11	22	0.00142601	000001X11X10100000X01X00011
X01X10X10000X11X10X01X10101	19	0.0069805	X01X10X10000X11X10X01X10X11
X00X11011001X11X00010X10X11	19	0.00175917	X00X11011001X11X00011X10101
X11X00X11010X11000001X01X11	11	0.0025174	X11X00X11010X11X01000X11X01
X01X00X11X01X11X01000X11X01	11	0.00286432	X01X00X11X01X11000001X01X11
100010X00X10X10110000011000	13	0.00195655	100010X00X10X10010110101X00
100010X00X10X00010110101X00	13	0.00124561	100010X00X10X00110000011000
111X00X00X00010X00X00X00X11	12	0.00112336	111X00X00X00001X01110X11001
X01000X10X01001X01110X11001	12	0.00234006	X01000X10X01010X00X00X00X11
X00011011X11X01001X10X01011	10	0.00183787	X00011011X10000001010X10X01
X01001011X00000001010X10X01	10	0.00275782	X01001011X01X01001X10X01011
010000X10X11110001X11X10010	16	0.0020258	010000X10X11110001X01010X00
001X11X00X01X11X01X01010X00	16	0.00211533	001X11X00X01X11X01X11X10010
X01X11000X01X11X01000X11X01	25	0.00206772	X01X11000X01X11X01000X11X00
X10X10X10010000000X11X01X00	25	0.0029021	X10X10X10010000000X11X01X01

Not:Jenerasyon 2 & Toplu ıstatistikler:
max=0.0069805 min=0.000809509 avj=0.00238242
top=0.238242 nmutasyon=31 ncross=100
Not:Jenerasyon 3 & Toplu ıstatistikler:
max=0.00924671 min=0.001186 avj=0.00267061
top=0.267061 nmutasyon=48 ncross=150
Not:Jenerasyon 4 & Toplu ıstatistikler:
max=0.00924671 min=0.000986211 avj=0.0031598
top=0.31598 nmutasyon=65 ncross=200
Not:Jenerasyon 5 & Toplu ıstatistikler:
max=0.00924671 min=0.00193413 avj=0.0042336
top=0.42336 nmutasyon=82 ncross=250
Not:Jenerasyon 6 & Toplu ıstatistikler:
max=0.00924671 min=0.00173153 avj=0.0035558
top=0.35558 nmutasyon=99 ncross=300
Not:Jenerasyon 7 & Toplu ıstatistikler:
max=0.00924671 min=0.00184754 avj=0.00265544

top=0.265544 nmutasyon=115 ncross=350
Not:Jenerasyon 8 & Toplu isticistikler:
max=0.00924671 min=0.00163909 avj=0.00255534
top=0.255534 nmutasyon=132 ncross=400
Not:Jenerasyon 9 & Toplu isticistikler:
max=0.00924671 min=0.00184754 avj=0.00272647
top=0.272647 nmutasyon=139 ncross=450
Not:Jenerasyon 10 & Toplu isticistikler:
max=0.00924671 min=0.0015066 avj=0.00304649
top=0.304649 nmutasyon=164 ncross=500
Not:Jenerasyon 11 & Toplu isticistikler:
max=0.00924671 min=0.0013221 avj=0.00329025
top=0.329025 nmutasyon=196 ncross=547
Not:Jenerasyon 12 & Toplu isticistikler:
max=0.00924671 min=0.00127568 avj=0.00347297
top=0.347297 nmutasyon=227 ncross=593
Not:Jenerasyon 13 & Toplu isticistikler:
max=0.00924671 min=0.00169151 avj=0.00410579
top=0.410579 nmutasyon=264 ncross=640
Not:Jenerasyon 14 & Toplu isticistikler:
max=0.0104047 min=0.00184903 avj=0.0043232
top=0.43232 nmutasyon=296 ncross=687
Not:Jenerasyon 15 & Toplu isticistikler:
max=0.0113262 min=0.00217576 avj=0.00493469
top=0.493469 nmutasyon=327 ncross=733
Not:Jenerasyon 16 & Toplu isticistikler:
max=0.0113262 min=0.00119669 avj=0.00564334
top=0.564334 nmutasyon=368 ncross=782
Not:Jenerasyon 17 & Toplu isticistikler:
max=0.0113262 min=0.00135358 avj=0.00666712
top=0.666712 nmutasyon=384 ncross=832
Not:Jenerasyon 18 & Toplu isticistikler:
max=0.0113262 min=0.00182684 avj=0.00667859
top=0.667859 nmutasyon=396 ncross=882
Not:Jenerasyon 19 & Toplu isticistikler:
max=0.0113262 min=0.00244481 avj=0.00743433
top=0.743433 nmutasyon=408 ncross=932
Not:Jenerasyon 20 & Toplu isticistikler:
max=0.0113262 min=0.00303403 avj=0.00796262
top=0.796262 nmutasyon=419 ncross=982
Not:Jenerasyon 21 & Toplu isticistikler:
max=0.0113262 min=0.00282727 avj=0.00814023
top=0.814023 nmutasyon=430 ncross=1032
Not:Jenerasyon 22 & Toplu isticistikler:
max=0.0113262 min=0.00204989 avj=0.00885593
top=0.885593 nmutasyon=448 ncross=1082
Not:Jenerasyon 23 & Toplu isticistikler:
max=0.0113262 min=0.00229601 avj=0.0086475
top=0.86475 nmutasyon=469 ncross=1132
Not:Jenerasyon 24 & Toplu isticistikler:
max=0.0113262 min=0.00297997 avj=0.00859839
top=0.859839 nmutasyon=488 ncross=1182
Not:Jenerasyon 25 & Toplu isticistikler:
max=0.0113262 min=0.00299147 avj=0.00846209
top=0.846209 nmutasyon=509 ncross=1232
Not:Jenerasyon 26 & Toplu isticistikler:
max=0.0113262 min=0.00224972 avj=0.00857824
top=0.857824 nmutasyon=528 ncross=1282
Not:Jenerasyon 27 & Toplu isticistikler:
max=0.0113262 min=0.00224972 avj=0.00852046

top=0.852046 nmutasyon=550 ncross=1331
Not:Jenerasyon 28 & Toplu istatistikler:
max=0.01731 min=0.00164896 avj=0.00819133
top=0.819133 nmutasyon=580 ncross=1380
Not:Jenerasyon 29 & Toplu istatistikler:
max=0.01731 min=0.00345179 avj=0.008591
top=0.8591 nmutasyon=609 ncross=1430
Not:Jenerasyon 30 & Toplu istatistikler:
max=0.01731 min=0.00299147 avj=0.00905543
top=0.905543 nmutasyon=625 ncross=1480
Not:Jenerasyon 31 & Toplu istatistikler:
max=0.01731 min=0.00291865 avj=0.00919368
top=0.919368 nmutasyon=648 ncross=1530
Not:Jenerasyon 32 & Toplu istatistikler:
max=0.0195534 min=0.00249212 avj=0.00941212
top=0.941212 nmutasyon=668 ncross=1580
Not:Jenerasyon 33 & Toplu istatistikler:
max=0.0195534 min=0.00287034 avj=0.0101026
top=1.01026 nmutasyon=689 ncross=1630
Not:Jenerasyon 34 & Toplu istatistikler:
max=0.0195534 min=0.00274528 avj=0.0102981
top=1.02981 nmutasyon=710 ncross=1680
Not:Jenerasyon 35 & Toplu istatistikler:
max=0.0195534 min=0.00285978 avj=0.0115534
top=1.15534 nmutasyon=731 ncross=1730
Not:Jenerasyon 36 & Toplu istatistikler:
max=0.0195534 min=0.00343652 avj=0.0120803
top=1.20803 nmutasyon=745 ncross=1779
Not:Jenerasyon 37 & Toplu istatistikler:
max=0.0195534 min=0.00543167 avj=0.0138842
top=1.38842 nmutasyon=757 ncross=1827
Not:Jenerasyon 38 & Toplu istatistikler:
max=0.0195534 min=0.00352417 avj=0.0146232
top=1.46232 nmutasyon=767 ncross=1874
Not:Jenerasyon 39 & Toplu istatistikler:
max=0.0195534 min=0.00543167 avj=0.0154413
top=1.54413 nmutasyon=780 ncross=1922
Not:Jenerasyon 40 & Toplu istatistikler:
max=0.0195534 min=0.00408603 avj=0.0163812
top=1.63812 nmutasyon=791 ncross=1970
Not:Jenerasyon 41 & Toplu istatistikler:
max=0.0195534 min=0.00408603 avj=0.0163867
top=1.63867 nmutasyon=802 ncross=2018
Not:Jenerasyon 42 & Toplu istatistikler:
max=0.0195534 min=0.00221405 avj=0.0156209
top=1.56209 nmutasyon=823 ncross=2067
Not:Jenerasyon 43 & Toplu istatistikler:
max=0.0195534 min=0.0022213 avj=0.015142
top=1.5142 nmutasyon=850 ncross=2117
Not:Jenerasyon 44 & Toplu istatistikler:
max=0.0195534 min=0.00400076 avj=0.0153968
top=1.53968 nmutasyon=878 ncross=2167
Not:Jenerasyon 45 & Toplu istatistikler:
max=0.0195534 min=0.00224734 avj=0.0140986
top=1.40986 nmutasyon=906 ncross=2217
Not:Jenerasyon 46 & Toplu istatistikler:
max=0.0195534 min=0.00204663 avj=0.0140922
top=1.40922 nmutasyon=932 ncross=2267
Not:Jenerasyon 47 & Toplu istatistikler:
max=0.0195534 min=0.00160107 avj=0.0138916

top=1.38916 nmutasyon=959 ncross=2317
Not:Jenerasyon 48 & Toplu isticistikler:
max=0.0195534 min=0.00125975 avj=0.0144026
top=1.44026 nmutasyon=990 ncross=2366



Jenerasyon 49

Eski Diziler(Ebeveynler)	xyeri	Uygunluk	Yeni Diziler(Cocuklar)
X01X01X11X10X11X00X00X01111	21	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X10X00X01X11	21	0.00736515	X01X00X11X10X11X10X00X01111
X01X00X11X10X11X10X00X01X11	9	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	9	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X01X00X01X11	11	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0146092	X01X01X11X10X11X01X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X00X00X01X11	19	0.00743232	X01X01X1111X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	8	0.00684151	X01001X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	8	0.0187988	X01X01X11X10X11X10X01X01X11
X01X00X01X10X11X00X00X01X11	13	0.00358185	X01X00X01X10X11X00X01X01X11
X01X00X11X10X11X00X01X01X11	13	0.0177436	X01X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X01X10X11X10X00X01X11	11	0.0106527	X01X01X01X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	17	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X00X00X01X11	17	0.0162324	X00X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.00589681	X00X01X11X10X11X11X00X01X11
X01X01X11X10X11X00X00X01X11	3	0.00366489	X01101X11X10X11X10X00X01X11
X11X01X11X10X11X10X00X01X11	3	0.0113262	X11X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X01X01X11	22	0.0165955	X01X01X11X10X11X10X01X00X11
X01X00X11X10X11X00X00X00X11	22	0.00261825	X01X00X11X11X11100X00X01X11
X01X01X11X10X11X10X01X01X11	12	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	12	0.00958188	X01X00X11X10X01X10001X01X11
X01X01011X10X11X00X01X01X11	25	0.003424	X01X01011X10X11X00X01X01X11
X01X00X11X10X11X00X00X01X11	25	0.00218418	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X01X01X11	8	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	8	0.0187988	X01X01X11X10X11X10X01X01X11
X01X00X01X10X11X00X00X01X11	26	0.0116834	X01X00X01X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	26	0.0177436	X01X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	3	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	3	0.0172913	X01X01X11X10X11X10X00X01X11
X11X01X11X10X11X10X00X01X11	4	0.0105283	X11X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	4	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X00X01X11	20	0.00695331	X00X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0102212	X01X01X10X10X11X10X10X01X11
X01X00X11X10X11X10X00X01X11	11	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	2	0.0177436	X01X00X11X10X11X00X00X01X11
101X00X11X10X11X00X00X01X11	2	0.00436874	101X01X11X10X11X10X00X01X11
X01XG1X11X10X11X00X00X01111	7	0.00405729	X01X01X11X10X11X10X00X01X10
X01X01X11X10X11X10X00X01X10	7	0.00684214	X01X01X11X10X11X00X00X01111
X01X01X11X10X11X00X00X01111	21	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X10X00X01X11	21	0.00736515	X01X00X11X10X11X10X00X01111
X01X00X11X10X11X10X00X01X11	9	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	9	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X01X00X01X11	11	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0146092	X01X01X11X10X11X01X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01Z11X10X11X00X00X01X11	19	0.00743232	X01X01X1111X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	8	0.00684151	X01001X11X10X11X10X00X01X11
X01XG1X11X10X11X10X00X01X11	8	0.0187988	X01X01X11X10X11X10X01X01X11
X01X00X01X10X11X00X00X01X11	13	0.00358185	X01X00X01X10X11X00X01X01X11

X01X00X11X10X11X00X01X01X11	13	0.0177436	X01X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X01X10X11X10X00X01X11	11	0.0106527	X01X01X01X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	17	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X00X00X01X11	17	0.0162324	X00X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.00589681	X00X01X11X10X11X11X00X01X11
X01X01X11X10X11X00X00X01X11	3	0.00366489	X01101X11X10X11X10X00X01X11
X11X01X11X10X11X10X00X01X11	3	0.0113262	X11X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X01X01X11	22	0.0165955	X01X01X11X10X11X10X01X00X11
X01X00X11X10X11X00X00X00X11	22	0.00261825	X01X00X11X11X11100X00X01X11
X01X01X11X10X11X10X01X01X11	12	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	12	0.00958188	X01X00X11X10X01X10001X01X11
X01X01011X10X11X00X01X01X11	25	0.003424	X01X01011X10X11X00X01X01X11
X01X00X11X10X11X00X00X01X11	25	0.00218418	X01X00X11X10111X00X10X01X11
X01X01X11X10X11X10X01X01X11	8	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	8	0.0187988	X01X01X11X10X11X10X01X01X11
X01X00X01X10X11X00X00X01X11	26	0.0116834	X01X00X01X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	26	0.0177436	X01X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	3	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	3	0.0172913	X01X01X11X10X11X10X00X01X11
X11X01X11X10X11X10X00X01X11	4	0.0105283	X11X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	4	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X00X01X11	20	0.00695331	X00X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0102212	X01X01X10X10X11X10X10X01X11
X01X00X11X10X11X10X00X01X11	11	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	2	0.0177436	X01X00X11X10X11X00X00X01X11
101X00X11X10X11X00X00X01X11	2	0.00436874	101X01X11X10X11X10X00X01X11
X01X01X11X10X11X00X00X01111	7	0.00405729	X01X01X11X10X11X10X00X01X10
X01X01X11X10X11X10X00X01X10	7	0.00684214	X01X01X11X10X11X00X00X01111
X01X01X11X10X11X00X00X01111	21	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X10X00X01X11	21	0.00736515	X01X00X11X10X11X10X00X01111
X01X00X11X10X11X10X00X01X11	9	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	9	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X01X00X01X11	11	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0146092	X01X01X11X10X11X01X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X00X00X01X11	19	0.00743232	X01X01X11110X11X00X00X01X11

Not: Jenerasyon 49 & Toplu istatistikler:
max=0.0195534 min=0.00218418 avj=0.0133359
top=1.33359 nmutasyon=1023 ncross=2414

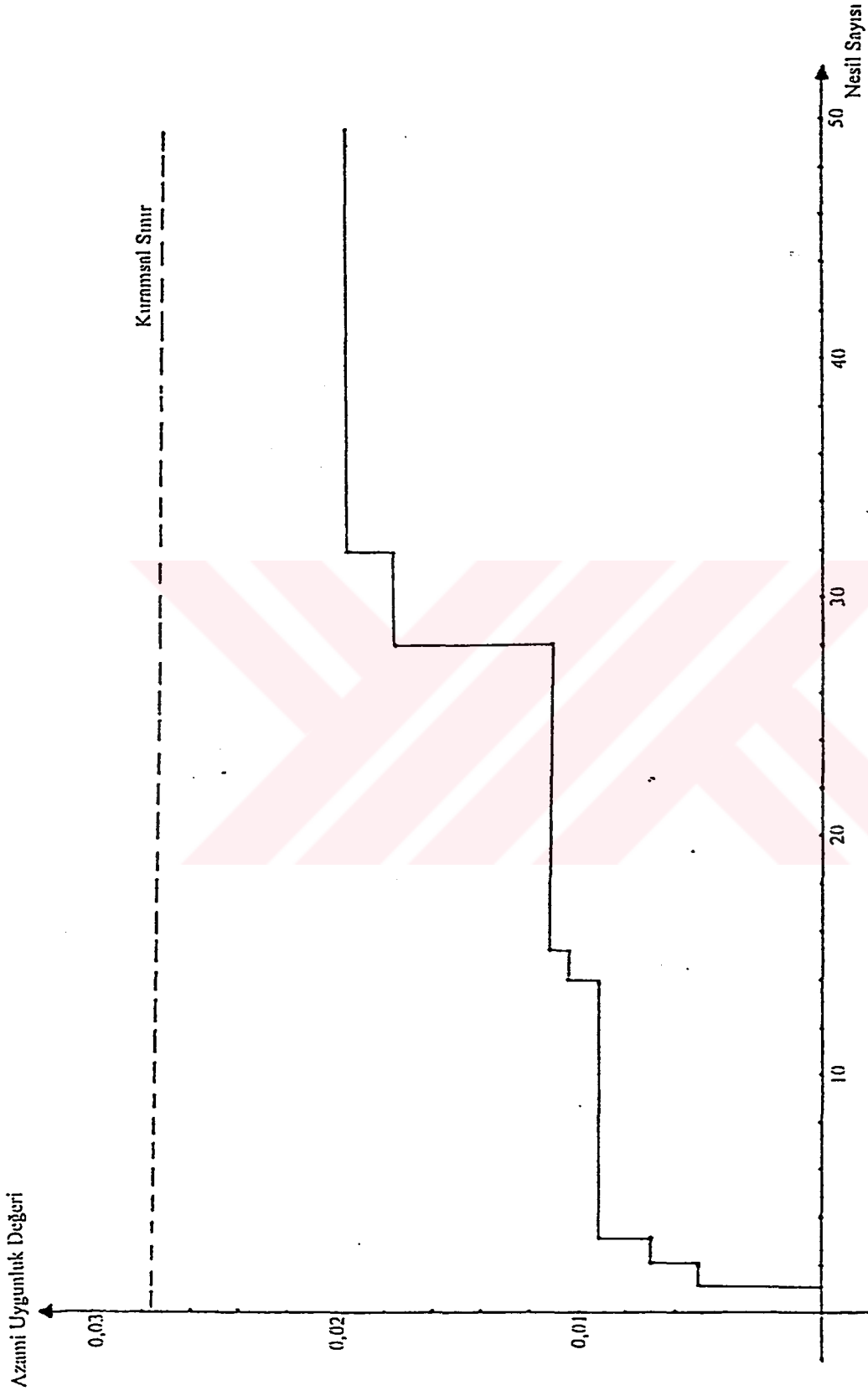
Jenerasyon 50

Eski Diziler(Ebeveynler)	xyeri	Uygunluk	Yeni Diziler(Cocuklar)
X01X01X11X10X11X00X00X01111	12	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	12	0.00684214	X01X01X11X10X11X00X00X01111
X01X01X11X10X11X01X00X01X11	20	0.00984564	X01X01X11X10X11X00X00X01X11
X01X01X1111X11X00X00X01X11	20	0.00743232	X01X01X1111X11X00X00X01X11
X01X01X11X10X11X10X01X01X11	14	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	14	0.00408603	X01X01X11X10X11X00X01X01X11
X01X01X11X10X11X10X00X01X11	19	0.0161209	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	17	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	2	0.0177436	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X00X00X01X11	2	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X01X00X01X11	13	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	13	0.0135747	X01X00X11X10X11X01X00X01X11
X01X01X11X10X11X00X00X01X11	1	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	1	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	22	0.00684151	X01X01X11X10X11X10X00X01X11
X01X00X11X1011X00X10X01X11	22	0.00657881	X01X00X11X1001X00X10X01X11
X01X01X11X10X11X00X00X01X11	3	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	3	0.00984564	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X01X00X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0165955	X01X01X11X10X11X10X01X00X11
X01X01X11X10X11X10X00X01X11	9	0.00431386	X01X01X11X10X11X00100X01X11
X01X01X11X10X11X10X00X01X11	9	0.0172913	X01X01X11X10X11X10X00X01X11
X01X0101X10X11X00X01X01X11	25	0.003424	X01X0101X10X11X00X01X01X11
X01X00X11X10X11X10X00X01X11	25	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	4	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	4	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	23	0.0158607	X01X00X11X10X11X10X00X01X11
X01001X11X10X11X10X00X01X11	23	0.00435593	X01001X1111X10X11X10X00X01X11
X00X00X11X10X11X00X00X01X11	1	0.0195534	X01X01X11X10X11X00X00X01X11
X11X01X11X10X11X00X00X01X11	1	0.0112984	X10X00X11X10X11X00X00X01X11
X01X01X11X10X11X01X00X01X11	5	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	5	0.0135747	X01X00X11X10X11X01X00X01X11
X01X01X11X10X11X10X00X01X11	1	0.00799111	X01X00X11X10X11X01X00X01X01
X01X00X11X10X11X00X00X01X11	1	0.00575887	001X01X11X10X11X10X00X01X11
101X01X11X10X11X10X00X01X11	8	0.00494447	101X01X11X10X11X00X00X01X11
X00X00X11X10X11X00X00X01X11	8	0.00453066	X00X00X11X10X11X10X00X00X11
X00X00X11X10X11X00X00X01X11	22	0.0159003	X00X00X01X10X11X00X00X01X11
X01X01X11X10X11X00X00X01X11	22	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	9	0.00464098	X01X00X11X10X11X10X00X00X11
X01X01X11X10X11X10X00X01X11	9	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X00X00X01X11	5	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X00X00X01X11	5	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X00X00X01X11	7	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	7	0.0177436	X01X00X11X10X11X00X00X01X11
X01X01X11X10X11X10X01X01X11	15	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	15	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	3	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	3	0.0187988	X01X01X11X10X11X10X01X01X11
101X01X11X10X11X10X00X01X11	2	0.00258853	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X01X01X11	2	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X01X10001X01X11	1	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	1	0.00958188	X01X00X11X10X01X10001X01X11
X01X01X11X10X11X10X01X01X11	13	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	13	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X00X00X01X11	1	0.0120083	X01X01X10X10X11X10X01X01X11

X01X01X11X10X11X10X01X01X11	1	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	19	0.0187968	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X10X01X01X11	19	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	21	0.0039732	X01X00X11X10X11X10X00X01X10
X01X01X11X10X11X10X00X01X10	21	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X00X00X01X11	15	0.00475613	X01X01X11X10X11X10X00X00X11
X01X01X11X10X11X10X00X01X11	15	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X00X01X01X11	22	0.003424	X01X01X11X10X11X00X01X01X11
X01X01X11X10X11X00X01X01X11	22	0.00633271	X01X01X11X10X11X01X00X01X01
X01X01X11X10X11X00X00X01X11	11	0.0195534	X01X01X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	11	0.0158607	X01X00X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	15	0.0177436	X01X00X11X10X11X00X00X01X11
X01X00X11X10X11X00X00X01X11	15	0.0039732	X01X00X11X10X11X10X00X01X10
X01X00X11X10X11X10X00X01X11	20	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X10X01X01X11	20	0.0187988	X01X01X11X10X11X10X01X01X11
X01X01X11X10X11X00X00X01X11	14	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X00X00X01X11	14	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	5	0.00765942	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X00X00X01X11	5	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	7	0.0158607	X01X00X11X10X11X10X00X01X11
X01101X11X10X11X10X00X01X11	7	0.00366489	X01101X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.0198612	X01X01X11X10X11X11X01X01X11
X01X01X11X10X11X10X01X01X11	17	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	6	0.0039732	X01X00X11X10X11X10X00X01X10
X01X01X11X10X11X10X00X01X10	6	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	11	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01111	7	0.00650635	X01X00X11X10X11X00X00X01111
X01X01X11X10X11X00X00X01111	7	0.0054152	X01X01X11X00X11X10X00X01111
X01X00X11X10X11X10X00X01X11	4	0.0172913	X01X01X11X10X11X10X00X01X11
X11X01X11X10X11X10X00X01X11	4	0.0097275	X11X00X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X11	20	0.0158607	X01X00X11X10X11X10X00X01X11
X01X01X11X10X11X00X00X01X11	20	0.0195534	X01X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	5	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	5	0.0172913	X01X01X11X10X11X10X00X01X11
X01X01X11X10X11X01X00X01X11	10	0.00494447	101X01X11X10X11X00X00X01X11
X01X01X11X10X11X10X00X01X11	10	0.0146092	X01X01X11X10X11X01X00X01X11
X01X00X11X10X11X10X00X01X11	7	0.0158607	X01X00X11X10X11X10X00X01X11
X01101X11X10X11X10X00X01X11	7	0.00366489	X01101X11X10X11X10X00X01X11
X01X01X11X10X11X10X00X01X11	17	0.0198612	X01X01X11X10X11X11X01X01X11
X01X01X11X10X11X10X01X01X11	17	0.0172913	X01X01X11X10X11X10X00X01X11
X01X00X11X10X11X10X00X01X10	6	0.0039732	X01X00X11X10X11X10X00X01X10
X01X01X11X10X11X10X00X01X10	6	0.0172913	X01X01X11X10X11X10X00X01X11

Not: Jenerasyon 50 & Toplu istatistikler:
max=0.0195534 min=0.00258853 avj=0.013959
top=1.3959 mutasyon=1047 ncross=2464

Bulunan en iyi yol:
1B-X- -3-X- -4-X-10-X- -4-X- -10-X- -1-X- -7-X- -2-X- -10-X5



Sıra	GA (1/T)	BF (1/T)	GA/BF
1	0,0177224	0,0210172	0,843233
2	0,0077803	0,0082204	0,946458
3	0,0155622	0,0199959	0,778269
4	0,0185770	0,0206760	0,898481
5	0,0187055	0,0221074	0,846119
6	0,0184606	0,0209946	0,879302
7	0,0179447	0,0186318	0,963122
8	0,0170196	0,0216682	0,785464
9	0,0191463	0,0226845	0,844025
10	0,0197366	0,0197366	1
11	0,0182029	0,0204480	0,890204
12	0,0186302	0,0205728	0,905574
13	0,0182537	0,0193271	0,944461
14	0,0188047	0,0206460	0,910815
15	0,0163015	0,0182250	0,894458
16	0,0171861	0,0203850	0,843075
17	0,0204331	0,0204331	1
18	0,0188220	0,0192009	0,980266
19	0,0202385	0,0206921	0,978078
20	0,0177298	0,0205610	0,862302

30 Nesil Sonunda GA ile BF Uygunluk Değerlerinin (1/T) Karşılaştırması



ÖZGEÇMİŞ

Ersan Aksoy, 1974 yılı Almanya doğumludur. Orta öğrenimini Trabzon Anadolu Lisesi'nde tamamladı. 1992 yılında İstanbul Teknik Üniversitesi Elektrik Elektronik Fakültesi Elektronik ve Haberleşme Mühendisliği bölümüne kayıt oldu. 1998 yılında aynı üniversitenin Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim dalı, Elektronik ve Haberleşme Mühendisliği Programı'nda yüksek lisansa başladı.

