

**AKILLI ULAŖIM SİSTEMLERİ İÇİN
YAPAY BAĐIŐIKLIK SİSTEMLERİ VE GENETİK
ALGORİTMA
İLE YENİ STOKASTİK EN KISA YOL
ALGORİTMALARININ GELİŐTİRİLMESİ**

Neslihan KARSLI

**Doktora Tezi
İnŐaat MühendisliĐi Anabilim Dalı
Doç. Dr. Sinan HINISLIOĐLU
2010
Her hakkı saklıdır**

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

AKILLI ULAŞIM SİSTEMLERİ İÇİN YAPAY BAĞIŞIKLIK
SİSTEMLERİ VE GENETİK ALGORİTMA İLE YENİ STOKASTİK EN
KISA YOL ALGORİTMALARININ GELİŞTİRİLMESİ

Neslihan KARSLI

İNŞAAT MÜHENDİSLİĞİ ANABİLİM DALI

ERZURUM

2010

Her hakkı saklıdır

Doç.Dr. Sinan HINISLIOĞLU danışmanlığında, Neslihan KARSLI tarafından hazırlanan bu çalışma 09.10.2010 tarihinde aşağıdaki jüri tarafından. İnşaat Anabilim Dalı'nda doktora tezi olarak kabul edilmiştir.

Başkan : Doç.Dr. SİNAN HINISLIOĞLU

İmza :

Üye : Doç.Dr. Darçın AKIN

İmza :

Üye : Doç.Dr.Mehmet AKTAN

İmza :

Üye : Yrd.Doç.Dr. Mehmet H. ÖZYAZICIOĞLU

İmza :

Üye : Yrd.Doç.Dr. Osman Ünsal BAYRAK

İmza :

Yukarıdaki sonucu onaylarım

Prof.Dr. Ömer AKBULUT
Enstitü Müdürü

ÖZET

Doktora Tezi

AKILLI ULAŞIM SİSTEMLERİ İÇİN YAPAY BAĞIŞIKLIK SİSTEMLERİ VE GENETİK ALGORİTMA İLE YENİ STOKASTİK EN KISA YOL ALGORİTMALARININ GELİŞTİRİLMESİ

Neslihan KARSLI

Atatürk Üniversitesi
Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Sinan HINISLIOĞLU

Akıllı Ulaşım Sistemlerinin (AUS) gelişmesine bağlı olarak gerçek zamanlı trafik yönetimi ve rota rehber modellerinde en kısa yol (EKY) bilgisine ihtiyaç duyulmaktadır. Klasik EKY algoritmaları kenar verilerini karşılaştırarak EKY bilgisine ulaştığından ağın geneli için ortaya konan stokastik problemlerde yetersiz kalmaktadır. Bu çalışmanın amacı stokastik bir ulaşım ağında EKY problemini Genetik Algoritma (GA) ve Yapay Bağışıklık Sistemi (YBS) yardımıyla çözerek Akıllı Ulaşım Sistemleri için ihtiyaç duyulan EKY bilgisini sağlamaktır.

Bu çalışmada farklı büyüklükteki ağlar için üç farklı GA ve YBS algoritması oluşturulmuştur. Ulaşım ağlarında her bir kenarın zaman ve yakıt değerleri için ihtimal dağılımı verilerek çevre şartlarından etkilenen stokastik bir yapı modellenmiştir. Stokastik bir ulaşım ağında EKY problemini çözmek için iki farklı karar kriterine göre modeller oluşturulmuştur. 1. Modelde verilen süre veya yakıtla gitme ihtimali en yüksek yol ve 2. Modelde ise hedef ihtimalle en az süre veya yakıtla gidilen yolu bulmak amaçlanmıştır. Öncelikle GA ve YBS'nin populasyon sayısı, nesil sayısı, çaprazlama olasılığı, mutasyon olasılığı ve klonlama oranı gibi parametrelerinin en uygun seviyelerini belirlemek için Taguchi Metodu kullanılmıştır. Önerilen modellerin etkinliğini göstermek için deterministik ağda EKY problemi GA ve YBS yöntemleriyle çözümlenerek elde edilen sonuçlar klasik EKY algoritmalarından olan Dijkstra ve Floyd algoritmaları ile karşılaştırılmıştır. Ayrıca her bir ağ, çok amaçlı EKY problemi olarak Pareto tekniği ile çözülmüştür. En az maliyetli yol, Pareto çözümleri arasında maliyet değerlendirilmesi yapılarak, AUS kullanıcılarına önerilmiştir. Sonuç olarak, önerilen GA ve YBS modellerinin stokastik EKY problemlerini çözmede etkili oldukları ve YBS modelinin problemi GA'dan daha kısa hesap süresinde çözdüğü görülmüştür.

2010, 173 Sayfa

Anahtar Kelimeler: En kısa yol problemi, akıllı ulaşım sistemleri, genetik algoritma, yapay bağışıklık sistemleri, pareto optimalitesi, Taguchi metodu, Dijkstra algoritması, Floyd algoritması, stokastik simülasyon.

ABSTRACT

Ph.D. Thesis

DEVELOPMENT OF NEW STOCHASTIC SHORTEST PATH ALGORITHMS BY ARTIFICIAL IMMUNE SYSTEMS AND GENETIC ALGORITHM FOR INTELLIGENT TRANSPORTATION SYSTEMS

Neslihan KARSLI

Atatürk University
Graduate School of Natural and Applied Sciences
Department of Civil Engineering

Supervisor: Assoc. Prof. Dr. Sinan HINISLIOĞLU

Based on the development of the intelligent transportation systems (ITS), the shortest path information is required for the real time traffic management and route guidance models. Classical shortest path algorithms are ineffective in the stochastic problems related to the whole network due to providing shortest path information by comparing the link data. The aim of this study is to provide the shortest path information in the stochastic network by Genetic Algorithm (GA) and Artificial Immune System (AIS) for ITS.

In this study, three different GA and AIS algorithms are built for the different size of networks. A stochastic structure affected by the environmental, road and traffic conditions and driver is modeled by giving the probability density functions for the time and fuel data concerning each edge. Models are designed according to two different decision criteria to solve the shortest path problem in a stochastic network. It is aimed to search that in the first model, the path which has highest probability for a given travel time or fuel consumption, in the second model, the path which has the lowest travel time and fuel consumption for a given target probability. First of all, Taguchi method is used to determine the most suitable levels of the factors such as number of population and crossover probability, mutation generation, crossover and mutation probability, and cloning ratio for GA and AIS. The results from GA and AIS models are compared to the ones from classical Dijkstra and Floyd algorithms for a deterministic network. After each network is also solved as multi-objective shortest path problem by Pareto technique, shortest path with minimum cost is recommended to ITS users by making cost evaluation among all pareto solutions. As a result, it is seen that proposed GA and AIS models are effective in solving the stochastic shortest path problems and, AIS model solves the problem in a shorter computing time than the GA model.

2010, 173 pages

Keywords: Shortest path problem, intelligent transportation systems, genetic algorithm, artificial immune system, Pareto optimality, Taguchi method, Dijkstra algorithm, Floyd algorithm, stochastic simulation.

TEŐEKKÜR

Bu tez alıőmasında bilgi ve deneyimleri ile yol gsteren, alıőmam boyunca desteęini benden esirgemeyen danıőman hocam Sayın Do.Dr. Sinan HINISLIOęLU'na teőekkürlerimi sunarım.

Sayın Do. Dr. Darın AKIN ve Do. Dr. Mehmet AKTAN'a alıőmamda yaptıkları deęerli katkılarından dolayı teőekkür ederim.

Ayrıca alıőmalarım boyunca sürekli özveride bulunan ve alıőabilmem iin gerekli ortamın oluőması iin elinden geleni yapan eőim Ebubekir TEPE'ye, varlıęıyla mutluluk veren oęlum Muhammet Kerem TEPE'ye ve desteklerini benden esirgemeyen anne ve babama sonsuz teőekkürlerimi bir bor bilirim.

Neslihan KARSLI

Eylül 2010

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	xiii
1. GİRİŞ.....	1
1.1. En Kısa Yol Problemi.....	2
1.1.1. Graf veri modeli	5
1.2. Kapsam.....	8
2. KAYNAK ÖZETLERİ	11
2.1. Akıllı Ulaşım Sistemleri	11
2.2. Genetik Algoritma.....	14
2.3. Yapay Bağışıklık Sistemleri (YBS)	20
2.4. Çok Amaçlı Optimizasyon	22
2.5. Kaynak Özetleri Değerlendirmesi	24
3. MATERYAL ve YÖNTEM.....	26
3.1. Materyal.....	26
3.1.1. Ağlar ve kenar bilgileri.....	26
3.1.2. EKY problemleri	36
3.1.3. Maliyet bilgileri	37
3.2. Yöntem	37
3.2.1. Dijkstra algoritması	38
3.2.2. Floyd algoritması.....	41
3.2.3 Genetik algoritma (GA).....	44
3.2.3.a. Değişken ve hedef fonksiyonların tanımlanması.....	46
3.2.3.b. Değişkenlerin sunumu	47
3.2.3.c. Populasyonun oluşturulması ve hedef değerlerinin bulunması	47
3.2.3.d. Eşlerin seçilmesi.....	48

3.2.3.e. Çaprazlama	50
3.2.3.f. Mutasyon.....	52
3.2.3.g. Yaklaşma denemesi	53
3.2.3.h. GA parametreleri	54
3.2.4. Yapay bağışıklık sistemleri	56
3.2.4.a. YBS'nin yapısı	58
3.2.4.b. Klonal seçim algoritması.....	60
3.2.5. Çok amaçlı optimizasyon	62
3.2.5.a. Çok amaçlı evrimsel algoritmalarda uygunluk ataması ve seçim	64
3.2.6. Taguchi metodu	67
3.3. Çalışmada Kullanılan Model ve Algoritmalar	70
3.3.1. Taguchi optimizasyonu	70
3.3.2. Algoritmalarda kullanılan uygunluk fonksiyonları	71
3.3.3. Çalışmada kullanılan algoritmalar.....	74
3.3.3.a. Çalışmada kullanılan GA'lar	75
3.3.3.b. Çalışmada kullanılan YBS algoritmaları.....	79
3.3.3.c. GA ve YBS'de kullanılan değişkenler	83
4. ARAŞTIRMA BULGULARI ve TARTIŞMA.....	84
4.1. Taguchi Optimizasyonu Sonuçları	84
4.1.1. GA parametreleri için Taguchi optimizasyonu	85
4.1.2. YBS parametreleri için Taguchi optimizasyonu	90
4.1.3. Taguchi optimizasyonu doğrulama deneyleri	96
4.2. EKY Değerlerinin Elde Edilmesi	98
4.2.1. 50 Düğümlü ağ	98
4.2.1.a. 50 düğümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi.....	98
4.2.1.b. 50 düğümlü ağda Pareto çözümlerinin elde edilmesi.....	103
4.2.2. 100 Düğümlü ağlar	108
4.2.2.a. 100 Düğümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi.....	108
4.2.2.b. 100 düğümlü ağda Pareto çözümlerinin elde edilmesi.....	118

4.2.3. 250 Dügümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi.....	125
4.2.4. 500 Dügümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi.....	126
4.2.5. 1000 Dügümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi	127
4.3. Çalışma Süresinin Değerlendirilmesi	128
4.3.1. GA ve YBS algoritmalarının çalışma zamanı açısından karşılaştırılması	128
4.3.2. GA ve YBS algoritma parametrelerinin işlem süresine etkileri.....	131
4.4. Tartışma.....	134
5. SONUÇ ve ÖNERİLER.....	137
5.1. Sonuçlar.....	137
5.2. Öneriler.....	139
KAYNAKLAR	140
EKLER.....	146
EK 1.	146
EK 2.	147
EK 3.	148
EK 4.	150
EK 5.	152
EK 6.	153
EK 7.	155
EK 8.	156
EK 9.	157
EK 10.	159
EK 11.	161
EK 11.	162
EK 12.	164
EK 13.	168
ÖZGEÇMİŞ	175

SİMGELER ve KISALTMALAR DİZİNİ

C	Mesafe matrisi
CI	Güven aralığı
D	Düğüm kümesi
DS	Düğüm sırası matrisi
df_e	Hatanın serbestlik derecesi
e_i	i. deneydeki rastsal hata
F	Tablo değeri
k	Amaç fonksiyonlarının kümesi
K	Kenarlar kümesi
I	Şehir sayısı
m	Performans değerinin genel ortalaması
n	Yineleme sayısı
N	Toplam deney sayısı
n_{eff}	Etkin tekrar sayısı
nt	Doğrulama deneylerindeki tekrar sayısını göstermektedir
O	İhtimal
R	Doğrulama deney sayısı
S/N	Sinyal/Gürültü
S	Sayısı
SD_{MSe}	Hata kareler ortalamasının serbestlik derecesi toplamı
t	Zaman
U	Uzaklık matrisi
V	Hız
V_e	Varyans değeri
V_t	Ortalama tahminde kullanılan parametrelerin toplam serbestlik derecesi
x	Uzunluk
X	Karar uzayı
X_i	Parametre-seviye kombinasyonunun sabit etkisi
y	Amaç vektörü

Y	Amaç uzayı
Y_i	i. deneyin performans istatistiđi
Y_o	Nominal arzu edilen deđer
z	Kısıtların kümesi

Kısaltmalar

AUS	Akıllı ulaşım sistemleri
BFS	Genişliğine ilk araştırma
DFS	Derinliğine ilk araştırma
EKM	En küçük maliyet
EKY	En kısa yol
GA	Genetik Algoritma
GPS	Cođrafik konumlandırma sistemi
VTPI	Victoria Ulaştırma Siyasetleri Enstitüsü
YBS	Yapay Bağışıklık Sistemi

ŞEKİLLER DİZİNİ

Şekil 1.1. AUS'nin faydaları (Vanderschuren 2003).....	2
Şekil 1.2. Graf türleri	6
Şekil 1.3. Grafın matris üzerinde tutulması	7
Şekil 1.4. DFS yöntemi	8
Şekil 1.5. BFS yöntemi	8
Şekil 3.1. Yönlü graf oluşturma algoritması	27
Şekil 3.2. 50 düğümlü ağın kenar mesafelerinin histogramı.....	28
Şekil 3.3. 100 düğümlü ağ için kenar mesafelerinin histogramı.....	28
Şekil 3.4. 50 düğümlü ağda ortalama kenar seyahat süreleri histogramı.....	29
Şekil 3.5. 100 düğümlük ağ için kenarların ortalama seyahat süreleri histogramı	29
Şekil 3.6. Yakıt tüketimi hız ilişkisi (Bakırcı 2005)	30
Şekil 3.7. Taşıt hızı-yakıt tüketimi ilişkisi	31
Şekil 3.8. 50 düğümlü ağda kenarların ortalama yakıt tüketimi histogramı	32
Şekil 3.9. 100 düğümlük ağda kenarların ortalama yakıt tüketimi histogramı	32
Şekil 3.10. 50 düğümlü ağda kenar mesafe değerleri	33
Şekil 3.11. 50 düğümlü ağda kenar seyahat süresi dağılım değerleri	34
Şekil 3.12. 50 düğümlü ağda kenar yakıt tüketimi dağılım değerleri	35
Şekil 3.13. Maliyetli graf	38
Şekil 3.14. Dijkstra algoritması, (1. Adım).....	39
Şekil 3.15. Dijkstra algoritması, (2. Adım).....	39
Şekil 3.16. Dijkstra algoritması, (3. Adım).....	40
Şekil 3.17. Dijkstra algoritması, (4. Adım).....	40
Şekil 3.18. Floyd algoritması için örnek graf.....	41
Şekil 3.19. GA'nın ilerleme adımları	45
Şekil 3.20. Rulet çemberi	49
Şekil 3.21. İkili kodlamada tek nokta çaprazlama	51
Şekil 3.22. Permutasyon kodlamada tek nokta çaprazlama	51
Şekil 3.23. İki nokta çaprazlama	51

Şekil 3.24. Çok nokta çaprazlama.....	52
Şekil 3.25. Üniform çaprazlama.....	52
Şekil 3.26. İkili kodlamada mutasyon.....	53
Şekil 3.27. Permutasyon kodlamada mutasyon	53
Şekil 3.28. Seçkinlik	55
Şekil 3.29. Arşivleme ile seçicilik.....	55
Şekil 3.30. Birincil, ikincil ve çapraz reaksiyon bağışıklık cevapları.....	57
Şekil 3.31. YBS için katmanlı çatısı	58
Şekil 3.32. Şekil uzayı gösterimi	59
Şekil 3.33. YBS algoritmaları	60
Şekil 3.34. Çok amaçlı optimizasyon problemi	63
Şekil 3.35. Pareto sıralama.....	66
Şekil 3.36. Amaç uzayında pareto optimal çözümler	67
Şekil 3.37. Budama operatörünün şematik gösterimi	75
Şekil 3.38. 50 düğümlü ağ için GA akış şeması	76
Şekil 3.39. 100 düğümlü ağ için GA akış şeması	77
Şekil 3.40. Büyük ağlar için GA akış şeması.....	78
Şekil 3.41. Büyük ağlarda algoritmanın ilerleme adımları	79
Şekil 3.42. 50 düğümlü ağ için YBS akış şeması	80
Şekil 3.43. 100 düğümlü ağ için YBS akış şeması	81
Şekil 3.44. Büyük ağlar için YBS akış şeması.....	82
Şekil 4.1. 50 düğümlük ağda GA parametrelerinin ortalama S/N etkileri.....	86
Şekil 4.2. 100 düğümlük ağda GA parametrelerinin ortalama S/N etkileri.....	87
Şekil 4.3. 250 düğümlük ağda GA parametrelerinin ortalama S/N etkileri.....	88
Şekil 4.4. 500 düğümlük ağda GA parametrelerinin ortalama S/N etkileri.....	89
Şekil 4.5. 1000 düğümlük ağda GA parametrelerinin ortalama S/N etkileri.....	90
Şekil 4.6. 50 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri.....	92
Şekil 4.7. 100 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri.....	93
Şekil 4.8. 250 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri.....	94
Şekil 4.9. 500 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri.....	95
Şekil 4.10. 1000 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri.....	96
Şekil 4.11. 50 düğümlü ağda rotaların kenar uzunluklarının	100

Şekil 4.12. 50 düğümlü ağda rotaların kenar seyahat sürelerinin karşılaştırılması.....	100
Şekil 4.13. 50 düğümlü ağda kenarlardaki yakıt tüketim miktarlarının karşılaştırılması	101
Şekil 4.14. 50 düğümlü ağda kenarlardaki ortalama taşıt hızlarının karşılaştırılması	101
Şekil 4.15. 50 düğümlü graf üzerinde rotaların gösterilmesi	102
Şekil 4.16. 50 düğümlü ağda problem 1 için Pareto çözümleri grafiği.....	104
Şekil 4.17. 50 düğümlü ağda problem 2 için Pareto çözümleri grafiği.....	105
Şekil 4.18. 50 düğümlü ağda problem 3 için Pareto çözümleri grafiği.....	107
Şekil 4.19. 100 düğümlü ağda rota 1 ve rota 2'nin kenar uzunluklarının karşılaştırılması	114
Şekil 4.20. 100 düğümlü ağda rota 3, rota 4 ve rota 5'in kenar uzunluklarının karşılaştırılması	114
Şekil 4.21. 100 düğümlü ağda rota 1, ve rota 2'nin kenar seyahat sürelerinin karşılaştırılması	115
Şekil 4.22. 100 düğümlü ağda rota 3, rota 4, ve rota 5'in kenar seyahat sürelerinin karşılaştırılması	115
Şekil 4.23. 100 düğümlü ağda rota 1, ve rota 2'nin kenar yakıt tüketimlerinin karşılaştırılması	116
Şekil 4.24. 100 düğümlü ağda rota 3, rota 4, ve rota 5'in kenar yakıt tüketimlerinin karşılaştırılması	116
Şekil 4.25. 100 düğümlü ağda rota 1, ve rota 2'nin ortalama taşıt hızlarının karşılaştırılması	117
Şekil 4.26. 100 düğümlü ağda Rota 3, Rota 4, ve Rota 5'in ortalama taşıt hızlarının karşılaştırılması	117
Şekil 4.27. 100 düğümlü ağda problem 1 için Pareto çözümleri grafiği.....	119
Şekil 4.28. 100 düğümlü ağda problem 2 için Pareto çözümleri grafiği.....	121
Şekil 4.29. 100 düğümlü ağda problem 3 için Pareto çözümleri grafiği.....	124
Şekil 4.30. Deterministik GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması	129
Şekil 4.31. Stokastik Model-1 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması	130

Şekil 4.32. Stokastik Model-2 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması	131
Şekil 4.33. 50 düğümlü ağ için GA ve YBS de modellerin işlem süresi açısından karşılaştırılması	131
Şekil 4.34. 50 düğümlü ağda GA parametrelerinin çalışma süresi üzerine etkileri.....	132
Şekil 4.35. 50 düğümlü ağda YBS parametrelerinin işlem süresi üzerine etkileri	133

ÇİZELGELER DİZİNİ

Çizelge 3.1. Ağ kenar sayıları	27
Çizelge 3.2. Kenar bilgileri	27
Çizelge 3.3. Rulet tekerleği oluşturulması (1. Adım)	49
Çizelge 3.4. Rulet tekerleği oluşturulması (2. Adım)	49
Çizelge 3.5. Rulet tekerleği oluşturulması (3. Adım)	50
Çizelge 3.6. Klonal seçim teorisinde kullanılan terimler	61
Çizelge 3.7. GA için kullanılan parametre ve seviye değerleri.....	70
Çizelge 3.8. YBS için kullanılan parametre ve seviye değerleri.....	70
Çizelge 3.9. GA için L_9 ortogonal dizisine göre yapılacak çalışma planı.....	71
Çizelge 3.10. YBS için L_9 ortogonal dizisine göre yapılacak çalışma planı	71
Çizelge 3.11. Çaprazlama metotları	74
Çizelge 3.12. Mutasyon metotları	74
Çizelge 4.1. GA için deterministik EKY değerleri	85
Çizelge 4.2. GA için S/N değerleri	85
Çizelge 4.3. 50 düğümlük ağda GA parametreleri için optimum seviyeler.....	86
Çizelge 4.4. 100 düğümlük ağda GA parametreleri için optimum seviyeler.....	87
Çizelge 4.5. 250 düğümlük ağda GA parametreleri için optimum seviyeler.....	88
Çizelge 4.6. 500 düğümlük ağda GA parametreleri için optimum seviyeler.....	89
Çizelge 4.7. 1000 düğümlük ağda GA parametreleri için optimum seviyeler.....	90
Çizelge 4.8. YBS için deterministik EKY değerleri	91
Çizelge 4.9. YBS için S/N değerleri	91
Çizelge 4.10. 50 düğümlük ağda YBS parametreleri için optimum seviyeler.....	92
Çizelge 4.11. 100 düğümlük ağda YBS parametreleri için optimum seviyeler.....	93
Çizelge 4.12. 250 düğümlük ağda YBS parametreleri için optimum seviyeler.....	94
Çizelge 4.13. 500 düğümlük ağda YBS parametreleri için optimum seviyeler.....	95
Çizelge 4.14. 1000 düğümlük ağda YBS parametreleri için optimum seviyeler.....	96
Çizelge 4.15. GA için doğrulama deneyleri.....	97
Çizelge 4.16. YBS için doğrulama deneyleri.....	97

Çizelge 4.17. 50 düğümlü ağda deterministik EKY değerleri	98
Çizelge 4.18. 50 düğümlü ağda stokastik EKY değerleri	98
Çizelge 4.19. 50 düğümlü ağda EKY rotaları	99
Çizelge 4.20. 50 düğümlü ağda Rota1 için kenar maliyet değerleri	99
Çizelge 4.21. 50 düğümlü ağda Rota2 için kenar maliyet değerleri	99
Çizelge 4.22. 50 düğümlü ağda problem 1 için Pareto çözümleri	103
Çizelge 4.23. 50 düğümlü ağda problem 1 in Pareto çözümlerinin maliyet hesabı.....	104
Çizelge 4.24. 50 düğümlü ağda problem 2 için Pareto çözümleri	105
Çizelge 4.25. 50 düğümlü ağda problem 2'nin Pareto çözümlerinin maliyet hesabı ..	106
Çizelge 4.26. 50 düğümlü ağda problem 3 için Pareto çözümleri	106
Çizelge 4.27. 50 düğümlü ağda problem3'ün Pareto çözümlerinin maliyet hesabı.....	107
Çizelge 4.28. 100 düğümlü ağda deterministik EKY değerleri	108
Çizelge 4.29. 100 düğümlü ağda stokastik EKY değerleri	108
Çizelge 4.30. 100 düğümlü ağda Rota 1 için kenar değerleri	109
Çizelge 4.31. 100 düğümlü ağda Rota 2 için kenar değerleri	110
Çizelge 4.32. 100 düğümlü ağda Rota 3 için kenar değerleri	111
Çizelge 4.33. 100 düğümlü ağda Rota 4 için kenar değerleri	112
Çizelge 4.34. 100 düğümlü ağda Rota 5 için kenar değerleri	113
Çizelge 4.35. 100 düğümlü ağda problem 1 için Pareto çözümleri	118
Çizelge 4.36. 100 düğümlü ağda problem 1 in Pareto çözümlerinin maliyet hesabı...	119
Çizelge 4.37. 100 düğümlü ağda problem 2 için Pareto çözümleri	120
Çizelge 4.38. 100 düğümlü ağda problem 2 nin Pareto çözümlerinin maliyet hesabı..	121
Çizelge 4.39. 100 düğümlü ağda problem 3 için Pareto çözümleri	122
Çizelge 4.40. 100 düğümlü ağda problem 3'ün Pareto çözümlerinin maliyet hesabı...	124
Çizelge 4.41. 250 düğümlü ağda deterministik EKY değerleri	125
Çizelge 4.42. 250 düğümlü ağda stokastik EKY değerleri	125
Çizelge 4.43. 500 düğümlü ağda deterministik EKY değerleri	126
Çizelge 4.44. 500 düğümlü ağda stokastik EKY değerleri	126
Çizelge 4.45. 1000 düğümlü ağda deterministik EKY değerleri	127
Çizelge 4.46. 1000 düğümlü ağda stokastik EKY değerleri	127
Çizelge 4.47. Deterministik GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması	128

Çizelge 4.48. Stokastik Model-1 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması.....	129
Çizelge 4.49. Stokastik Model-2 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması.....	130
Çizelge 4.50. GA parametrelerinin işlem süresine etkisi.....	132
Çizelge 4.51. YBS parametrelerinin program çalışma süresine etkisi.....	133

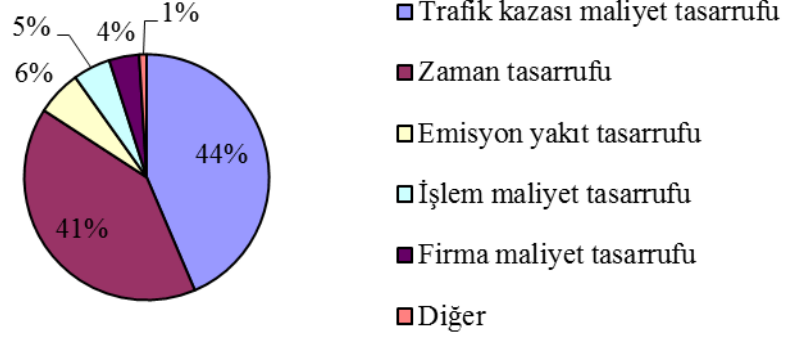
1. GİRİŞ

İnsan ve eşya ulaşımının zamanında ve güvenilir bir şekilde gerçekleştirilmesi doğal bir beklentidir. Nüfus artışı, şehirleşme oranının artması ve taşıt sayısında meydana gelen artış gibi nedenler ulaşım ağlarında gittikçe büyüyen tıkanıklık problemleri oluşturmuştur. Tıkanıklık, ulaşım altyapısının gelen talepleri karşılayacak kapasiteye sahip olmaması durumunda meydana gelir. Tıkanıklığın artması beraberinde seyahat süresinin, hava kirliliğinin ve yakıt tüketiminin artması ve ulaşım alt yapısının etkinliğinin azalması sorunlarını meydana getirmektedir. Ayrıca hazır mamullerin ve ham maddelerin yerlerine zamanında ulaştırılması için gerekli olan ulaşım alt yapısının güvenilirliğini azaltarak ticareti etkilemektedir. Bu durum ülke ekonomisinde kayıplara yol açmaktadır.

Tıkanıklığı önlemek için mevcut yolların şeritleri artırılarak ya da yeni yollar inşa edilerek alt yapı kapasitesini artırma yoluna gidilir. Fakat çevresel şartlar ve bütçedeki kısıtlamalar alt yapıyı büyütme çalışmalarını sınırlar. Sonuç olarak ya mevcut alt yapı için gelen talepler değiştirilerek ya da mevcut alt yapıyı daha etkin bir şekilde kullanan trafik yönetim çözümleri kullanılarak tıkanıklık probleminin üstesinden gelinmeye çalışılır (Lin 2008).

Son yıllarda kara ulaşım sistemlerinin kapasite, güvenlik ve etkinliğini artırmak için Akıllı Ulaşım Sistemleri (AUS) kullanılmaya başlanmıştır. AUS, seyahat etkinliğini ve hareketliliğini geliştirmek, güvenliği artırmak, enerji kayıplarını azaltmak, ekonomik yararlar sağlamak ve çevreyi korumak gibi önemli amaçlar güder. Bu amaçları gerçekleştirmek için gelişmiş bilgi işleme, iletişim, algılama, elektronik ve bilgisayar kontrol teknolojileri kullanılır. Özellikle AUS'nin bir parçası olan Taşıt Rota Rehber Sistemleri insanların veya eşyanın varacağı noktaya en iyi nasıl ulaşacağı hakkında rehberlik eder. Böylece güvenilir, hızlı ve ekonomik ulaşım imkanı yanında trafik tıkanıklığının, gecikmelerin, yakıt tüketiminin, hava ve gürültü kirliliğinin azalması sağlanır.

Amerika Birleşik Devletlerinde 1996-2015 yıllarını kapsayan AUS projelerinden beklenen faydalar Şekil 1.1’de özetlenmiştir (Vanderschuren 2003).



Şekil 1.1. AUS’nin faydaları (Vanderschuren 2003)

AUS’nin gelişmesi ile gerçek zamanlı trafik yönetimi ve rota rehber modellerine duyulan ihtiyaç ulaştırma mühendisliğinde en kısa yol (EKY) problemlerinin çözümüne olan ihtiyacı artırmıştır. AUS’nin temel fonksiyonlarından birisi ağdaki mevcut trafik şartları hakkındaki bilgiyi kullanarak tüm sistem kullanıcılarının seyahat zamanını optimize etmektir. Bunu yaparken trafik ağlarından ham bilgi toplanır, bilgisayar ya da merkezi kontrolör tarafından işlenir ve kullanıcının anlayacağı şekilde genellikle rota bilgisi olarak sunulur (Ziliaskopoulos *et al.* 1997).

1.1. En Kısa Yol Problemi

EKY problemi başlangıç ile varış noktası arasındaki alternatif yollardan minimum mesafe, zaman ya da maliyete sahip yolu bulmak olarak tanımlanır. EKY problemi, şebeke akış optimizasyon teorisindeki en temel problemlerden birisidir. Şebeke akış problemi bir başlangıç noktasından varış noktasına en yüksek akışın sağlanması ihtiyacından ortaya çıkmıştır. Uygulama birçok problemin modellenmesinde kullanılmaktadır. EKY problemleri hakkında 1950 yıllarının sonlarından bu yana birçok bilimsel çalışma yapılmıştır. Ulaşım, yönlendirme, iletişim, bilgisayar ve tedarik zinciri yönetimi gibi birçok alanda uygulama sahası bulmuştur (Hernandes 2007).

Karayolu ulařımında sűrűcűlerin ve yolcuların gidilecek herhangi bir noktaya EKY'den gitme talebi, EKY problemini ortaya ıkarmıřtır. Ayrıca son yıllarda AUS'nin alt dalı olan İleri Seyahat Bilgi Sistemleri, Tařıt Rota ve Rehber Sistemleri, Acil Olay Yűnetim Sistemleri ve Nakliye ve Filo Yűnetim Sistemlerinde meydana gelen geliřmelerden dolayı EKY bilgisine ihtiya duyulmaktadır.

Bir olay veya ifadenin dűğűm ve izgiler kullanılarak gűsterilmesine graf veri modeli denmektedir. Dűğűmler varlıkları izgiler ise varlıklar arasındaki iliřkiyi temsil etmektedir. izgiler kenar olarak da isimlendirilirler. Bilgisayar uygulama alanında ve bilimsel-teknik modellemelerde graflar olduka fazla kullanılmaktadır. Őrneğın bir űhrin yol akıř kontrolű, ime suyu alt yapısı, bilgisayar ağı ve ulařım tűrlerinin hemen hepsinde rota problemleri graf veri modeli ile gűsterilebilir. Graf veri modeli EKY problemine algoritmik bir űzűm sunmaktadır.

EKY problemleri ağılarda kenarlara atanan bilginin tűrűne gűre genel olarak deterministik ve stokastik olmak űzere ikiye ayrılır. Deterministik EKY problemlerinde ağıdaki kenar maliyetlerinin sabit olduėu kabul edilir. evre űartlarından etkilenmez. Stokastik EKY problemlerinde kenar bilgisi deėiřkendir. űnkű gerek dűnyada sűrekli deėiřim sűz konusudur. Bu deėiřime paralel olarak kenar maliyetleri de sűrekli deėiřmektedir. Őrneğın kenar maliyetinin seyahat sűresi olarak dűřűnűlen bir yolda seyahat sűresi hava űartlarına ve tařıtların yoėunluėuna baėlı olarak gűn ierisinde sűrekli deėiřmektedir.

EKY problemlerini űzmek iin geliřtirilmiř birok klasik yűntem mevcuttur. Bunlara Dijkstra, Floyd ve Bellman Ford Algoritması űrnek verilebilir. Dijkstra algoritmasında bir bařlangı dűğűműnden diėer tűm dűğűmlere olan EKY belirlenirken Floyd algoritmasında bűtűn dűğűmlerden her bir dűğűme EKY belirlenir.

Klasik yűntemler bűtűn bir ağıdaki her bir dűğűmű tek tek inceleyerek sonuca ulařır. Fakat klasik yűntemler stokastik ağı yapılarında yetersiz kalmaktadır. Bu problemi űzmek iin son yıllarda EKY problemlerinin űzűműnde doėadan esinlenerek ortaya

çıkılmış sezgisel yöntemler kullanılmaktadır. Bu yöntemlere Genetik Algoritmalar (GA), Karınca Kolonileri Optimizasyonu, Tabu Arama ve Yapay Bağışıklık Sistemleri (YBS) örnek olarak verilebilir.

GA, doğal seçim yöntemine dayanan bir arama ve optimizasyon yöntemidir. Çözüm uzayının tamamını değil sadece bir bölümünü tararlar böylece etkin arama yaparak daha kısa sürede çözüme ulaşırlar. GA bir çözüm uzayındaki her noktayı, kromozom olarak kodlarlar. Çözüm uzayında birden fazla noktada aynı anda çalışıldığından populasyon adı verilen bir kromozom kümesi mevcuttur. Populasyondaki her bir kromozomun değeri uygunluk fonksiyonu ile hesaplanır. GA her nesilde çaprazlama ve mutasyon gibi genetik operatörleri kullanarak yeni bir populasyon oluşturur. Birkaç nesil sonunda, populasyon daha iyi uygunluk değerine sahip bireyler bulundurur (Jang 1997).

YBS, insandaki bağışıklık sisteminden taklit edilerek ortaya çıkarılmış sezgisel bir yöntemdir. Bağışıklık sistemlerinin temel özellikleri olan öğrenme, hatırlama, sınıflandırma ve algılama özelliklerini taklit ederler. YBS, GA ile benzerlik göstermektedir. Çözüm uzayının kodlanması, uygunluk fonksiyonu kullanılması ve mutasyon operatörünün varlığı GA ile ortak yönleridir. YBS de çaprazlama operatörü kullanılmamaktadır. Örüntü tanıma, iş programlama, robotik veri analizi gibi birçok alanda başarıyla uygulanmıştır (Castro and Zuben 2000).

EKY problemlerinde şayet optimize edilecek tek bir amaç varsa problem tek amaçlı, birden fazla amaç bulunması halinde ise çok amaçlı optimizasyon problemi olarak adlandırılmaktadır. Gerçek hayattaki birçok problem genellikle birbiriyle çelişen birden fazla amacın eş zamanlı optimizasyonundan oluşmaktadır. Amaçların her birini aynı anda sağlayan tek bir çözümün bulunması her zaman mümkün olmayabilir. Bu durumda problem bilgisine sahip olan karar vericiden, amaçların her biri için kabul edilebilir düzeyde olan alternatif çözümlerden seçim yapması istenir. Bu çözümlerin her birine Pareto-Optimal çözüm denir (Sağ ve Çunkaş 2009).

GA ve YBS de kurulan modelin başarısını etkileyen birçok parametre bulunmaktadır. Bu parametrelerin optimum seviyesini belirlemek için deney tasarımı teknikleri uygulanmalıdır. Tam faktöriyel deney tasarımında, bir faktör değiştirilirken diğer bütün faktörler sabit tutulur. Faktör ve seviye sayısının fazla olması durumunda deney sayısı çok fazladır ve pratikte yapılması da mümkün değildir. Aynı zamanda eğer faktörler arasında iç etkileşim varsa, geleneksel deney tasarımına göre bulunan optimum şartlar, gerçek optimum şartlar olmayabilir. Deney sayısını azaltmak için birçok deney tasarımı önerilmiştir. Bu deney tasarım teknikleri arasında, Taguchi Metodu sistematik dizaynlarda parametre optimizasyonu için başarılı şekilde uygulanmaktadır (Hınıslıoğlu and Bayrak 2004).

1.1.1. Graf veri modeli

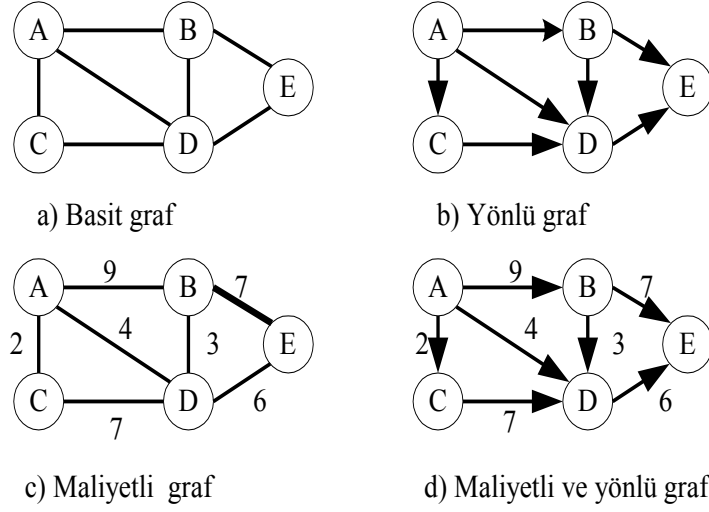
Graf, düğümler ve düğümlere bağlı kenarlardan oluşmaktadır. Buna göre bir modelde bulunan varlıklar düğümler ile bu varlıklar arasındaki ilişkiler ise kenarlar ile ifade edilmektedir. Örneğin bir ulaşım ağında şehirler düğümleri, şehirleri birbirine bağlayan yollar ise kenarları temsil edebilir. Düğümler kümesi (D), Kenarlar kümesi (K) olmak üzere graf veri modeli matematiksel olarak aşağıdaki şekilde ifade edilir (Çölkesen 2004).

$$G = (D, K)$$

$$D = \{d_0, d_1, d_2, \dots, d_n\}$$

$$K = \{k_0, k_1, k_2, \dots, k_n\}$$

Graflar, kenarlarının sahip olduğu özelliklere göre çeşitli şekillerde isimlendirilirler. Eğer bir graf üzerindeki tüm kenarların maliyeti bir veya birbirine eşitse ve yön bilgisi yoksa basit graf olarak isimlendirilir. Graf üzerindeki kenarlar bağlantının nereden başlayıp nerede sonlanacağını belirten yön bilgisine sahipse yönlü graf adını alır. Graf üzerindeki kenarların ağırlıkları eşit değilse ve her biri farklı bir değer alabiliyorsa bu graf maliyetli graf olarak isimlendirilir. Maliyetli ve yönlü graflar kısaca ağ olarak adlandırılır. Bu belirtilen graf türleri Şekil 1.2'de gösterilmektedir (Çölkesen 2004).

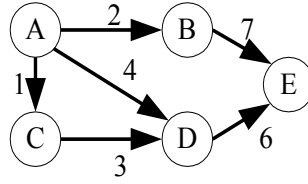


Şekil 1.2. Graf türleri

Bir grafın bilgisayara tanıtılması ve bellekte tutulması için dört farklı yöntem vardır. Bunlar;

- Matris üzerinde,
- İki dizi üzerinde,
- Bağlantılı liste ile,
- Dizili bağlantılı liste ile.

Bu yöntemlerden hangisinin kullanılacağı problemin türüne ve grafın seyrek veya yoğun olmasına göre farklılık göstermektedir. Bazen ortaya çıkan ihtiyaca göre bu yöntemler birleştirilerek uygulanabilir. Bunlardan en kolay ve yaygın olanı matris üzerinde ifade etmektir. Grafın matris üzerinde tutulmasının en önemli avantajı istenilen bağlantı bilgisine anında erişilmesidir. Matris üzerinde ifade edilirken komşuluk matrisi kullanılmaktadır. Komşuluk matrisi, düğümlerden düğüme bağlantıyı gösteren kare bir matristir. Şekil 1.3'te maliyetli ve yönlü grafın matris üzerinde tutulmasına örnek verilmiştir.



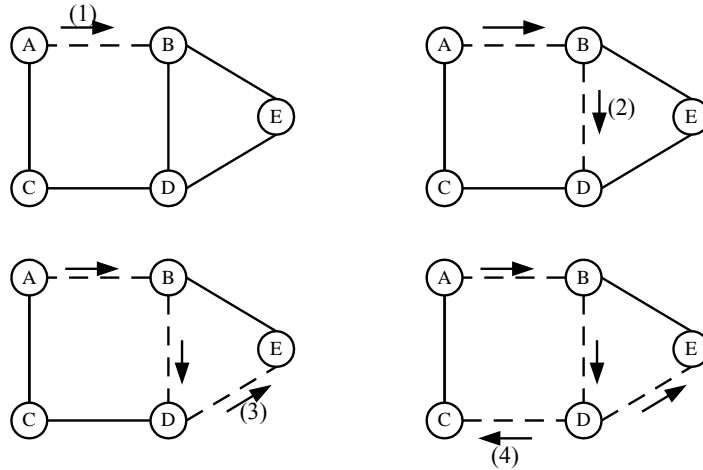
Komşuluk matrisi

	A	B	C	D	E
A	-	2	1	4	-
B	-	-	-	-	7
C	-	-	-	3	-
D	-	-	-	-	6
E	-	-	-	-	-

Şekil 1.3. Grafın matris üzerinde tutulması

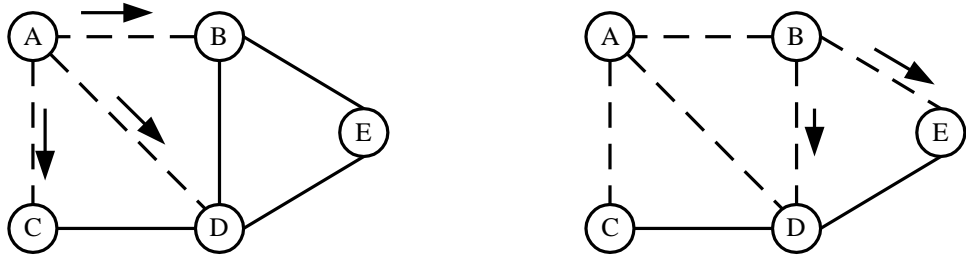
Graf problemlerini çözmek için geliştirilen graf algoritmaları, graf üzerinde dolaşma yöntemleri ve Greedy yaklaşımı oluşturmaktadır. Graf üzerinde dolaşma, grafın düğümleri ve kenarları üzerinde problemi çözecek şekilde hareket etmektir. Örneğin, tüm düğümleri en az maliyetle dolaşma, iki düğüm arasında en kısa yolu bulma, gezi yapma, en yüksek akış kapasitesini hesaplama gibi konular graf üzerinde dolaşma konusuna girer. Graf üzerinde dolaşma yapan birçok yöntem vardır. Bunlardan en önemli iki tanesi DFS (Depth First Search) ve BFS (Breadth First Search) yöntemleridir (Çölkesen 2004).

DFS yöntemi önce derinlik araması olarak nitelendirilir. Başlangıç düğümünün bir kenarından başlayıp, gidilebilecek en uzak düğüme kadar bu davranış sürdürülür. İşlem sonunda hiç uğranılmayan düğüm varsa bir geriye dönülerek uğranılmayan düğüme gidilir. Bu yöntemin ilerleme sırası Şekil 1.4 üzerinde gösterilmiştir.



Şekil 1.4. DFS yöntemi

BFS yöntemi önce genişlik araması olarak nitelendirilir. DFS yönteminin tersine başlangıç düğümünden gidilebilecek tüm komşu düğümlere gidilmesiyle başlanır. Bu komşu düğümlerden biri seçilerek o düğüme bağlı diğer düğümlere gidilir. Bu işlem en son düğüme ulaşıncaya kadar devam eder. Bu yöntemin ilerleme yönü Şekil 1.5 üzerinde gösterilmiştir (Çölkesen 2004).



Şekil 1.5. BFS yöntemi

1.2. Kapsam

Bu çalışmada farklı büyüklükteki ağlar için EKY problemi GA ve YBS kullanılarak tek amaçlı ve çok amaçlı problemler için çözülmüştür.

Tek amaçlı problemler deterministik ve stokastik olmak üzere iki grupta toplanmıştır. Deterministik problemlerde “en kısa mesafeli yol”, “en kısa ortalama seyahat süreli yol” ve “en az ortalama yakıt tüketilen yol” hangisidir sorusuna cevap aranmıştır.

Stokastik problemlerde ulaştırma sistemlerinde bulunan bir takım belirsizliklerden dolayı ağın kenar bilgisi olarak ihtimal dağılım bilgisi kullanılmıştır. Stokastik problemler için farklı karar kriterlerine göre 2 ayrı model ortaya konulmuştur. 1. modelde, verilen bir T_0 (tüketilen yakıt miktarı ya da seyahat süresi) değerinden daha az değerde gidilme ihtimali en yüksek yol bilgisine ulaşılırken, 2. modelde, verilen güven seviyesinde en az seyahat süresinde veya en az yakıt tüketilerek gidilen yol bilgisine ulaşılmaktadır.

Bu çalışmada EKY problemi maliyetli ve yönlü graflar (ağlar) için geliştirilmiştir. Düğüm sayıları 50, 100, 250, 500 ve 1000 olmak üzere beş tür ağ üzerinde EKY problemi çözülmüştür. Bu beş farklı ağı oluşturmak için MATLAB’da başlangıçta verilen düğüm sayısı ve bağlantı derecesi verilerine göre rastgele ulaşım ağı oluşturan bir program yazılmıştır. Bu ağlardan 50 düğümlü ağ “küçük”, 100 düğümlü ağ “orta” ve 250, 500 ve 1000 düğümlü ağlar büyük ağ olarak nitelendirilmiştir.

Küçük, orta ve büyük ağlar için operatörlerin kullanımında bir takım farklılıklar oluşturularak üç farklı GA ve YBS algoritmaları MATLAB’da oluşturulmuştur. Büyük ulaşım ağlarında kurulan modellerin doğru sonuca yaklaşmasını sağlamak için ara hedef düğümler belirlenmiştir. Büyük ağlarda ara başlangıç ve hedef düğümleri ile bağlantısı olmayan düğüm ve kenarların budanması için ayrıca bir algoritma geliştirilmiştir. Böylece kurulan modellerin gereksiz alanlarda çalışılmasının önüne geçilerek etkinliği artırılmıştır.

GA’nın mutasyon oranı, çaprazlama oranı, popülasyon sayısı ve nesil sayısı ve YBS’nin mutasyon oranı, klonlama oranı, popülasyon sayısı ve nesil sayısı parametrelerinin uygun seviyelerini belirlemek için Taguchi Metodu kullanılmıştır. Kurulan modellerin doğruluğunu göstermek için deterministik bir ağda EKY problemi klasik EKY algoritmalarından olan Dijkstra ve Floyd Metodu ile çözümlenerek elde edilen değerler GA ve YBS ile elde edilen EKY bilgisiyle karşılaştırılmıştır.

Çalışmada ayrıca GA ve YBS ile çok amaçlı optimizasyon gerçekleştirilmek üzere üç tür problem belirlenmiştir. Birinci problem “en az seyahat sürelili ve en az mesafeye sahip yol hangisidir?”, ikinci problem “en az yakıt tüketilen ve en az mesafeye sahip yol hangisidir?” ve üçüncü problem “en az yakıt tüketilen ve en az seyahat sürelili yol hangisidir?” olarak belirlenmiştir. Belirlenen bu problem türlerine göre çok amaçlı optimizasyon gerçekleştirilerek Pareto EKY çözümleri elde edilmiştir. Elde edilen pareto çözümlerinden hangisinin seçileceği konusunda AUS kullanıcısına yol göstermek adına maliyet değerlendirilmesi yapılmıştır.

Bu çalışmanın ikinci bölümünde kaynak özetlerine yer verilmiştir. EKY problemleri ile ilgili yapılan kaynak taraması AUS, GA, YBS ve Çok Amaçlı Optimizasyon olmak üzere dört başlık altında sunulmuştur. Üçüncü bölümde çalışmada kullanılan materyal ve kullanılan yöntemler anlatılmıştır. Çalışmanın materyal bölümü EKY probleminin çözüldüğü ağlar ve kenar bilgilerini kapsamaktadır. Yöntem bölümünde ise Dijkstra ve Floyd Algoritmaları, GA, YBS, Taguchi Optimizasyonu ve Çok Amaçlı Optimizasyon tanımlanmıştır. Dördüncü bölümde araştırma bulguları ve tartışma yer almaktadır. Beşinci bölümde ise gerçekleştirilen çalışmadan elde edilen sonuçlar ve öneriler sunulmuştur.

2. KAYNAK ÖZETLERİ

Bu bölümde EKY problemi ile ilgili literatür taramasına değinilmiştir. Amacımız EKY probleminin çözümü hakkında literatürde var olan çalışmalarını değerlendirerek kendi çalışmamızın bilime katkısını ortaya koymaktır. Bu bağlamda EKY problemlerinin çözümünde kullanılan AUS, GA, YBS ve Çok Amaçlı Optimizasyon yöntemleri ile ilgili şimdiye kadar yapılan çalışmalar sunulmuştur.

2.1. Akıllı Ulaşım Sistemleri

Bilgisayar teknolojisindeki gelişmelerin ulaştırma sektörünü etkilemesine bağlı olarak özellikle son yıllarda AUS adı altında bilişime dayalı sistemler geliştirilmiştir.

Shimizu and Kobayashi (1994) tarafından sürücünün o anki pozisyonundan varacağı noktaya kadar en kısa ortalama seyahat süresine sahip rotayı veren rota rehber algoritması önerilmiştir. Sinyal devresi, yeşil süre ve tekrarlanma süresi olmak üzere üç adet sinyal kontrol parametresi kullanılmıştır. Bu parametreler trafik ağında trafik tıkanıklık uzunluğunun toplamını minimize etmek için sistematik ve adaptif olarak kontrol edilmiştir. Ortalama seyahat süresinin yol akış şartları üzerinde oldukça etkili olduğu gözlenmiştir. Rota rehber algoritmasının sürücüler için kabul edilebilir rotalar önerdiği ve trafik sıkışıklığını azaltmada etkili olabileceği tespit edilmiştir.

Ziliaskopoulos *et al.* (1997), gerçek zamanlı AUS uygulamalarında kullanılabilen zamana bağlı en kısa zamanlı yol algoritmalarının paralel tasarımı konusunda çalışmışlardır. İki adet paylaşımli hafıza ve 1 adet mesaj geçme algoritması kodlanarak gerçek ve rastgele ağlarda performans etkisi gösterilmiştir.

Cavit (1998), büyük ulaşım graflarında AUS ve coğrafik bilgi sistemleri (CBS) kapsamında kullanılmak üzere yönlendirme için etkili algoritmalar önermiştir. Büyük ölçekli ağlarda ön işlem için gerekli olan hafıza sınırlı olacağı yada başlangıç hesapları için çok zaman gerekli olduğu belirtilerek bu durumun üstesinden gelmek için çalışmada öncelikle hiyerarşik data yapısı kullanan statik bir algoritma önerilmiştir. Bunu başarmak için genel hedef parçalama tekniği uygulanmıştır. Algoritma Los angles ulaşım sistemlerinin graflarında test edilerek bellek ihtiyacı noktasında bilinen algoritmalarından daha iyi sonuçlar verdiği belirlenmiştir. Geliştirilen ikinci algoritmada değişen trafik şartları uyarlanmıştır. Bellek ihtiyacının ve ön işlem süresinin diğer tekniklere göre daha az miktarlarda olduğu tespit edilmiştir.

Subramanian (1999), akıllı ulaştırma ağları için dinamik yönlendirme algoritmaları geliştirmiştir. Trafik tıkanıklığının ekonomik, sosyal ve politik etkilerinin olduğu ve trafik yönetiminde oldukça büyük problemler oluşturduğu ileri sürülerek kabul edilebilir seviyelerde tıkanıklığı azaltmak için bir takım önlemler ve karma teknikler araştırma görevi AUS araştırma ve geliştirme programlarında gerçekleştirildiği belirtmiştir. Çalışmanın temel amacı, akıllı ulaşım ağlarında pratik yönlendirme aracı olarak zamana bağlı EKY algoritmaları ve k. EKY algoritmalarının etkinliğini göstermek olarak ifade edilmiştir. Dinamik ağlar için çoklu optimal rotalar hesap eden sezgisel bir algoritma önerilmiştir. Algoritmanın hesapsal etkinliğini göstermek için farklı büyüklükte ve yoğunluktaki ağlar üzerinde istatistiksel ve regresyon testleri yapılmıştır. Sonuçlar algoritmanın EKY'yi hesaplamada performansının etkili olduğunu ve gerçek zamanlı uygulamalarda kullanılabileceğini göstermiştir.

Vanderschuren (2003), alınan tedbirlerin etkisini tahmin etmenin AUS'nin yeni bir alan olması ve uygulama alanının sınırlı olmasından dolayı güç olduğunu ileri sürerek AUS için bir takım modeller sunmuş ve bu modellerin beklenen etkilerini kurgulanan çalışmalara göre ortaya koymuştur. Mikro simülasyona dayalı AUS modellerinin etkilerinin tahmini oldukça ümit verici olarak bulunmuştur. Özellikle seyahat zamanının azalması ve yolun kapasitesinin artması ulaşılabilir hedef olarak görülmüştür.

Taniguchi and Shimamoto (2004), çalışmalarında sürekli değişken olarak seyahat zamanı bilgisini içeren gerçek zamanlı trafik bilgisini kullanarak dinamik taşıt yönlendirme ve listeleme yöntemi sunmuşlardır. Dinamik trafik simülasyonu seyahat zamanlarını güncellemek için kullanılmıştır. Model test yol ağında uygulanmıştır. Tahmin modelleriyle karşılaştırıldığında, değişken zamana dayalı gerçek zamanlı bilgiyi kullanan dinamik taşıt yönlendirme ve listeleme prosedürünün uygulanmasının toplam maliyeti ve taşıtların çalışma süresini de azaltmakta olduğu belirlenmiştir.

Velega *et al.* (2009), topolojik harita eşleştirme algoritmalarının performansını artırmak için bir takım optimizasyon teknikleri kullanmışlardır. Gerçek hayattaki alan bilgilerinden elde edilen ağırlık bilgisiyle, ağırlıklı topolojik harita eşleştirme algoritmaları geliştirmişlerdir. Özellikle kavşaklarda eşleştirme performansını artırmak için dönüş kısıtı ve kenar bağlantıları olmak üzere iki tür ağırlık algoritmaya dahil edilmiştir. Yanlış eşleştirmeleri azaltmak için uygunluk prosedürü eklenmiştir. İleri harita eşleştirme algoritması yoğun şehir ve şehir dışındaki yerleşim alanlarından elde edilen alan bilgisi ile test edilmiştir. Algoritma elde edilen alan bilgileri ile bağlantıları yoğun şehir merkezlerinden Washington DC ve Londra %96,8 ve %95,3 oranında, şehir dışındaki yerleşim alanlarında ise %96,71 doğruluk oranında tanımlamıştır. Bu çalışma mevcuttaki topolojik harita eşleştirme algoritmalarına üstünlük sağlamıştır. Ayrıca piyasadaki mevcut çoğu AUS hizmetlerinin navigasyon modüllerini destekleme potansiyeline sahiptir.

Miller (2009), AUS için en hızlı yolları hesaplayan dinamik bir algoritma geliştirmiştir. Bu çalışmada ayırık trafik konumlarından ziyade sürekli akış bilgisini dikkate alarak AUS için bilgi toplama yöntemine yeni bir yaklaşım sunmuştur. Döngü dedektörleri ve video kameralar ile bilgi toplamak için temel yöntemler sağlanmıştır.

Zhu (2010), dinamik otoyol yönlendirme sistemleri için yeni yönlendirme metotlarının dizaynını ve geliştirilmesi konusunda çalışmıştır. Çalışmasında mevcut yönlendirme algoritmalarının seyahat süresini ya da mesafeyi minimize etmek olduğu ifade edilerek yakıt maliyetini ve taşıt emisyonunu minimize eden çevreyle dost yönlendirme

stratejilerinin dikkate alınması gerektiğine dikkat çekilmiştir. Yol tiplerinin taşıt emisyonu ve yakıt tüketiminde önemli rol oynadığı belirtilerek yol tipi sınıflandırma algoritması geliştirilmiştir. Önerilen modelin, yalnızca rutin trafik olaylarında kullanılmakla sınırlı kalmayacağı aynı zamanda meydana gelebilecek acil olaylarda insanların daha güvenli bölgelere tahliyesi için kullanılabilmesi önerilmiştir.

2.2. Genetik Algoritma

GA, yapay zeka hesaplama tekniğinin bir parçasını oluşturmaktadır. EKY problemlerinin çözümünde başarıyla uygulanmışlardır.

Gomez–Albarran *et al.* (1997), çalışmalarında GA kullanarak klasik yönlendirme algoritmalarından olan Lee Algoritmasını geliştirmeyi amaçlamışlardır. GA ve Lee Algoritması birleştirilerek GALO modeli önerilmiştir. Sonuç olarak GALO Algoritması ile Lee Algoritmasına göre araştırma uzayının %30-%50 arasında azalmasını sağlamışlardır.

Jung (2000), zamana bağlı taşıt yönlendirme problemleri için matematiksel bir model önermiş ve problemi çözmek için GA kullanmıştır. Problemin formülasyonunda farklı kapasiteli taşıtlar, zaman sınırlı talep dağıtımı, gerçek zamanlı servis ihtiyaçları, talep düğümleri arasında seyahat zamanlarındaki gerçek zamanlı değişimler dikkate alınmıştır. Bu çalışmada taşıtların maliyeti, müşteri uygunsuzluğu maliyeti ve yönlendirme maliyeti bileşenlerinden oluşan toplam maliyetin minimize edilmesi amaçlanmıştır. Bir ağ üzerinde zamana bağlı ve deterministik en kısa yollar hesap edilmiş sonuç olarak zamana bağlı en kısa yol planının daha düşük maliyetli olduğu belirlenmiştir.

Gomez-Sanchez (2001), çalışmasında EKY problemini çözmek için yeni bir algoritma geliştirmiştir. Algoritmanın temel amacı çok seçenekli kararların varlığında takip edilebilecek en iyi rotaları belirlemektir. Algoritma çözümü stokastik elemanlar içeren ağ modeline uygun simülasyon ortamında gerçekleştirilmiştir. Yollar ve karşılık gelen

maliyetler hedef düğüme ulaşıncaya kadar düğümden düğüme eklenerek elde edilmiştir. Algoritmada ağı tanımlanması için dinamik bir yapının kullanılması, sabit uzunluklu vektör kullanımını benimseyen GA'lara göre kromozom büyüklüğünü, muhtemel kombinasyonların sayısını ve araştırma sayısını azaltarak etkili bir çalışma sağlamıştır. Elde edilen sonuçlar, daha küçük populasyonlarda doğru EKY çözümlerine ulaşılmışının mümkün olduğunu göstermiştir.

Ahn and Ramakrishna (2002), çalışmalarında EKY yönlendirme problemleri için GA yaklaşımı sunulmuştur. Problemi kodlamak için değişik uzunluktaki kromozom ve genler kullanılmıştır. Önerilen algoritmada tamir fonksiyonu kullanılarak uygun olmayan çözümler düzeltilerek uygun hale getirilmiştir. Çaprazlama ve mutasyon operatörleri algoritmaya çözüm kalitesinin geliştirilmesi ve sonuca daha fazla yakınsaması gibi önemli faydalar sağlamışlardır. Bu çalışmada ayrıca istenilen kalitedeki çözüme ulaşmayı sağlayacak populasyon boyutlandırma denklemi kullanılmıştır. Denklem daha fazla yakınsama ve genelleştirmeye dayandırılmıştır. Denklem populasyon büyüklüğü, çözüm kalitesi ve algoritmada kullanılan diğer parametrelere bağlıdır. Bilgisayar simülasyonları ile kullanılan algoritmanın başka algoritmalara göre daha iyi çözüm kalitesine sahip olduğunu ve optimal çözümlere daha iyi yakınsadığı gösterilmiştir. Ayrıca EKY problemleri için optimal çözümlerin elde edilmesinde yeter sayıda populasyon ile çalışılmasını sağlamak için populasyon boyutlandırma denkleminin kullanılabileceği vurgulanmıştır.

Pattanemekar *et al.* (2003), ulaşım ağlarında dinamik ve stokastik ağ üzerinde çalışmışlardır. Geleceğe yönelik tahmin yapılırken iki tür belirsizliğin ortaya çıkacağını öne sürmüşlerdir. Bu belirsizlikler her bir kenarın seyahat zaman varyansı ve ortalama seyahat zamanı tahmin hatasıdır. Çalışmalarında bu belirsizlikleri tespit etmek için matematiksel model ortaya koymuşlardır. Önerilen modeller Houston'da bulunan test aracından elde edilen seyahat zaman bilgisi ile karşılaştırılmıştır. Sonuçlar önerilen algoritmanın seyahat süresini tahmin etmede etkili olduğunu göstermiştir.

Wanichworanant (2003) tarafından yapılan çalışmada geleneksel internet protokolü (IP) ağlarında statik kenar maliyet ölçümüne dayalı olarak en kısa yolun hesaplandığı, bunun sonucunda ağır yükler altında tıkanıklıktan dolayı dengesiz trafik dağılımının ve uzun gecikmelerin meydana geldiği ifade edilmiştir. Bu problemlerin temel sebebinin bant genişliği ve trafik karakteristiklerinin yönlendirme kararları alınırken dikkate alınmadığını, bu durumun üstesinden gelmek ve bilgisayar gecikme süresini azaltmak ve optimal yol dengesini sağlamak amacıyla GA'nın kullanıldığı bir optimizasyon algoritması oluşturulmuştur. Sonuç olarak Trafik Mühendisliğine dayalı GA'nın internet protokolüne sorunsuz bir şekilde dahil edilebileceği ifade edilmiştir.

Hamdan (2004), çalışmasında bilgisayar ağlarında tek bir kaynaktan aynı verinin birçok varış noktasına gönderilmesi işlemi için en uygun rotanın bulunması konusunu içeren sınırlı çoğa gönderim problemi için GA'ya dayalı bir yaklaşım sunmuştur. Çoğa yönlendirmede temel amaç, ya en az toplam maliyetli ya da her bir yolun kaynaktan varışa en az maliyetli rotasını bulmak olarak tanımlanmıştır. Önerilen algoritma, gecikme ve gecikme varyasyon sınırlı çoğa gönderim yönlendirme problemine uygulanmıştır. Gecikme varyasyon sınırı herhangi iki düğüm arasındaki gecikme farkına dayalıdır. Yapılan doğrulama deneyleri ile algoritmanın performansı gösterilmiş ve geleneksel çoğa gönderim algoritmaları ile karşılaştırılmıştır. Sonuç olarak algoritmanın etkili sonuçlar ortaya koyduğu görülmüştür.

Ji (2005), stokastik kenar uzunluğuna sahip 23 düğümlü EKY problemi üzerinde çalışmıştır. Farklı karar kriterlerine göre üç yeni model önermiştir. Bunlar; beklenen değer modeli, ihtimalle sınırlandırılmış programlama ve ihtimale bağlı programlamadır. Bu modelleri çözmek için stokastik simülasyonla genetik algoritmayı birleştirerek melez akıllı algoritma geliştirmiştir. Sonuç olarak bu algoritmanın genetik algoritmanın parametrelerine göre daha etkili olduğu tespit edilmiştir.

Kim *et al.* (2005), dinamik ve stokastik ağda gerçek zamanlı trafik bilgisinin optimal taşıt yönlendirmeye etkisini incelemiştirlerdir. Gerçek zamanlı bilgi teknolojisiyle ulaşım sistemini entegre etmek için sistematik bir yaklaşım sunmuşlardır. Şehir yol ağında

gerçekleştirilen nümerik çalışmayla bu sistemin kullanılması halinde taşıtların zamanında hizmet ulaştırma seviyeleri artarken kullanılma süresinin ve toplam maliyetin azaldığını ispatlamışlardır.

He *et al.* (2005), dinamik ve stokastik ulaşım ağında etkili en iyi rotayı bulmaya çalışmışlardır. Kenar ile yol özellikleri arasındaki ilişki bulunarak bütün bir yolun seyahat zaman dağılımı türetilmiş ve uyarlamalı en iyi yol kümesi belirlenerek etkili bir simülasyon metodu uygulanmıştır. Çalışmada en uygun rotayı bulmak için yalnızca seyahat sürelerinin değil aynı zamanda bütün bir yolun seyahat zaman dağılımı bilgisinde etkili olduğu sonucu ortaya koymuştur.

Chuang and Kung (2005) tarafından bulanık EKY problemiyle ilgili sezgisel yeni bir algoritma önerilmiştir. Bu algoritmada her bir kenar uzunluğu üçgen bulanık küme olarak verilmiştir. Önerilen algoritma ile bir ağdaki bütün muhtemel yollar arasında bulanık yol uzunlukları ve bulanık en kısa yollar hesap edilmiştir. Hesaptaki kompleksliği azaltmak için sezgisel bir prosedür önerilmiştir. Bu prosedür bulanık EKY uzunluğu ile her bir bulanık yol uzunluğu arasındaki benzerlik derecesinin hesabına dayandırılmıştır.

Fu *et al.* (2006), geçmişte yapılan çeşitli sezgisel EKY algoritmalarına genel bir bakış sunmuşlardır. Amaçları farklı sezgisel stratejilerin temel özelliklerini tanımlamak, birleştirilmiş bir sınıflandırma çatısı geliştirmek ve hesap tekniklerini özetlemektir. Farklı araştırma stratejileri üç aşamada kategorize edilmiştir. Bunlardan birincisi araştırılan alanı sınırlandırmak, ikincisi araştırma alanını hedefe göre ayrıştırmak ve üçüncüsü ise araştırılan kenarları sınırlandırmaktır.

Liang *et al.* (2007), dinamik rota rehber sistemlerinde GA'nın uygulanması konusunda çalışmışlardır. Rastgele bir A* algoritması oluşturularak GA ile dinamik taşıt yönlendirme algoritmaları arasında en büyük problem olan başlangıç popülasyonunun nasıl oluşturulacağı konusuna çözüm getirilmiştir. Geliştirilen model ve algoritmalar

elektronik harita üzerine yerleştirilerek hesap performansı analiz edilmiştir. Sonuçlar, üretilen modelin rota rehber problemleri için etkili olduğunu göstermiştir.

Grosan and Abraham (2007) tarafından iki amaçlı sezgisel yaklaşım kullanılarak en az kapasiteli yol probleminin çözümü için yeni bir algoritma önerilmiştir. Önerilen algoritmada doğru yol sonuçlarına ulaşmak için farklı bir yol ve yol uyarlamak için bir geliştirme tekniği denenmiştir. Algoritmanın etkinliğini göstermek için rastgele oluşturulan 50 ve 200 düğümlü ağlarda bir takım nümerik deneyler yapılmıştır. Amprik sonuçlar GA kullanılarak elde edilen sonuçlarla karşılaştırılmıştır. Sonuçlar algoritmanın büyük ağlar için etkili sonuçlar verdiğini ortaya koymuştur.

Kang (2008), çalışmasında satın alma, seyahat programı ve seyahat maliyetleri gibi minimum maliyetli ziyaretler için kullanıcılara rota ve konum yönünden hizmet veren danışman hizmet probleminin parasal maliyet ve seyahat maliyeti olmak üzere iki tür bileşeni olduğunu ileri sürmüştür. Çok amaçlı yaklaşımın parasal maliyet ya da zaman maliyeti noktasında diğer çözümlerden daha az maliyete sahip çoklu çözümleri bulmayı sağladığını göstermiştir. Yaptığı araştırmada birleştirilmiş maliyet ve çok amaçlı yaklaşımlar için tam ve sezgisel bir çözüm algoritması geliştirilmiştir. Tam algoritmalar adaptif dinamik programlama içerirken, sezgisel algoritmalar GA, EKY metodunu içerir. Önerilen algoritmaların hesap performansı büyük ölçekli metropolitan ağlarda gerçekleştirilen deneylerle ölçülmüş ve sağlam bir algoritma olduğu doğrulanmıştır.

Lin *et al.* (2009), Rota Rehber Sistemleri ve GPS gibi teknolojilerin gelişmesine bağlı olarak rota rehberinin doğruluğu ve etkinliğinin trafik şartlarının doğruluğuna dayandığını, bu durumun ancak problemin çok fazla değişkenle tanımlanmasıyla çözülebileceğini ileri sürmüşlerdir. Rota rehber sistemleri tarafından kullanılan değişkenler ne kadar çok artarsa hesap zamanının da o kadar çok artacağı ve elle taşınan GPS tarzı aletlerin sınırlı kaynaklara sahip olduğundan, gerçek zamanlı sistemlerde optimal en iyi çözüm olarak bilinen Dijkstra algoritmasını kullanarak en iyi çözüme ulaşmanın mümkün olmayacağı belirtilmiştir. Bu problemin üstesinden gelmek için GA kullanılmıştır. Gerçek trafik şartları ve değişen hız sınırları dikkate alınarak farklı iki

senaryo ışığında sürüş halinde en kısa yolu bulmak için GA kullanılmıştır. GA'nın etkinliği, çok büyük kenar sayılarına sahip gerçek şehir haritalarına uygulanarak ispatlanmıştır.

Kumar *et al.* (2009), çalışmalarında yol hesaplama algoritmalarında GA kullanarak AUS için yol planlamışlardır. Statik bir çevrede gerçek bir harita üzerinde verilen bir varışa kadar rota belirleme problemi üzerinde çalışmışlardır. Önerilen çözümde GA kullanılmıştır. GA'ya dayalı olarak talebe göre uyarlanmış bir metot önerilmiştir. GA'ya dahil edilen Dijkstra ve A* algoritmaları GA'ya dahil edilmeyen GA ve A* algoritmaları ile karşılaştırılmış ve EKY problemlerinde hesap zamanı ve kompleksliğini azaltmada etkili olduğu bulunmuştur.

Boonsuwan (2009), kompleks ağ optimizasyonu için evrimsel algoritma yaklaşımı konusunda çalışma yapmıştır. Çalışmasında temel ağ optimizasyon problemlerinden olan EKY yönlendirme, minimum budama ağacı ve iki kriterli EKY yönlendirme problemlerini çözmek için evrimsel algoritmalara dayalı stokastik bir teknik geliştirmiştir. Bu çalışmadaki temel amaç, gerçek ağ optimizasyon problemlerini etkili bir şekilde çözebilen bir algoritma geliştirmektir. Çalışmada, çözüm uzayında stokastik araştırma yapılmıştır. Potansiyel çözümler önceden tanımlı uygunluk fonksiyonları ile test edilmiştir. Yeni bireyler oluşturulurken ebeveynler ile yavrular arasındaki uygunluk fonksiyonları arasındaki farka dayalı bulanık mantık kontrolü ile yaşamı devam ettirme ihtimali belirlenmiştir. Çalışmada ayrıca çok amaçlı optimizasyon için etkileşimli ağırlıklı uygunluk fonksiyonun kullanıldığı GA önerilmiştir. Sonuçlar kurulan algoritmaların geleneksel yöntemlere göre daha iyi sonuç verdiğini göstermiştir.

Badawy (2010) tez çalışmasında, güneş enerjili kablosuz ağların gelişmesi ve işlemleri konusuna yoğunlaşmıştır. Ağ düğümlerine uygun güneş paneli ve pil konfigürasyonu sağlama probleminin üstesinden gelmek için, geçici EKY yönlendirmenin kullanımına dayalı ve hedef gelişim zaman periyodu için düğüm enerji akışını hesaba katan kaynak sağlama algoritması önerilmiştir. Kaynak atama prosedürü ile enerji paylaşımı yönlendirmesi için, GA kullanan bir metod ileri sürülmüştür. Sonuçlar, önerilen kaynak

sağlama algoritmasının geleneksel yöntemlere göre büyük maliyet kazancı sağladığını ortaya koymuştur.

2.3. Yapay Bağışıklık Sistemleri (YBS)

Timmis *et al.* (2000), veri analizine esas teşkil etmek üzere YBS kullanmışlardır. YBS ile kurulan ağların basit sınıflandırma aracı olarak kullanıldığı zaman etkili oldukları gösterilmiştir.

Castro and Zuben (2001), bağışıklık cevabı oluştururken uygunluk olgunlaşması sürecini tam olarak hesaba katan klonal seçim prensibini kullanmışlardır. Makine öğrenmesi ve numune tanıma gibi genel işlevleri yerine getirmek üzere CLONALG adında bir algoritma önererek daha sonra çok terimli zamanlı optimizasyon problemine bu algoritmayı uyarlamışlardır.

Castro and Timmis (2002), numune tanıma işlevini yerine getirmek için YBS önermiştir. Literatürde bu konuda yapılan YBS algoritmaları hakkında bilgi verilmiş ayrıca Yapay Sinir Ağları ile YBS arasında bir karşılaştırma yapmışlardır.

Keko *et al.* (2003), elektrik dağıtım ağlarının planlanmasında kullanılan gezgin satıcı problemini GA ve YBS kullanarak çözümlenmişlerdir. GA'nın başlangıçta problem hakkında var olan bir takım bilgileri kullanma noktasında yetersiz kalma problemini gidermek için YBS'ye özgü aşılama tekniğini kullanmışlardır.

Keko *et al.* (2004), dağıtım ağı yönlendirme problemlerini YBS ile çözme konusunda çalışma yapmışlardır. Çok terimli zamanlı ayrık optimization problemlerinden olan depo-taşıtlı yönlendirme problemlerini GA ve YBS kullanarak çözümlenmişlerdir.

Wang *et al.* (2004), yapay bağışıklık optimizasyon algoritmaları ve uygulamaları konusunda bir inceleme yapmışlardır. Son on yılda Yapay Bağışıklık Teorisinin

gelişimi ve uygulamaları konusunda araştırma yapılmış, diğer evrimsel algoritmalara göre üstün olan yönleri ifade edilmiştir.

Liu *et al.* (2006), tarafından çoklu yönlendirmeye dayalı çok ajanlı bağışıklık klonal seçim algoritması (MAISCA) önerilmiştir. MAISCA algoritmasında bağışıklık klonal seçim algoritmasının araştırma stratejisi ile çok ajanlı olma hali birleştirilmiştir. MAISCA algoritması GA ile karşılaştırılmış, sonuçlarının daha doğru ve daha kısa bir zamanda elde edildiği fikrine ulaşılmıştır.

Berbert *et al.* (2007), düşük maliyetli farklı topolojiye sahip k-spanning ağacının kümesini bulmak için YBS kullanmışlardır. Graf ne kadar çok düğüme sahip olursa çözüm uzayının o kadar büyük olduğu, bu durumu klasik algoritmalarla çözmenin mümkün olmadığı öne sürülerek, YBS kullanılmış ve oldukça etkili çözümler elde etmişlerdir.

Bakhouya and Gaber (2007), doğal bağışıklık sisteminin klonal, pozitif ve negatif seçim mekanizmasından esinlenen klonlamaya dayalı yeni bir algoritma önermişlerdir. Algoritmadaki klonlama stratejisi, adaptif özelliğe sahip olan Greedy yaklaşımına göre düzenlenmiştir. Algoritma gezgin satıcı problemine uygulanmış, etkinliğini denetlemek için karınca kolonileri algoritması kullanılmıştır. Sonuç olarak önerilen algoritmanın gezgin satıcı problemi için uygun olduğu ve karınca kolonisi algoritmasına göre sonuca daha çok yakınsadığı tespit edilmiştir.

Gong *et al.* (2008), çalışmalarında nümerik optimizasyon için PAIS isminde populasyon tabanlı yapay bağışıklık sistemi önermişlerdir. PAIS insan bağışıklık sistemini temsil eden bir dinamik süreç olarak modellenmiştir. Algoritmada antijen, antikor, antikorlar arasındaki etkileşimleri tanımlayan reaksiyon kuralları kümesi ve reaksiyon kurallarının algoritmaya nasıl uygulanacağını gösteren dinamik bir algoritma kullanmışlardır. PAIS'nin performansını doğrulamak için 20-10 000 boyutlu dokuz farklı Benchmark fonksiyonunun ve pratik optimizasyon problemlerinin çözümü yapılmıştır. Sonuçlar bu problemleri çözmede oldukça başarılı bir performans gösterdiğini ortaya koymuştur.

Vijayalakshmi and Radhakrishnan (2008), hibrit GA dayalı yeni çoklu yönlendirme algoritması önermişlerdir. Kısıtları uygulamak ve hedef faktör metodunda karşılaşılan sorunların üstesinden gelmek için algoritmaya YBS algoritması dahil edilmiştir. YBS algoritmasında antijenler ile antikolar arasındaki etkileşimi kullanmışlardır. Deneysel sonuçlar ile algoritmanın çeşitli büyüklükteki rastgele ağlar için oldukça etkili olduğu gösterilmiştir.

2.4. Çok Amaçlı Optimizasyon

Zhang *et al.* (1999), çalışmalarında stokastik ağlar için bir model önermişlerdir. Bu modelde yalnızca deterministik değerler dikkate alınmamış aynı zamanda rastgele değişkenlerin ortalama ve varyansı da dikkate almışlardır. Kurulan modelde YBS, kaos operatörü ve Parçacık Koloni Optimizasyon yöntemi birlikte kullanılmıştır. Sonuç olarak gerçek yaşama daha yakın gerçekçi ve etkili sonuçlar elde etmişlerdir. GA ve Parçacık Koloni Optimizasyonuna göre daha iyi sonuçlar elde edilmiştir.

Zitzler *et al.* (2002), çok amaçlı optimizasyon problemleri için pareto optimal kümeyi bulan yada optimal kümeye yaklaşan Strength Pareto Evrimsel Algoritmasını (SPEA) sunmuşlardır. Daha sonra bu algoritma geliştirilerek SPEA2 algoritması önerilmiştir. Bu iki algoritma başka algoritmalarla karşılaştırıldığında çok iyi sonuçlar verdiği gözlenmiştir.

Soltoni *et al.* (2003), çalışmalarında inşaat şantiyelerinde çok amaçlı kriterlere göre yol planlanmasını incelemişlerdir. Şantiye planlayıcılarının inşaat projesi boyunca şantiye işlemleri ve taşıtlar için kısa mesafe, düşük risk ve yüksek görüş mesafesi özelliğine sahip yolları seçme eğiliminde olduğu belirtilerek bu amaçla Dijkstra, A* ve GA algoritmalarının performansı karşılaştırılmıştır. GA'ya dayalı ihtimal optimizasyon yaklaşımının, optimum yada optimuma en yakın çözümler verdiği belirlenmiştir.

Gandibleux *et al.* (2005), çok amaçlı EKY problemi için 1984 yılında Martin tarafından bulunan Label Setting algoritmasını yönlü graflar için geliştirmişlerdir. Geliştirilen

algoritma ile verilen bir kaynak düğümünden ağdaki diğer tüm düğümlere olan bütün etkili yollar hesap edilmiştir. Ağın her bir kenarına pozitif maliyet bilgisi atanmıştır. Rastgele üretilen ağlar üzerinde yapılan deneylerde etkili sonuçlar bulunmuş ve algoritmanın çalışma zamanından tasarruf etmişlerdir.

Hamacher *et al.* (2006), tarafından zamana bağlı iki amaçlı EKY problemleri için algoritma geliştirilmiştir. Elde edilen sayısal değerler literatürde bulunan algoritmalarla karşılaştırılarak daha iyi sonuçlar verdiği gösterilmiştir.

He *et al.* (2007) yaptıkları çalışmalarında, yönlendirilmemiş ağlar için çok amaçlı EKY evrimsel algoritmasını (MOSPEA) önermişlerdir. Bastırılmamış çözümlerin dış küme olarak kullanılması ve farklı seçim stratejilerinin kullanılması yönüyle diğer algoritmalarından ayrılmıştır. MOSPEA'nın stokastik yakınsama özelliği analiz edilmiştir. Yaptıkları çalışma ile MOSPEA'nın çok amaçlı en kısa yollar için uygulanabilir bir yaklaşım olduğunu göstermişlerdir.

Tsaggoruis and Zaroliagis (2007), çok amaçlı en kısa yollar için lineer olmayan hedeflere sahip çok amaçlı optimizasyon işlevi gören FPTAS modelini geliştirmişlerdir. Bu modelin trafik optimizasyonu ve yönlendirme stratejileri gibi önemli uygulamalara sahip sınırlı optimal yollarda ve uzantısız en kısa yollarda uygulamaları gerçekleştirilmiştir. Elastik talep ve değerlere sahip ağırlıklı çoklu ticari akış problemlerinde iyi sonuçlar verdiği gösterilmiştir.

Panglinan and Janssens (2007), çok amaçlı EKY problemi için genel bir bakış sunmuşlardır. Algoritma EKY problemine uygulanarak çözümlerin çeşitliliği, hesap karmaşıklığı ve sonuçların uygunluk seviyesi incelenmiştir. Sonuçlar, çok terimli zamanda, evrimsel algoritmanın EKY problemine farklı çözümler getireceğini göstermiştir.

Pangilinan *et al.* (2007) tarafından yapılan çalışmada, çok amaçlı evrimsel algoritmalarının çok amaçlı EKY problemlerine uygulanabileceği gösterilmiştir. Çok

amaçlı EKY problemlerinin kolay kontrol edilemediği ve hafıza sorunlarının yaşandığı noktalarda iyi bir alternatif olacağını kanıtlamışlardır.

Sağ ve Çunkaş (2009), çok amaçlı optimizasyon problemlerinin çözümünde kullanılan GA tekniklerini incelemiştir. Bu alanda var olan sekiz ayrı algoritmanın test problemleri kullanılarak, çevrimdışı bir değerlendirme kriterinin ışığında performansları karşılaştırılmıştır. Sekiz algoritmanın tamamı, geliştirilen editör yardımıyla çalıştırılmış ve her biri için ayrı bir Pareto-Optimal çözümler kümesi bulunmuştur. Elde edilen sonuçlar NSGAI algoritmasının Pareto-Optimal çözümler üretmede en iyi performansa sahip olduğunu göstermiştir.

Gürsu ve İnce (2009) çalışmalarında, modelledikleri GA'da düzenli dizilerin referansında, çözüm uzayına düzgün dağılmış başlangıç popülasyonu üretmişlerdir. Ayrıca Pareto optimal çözümleri aramak için GA'da kullanılan yeni bir çaprazlama operatörünü de tanıtmışlardır.

Ghoseiri and Nadjari (2009), iki amaçlı EKY problemini çözmek için çok amaçlı karınca kolonisi optimizasyonuna dayalı bir algoritma sunmuşlardır. Algoritmanın etkinliğini ve çözümlerin kalitesini kontrol etmek için deneysel analizler gerçekleştirmişlerdir. Test ağı olarak büyük ve küçük ağlar kullanılmış, elde edilen değerler çok amaçlı Label Correcting algoritması ile karşılaştırılmıştır. Sonuçlar, üretilen algoritmanın kaliteli bastırılmayan çözümler ürettiğini ve büyük ağlarda çalışma zamanından kazanç sağladığını göstermiştir.

2.5. Kaynak Özetleri Değerlendirmesi

Sürücülerin yeni nesil teknolojileri kullanmak suretiyle yönlendirilmelerinin sağlanması AUS çalışmalarına girmektedir. Yapılan literatür taramasında AUS için geliştirilen rota rehber algoritmalarının seyahat süresini ve trafik sıkışıklığını azalttığı ve yolların kapasitesini artırdığı dolayısıyla bu durumun maliyetleri azalttığı tespit edilmiştir. AUS için EKY hesap edilirken klasik yöntemlerin büyük ağlarda fazla hafıza

gerektirdiğinden mobil aletlerinin kullanımının kısıtlandığını dolayısıyla GA ve YBS gibi sezgisel algoritmalarla EKY nin tahmin edilmesinin hafıza sorununu çözeceği ifade edilmiştir. Özellikle YBS'nin yeni uygulanan bir yöntem olması dolayısıyla literatürde var olan çalışmaların sınırlı olduğu ve daha çok gezgin satıcı problemi, depo taşıt problemi türünde çoğa gönderim problemlerinde uygulandığı görülmüştür. GA'lar da kullanılan populasyon sayısı parametresini optimize etmek için bir takım denklemler uygulanmıştır.

Yaptığımız çalışmada kullanılan ağların kenar maliyetleri stokastik olduğu gibi bulunan EKY de stokastik hedeflere göre belirlenmiş ve farklı büyüklükteki ağlar için kullanılmak üzere farklı GA ve YBS algoritmaları geliştirilmiştir. Büyük ağlarda araştırma uzayı büyük olduğundan etkili sonuçlar alınabilmesi için ara hedef düğümler belirlenerek araştırma uzayı küçük parçalara ayrılmıştır. Çalışmalardan elde edilen sonuçlar performans ve işlem süresi bakımından birbirleriyle karşılaştırılarak GA ve YBS nin birbirine üstün olma durumları tespit edilmiştir. Ayrıca GA ve YBS algoritmalarının parametrelerinin optimum seviyeleri Taguchi optimizasyon tekniği ile belirlenmiştir. Böylece yaptığımız çalışma ile klasik EKY yöntemlerinin sınırlı kaldığı stokastik problemlerde AUS için GA ve YBS ile stokastik EKY değerleri elde edilmesi sağlanarak bilinen metotların yeni bir alana uygulanması sağlanmıştır.

3. MATERYAL ve YÖNTEM

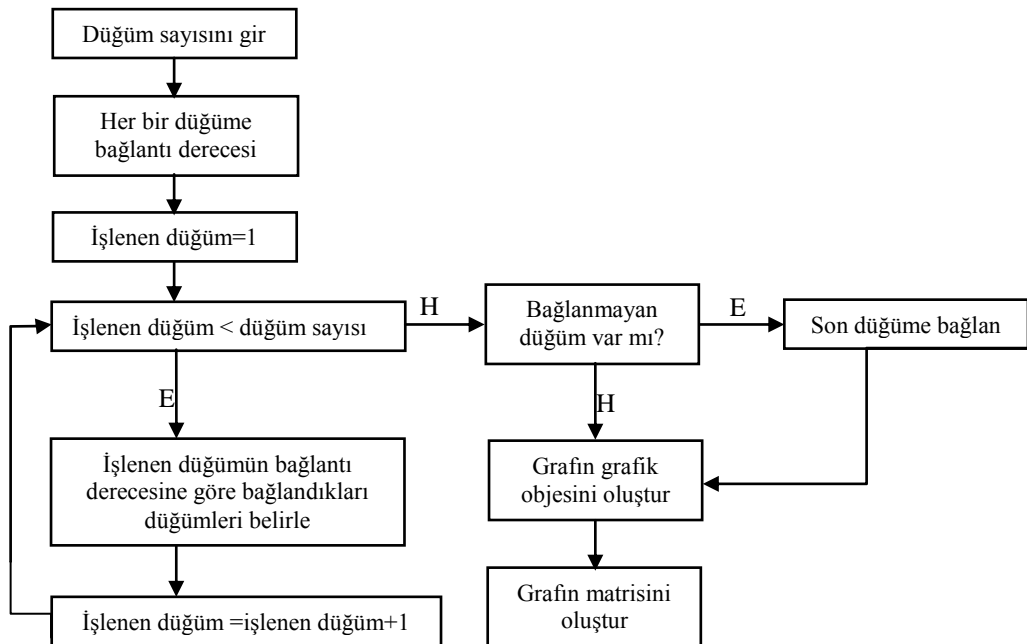
Çalışmanın bu bölümünde EKY problemlerinde materyal olarak kullanılan ağ, kenar bilgileri, problem türleri ve maliyet bilgileri sunulduktan sonra, problemin çözümünde yararlanılan Taguchi metodu, GA, YBS, çok amaçlı optimizasyon metotlarına yer verilecektir.

3.1. Materyal

EKY problemlerinin çözümü için kullanılan ağlar ve ağlara ait kenar bilgileri materyal olarak kullanılmıştır.

3.1.1. Ağlar ve kenar bilgileri

Bu çalışmada EKY problemi, maliyetli ve yönlü graflar için modellenmiştir. Kurulan modellerin etkinliğini göstermek için farklı büyüklükte graflar kullanma gereği ortaya çıktığından rastgele ağ oluşturan yönlü graf algoritması kurulmuştur (Şekil 3.1).



Şekil 3.1. Yönlü graf oluşturma algoritması

Bu algoritma MATLAB’da programlanarak 50, 100, 250, 500 ve 1000 düğümlü ağlar oluşturulmuştur. Bu çalışmada düğüm sayısı 50 olan ağ küçük, düğüm sayısı 100 olan ağ orta ve 250, 500 ve 1000 düğümlü olanlar büyük ağ olarak nitelendirilmiştir. Ağlar EK1-4’de gösterilmiştir. Ağların kenar sayıları Çizelge 3.1’de verilmiştir.

Çizelge 3.1. Ağ kenar sayıları

Ağ türü	Düğüm sayısı	Kenar sayısı
Küçük	50	102
Orta	100	203
Büyük	250	495
	500	1026
	1000	1945

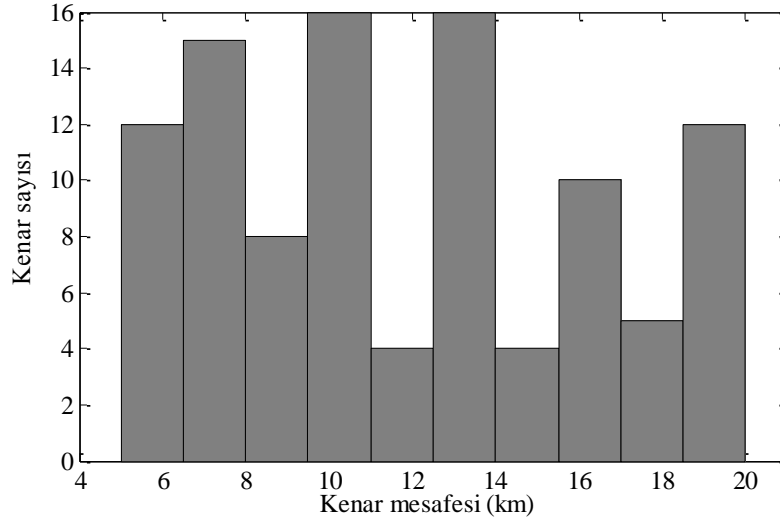
Ağlar sisteme boşluk (sparse) matrisleri ile tanıtılmıştır. Komşuluk matrislerinde bütün düğümler için bağlantı bilgisi oluşturulurken, boşluk matrislerinde sadece birbirleriyle bağlantılı olan düğümler için bağlantı bilgisi oluşturulmuştur. Boşluk matrislerinin kullanımıyla bilgisayarda bellek alanından kazanç sağlanması hedeflenmiştir.

Üretilen ağlarda mesafe, seyahat süresi ve yakıt tüketimi olmak üzere 3 tür kenar bilgisi kullanılmıştır (Çizelge 3.2).

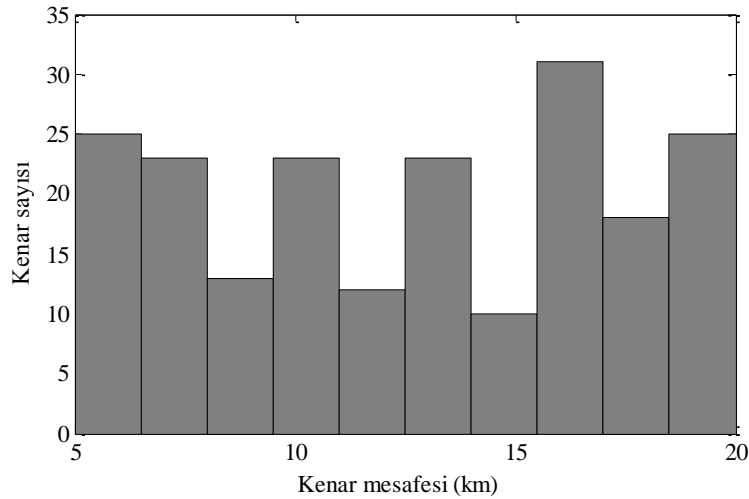
Çizelge 3.2. Kenar bilgileri

Değişken	Birimi	Türü
Mesafe	kilometre	deterministik
Seyahat süresi	dakika	stokastik
Yakıt tüketimi	litre	stokastik

Mesafe: [5-20] km arasında rastgele olarak üretilmiştir. 50 ve 100 düğümlü ağlar için kenar mesafe bilgileri **EK 5** ve **EK 8**'de verilmiş ve histogramları Şekil 3.2 ve 3.3'te gösterilmiştir.



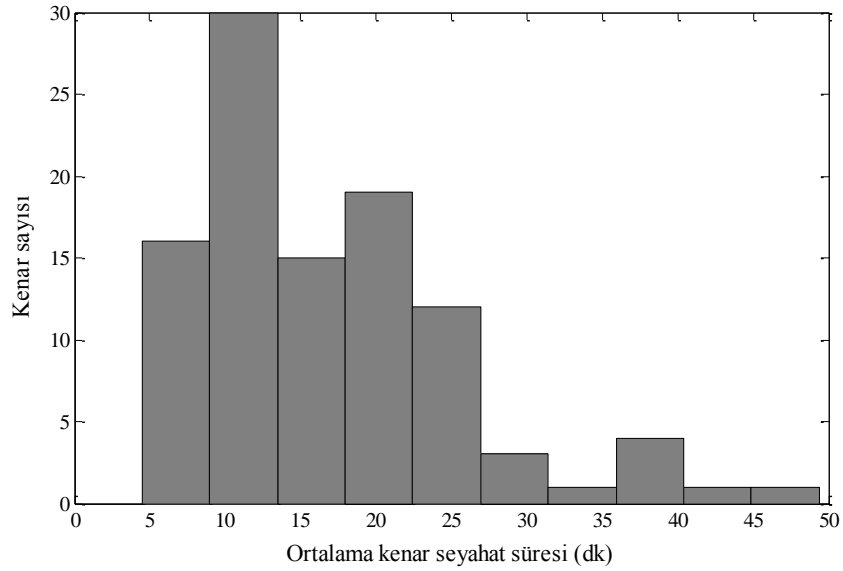
Şekil 3.2. 50 düğümlü ağın kenar mesafelerinin histogramı



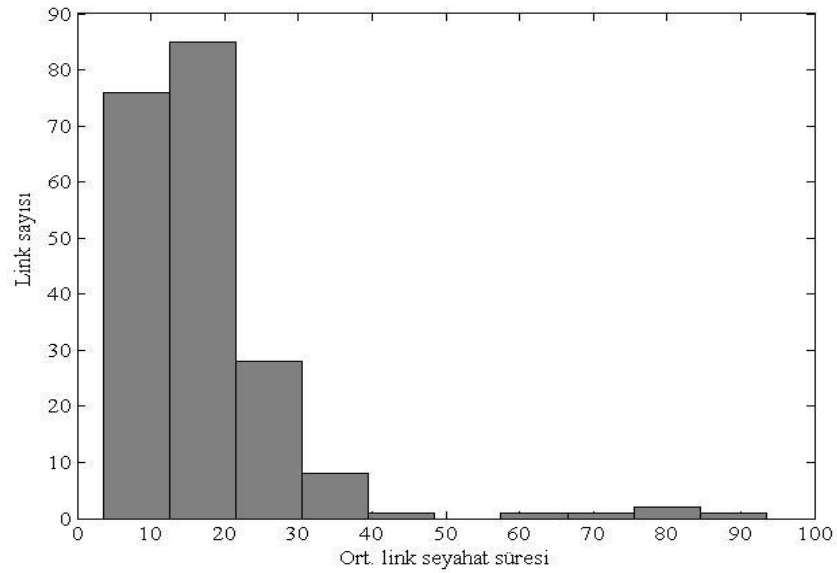
Şekil 3.3. 100 düğümlü ağ için kenar mesafelerinin histogramı

Seyahat süresi: Ulaşım sistemlerinde hava şartları, tıkanıklık vb. gibi bir takım olaylardan dolayı seyahat süresi sürekli olarak değiştiğinden, stokastik olarak tasarlanmıştır. Kenarlardaki seyahat süresinin normal dağılım, üniform dağılım ve üçgen dağılıma uyduğu kabulü yapılmıştır (Ji 2005). 50 ve 100 düğümlü ağlar için

kenar seyahat süresi dağılım bilgileri **EK6** ve **EK9**'da verilmiş, kenarlara ait ortalama seyahat süresi histogramı Şekil 3.4 ve 3.5'te gösterilmiştir.

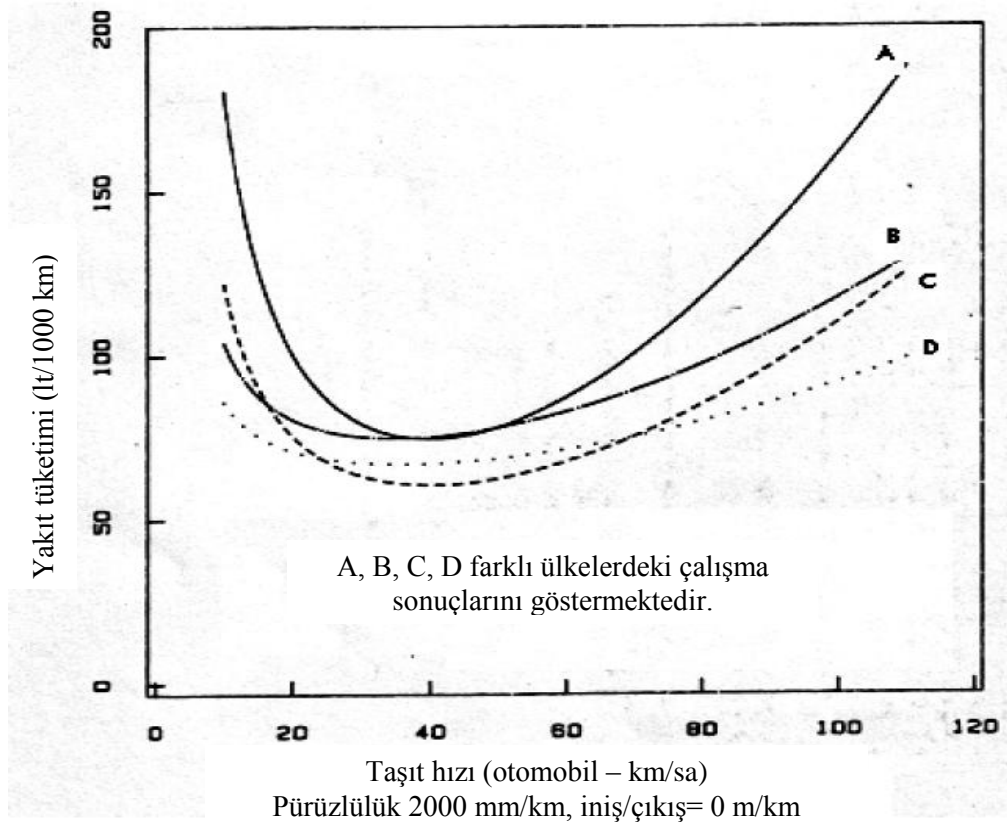


Şekil 3.4. 50 düğümlü ağda ortalama kenar seyahat süreleri histogramı



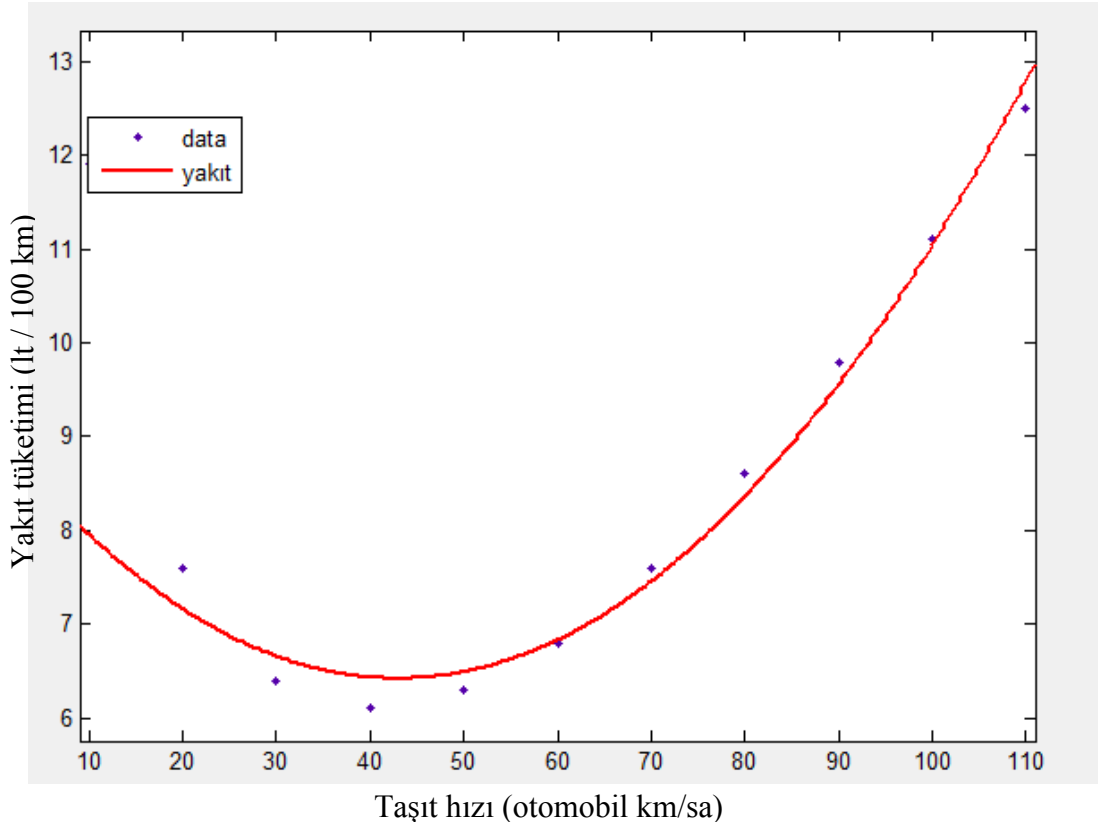
Şekil 3.5. 100 düğümlük ağ için kenarların ortalama seyahat süreleri histogramı

Yakıt tüketimi: Ülkemiz açısından en önemli taşıt işletme maliyet bileşenlerinden biridir. Birim uzunluk için yakıt tüketimi genel olarak kaplama tipi, eğim ve trafik akım tipine bağlı olarak değişmektedir. Yapılan çalışmalarda pürüzlülüğü 2000 mm/km ve iniş-çıkış 0 m/km olan yollarda otomobil başına yakıt tüketimi ile hız arasındaki ilişki Şekil 3.6'da gösterilmiştir (Bakırcı 2005).



Şekil 3.6. Yakıt tüketimi ile hız ilişkisi (Bakırcı 2005)

Model çalışmalarımızdaki ağlarda gözlenen hız değerlerinin ortalamaları 25-65 km/sa arasında olduğu ve bu aralıkta C parabolünün yakıt tüketim değerlerinin diğer parabolere karşılık gelen yakıt tüketim değerlerine göre düşük olmasından dolayı yapılan çalışmalarda C parabolü esas alınmıştır. Yakıt ile hız arasındaki ilişki ikinci dereceden bir parabol ile ifade edilmiş ve Şekil 3.7'deki grafik elde edilmiştir.



Şekil 3.7. Taşıt hızı-yakıt tüketimi ilişkisi

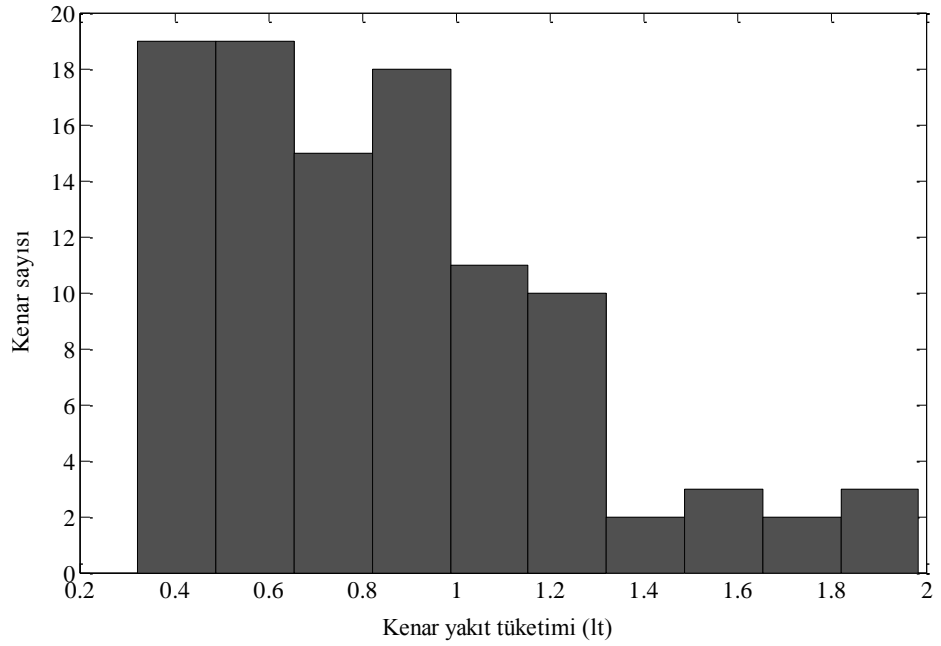
Taşıtların kenarlarda tükettiği yakıt miktarı taşıtların hızına bağlı olarak aşağıda ikinci dereceden denklem ile ifade edilmiştir.

$$yakıt = 0.001417 V^2 - 0.1216 V + 9.031$$

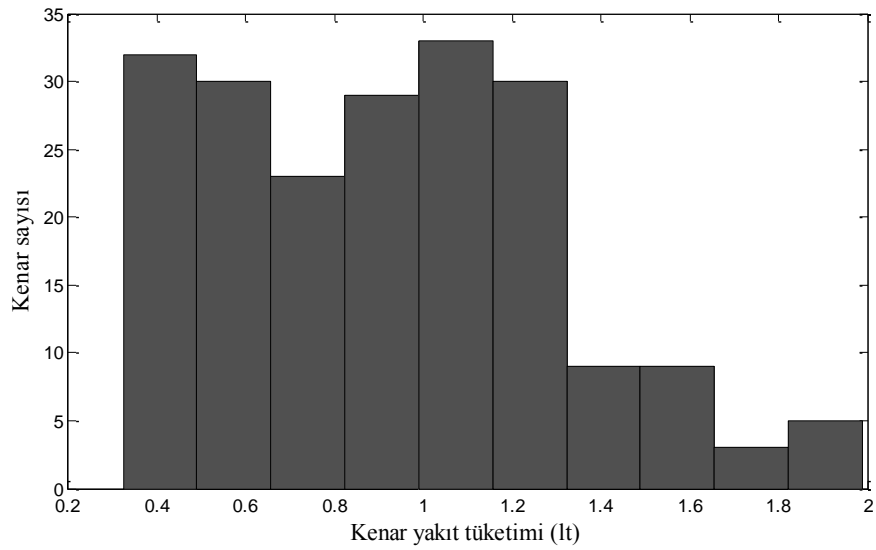
$$V = 60 * \frac{x}{t}$$

V: hız (km/sa)
x: uzunluk (km)
t: zaman (dk)

Taşıtların kenarlarda tükettiği yakıt miktarının normal dağılıma uyduğu görülmüştür. 50 ve 100 düğümlü ağlar için kenar yakıt tüketimi dağılım bilgileri **EK 7** ve **EK 10**'da verilmiş kenarlara ait ortalama yakıt tüketimi histogramı Şekil 3.8 ve 3.9'da gösterilmiştir.

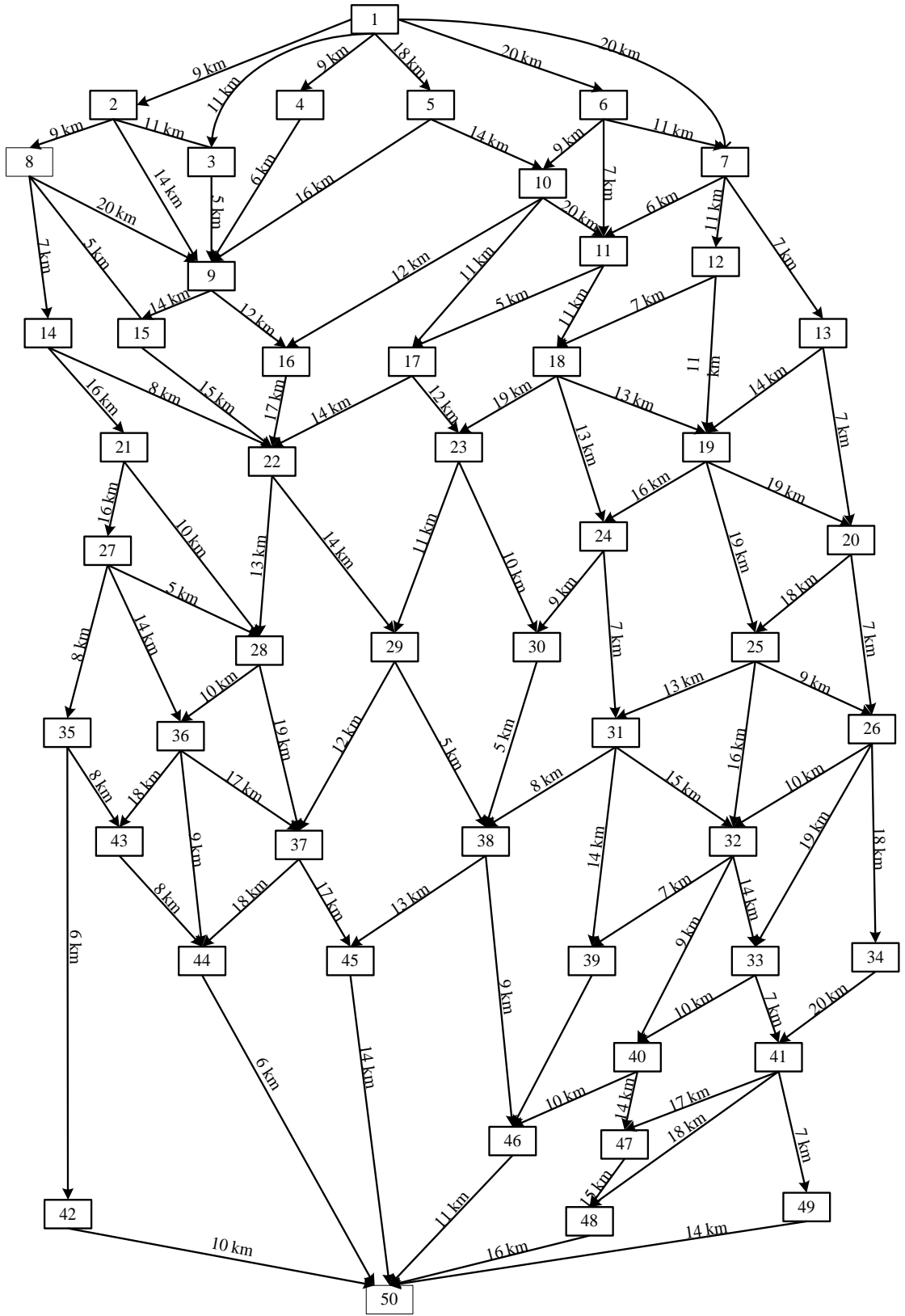


Şekil 3.8. 50 düğümlü ağda kenarların ortalama yakıt tüketimi histogramı

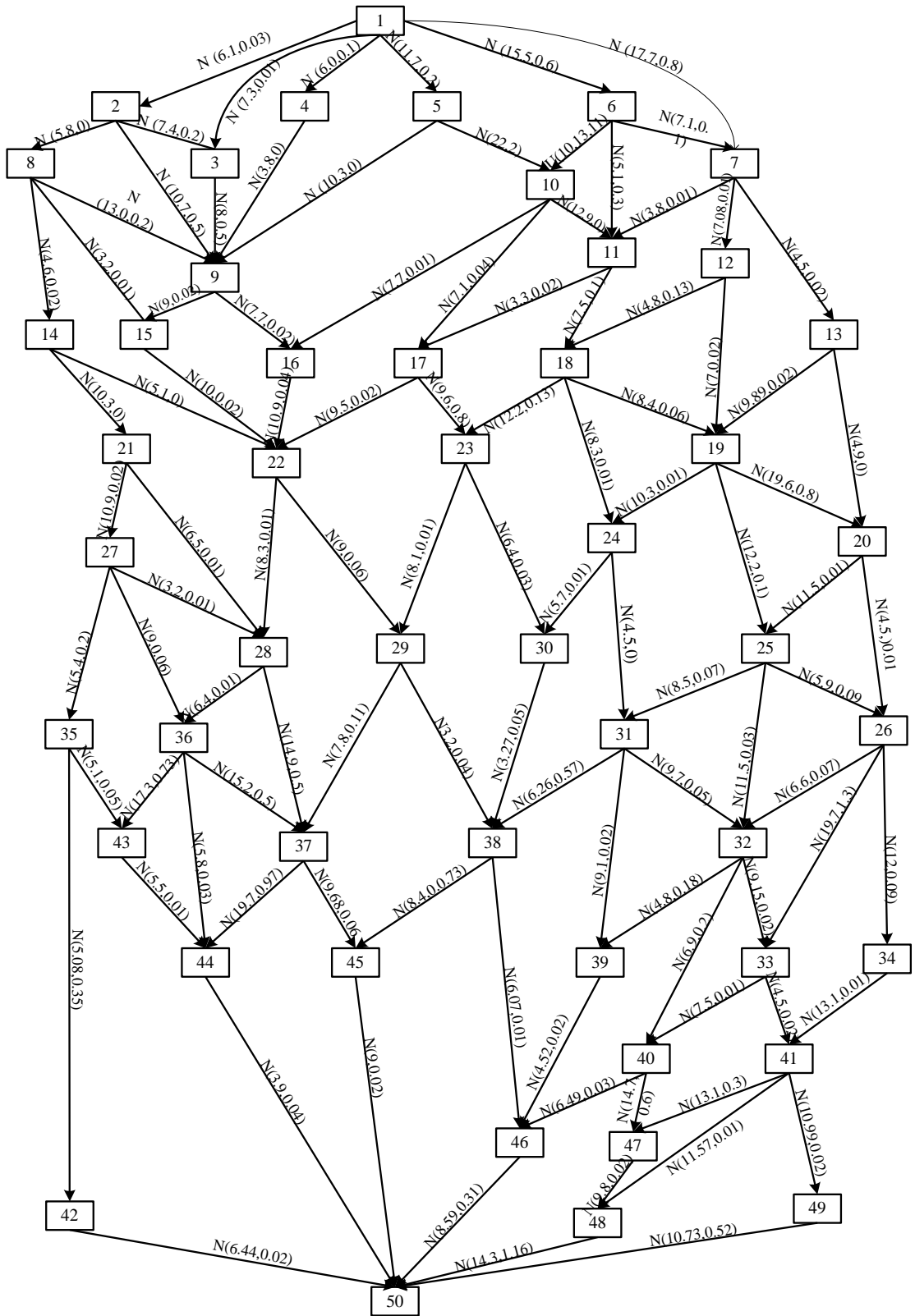


Şekil 3.9. 100 düğümlük ağda kenarların ortalama yakıt tüketimi histogramı

Şekil 3.10-3.12’de sırasıyla 50 düğümlü ağa ait kenar mesafeleri, seyahat süresi dağılımı ve yakıt tüketimi dağılım değerleri verilmiştir. Ağlar üzerindeki N normal dağılımı, UN üniform dağılımı ve U ise Üçgen dağılımı göstermektedir.



Şekil 3.10. 50 düğümlü ağda kenar mesafe değerleri



Şekil 3.12. 50 düğümlü ağda kenar yakıt tüketimi dağılım değerleri

250, 500 ve 1000 düğümlü ağların bağlantı sayıları oldukça fazla olduğundan dolayı karmaşıklığa yol açmamak için bu ağlara ait kenar bilgiler verilememiştir.

3.1.2. EKY problemleri

Çalışmada kullanılan EKY problemleri çalışmanın türüne göre farklılık göstermektedir. Tek amaçlı deterministik problemlerde amaç, verilen başlangıç ve hedef düğümü arasındaki;

1. En kısa mesafeli yolu
2. En kısa ortalama seyahat süreli yolu
3. En az ortalama yakıt tüketilen yolu bulmaktır.

Stokastik problemlerde ulaştırma sistemlerinde bulunan bir takım belirsizliklerden dolayı ağın kenar bilgisi olarak ihtimal dağılım bilgisi kullanılmıştır. Stokastik problemler için farklı karar kriterlerine göre 2 ayrı model ortaya konulmuştur.

1. Modelde, başlangıç ve varış düğümü arasında verilen bir T_0 (tüketilen yakıt miktarı ya da seyahat süresi) değerinden daha az değerde gidilme ihtimali en yüksek yolu bulmak
2. Modelde ise başlangıç ve varış düğümü arasında verilen güven seviyesinde en az değer (seyahat süresi, tüketilen yakıt miktarı) ile gidilen yolu bulmak amaçlanmıştır.

Çok amaçlı problemlerde üç tür problem üzerinde çalışılmıştır.

1. Problem: minimum seyahat süreli ve minimum mesafeye sahip en kısa yolu,
2. Problem: minimum yakıt tüketilen ve minimum mesafeye sahip yolu,
3. Problem: minimum seyahat süreli ve minimum yakıt tüketilen yolu bulmaktır.

3.1.3. Maliyet bilgileri

Pareto optimum çözümlerin değerlendirilmesinde kullanılmak üzere üç tür maliyet bilgisi kullanılmıştır. Bunlar; seyahat zamanı maliyeti, yakıt maliyeti ve taşıt işletme maliyetidir.

Seyahat zamanı maliyeti: Karayolu kullanıcıları için zamanın parasal değerini göstermektedir. Yapılan çalışmalarda otomobilde bulunan kişi sayısı ve kişilerin gelir seviyeleri bilgilerinden hareketle bir otomobilin saatlik seyahat zaman maliyeti 19,35 TL/saat olarak hesaplanmıştır (Tulum ve Terzi 2010).

Yakıt maliyeti: Taşıtların tükettiği yakıtın litre fiyatını göstermektedir. Petrol Ofisinin 2010 yılı ortalama benzin litre fiyatı 3,6 TL/litre olarak kabul edilmiştir.

Taşıt işletme maliyeti: Taşıt işletme maliyeti sabit ve değişken maliyet olmak üzere ikiye ayrılır. Sabit maliyetler; taşıt alımı ya da kiralama maliyeti, sigorta maliyeti, ruhsat ve taşıt vergileri maliyetlerinden oluşurken değişken maliyetler; tamir ve bakım maliyetleri ile yakıt tüketimi değerlerinden oluşmaktadır. Victoria Ulaştırma Siyasetleri Enstitüsünde (VTPI) yapılan incelemeler, yakıt tüketiminin değişken taşıt işletme maliyetinin yaklaşık %70'i olduğunu göstermiştir (VTPI 2009). Çalışmada elde edilen yakıt verilerine göre 100 km de yaklaşık 7 litre yakıt tüketimi kabulü ile değişken taşıt işletme maliyeti 0,36 TL/km olarak hesaplanmıştır.

3.2. Yöntem

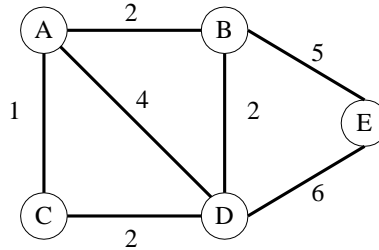
Çalışmada yöntem olarak klasik EKY algoritmalarından olan Dijkstra Algoritması, Floyd Algoritması, Taguchi Optimizasyonu, GA, YBS, Pareto Optimizasyon teknikleri kullanılmıştır.

3.2.1. Dijkstra algoritması

Klasik EKY algoritmalarından olan Dijkstra algoritması, Greedy yaklaşımına dayanır. Greedy yaklaşımı, graf üzerinde problem olan konu hakkında optimum sonucu bulabilmek amacıyla bir sonraki düğümü belirlemek için yapılan bir karar verme yöntemidir. Greedy yaklaşımı bir çok graf algoritmasında lokal seçim unsuru olarak kullanılmaktadır. Greedy yaklaşımının işleyişi en genel haliyle aşağıdaki şekilde ifade edilmektedir (Çölkesen 2004).

1. Adım: Başlangıç düğümü seçilir ve bu düğüm yol dizisine eklenir.
2. Adım: Gidilmeyen düğüm olup olmadığı sorgulanır. Gidilmeyen düğüm yoksa elimizdeki çözüm en iyi optimal çözümdür. Gidilmeyen düğüm varsa 3. adıma gidilir.
3. Adım: Bulduğumuz düğümüne komşu olan düğümlerden en az maliyete sahip olan düğüm seçilir ve yol dizisine eklenir. 2. adıma geri dönülür.

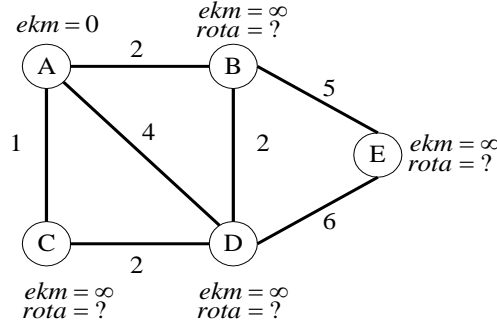
Dijkstra algoritması belli bir başlangıç noktasına göre en kısa yolu belirleyen bir algoritmadır. Bir başlangıç düğümünden diğer tüm düğümlere olan EKY'yi belirler. Maliyetli ve yönlü graflar için geliştirilmiştir Dijkstra algoritmasının uygulama adımlarını göstermek için örnek olarak Şekil 3.13'teki beş düğümlü maliyetli graf seçilmiştir.



Şekil 3.13. Maliyetli graf

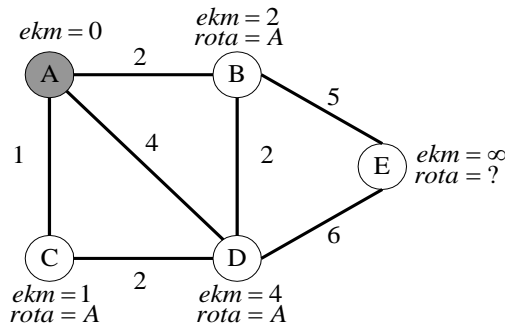
Yukarıdaki şekilde A, B, C, D ve E düğümleri, rakamlar ise kenarların maliyetlerini göstermektedir. Dijkstra algoritmasının uygulama adımları sırasıyla şöyle ifade edilebilir.

1. Adım: Her bir düğüm için en küçük maliyet ve rota bilgisi oluşturulur. En küçük maliyet (ekm) ile gösterilir. Çözüme başlarken başlangıç düğümünün en küçük maliyeti için sıfır diğerleri için sonsuz değerleri atanır. Rota bilgileri ise başlangıçta boş kümedir. Başlangıç düğümü Şekil 3.14'den de görüleceği üzere A'dır.



Şekil 3.14. Dijkstra algoritması, (1. Adım)

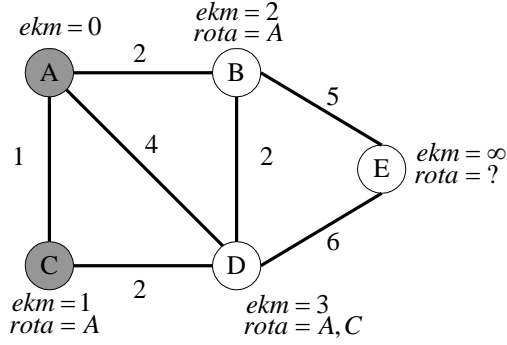
2. Adım: Başlangıç düğümü ile bağlantısı olan düğümler belirlenir. Bu düğümlerin en küçük maliyet ve rota bilgisi güncellenir. Güncelleme yaparken düğüme önceden atanmış en küçük maliyet bilgisi varsa küçük olan güncelde kalır, diğeri elenir. Rota ekm bilgisine göre değiştirilir. Örnekte A düğümü ile bağlantısı olan diğer düğümler B, C ve D düğümleridir. Ziyaret edilen düğümler koyu renkle gösterilir (Şekil 3.15).



Şekil 3.15. Dijkstra algoritması, (2. Adım)

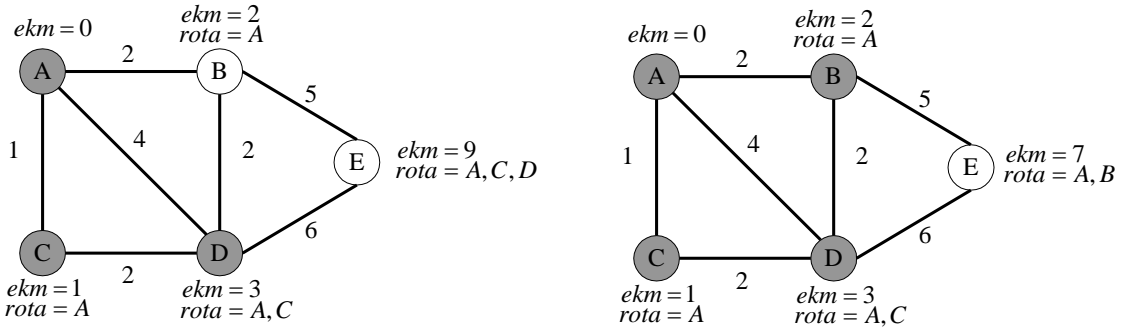
3. Adım: Başlangıç düğümü ile bağlantısı olan düğümlerin en küçük maliyetine göre küçük olan düğüme gidilir. Örneğimizde B, C ve D düğümleri içinde en küçük

maliyete C düğümü sahip olduğu için yeni düğüm olarak C seçilir. En küçük maliyet ve rota bilgileri güncellenir (Şekil 3.16).



Şekil 3.16. Dijkstra algoritması, (3. Adım)

4. Adım: Yeni düğüm seçildikten sonra gidilecek rotaların en küçük maliyet ve rota bilgisi güncellenir. Varış düğümüne ulaşıncaya kadar ikinci ve üçüncü adımlar tekrar edilir (Şekil 3.17).

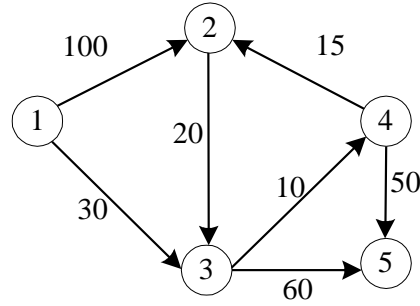


Şekil 3.17. Dijkstra algoritması, (4. Adım)

Sonuç olarak A düğümünden E düğümüne gitmek için 7 birimlik maliyette gidilir ve A, B ve E rotası izlenir.

3.2.2. Floyd algoritması

Floyd algoritması tüm düğümler arasındaki EKY'yi belirleyen bir algoritmadır. Yoğun graflarda kullanılması daha iyi sonuç verebilir. Dijkstra algoritması düğüm sayısı kadar çalıştırılırsa Floyd algoritması gibi çalışır. Şayet ağ seyrek ise Dijkstra algoritması, yoğun ise Floyd algoritması tercih edilmelidir. Algoritmada N düğümlü ağ NxN boyutunda kare matris olarak gösterilir. Şekil 3.18'deki grafın Floyd algoritması için uygulaması yapılmıştır (Çölkesen 2004).



Şekil 3.18. Floyd algoritması için örnek graf

1. Adım: Başlangıçta Uzaklıklar Matrisi (U) ve Düğüm Sırası Matrisi (DS) oluşturulur. U matrisinin (i,j) elamanı i düğümünden j düğümüne olan uzaklığı verir. Şayet i düğümü j düğümüne doğrudan bağlı değilse ∞ değerini alır. Bir düğümün kendisine bağlantısı olmayacağını göstermek için “-” işareti kullanılır. DS düğümlerin bağlanma sırasını tutar. Her i farklı j olmak üzere (i,j)=j yapılır. Grafın U ve DS’leri aşağıda gösterilmiştir.

		Uzaklık matrisi				
		1	2	3	4	5
1	–	100	30	∞	∞	
2	∞	–	20	∞	∞	
3	∞	∞	–	10	60	
4	∞	15	∞	–	50	
5	∞	∞	∞	∞	–	

		Düğüm sırası matrisi				
		1	2	3	4	5
1	–	2	3	∞	∞	
2	∞	–	3	∞	∞	
3	∞	∞	–	4	5	
4	∞	2	∞	–	5	
5	∞	∞	∞	∞	–	

2. Adım: **i** başlangıç düğümü, **k** ara düğüm ve **j** varış düğümü olmak üzere U matrisinin her bir satır ve sütunu sırasıyla kontrol edilerek aşağıdaki kurala göre yeniden düzenlenir.

Şayet $U(i,k) + U(k,j) < U(i,j)$ o zaman $U(i,j) = U(i,k) + U(k,j)$ ve $DS(i,j) = k$

$k=1$ için U ve DS;

Uzaklık matrisi

	1	2	3	4	5
1	–	100	30	∞	∞
2	∞	–	20	∞	∞
3	∞	∞	–	10	60
4	∞	15	∞	–	50
5	∞	∞	∞	∞	–

Düğüm sırası matrisi

	1	2	3	4	5
1	–	2	3	4	5
2	1	–	3	4	5
3	1	2	–	4	5
4	1	2	3	–	5
5	1	2	3	4	–

$U(i,k) + U(k,j) < U(i,j)$ şartını sağlayan herhangi bir bağlantı olmadığı için bir değişiklik yapılmaz.

$k=2$ için U ve DS;

Uzaklık matrisi

	1	2	3	4	5
1	–	100	30	∞	∞
2	∞	–	20	∞	∞
3	∞	∞	–	10	60
4	∞	15	∞	–	50
5	∞	∞	∞	∞	–

Düğüm sırası matrisi

	1	2	3	4	5
1	–	2	3	4	5
2	1	–	3	4	5
3	1	2	–	4	5
4	1	2	3	–	5
5	1	2	3	4	–

$U(4,2) + U(2,3) < U(4,3)$ şartı $15 + 20 < \infty$ sağlandığından $U(4,3) = 35$ ve $DS(4,3) = 2$

k=3 için U ve DS;

Uzaklık matrisi

	1	2	3	4	5
1	_	100	30	∞	∞
2	∞	_	20	∞	∞
3	∞	∞	_	10	60
4	∞	15	35	_	50
5	∞	∞	∞	∞	_

Düğüm sırası matrisi

	1	2	3	4	5
1	_	3	3	4	5
2	1	_	3	4	5
3	1	2	_	4	5
4	1	2	2	_	5
5	1	2	3	4	_

$U(1,3) + U(3,4) < U(1,4)$ şartı $30+10 < \infty$ sağlandığından $U(1,4)=40$ ve $DS(1,4)=3$

$U(2,3) + U(3,4) < U(2,4)$ şartı $20+10 < \infty$ sağlandığından $U(2,4)=30$ ve $DS(2,4)=3$

$U(2,3) + U(3,5) < U(2,5)$ şartı $20+60 < \infty$ sağlandığından $U(2,5)=80$ ve $DS(2,5)=3$

$U(1,3) + U(3,5) < U(1,5)$ şartı $30+60 < \infty$ sağlandığından $U(1,5)=90$ ve $DS(1,5)=3$

k=4 için U ve DS,

Uzaklık matrisi

	1	2	3	4	5
1	_	100	30	40	90
2	∞	_	20	30	80
3	∞	∞	_	10	60
4	∞	15	35	_	50
5	∞	∞	∞	∞	_

Düğüm sırası matrisi

	1	2	3	4	5
1	_	3	3	3	3
2	1	_	3	3	3
3	1	2	_	4	5
4	1	2	2	_	5
5	1	2	3	4	_

$U(1,4)+U(4,2) < U(1,2)$ şartı $40+15 < 100$ sağlandığından $U(1,2)=55$ ve $DS(1,2)=4$

$U(3,4)+U(4,2) < U(3,2)$ şartı $10+15 < \infty$ sağlandığından $U(3,2)=25$ ve $DS(3,2)=4$

k=5 için U ve DS;

	1	2	3	4	5
1	_	55	30	40	90
2	∞	_	20	30	80
3	∞	25	_	10	60
4	∞	15	35	_	50
5	∞	∞	∞	∞	_

	1	2	3	4	5
1	_	4	3	3	3
2	1	_	3	3	3
3	1	4	_	4	5
4	1	2	2	_	5
5	1	2	3	4	_

Artık graf üzerindeki herhangi iki düğüm noktası için gereken tüm bilgiler uzaklık matrisi ve düğüm sırası matrisi üzerinden okunabilir. Yalnız düğüm sırası matrisinden DS (i,j)=j oluncaya kadar düğümler takip edilir. Örneğin 1 düğümünden 5 düğümüne gitmek için en az maliyet, uzaklık matrisinden 90, takip edilecek yol ise 1-3-5 olarak okunmaktadır.

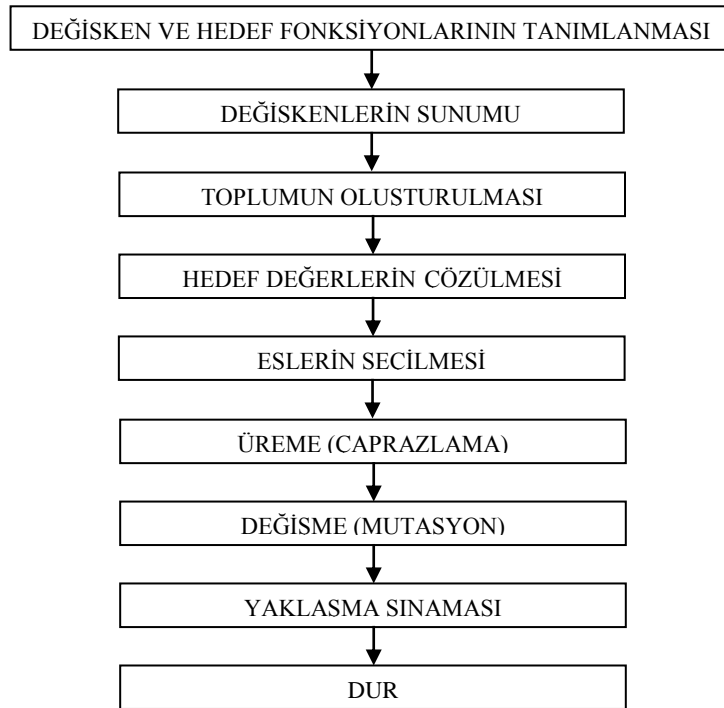
3.2.3 Genetik algoritma (GA)

GA yapay zeka tekniklerinden esinlenerek geliştirilmiş bir algoritmadır. GA ifadesi ilk olarak 1967 yılında Bagley'in öncülük ettiği tez çalışmasında kullanılmıştır. Michigan Üniversitesinde makine öğrenmesi konusunda araştırmalar yapan John Holland ve arkadaşlarının canlılarda görülen genetik süreci bilgisayar ortamında gerçekleştirme çalışmaları sonucunda 1970'li yıllarda ortaya çıkmıştır. Başlangıçta pratik bir yararı olmadığı düşünülen GA'lara Holland'ın öğrencisi olan David E. Goldberg'in GA konusunda yaptığı doktora teziyle National Science Foundation tarafından verilen Genç Araştırmacı ödülünü alması ve 1989 yılında konusunda klasik görülen kitabın yayınlanmasıyla popülaritesi artmıştır (Goldberg 1997).

GA'lar, geleneksel yöntemlerle çözümü zor ya da imkânsız olan problemlerin çözümünde kullanılmaktadır. Özellikle tabu araştırma, sırt çantası ve gezgin satıcı problemi gibi çok geniş bir çözüm uzayına sahip problemlerde GA'nın kullanılması

önemli bir avantaj sağlamaktadır. Bu tarz problemlerde çözüm uzayının tanımlanması klasik yöntemlerle uzun zaman almasına karşın, GA ile kısa sürede güvenilir sonuçlar alınabilmektedir. GA'lar ayrıca mühendislikte optimizasyon problemlerinde, tasarım, otomatik programlama, makine öğrenmesi, ekonomi, planlama vb. alanlarda da uygulanmaktadır (Deliktaş 2009).

GA yöntemleri, çözüm alanını rastgele biçimde bombardımana tutarak en iyi çözümü arayan bir yöntemdir. Çözüme ulaşabilmek için önce karar değişkeni uzayında rastgele noktalar topluluğu alınır. Daha sonra gösterilecek kuralların ışığı altında bu noktalar arasında eşleştirmeler yapılarak toplumun bazı üyeleri yok edilirken onların yerine yenileri oluşturulur. Yeni gelen bireylerin topluma katılması ile o toplumun öncekinden daha sağlıklı, yani hedefe daha yakın olması sağlanır. Böylece hedefe birçok noktadan ve kısa yollardan yaklaşılr (Şen 2004). Aşağıda GA'ların ilerleme adımları verilmiştir.



Şekil 3.19. GA'nın ilerleme adımları

3.2.3.a. Değişken ve hedef fonksiyonların tanımlanması

Karar uzayının değişik noktalarında bulunan aday çözüm noktalarının her birinin, aranılan çözüme ne kadar yakın olduğunu belirtebilmek için, mutlaka bir hedef fonksiyonuna ihtiyaç vardır. Problemlerin değişkenlerini içinde toplayan hedef fonksiyonu, matematik bir ifade olabileceği gibi yapılan deneylerden elde edilen sonuçlar ve sistematik kurallardan meydana gelen algoritmalar da olabilir (Şen 2004).

GA terminolojisinde, problemin değişkenlerinin her birine gen adı verilmektedir. Genlerin bir dizi halinde arka arkaya sıralanması ile meydana gelen genler dizisine kromozom adı verilir. Her bir kromozom bir birey olarak adlandırılır. GA'ların ilk aşaması, değişkenler dizisini temsil eden kromozomun oluşturulmasıdır. Eğer bir sorunun a_1, a_2, \dots, a_n gen olmak üzere, n adet gen değişkeni varsa kromozom, n adet gen elemanı olan bir diziden oluşur. Aşağıdaki şekilde gösterilir.

$$\text{Kromozom} = [a_1, a_2, \dots, a_n \text{ gen}]$$

Mevcut problem için muhtemel pek çok çözümü temsil eden bu çözüm kümesine, GA terminolojisinde popülasyon adı verilir. Popülasyonlar vektör, kromozom veya birey olarak adlandırılan veri dizilerinden oluşur. Aşağıda n adet kromozomdan oluşan bir popülasyon gösterilmektedir.

$$\text{Popülasyon} = [\begin{array}{l} \text{kromozom}_1 \\ \text{kromozom}_2 \\ \cdot \\ \cdot \\ \text{kromozom}_n \end{array}]$$

Başlangıçta popülasyon, çözüm uzayında rastgele bir şekilde oluşturulur. Sonraki nesillerde popülasyondaki kromozomlar değişim süreci içinde genetik algoritma işlemcileri tarafından yeniden belirlenirler (Şen 2004).

3.2.3.b. Değişkenlerin sunumu

Değişkenlerin ne şekilde gösterileceği konusu başka bir deyişle kromozomların kodlanma biçimi, problemin türüne göre farklı şekillerde olabilir. Bu kodlama türlerinden en çok kullanılanları şöyledir.

İkili Kodlama (Binary Encoding): GA'da ilk kullanılan ve en yaygın olan kodlama şeklidir. Her kromozom 1 ve 0'lardan oluşan diziler (string) halinde ifade edilir.

Kromozom= 1001011

Permutasyon Kodlaması (Permutation Encoding): Daha çok sıralama problemlerinde kullanılan kodlama yöntemidir. Gezgin satıcı problemi, görev ya da güzergah sıralaması problemlerinde kullanılabilir. Permutasyon kodlamada, her kromozom, sıralamada konum belirten numara karakter dizisinden oluşur.

Kromozom = 1 3 7 8 9 6

Değer Kodlaması (Value Encoding): İçerisinde gerçek sayısal değer, karakter veya nesne taşıyan problemlerde kullanılan kodlama şeklidir. İkili kodlama ile çözümün zor olduğu durumlarda kullanılır.

Kromozom A= 1,23 2,56 4,25 veya

Kromozom B= ABCDTYHSD veya

Kromozom C= "kuzey", "güney", "batı", "doğu"

3.2.3.c. Populasyonun oluşturulması ve hedef değerlerinin bulunması

GA'larda iki veya daha fazla kromozomun bir araya gelmesi ile oluşan sayıların hepsine birden populasyon başka bir deyişle toplum adı verilir. Populasyonu oluşturacak

kromozom sayısı GA'nın çözüm süresine ve doğru sonuca yakınsama miktarına etki eder. Populasyon büyüklüğü genelde çözüm uzayının büyüklüğüne ve hedef fonksiyonunun türüne göre seçilir. Genel olarak 15-20 den daha az alınmasının GA'nın etkinliğini azaltacağı ifade edilir.

Kromozomların hedeflenen çözüme yakınlık durumunu belirlemek için, uygunluk değerleri hesaplanır. Bu hesap için, başlangıç aşamasında belirlenen hedef fonksiyonları kullanılır. Hedef fonksiyonlarına aynı zamanda uygunluk fonksiyonu adı da verilmektedir. GA'nın başarısı bu fonksiyonun problemi etkili bir şekilde temsil etmesi, verim ve hassasiyetine bağlı olmaktadır.

3.2.3.d. Eşlerin seçilmesi

Bir sonraki nesilde hayatını devam ettirecek bireylerin belirlenmesi amacıyla, GA işlemlerine tabi tutulacak kromozomların mevcut populasyondan seçilmeleri gerekmektedir. Kromozomlar seçilirken uygunluk derecelerine göre rastgele bir şekilde seçilirler. Amaç, toplumun evrim geçirerek daha iyiye doğru gelişmesidir. Bu süreç sırasında seçilen eşler seçilmeyenlerin üzerine kopyalanır ve yeni bir toplum oluşturulur. Farklı birçok seçim operatörü geliştirilmiştir. Rulet tekerleği seçimi, turnuva seçimi ve sıralı seçim, örnek olarak verilebilir.

Rulet tekerleği seçiminde kromozomlar uygunluk değerlerine göre sıralanırlar. Sonra her bir kromozomun toplum içerisinde yüzdesi hesaplanır ve yığılımlı toplamı hesaplanarak rulet çemberi oluşturulur. Daha sonra 0 ile 1 arasında rastgele sayılar oluşturularak rulet çemberinde karşılık geldiği noktalar bulunarak işlem görece kromozomlar tespit edilir ve böylece yeni populasyon oluşturulmuş olur.

Çizelge 3.3'te permutasyon kodlama ile gösterilen 5 adet kromozomdan oluşan populasyon ve uygunluk değerleri gösterilmektedir.

Çizelge 3.3. Rulet tekerleği oluşturulması (1. Adım)

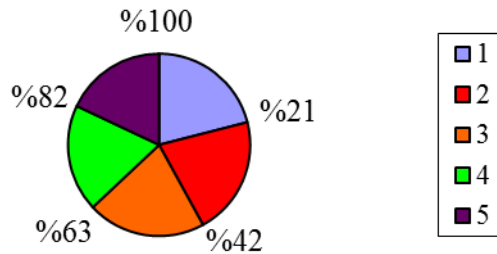
Kromozom	Uygunluk Değeri
1-2-6-9-16-20-23	61,44
1-5-8-13-15-18-22-23	51,84
1-4-11-14-21-23	53,79
1-5-8-13-19-22-23	59,06
1-5-11-17-20-23	61,68

Kromozomlar uygunluk değerlerine göre büyükten küçüğe doğru sıralanır. İhtimal ve yığışımli ihtimal değerleri hesaplanır (Çizelge 3.4).

Çizelge 3.4. Rulet tekerleği oluşturulması (2. Adım)

Kromozom No	Kromozom	Uygunluk Değeri	İhtimal %	Yığışımli İhtimal %
1	1-5-11-17-20-23	61,68	21	21
2	1-2-6-9-16-20-23	61,44	21	42
3	1-5-8-13-19-22-23	59,06	21	63
4	1-4-11-14-21-23	53,79	19	82
5	1-5-8-13-15-18-22-23	51,84	18	100
Toplam		287,81	100	

Çizelge 3.4'teki değerlere bağlı olarak, rulet çemberi Şekil 3.20'deki gibi oluşur.

**Şekil 3.20.** Rulet çemberi

Daha sonra rastgele sayılar üretilerek rulet çemberinde tekabül ettiği kromozom tespit edilir böylece yeni populasyon Çizelge 3.5'teki gibi oluşturulmuş olur.

Çizelge 3.5. Rulet tekerleği oluşturulması (3. Adım)

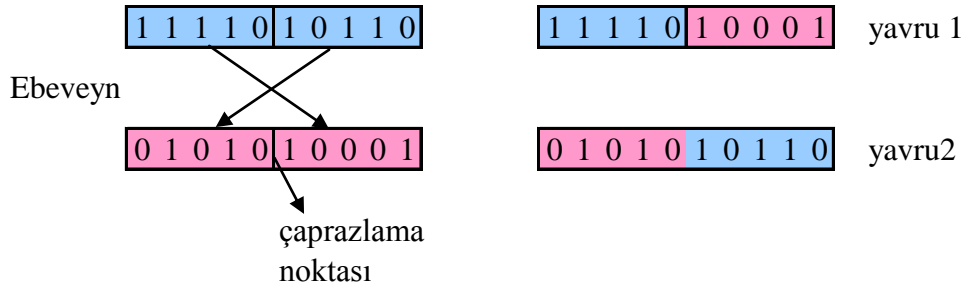
Rastgele Rulet Çevir	Seçilen Kromozomlar	Yeni Populasyon
35	2	1-2-6-9-16-20-23
76	4	1-4-11-14-21-23
59	3	1-5-8-13-19-22-23
61	3	1-5-8-13-19-22-23
84	5	1-5-8-13-15-18-22-23

Şayet uygunluk fonksiyonları arasında büyük farklar varsa, uygunluk değeri küçük olan kromozomların seçilme şansı azalacağından, rulet seçimi uygun olmayabilir. Bu durumda kromozomların uygunluğuna göre dizilerek seçildiği sıralı seçim yöntemi tercih edilebilir.

3.2.3.e. Çaprazlama

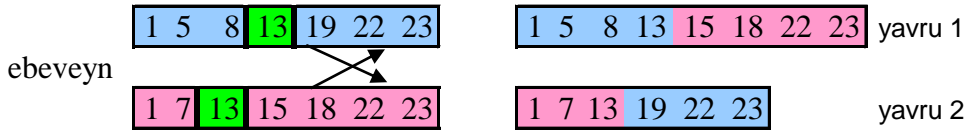
Rastgele seçilen kromozom çiftlerinin bir araya gelerek birleşmesi ile yeni ürünlerin ortaya çıkmasına çaprazlama denir. Çaprazlama, kromozomların genlerini birbirleriyle değiştirmelerini sağlayan bir işlemdir. Önce çaprazlamaya tabi tutulacak kromozom eşleri rastgele seçilir. Daha sonra bu kromozom eşlerinin hangi genlerden itibaren kesileceği rastgele seçimle belirlenir. Kesilen genler kromozom eşleri arasında değiştirilir. Böylece farklı nesiller oluşturulur. Farklı çaprazlama teknikleri vardır. Bunlardan en çok kullanılanı tek nokta çaprazlama, iki nokta çaprazlama, çok nokta çaprazlama ve üniform çaprazlamadır. Bu metotlardan hangisinin seçileceği ise problemin türüne göre farklılık arz etmektedir

Tek nokta çaprazlamada, kromozom eşleri üzerinde tek bir nokta seçilir ve o nokta referans alınarak ebeveyn kromozomlar çaprazlanır (Şekil 3.21).



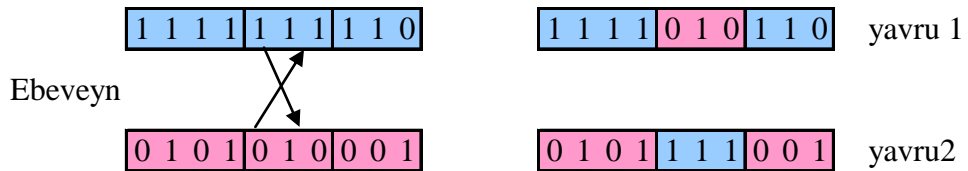
Şekil 3.21. İkili kodlamada tek nokta çaprazlama

Şayet kodlama türü olarak ikili değil de permutasyon kodlama kullanılmışsa, çaprazlama noktası olarak rastgele herhangi bir biti seçmek yerine her iki kromozomda ortak olan düğüm noktaları seçilmelidir. Aksi takdirde yönlendirme problemlerinde yavrular illegal rotalardan oluşacak, bu da çözümsüzlüğe sebep olacaktır. Şekil 3.22’de permutasyon kodlama da çaprazlama noktası olarak 13 düğümünün alındığı çaprazlama örneği gösterilmiştir.



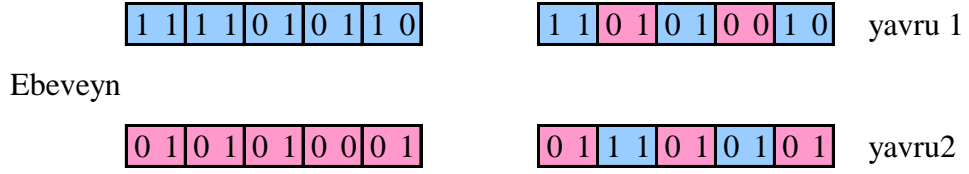
Şekil 3.22. Permutasyon kodlamada tek nokta çaprazlama

İki nokta çaprazlamada, iki referans nokta seçilerek bu noktalar referans alınmak kaydıyla ebeveyn kromozomlar çaprazlanır (Şekil 3.23).



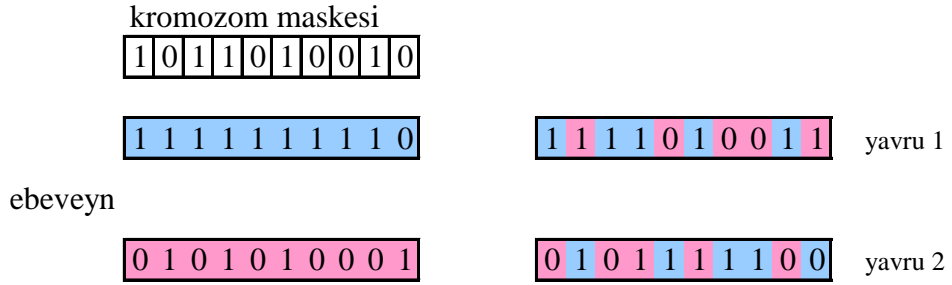
Şekil 3.23. İki nokta çaprazlama

Çok nokta çaprazlamada birçok referans nokta seçilir ve bu referanslara göre ebeveynlerin kromozomları çaprazlanır (Şekil 3.24).



Şekil 3.24. Çok nokta çaprazlama

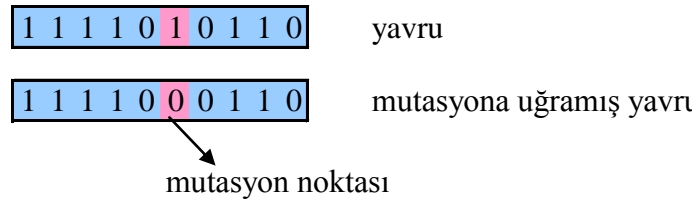
Üniform çaprazlama yönteminde çaprazlama maskesi kullanılır. Çaprazlama maskesinin uzunluğu her bir kromozomun uzunluğu kadardır. Maskedeki 1 ve 0'lar, türetilen kromozomun hangi geni hangi ebeveynden alınacağını belirler. Şekil 3.25'te her bir çaprazlama türü için örnek verilmiştir.



Şekil 3.25. Üniform çaprazlama

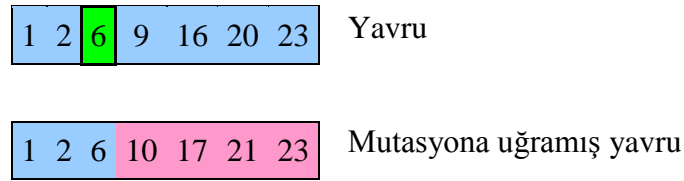
3.2.3.f. Mutasyon

Mutasyon GA'nın kromozom üzerinde rakam değişimi yapılarak yeni bir kromozom elde etme işlemidir. Bu işlem vasıtasıyla en iyi çözümün aranması sırasında GA'lar bir çözüm noktasının yakınındaki başka bir çözüm noktasını dener. Nesiller arası çeşitliliğin azalması durumunda mutasyon iyi bir etki oluşturabilir. Mutasyon operatörünün GA içindeki temel işlevi, nesiller arasındaki farklılaşmaya katkıda bulunmak ve çözüm uzayının farklı noktalarında çalışmasını sağlayarak hedefe hızlı bir şekilde yakınsamaktır. Mutasyon, kromozom üzerindeki herhangi bir bit değerini değiştirir (Şekil 3.26). Ele alınan problemin yapısına uygun olarak farklı mutasyon operatörleri tercih edilebilir (Kurt ve Semetay 2001).



Şekil 3.26. İkili kodlamada mutasyon

Permutasyon kodlama kullanılan bir rota probleminde, rastgele bir gen başlangıç noktası olarak seçildikten sonra şayet ikinci bir mutasyon noktası gösterilmemişse, son gen varış noktası olmak üzere ulaşım ağından başlangıç ve varış noktası için rastgele yeni bir rota oluşturulur. Şekil 3.27’de başlangıç noktası 6 varış noktası 23 olan yeni bir rota mutasyon sonucunda yavruya eklenmiştir.



Şekil 3.27. Permutasyon kodlamada mutasyon

3.2.3.g. Yaklaşma denemesi

Yaklaşma denemesi GA ile yapılan çözümlerin ne zaman en iyiye ulaşacağı değişik durma kriterleri ile tespit edilir. Bunlar:

- Önceden tespit edilen nesil sayısına ulaşılması,
- Toplumun uygunluk fonksiyonu değerinin ortalama veya standart sapmasının belirli bir hedefe ulaşması,
- Toplumların belirli zaman süresince hep aynı en iyi kromozom ile devam etmesi durumu,
- En iyi uygunluk değerine sahip kromozoma bakılarak bunun en iyi çözüm olduğuna karar verilmesi durumudur.

Şayet algoritma, durma kriterlerini henüz sağlamıyorsa, hedef değerlerinin sağlanması adımına geri dönülerek GA işlemleri tekrar yürütülür (Şen 2004).

3.2.3.h. GA parametreleri

GA'ların en önemli parametreleri çaprazlama oranı ve mutasyon oranıdır. Bunun yanında nesil sayısı ve popülasyon sayısı ve seçkinlik gibi parametreler de GA'ların performansını etkilemektedir.

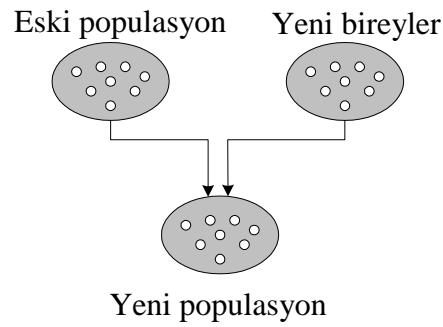
Çaprazlama oranı çaprazlamanın ne kadar yoğunlukta yapılacağını belirten bir parametredir. Çaprazlama, eski toplumun iyi bireylerinin bazı genlerinin yeni nesle aktarılmasının iyi olacağı düşüncesiyle yapılır. Çaprazlama oranı 0 ile 1 arasında olmak zorundadır. Çaprazlama oranının 0 olması yavruların atalarının aynısı olacağı, çaprazlama olasılığının 1 olması ise ataların taşıdığı bazı genlerin yeni nesle aktarılacağı anlamına gelmektedir (Şen 2004).

Mutasyon oranı kromozomların ne kadar sıklıkla mutasyon geçireceğini belirtir. Mutasyon olasılığı 0 ile 1 arasında alınmalıdır. Mutasyon oranının 0 olması durumunda yavrular çaprazlamadaki bireyler hiçbir değişikliğe uğratılmadan aynen kopyalanarak oluşturulur. 1 olması durumunda ise bütün kromozomların bazı genleri değişecektir. Mutasyon genellikle GA'nın yerel aşırılıklara düşmesini engeller. Mutasyonun büyük alınması durumunda GA rastgele aramaya dönüşebilir (Şen 2004).

Popülasyon büyüklüğü toplumdaki bireylerin (kromozom) sayısını gösterir. Eğer popülasyonda çok az kromozom varsa, arama uzayının çok küçük bir kısmı araştırılacak, ayrıca bireylerin çaprazlama olasılığı azalacağından farklı yeni yavruların elde edilmesi güçleşecektir. Eğer çok fazla birey varsa, bilgisayarın çalışması yavaşlayacaktır. Araştırmalar bazı sınırlardan sonra çok büyük toplumların kullanılmasının problemin hızlı bir şekilde çözülmesine yardımcı olmadığını göstermiştir (Şen 2004).

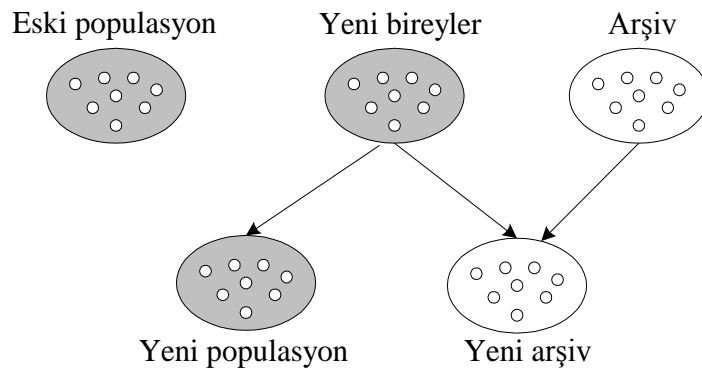
Nesil sayısı gelecekte oluşturulacak kuşakların tekrarlanma sayısını gösterir. Maksimum nesil sayısının 1000'den fazla alınmaması tavsiye edilir. Daha büyük alınması durumunda işlemci daha fazla hafıza alanına ihtiyaç duyacağından hızı yavaşlayacaktır.

Seçkinlik (Elitizm), optimizasyon süreci boyunca algoritmanın rastgelelik doğasından dolayı bir takım en iyi sonuçlar kaybedilebilir. Bu problemle başa çıkmak için seçkinlik operatörü kullanılır. Seçkinlik, en iyi kromozomların kaydedilip yeni nesle aktarılması işlemidir. Bu işlem, Şekil 3.28'de görüldüğü gibi bulunan en iyi çözümün yeni nesle aktarılmasını sağlayarak GA'nın global en iyiye ulaşma hızını artırır (Zitzler *et al.* 2004) .



Şekil 3.28. Seçkinlik

Yukarıdaki yöntem alternatif olarak, arşiv olarak isimlendirilen ikinci bir popülasyon oluşturulur. Böylece her bir jenerasyonda iyi bireyler arşive atılarak saklanır (Şekil 3.29). Arşiv, optimizasyon metodunda ayrıca bir dış bilgi saklama yeri olarak kullanılabilir (Zitzler *et al.* 2004) .



Şekil 3.29. Arşivleme ile seçicilik

Graf türü çalışmalarda, GA'da kromozom sunumunun ikili sayıdan ziyade tamsayı vektör sunumu olarak gösterilmesi daha uygundur. Şayet ikili sayı kullanılacak olursa mutasyon sonucunda aynı şehir bir turda iki kez geçebilir ya da illegal olan tura ulaşılabilir. Bu durumda araştırma uzayına geri dönülerek tekrardan kromozom oluşturan tamir algoritmalarına ihtiyaç duyulur. Tüm bu sakıncaları ortadan kaldırmak için kromozomların tamsayı vektör olarak kodlanması gerekmektedir (Michalewicz 1992).

Bu problemlerin hemen hepsi için, çok geniş bir çözüm uzayının tanımlanması gerekmektedir. Bu çözüm uzayının geleneksel yöntemlerle incelenmesi, çok uzun sürmesine karşın genetik algoritmayla kısa bir sürede kabul edilebilir sonuçlar alınabilmektedir.

3.2.4. Yapay bağışıklık sistemleri

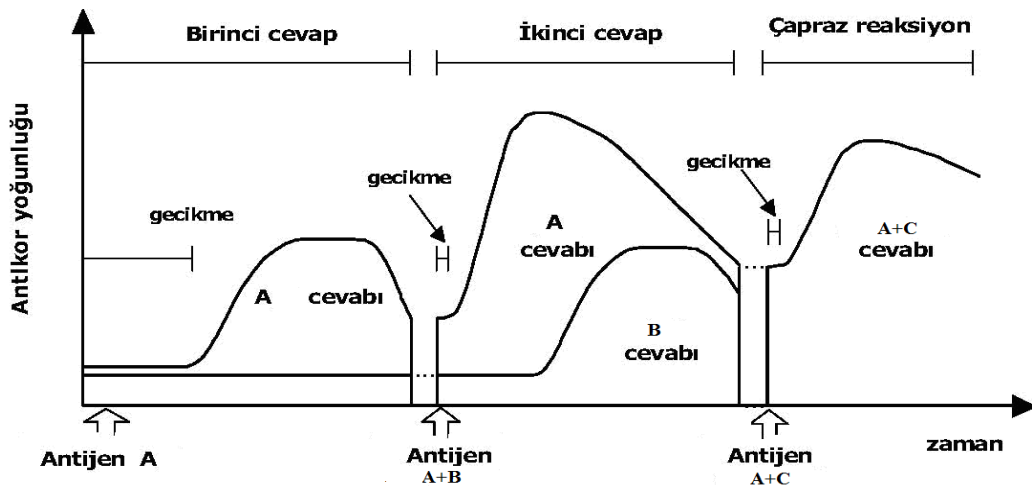
1900'lü yılların ikinci yarısından itibaren gelişmeye başlayan yapay zeka, insan gibi düşünen ve davranan makinelerin ortaya çıkarılması gibi amaçlara yönelik çalışmalarda büyük ölçüde insanın biyolojik sisteminin işleyişini referans almıştır. YBS'leri, doğal bağışıklık sistemlerinden esinlenerek oluşturulmuş, karmaşık problemlerin çözümünde ve sistemlerin geliştirilmesinde kullanılan sezgisel yöntemlerden biridir.

Doğal bağışıklık sistemi, vücudumuzu enfeksiyonlara yol açabilen virüs, bakteri, mantar ve parazit gibi mikroorganizmaların zarar verici etkilerine karşı koruyan çeşitli organ ve hücrelerden oluşan karmaşık bir sistemdir. Canlı organizmalarla bilgisayar sistemleri arasındaki bir takım benzerlikler bağışıklık sisteminin uygun bir yol gösterici olmasını sağlamıştır. Bağışıklık sisteminin araştırmalara yön veren en önemli özellikleri; öğrenme, algılama, hatırlama, genelleme ve sınıflandırma olarak sayılabilir.

Bağışıklık sisteminin en önemli özelliği olan öğrenme, genel olarak yabancı mikroorganizmaları yani antijenleri tanıyan lenfositlerin (bağışıklık hücresi) popülasyon büyüklüğünü ve benzerlikleri artırması faaliyetlerinden oluşmaktadır.

Normal bir bağışıklık sisteminde organizma, bir antijen ile bütün bir ömrü boyunca defalarca karşılaşır. Organizma antijen saldırısına ilk defa maruz kaldığı zaman her biri farklı benzerlik derecesine sahip antikorlar üretir ve az sayıda kopya oluşturarak lenfosit hücrelerinden olan B hücreleri ile karşılık verir. Bu ilk saldırıdan sonra yüksek benzerlikli antikor üreten hücreler hafıza hücreleri olarak saklanır. Böylece organizma aynı antijen saldırısına ikinci kez maruz kaldığı zaman büyük bir klon popülasyonu oluşturarak bağışıklık cevabının hem doğru hem de hızlı bir şekilde verilmesini sağlar.

Bağışıklık cevabını örneklemek için bir A antijeninin vücuttan içeriye girdiğini düşünelim. A antijeni, kendisini tanıyıp ona bağlanabilen antikorla karşılaşır ve bu antikorum sayısı belirli bir seviyeye kadar yükselir böylece bağışıklık sisteminin birincil cevabı oluşur. A antijeni vücuda tekrar girdiğinde daha önce kendisi için uygun antikor üretildiği için çok kısa bir gecikme sonucu birincil cevaptan daha kısa, yoğun ve etkili ikincil cevap verilir. Vücuda giren başka bir B antijeninin şayet yapısal olarak A antijeni ile hiçbir benzerliği yoksa cevap birincil cevap süresinde olacaktır. Şekil 3.30'da görüleceği gibi vücuda giren C antijeninin A antijeni ile arasında yapısal olarak bir benzerlik varsa bu durumda bağışıklık hafızası ilişkilendirebildiği için ikincil bir cevap verecektir, bu cevaba çapraz reaksiyon cevabı denir (Şahan 2004).

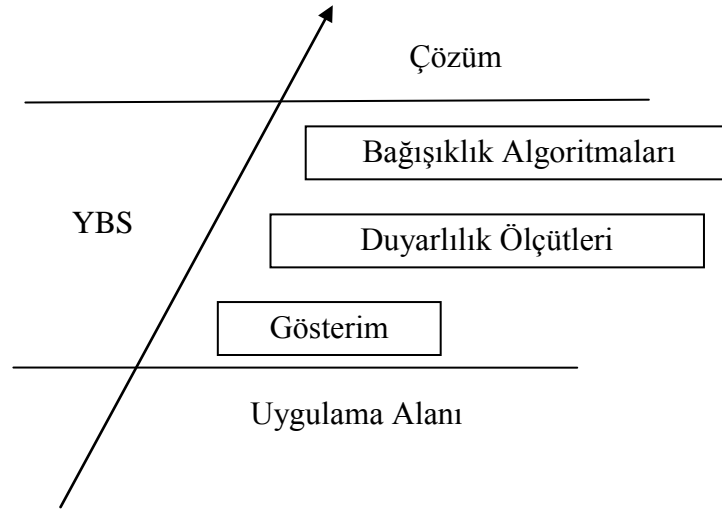


Şekil 3.30. Birincil, ikincil ve çapraz reaksiyon bağışıklık cevapları

1974 yılında Jerne'nin ortaya attığı Bağışıklık Ağı Teorisi yapay zeka araştırmacılarının ilgisini çekmiş böylece oluşturulan yeni algoritma ve uygulamalarla YBS'nin temelleri atılmıştır.

3.2.4.a. YBS'nin yapısı

YBS'de çözüme ulaşmak için bir takım aşamalardan geçilmesi gerekir. Bu aşamalar Şekil 3.31'de gösterilmiştir.



Şekil 3.31. YBS için katmanlı çatısı

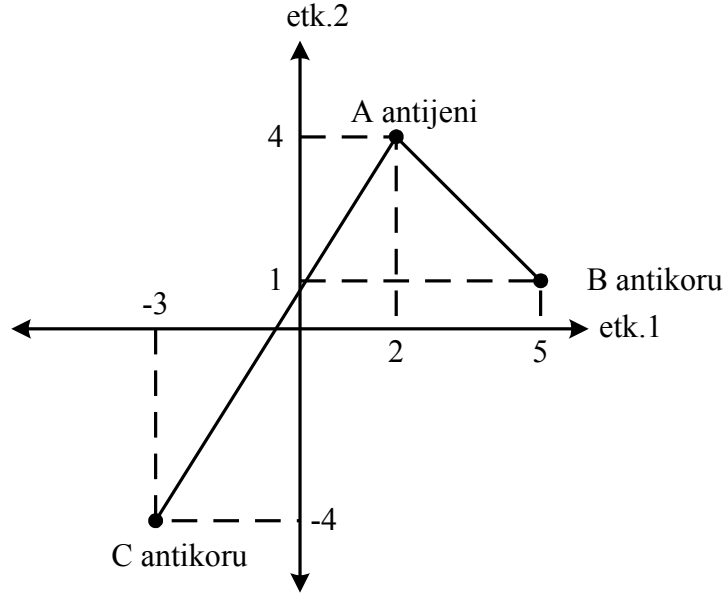
Gösterim: Problemi temsil edecek antijen ve antikorların ne şekilde gösterileceğine bu aşamada karar verilir. Matematiksel olarak bir antikor ya da antijen molekülü, L boyutlu bir şekil uzayında bir nokta olarak ifade edilebilir. Bu noktanın koordinatları dizi ile belirtilir.

$$A_b = [A_{b1}, A_{b2}, \dots, A_{bL}]$$

$$A_g = [A_{g1}, A_{g2}, \dots, A_{gL}]$$

Bu dizi içindeki her bir eleman, şekil uzayındaki bir eksene ait koordinat değeridir ve antikor-antijen arasındaki fizikokimyasal etkileşimlerden birini temsil eder.

Bir A antijeni ile B antikoru arasında iki çeşit fizikokimyasal etkileşim gerçekleşebilir kabulü yapılsın. Bu durumda A antijeni ile B antikoru göstermek için iki boyutlu bir şekil uzayı kullanılabilir. Şekil 3.32’de, A antijeni ile B ve C antikorumun etkileşimi gösterilmektedir.



Şekil 3.32. Şekil uzayı gösterimi

A antijeni ile C antikoru arasındaki uzaklık, B antikorundan fazla olduğu için, A antijeni ile C antikoru arasındaki etkileşimin şiddeti, A antijeni ile B antikoru arasındaki etkileşimin şiddetinden daha fazladır. Yani A antijeninin C antikoru ile daha çok benzediği söylenebilir (Şahan 2004).

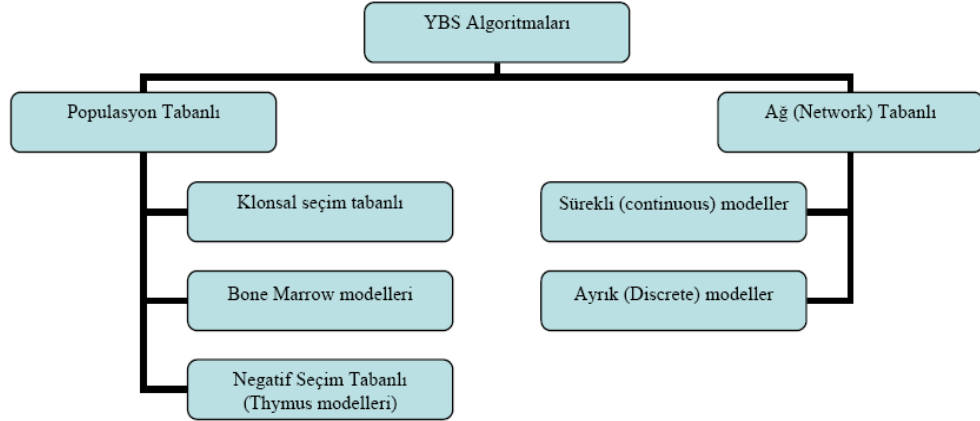
Duyarlılık ölçütleri: Antijen ve antikolar arasındaki etkileşim duyarlılık ile ifade edilir. Problemin türüne göre farklı duyarlılık ölçütleri vardır.

Öklit uzaklığı:
$$D = \sqrt{\sum_{i=1}^L (Ab_i - Ag_i)^2}$$

Hamming uzaklığı:
$$D = \sum_{i=1}^L \delta$$

$$\delta = \begin{cases} 1, & Ab_i \neq Ag_i \\ 0, & Ab_i = Ag_i \end{cases}$$

Bağışıklık algoritmaları: Problem türüne göre çok farklı algoritmalar geliştirilmiştir. Geliştirilen bu algoritmalar uygulama alanlarına ve yapılarına göre Şekil 3.33'te gruplandırılmıştır.



Şekil 3.33. YBS algoritmaları

YBS algoritmaları populasyon tabanlı ve ağ tabanlı olmak üzere iki temel gruba ayrılmaktadır. Populasyon Tabanlı algoritmalar da kendi içerisinde Klonal Seçim Tabanlı, Bone Marrow Modelleri ve Negatif Seçim Tabanlı olmak üzere üç gruba ayrılmaktadırlar. İkinci temel grup ise ağ tabanlı sistemlerin oluşturulduğu gruptur. Bağışıklık sistemi oldukça karmaşık bir yapıya sahiptir ve sahip olduğu dağıtılmış mekanizma bu karmaşık sistemdeki her bir birimin görevini net olarak belirler. Bu sayede bağışıklık sistemi oldukça zor görünen çoğu görevi iş bölümü aracılığı ile yerine getirebilir. Bağışıklık ağ modellerinin temel aldığı özellik, bu dağıtılmış mekanizmadır. Bağışıklık ağ modellerinde sistem, belirli görevleri yerine getirmekle sorumlu birçok birimden oluşur (Şahan 2004).

3.2.4.b. Klonal seçim algoritması

Klonal seçim algoritması klonal seçim, klonal büyüme, uygunluk olgunlaşması ve somatik hiper mutasyon mekanizmalarını içinde barındıran klonal seçim teorisini taklit eder (Brownlee 2007). Klonal Seçim Algoritması biçim tanıma, optimizasyon gibi kompleks problemlerin çözümüne uygulanmıştır. Bağışıklık sisteminin, repertuardan

bağımsız olarak hafıza hücrelerine sahip olması, en fazla benzerlik gösteren hücrelerin seçilip klonlanması, etkinlik gösteremeyen hücrelerin ölümü, benzerlik derecesini arttırmak için olgunlaştırma ve yeniden seçme, üretim ve çeşitliliğin korunması gibi özellikleri bu algorithmada uygulanmıştır. Klonal seçim teorisinde kullanılan terimleri Çizelge 3.6'daki gibi ifade edilebilir.

Çizelge 3.6. Klonal seçim teorisinde kullanılan terimler

Terim	Tanım
Antijen	Problem
Antikor, B-hücre, Lenfosit	Aday çözüm
Klon	Yeni örneklerin toplamı, nesil
Hafıza hücresi	Seçkin örnekler kümesi
Klonal seçim prensibi	Öğrenme prensibi
Hipermutasyon	Mutasyon, değişme
Repertuar	Populasyon, örneklerin toplamı
Uygunluk olgunlaşması	Yeniden örnekleme, geliştirme
Üreme	Klonlama
Seçim	Antijen-antikor eşleştirme
Benzerlik	Çözüm kalitesi, uygunluk

Klonal Seçim Algoritmasının adımları şu şekildedir:

1. Adım: Antikor repertuarı oluşturulur,
2. Adım: Antikorların benzerlik dereceleri belirlenir,
3. Adım: n adet en yüksek benzerlikte antikor seçilir,
4. Adım: Seçilen n adet antikorum benzerlik dereceleri oranında, yüksek benzerlik dereceli antikor daha fazla klonlanır,
5. Adım: Antikorların benzerlik derecesi yüksek olanlar, daha az olacak şekilde benzerlik mutasyonuna uğratılır,
6. Adım: Mutasyona uğramış klonların benzerlik dereceleri belirlenir,
7. Adım: En yüksek benzerlik dereceli antikorlar yeniden seçilir,
8. Adım: En düşük benzerlik derecesindeki bazı antikorlar yeni üretilen antikorlarla değiştirilir (Castro 2001).

3.2.5. Çok amaçlı optimizasyon

Optimizasyon teknikleri bir problemin en iyi çözümünü ortaya koymak için kullanılır. Sonuç olarak ortaya konulmuş olan çözüm optimum çözüm olarak adlandırılır. Amaç, her zaman için bu optimum çözümü yakalayabilmektir. Eğer optimize edilecek tek bir amaç varsa, problem tek amaçlı optimizasyon problemi, birden fazla amaç olması durumunda ise çok amaçlı optimizasyon problemi olarak tanımlanır. Çok amaçlı optimizasyon problemi aşağıdaki gibi formülle ifade edilir (Zitzler 1999).

$$\begin{aligned} \text{Max} \quad & y = f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{Kısıtlar} \quad & e(x) = (e_1(x), e_2(x), \dots, e_z(x)) \leq 0 \\ & x = (x_1, x_2, \dots, x_n) \in X \\ & y = (y_1, y_2, \dots, y_k) \in Y \end{aligned}$$

Burada;

x : karar vektörü

y : amaç vektörü

X : karar uzayı

Y : amaç uzayı

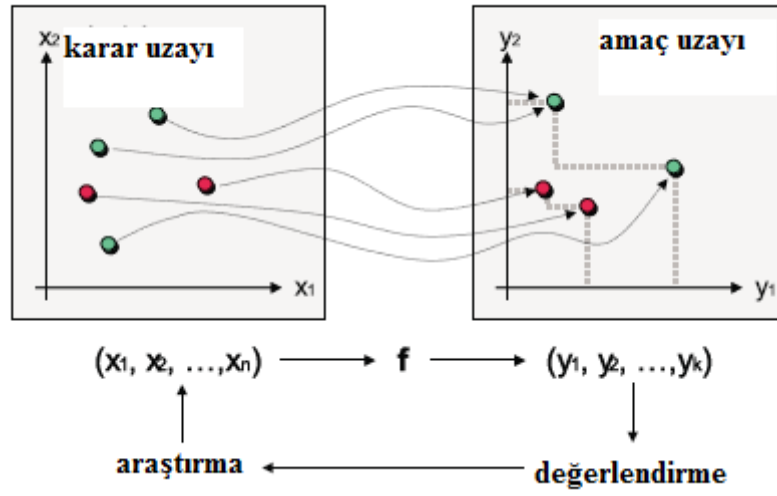
n : karar değişkenlerinin kümesi

k : amaç fonksiyonlarının kümesi

z : kısıtların kümesi

$e(x) \leq 0$ sınır şartı uygun çözümlerin kümesini belirler.

Çok amaçlı optimizasyon probleminin şematik gösterimi Şekil 3.34'teki gibi gösterilebilir.



Şekil 3.34. Çok amaçlı optimizasyon problemi

$e(x)$ sınırlarını sağlayan x karar vektörlerinin kümesi, uygun çözüm kümesi (X_f) olarak tanımlanır.

$$X_f = \{x \in X \mid e(x) \leq 0\}$$

Amaç uzayındaki uygun çözümler, başka bir deyişle X_f 'in görüntüsü, matematiksel olarak şu şekilde ifade edilir.

$$Y_f = f(X_f) = \bigcup_{x \in X_f} \{f(x)\}$$

Şayet amaç fonksiyonunu oluşturan karar değişkenleri sürekli ise bu tür problemler doğrusal programlama, kesikli ise kombinatoryal problemler olarak tanımlanır. Araç rotalama problemi, gezgin satıcı problemi gibi problemler kombinatoryal problemlere örnek verilebilir. Kombinatoryal problemlerin çözümünde sezgisel yöntemler kullanılmıştır. Klasik sezgisel yöntemler küçük boyutlu problemlerde optimum çözümler sağlasada problemin boyutu büyüdükçe bu yöntemler geçerliliğini yitirmektedir.

Optimizasyon problemlerinin çözümünde öncelikle doğrusal modellemeler kullanılmış ancak bu yöntemlerin her problemin çözümü için yeterli olmadığı görülmüştür. Son zamanlarda, çok amaçlı optimizasyon problemlerinin çözümünde evrimsel algoritmalar yaygın olarak kullanılmaya başlanmıştır. Çok amaçlı optimizasyon problemleri telekomünikasyon, ulaşım, bilgisayar bilimleri, ekonomi ve benzeri pek çok alanda uygulanmaktadır (Sağ 2009).

3.2.5.a. Çok amaçlı evrimsel algoritmalarda uygunluk ataması ve seçim

Evrimsel algoritmalar çok amaçlı problemlere uygulandığı zaman büyük bir problemle karşılaşılır. Bu problem çözüm kümesini araştırmada rehberlik eden uygunluk atama ve seçim işleminin ne şekilde yapılacağıdır.

a) Amaçları değiştirerek seçme

Bu metotta tek bir skaler değer vererek amaçları birleştirmek yerine, seçim fazı boyunca amaçlar arasında geçiş yapılır. Yani bir birey üretim için seçileceği zaman farklı bir amaç popülasyon üyesinin çiftleşme havuzuna gidip gitmeyeceğine karar verilecektir. Schaffer (1985), çiftleşme havuzunun farklı amaçlara göre eşit oranlarda doldurulması gerektiğini önerirken, Fourman (1985), amaçlara verilecek önem derecesine göre bir seçim planı yapılmasını önermiştir. Kursawe (1991), her bir amaca ihtimal atayarak bir sonraki adımdaki seçimde kullanılıp kullanılmayacağını belirlemiştir. Atanan ihtimal kullanıcı, tanımlı ya da rastgele olabilir. Burada ifade edilen yaklaşımların hepsi aşırı çözümler elde etme eğilimindedir ve şekil uzayında konveks ya da konkav olmayan çözümler elde edilmesini sağlayabilir (Zitzler 1999).

b) Varyasyon parametresi ile kümeleme seçimi

Bu seçimde amaçlar çeşitli parametreler eşliğinde tek bir fonksiyon içinde toplanır.

$$\text{Max } y = f(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) + \dots + w_k \cdot f_k(x)$$

$$x \in X_f$$

Parametreler(w_k) bütün bir çalışma boyunca sabit kalacağı gibi kimi uygulamalarda parametreler değiştirilerek her bir bireyin farklı amaç fonksiyonu ile çalışması sağlanmıştır. Çok amaçlı optimizasyonun etki alanının sınırlanması bu yöntemin dezavantajı olarak gösterilmektedir (Zitzler 1999).

c) Pareto esash seçim

Gerçek dünyada çoklu ve birbirine zıt amaçlar içeren ve oldukça karmaşık araştırma uzayına sahip olan birçok problemle sıkça karşılaşmaktadır. Bu tarz problemlerde tek bir optimal çözüm elde etmek yerine birbiriyle uyumlu birden fazla çözüm başka bir deyişle pareto optimal çözümler elde edilmesi uygun olmaktadır. Çok amaçlı optimizasyonda tüm amaçlar için kabul edilebilir düzeyde olan çözümlerin bulunması amaçlanır. Bunu sağlayan çözümlere Pareto-Optimal çözümler denir.

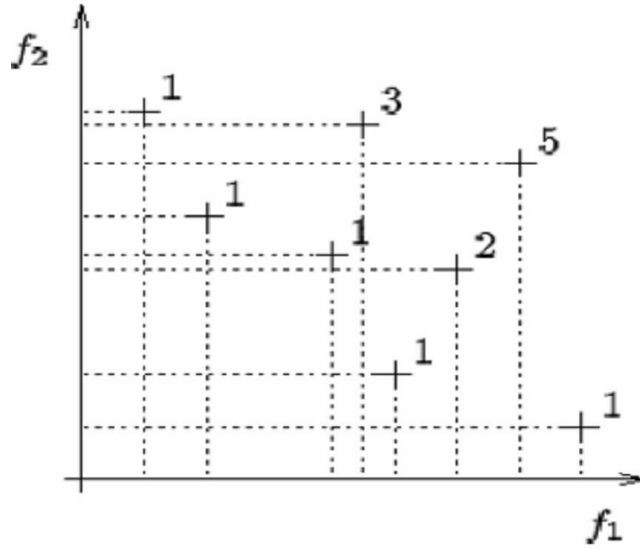
Pareto çözümler farklı yöntemlerle elde edilir. Bunlardan birisi pareto bastırılmayan çözümlerin elde edilmesidir. İki karar vektörü, a ve b olsun. Bu iki vektörün birbirlerine baskın olma durumları aşağıdaki şekilde formüle edilir.

$$a > b \text{ (} a, b' \text{ye baskındır)} \quad \text{if } f(a) > f(b)$$

$$a \geq b \text{ (} a, b' \text{ye zayıf baskındır)} \quad \text{if } f(a) \geq f(b)$$

$$a \sim b \text{ (} a, b' \text{ye benzerdir)} \quad \text{if } f(a) \geq f(b) \text{ veya } f(b) \geq f(a) \text{ değilse}$$

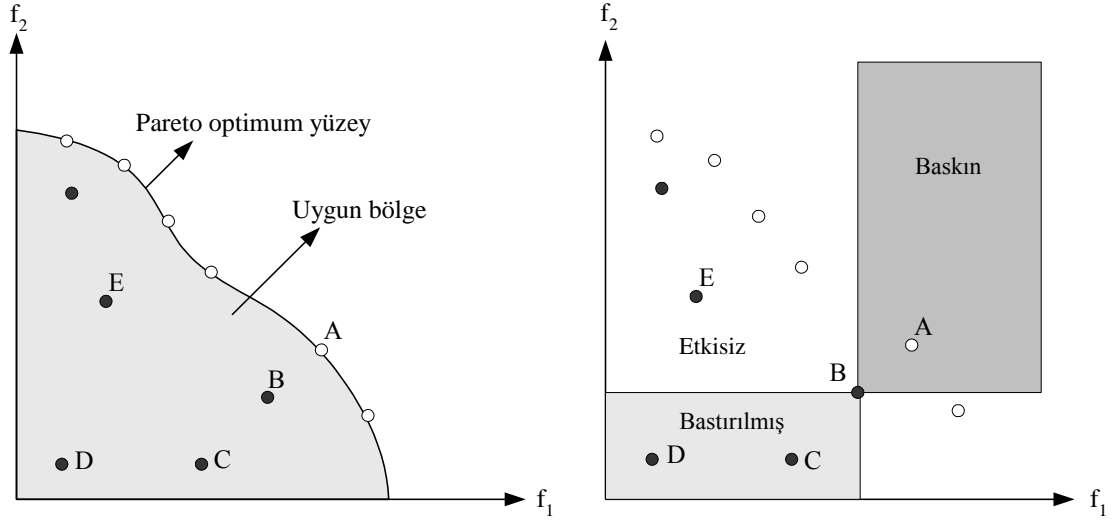
Pareto bastırılmayan çözümler elde edilirken bütün bastırılmamış bireyler birinci sıraya toplanır ve geçici olarak popülasyondan çıkartılır. Sonra gelen bastırılmamış bireyler ikinci sıraya toplanır ve işlem bütün bireyler sıralanıncaya kadar devam ettirilir. Böylece bireylerin sırası uygunluk değerlerini temsil eder. Bu durum Şekil 3.35'te gösterilmiştir.



Şekil 3.35. Pareto sıralama

Diğer metotlarda bireylerin uygunluğu popülasyondan bağımsız hesaplanırken paretoya dayalı seçimde birbirlerine bağlı olarak hesaplanır. Araştırma uzayı çok boyutlu ise çözüm bulmada etkili olmayabilir.

Yapılacak bir yolun performans ve ucuzluğunun maksimum yapılmak istendiği bir örneği ele alalım. f_1 performansı, f_2 ise ucuzluğu gösterebilir. Bu durumda her iki amacı maksimum yapan pareto optimal çözüm Şekil 3.36'da gösterildiği gibi olacaktır.



Şekil 3.36. Amaç uzayında pareto optimal çözümler

Şekil 3.36'da B noktası ile gösterilen karar vektörü, C ve D ile gösterilen karar vektörlerini bastırır. Çünkü her iki çözüm noktasına göre hem maliyet hem de ucuzluk bakımından üstündür. Fakat B noktası, A ile gösterilen karar vektörüne göre hem maliyet hem de ucuzluk yönünden zayıflık gösterir. Bu yüzden A noktası B yi bastırır denir. Sonuç olarak A karar vektörü B, C, D ve E karar vektörlerine üstünlük gösterir. Bu nedenle A noktasına pareto optimal çözüm denir. Şekil 3.36'da beyaz noktalar pareto optimal noktaları göstermektedir. Bütün pareto optimal çözümlerin kümesine pareto optimal küme ve bu kümeye karşılık gelen amaç vektörlerine ise pareto optimal yüzey denmektedir (Zitzler 1999).

3.2.6. Taguchi metodu

Taguchi metodu Genichi Taguchi tarafından 1960'lı yıllarda kalitenin geliştirilmesi konusunda deneysel tasarıma bağlı olarak geliştirilmiş, istatistiksel bir yöntemdir. Deney sayısının azaltılması, kontrol faktörlerinin kalite karakteristiği üzerindeki etkilerinin ve optimum faktör seviyelerinin kolaylıkla elde edilmesi, bu yöntemin en önemli avantajları arasında sayılmaktadır (Pınar ve Güllü 2010).

Taguchi metodunun uygulama adımları:

1. Adım: Faktör ve etkileşimleri belirlenir,
2. Adım: Faktör seviyeleri seçilir,
3. Adım: Uygun ortogonal dizi seçilir,
4. Adım: Seçilen ortogonal matrise faktör ve etkileşimleri atanır,
5. Adım: Deneyle yapılır,
6. Adım: Veriler analiz edilir,
7. Adım: Faktörlerin optimum seviyeleri tespit edilir,
8. Adım: Doğrulama deneyleri yapılır,
9. Adım: Metodun güven aralığı tespit edilir.

Taguchi deneysel tasarımı, kontrol faktörleri ile kontrol edilemeyen faktörler (gürültü faktörleri), arasındaki ilişkiyi tanımlar. Ortogonal dizi olarak isimlendirilen bir takım dizayn matrisleri vasıtasıyla veriler ayrı ayrı işlenir. Taguchi metodunda elde edilen sonuçlar S/N oranı kullanılarak incelenir. Performans karakteristiği olarak da tanımlanan S/N oranı, kalite hedef değerine göre farklı şekillerde hesap edilir. S/N oranının hesaplanmasında hedeflenen kalite değerinin niteliği belirleyici unsurdur. Üç temel unsurdan bahsedilir. Bunlar en küçük en iyi, en büyük en iyi ve nominal en iyidir (Savaşkan ve diğ. 2004).

$$\text{En büyük en iyi } S/N = -10 \text{Log} \left[\frac{1}{n} \sum_{1}^n \frac{1}{Y_i^2} \right]$$

$$\text{En küçük en iyi } S/N = -10 \text{Log} \left[\frac{1}{n} \sum_{1}^n Y_i^2 \right]$$

$$\text{Nominal en iyi } S/N = -10 \text{Log} \left[\frac{1}{n} \sum_{1}^n (Y_i - Y_o)^2 \right]$$

S/N : Performans istatistiği (desibel)

n : Yineleme sayısı

Y_i : i. deneyin performans istatistiği

Y_0 : Nominal arzu edilen deęer

Yapılan alıřmada arzu edilen en iyi deęer mevcut olduęu iin, nominal en iyi performans karakteristięi kullanılmıřtır.

S/N performans istatistięini maksimum yapan parametre seviyeleri optimumdur. Ancak Taguchi Metodu iin optimum parametre seviyelerini veren herhangi bir alıřma yapılmamıř ise optimum performans deęeri ařaęıdaki modelden yararlanılarak tahmin edilebilir (Hınıřlıoęlu and Bayrak 2004).

$$Y_i = \mu + X_i + e_i$$

μ : Performans deęerinin genel ortalaması

X_i : Parametre-seviye kombinasyonunun sabit etkisi

e_i : i. deneydeki rastgele hata

Doęrulama deneylerinden elde edilen sonuların anlamlı olup olmadıklarını belirleyebilmek iin belirlenen hata seviyesinde gven aralıęı;

$$\mu = \sqrt{F_{\alpha;1,SD_{MSe}} MSe \left[\frac{1+m}{N} + \frac{1}{n_t} \right]}$$

baęıntısıyla hesaplanır. Burada;

F : Tablo deęeri,

α : Hata seviyesini,

SD_{MSe} : Hata kareler ortalamasının serbestlik derecesi toplamını,

N : Toplam deney sayısı

n_t : Doęrulama deneylerindeki tekrar sayısını gstermektedir.

3.3. Çalışmada Kullanılan Model ve Algoritmalar

3.3.1. Taguchi optimizasyonu

Bu çalışmada kullanılan S/N oranı nominal en iyi performans istatistiği kullanılarak hesap edilmiştir. GA için mutasyon oranı, çaprazlama oranı, populasyon sayısı ve nesil sayısı olmak üzere dört farklı parametre için üç farklı seviye belirlenmiştir. YBS için mutasyon oranı, klonlama oranı, populasyon sayısı ve nesil sayısı olmak üzere dört farklı parametre için üç farklı seviye belirlenmiştir. Taguchi optimizasyonunda kullanılan parametreler ve seviye değerleri Çizelge 3.7 ve 3.8’de gösterilmiştir.

Çizelge 3.7. GA için kullanılan parametre ve seviye değerleri

Parametre	Seviye		
	1	2	3
Mutasyon Oranı	0.1	0.5	1
Çaprazlama Oranı	0.1	0.5	1
Populasyon Sayısı	10	50	100
Nesil Sayısı	10	100	300

Çizelge 3.8. YBS için kullanılan parametre ve seviye değerleri

Parametre	Seviye		
	1	2	3
Mutasyon Oranı	0.1	0.5	1
Klonlama Oranı	0.1	0.2	0.5
Populasyon Sayısı	10	30	100
Nesil Sayısı	10	75	200

Bu çalışmada deney koşulları L_9 ortogonal dizisine göre Çizelge 3.9 ve 3.10’daki gibi şekilde planlanmıştır.

Çizelge 3.9. GA için L_9 ortogonal dizisine göre yapılacak çalışma planı

Deney No	Mutasyon Oranı	Çaprazlama Oranı	Populasyon Sayısı	Nesil Sayısı
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Çizelge 3.10. YBS için L_9 ortogonal dizisine göre yapılacak çalışma planı

Deney No	Mutasyon Oranı	Klonlama Oranı	Populasyon Sayısı	Nesil Sayısı
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

3.3.2. Algoritmalarda kullanılan uygunluk fonksiyonları

Bu çalışmada EKY problemi yönlü ve devresiz ağlar için çözülmüştür. Bu ağlar V düğümler kümesi, A kenarların kümesi olmak üzere $G=(V,A)$ ile gösterilsin. i, j G ağındaki düğüm noktaları olmak üzere i den j ye tek bir yönlendirilmiş (i,j) kenar olduğu farz edilir. Yolun x ile temsil edildiğini farz edelim. Bu durumda;

$$x = \{x_{ij} | (i, j) \in A\}$$

Şayet (i, j) ağa ait bir kenar ise x , 1 değerini alır aksi takdirde 0 olur.

$w_{ij} = (i, j)$ kenarının ağırlığını gösterebilir. Kenar ağırlığı problemin türüne göre değişmektedir.

Deterministik EKY probleminde w_{ij} düğümler arasındaki mesafeyi (km), kenarların ortalama seyahat süresini (dk) ya da kenarlardaki ortalama yakıt tüketimini (lt) göstermektedir.

Stokastik EKY probleminde w_{ij} düğümler arasındaki kenarların seyahat süresi dağılımını ve taşıtların yakıt tüketim dağılımını göstermektedir (**EK 6, 7, 9 ve 10**). Bu durumda:

Deterministik EKY uzunluğu:

$$\text{Uygunluk fonksiyonu, } T(x, w) = \min \sum_{(i,j) \in A} x_{ij} w_{ij}$$

Kısıtlar;

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1, i = 1 \\ 0, 2 \leq i \leq n-1 \\ -1, i = n \end{cases}$$

$$x_{ij} = \begin{cases} 1, (i, j) \in A \\ 0, (i, j) \notin A \end{cases}$$

Stokastik en kısa yol uzunluğu:

1 Model: Verilen başlangıç T_o (seyahat süresi ya da yakıt tüketim miktarı) değerinden daha az değerde gidilme ihtimali en yüksek yol olarak tanımlanır. Şayet x yolu aşağıdaki şartı sağlarsa ihtimale dayalı EKY modeli olarak ifade edilir.

$$\Pr\{T(x, w) \leq T_o\} \geq \Pr\{T(x', w) \leq T_o\}$$

$$\text{Uygunluk fonksiyonu; } \max \Pr\left\{ \sum_{(i,j) \in A} x_{ij} w_{ij} \leq T_o \right\}$$

Kısıtlar;

$$x_{ij} = \begin{cases} 1, (i, j) \in A \\ 0, (i, j) \notin A \end{cases}$$

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1, i = 1 \\ 0, 2 \leq i \leq n-1 \\ -1, i = n \end{cases}$$

2. Model: Verilen α ihtimalle en az sürede veya en az yakıtta gidilen yol olarak tanımlanır. x yolu aşağıdaki şartı sağlarsa α EKY olarak isimlendirilir.

\bar{T} = Verilen α ihtimaline karşılık gelen eky değeri

$$\min\{\bar{T} \mid \Pr\{T(x, w) \leq \bar{T}\} \geq \alpha\} \leq \min\{\bar{T} \mid \Pr\{T(x', w) \leq \bar{T}\} \geq \alpha\}$$

Bu durumda uygunluk fonksiyonu ve kısıtlar aşağıdaki şekilde ifade edilir.

Uygunluk fonksiyonu $\min \bar{T}$,

Kısıtlar;

$$\Pr\{T(x, w) \leq \bar{T}\} \geq \alpha,$$

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1, i = 1 \\ 0, 2 \leq i \leq n-1 \\ -1, i = n \end{cases},$$

$$x_{ij} = \begin{cases} 1, (i, j) \in A \\ 0, (i, j) \notin A \end{cases}$$

3.3.3. Çalışmada kullanılan algoritmalar

Çalışmamızda EKY problemini çözmek için GA ve YBS algoritmaları kullanılmıştır. Farklı büyüklükte ağlar için problemin yapısına uygun olarak değişik çaprazlama ve mutasyon operatörleri kullanılarak farklı algoritmalar üretilmiş ve optimum sonuçlar elde edilmesi hedeflenmiştir. Ağ türüne göre kullanılan farklı çaprazlama ve mutasyon metotları Çizelge 3.11 ve 3.12'de verilmiştir.

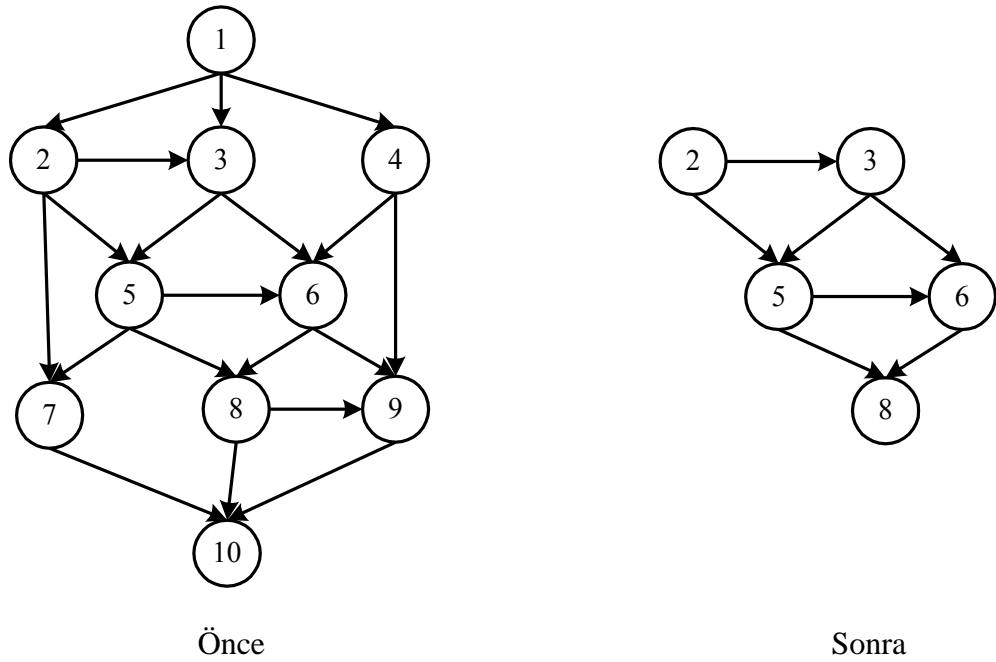
Çizelge 3.11. Çaprazlama metotları

Graf Türü	Çaprazlama Metodu
Küçük	Tek nokta
Orta	Tek nokta
Büyük	İki nokta

Çizelge 3.12. Mutasyon metotları

Graf Türü	Mutasyon Metodu
Küçük	Tek nokta
Orta	Tek nokta
Büyük	İki nokta

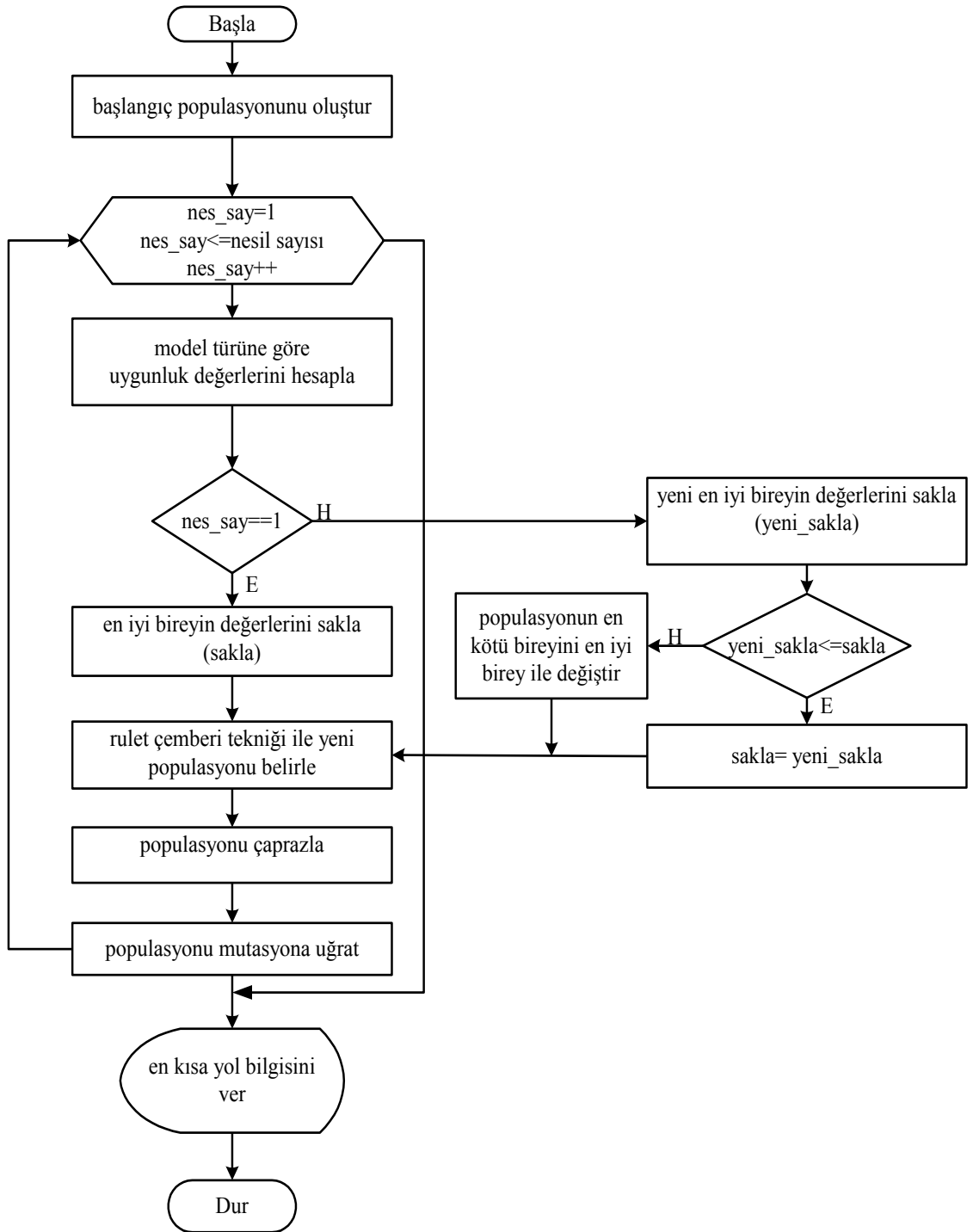
Ayrıca büyük ağlarda ara hedef düğüm planı ve iki nokta mutasyon metodu uygulandığından, rastgele oluşturulan kromozomların legal rotaları vermesi için hem GA hemde YBS' de düğüm budama operatörü devreye sokulmuştur. Böylece tamir algoritması kullanma ihtiyacı ortadan kaldırılmıştır. Bu operatörün rolü, verilen ara başlangıç ve hedef düğümü arasındaki birbirleriyle bağlantılı olan düğümleri alıp ilgili olmayanları budayarak yeni bir komşuluk matrisi oluşturmaktır. Örneğin 2 numaralı düğümden 8 numaralı düğüme gitmek için ağ budama algoritmasının uygulaması Şekil 3.37'de gösterilmiştir.



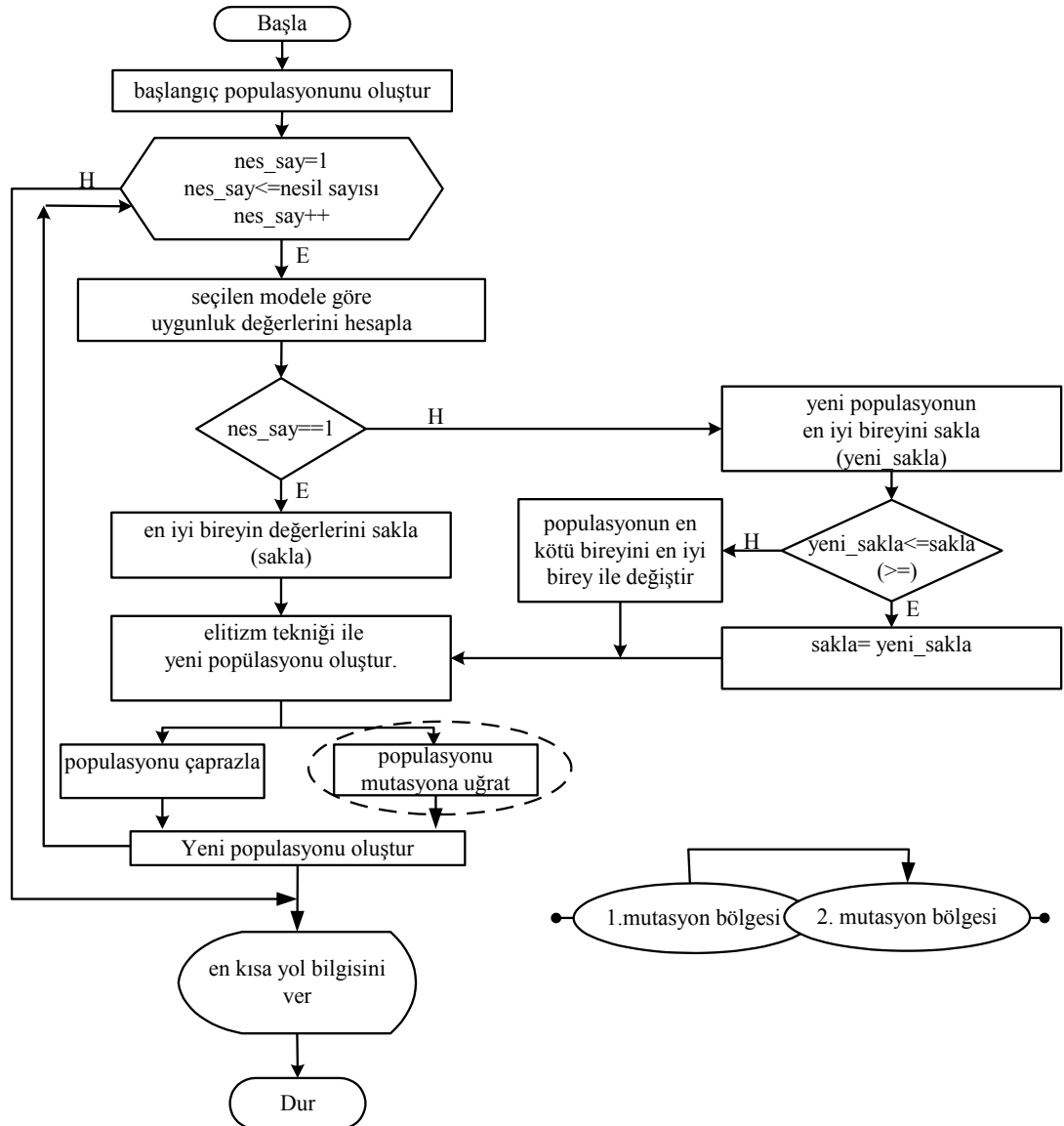
Şekil 3.37. Budama operatörünün şematik gösterimi

3.3.3.a. Çalışmada kullanılan GA'lar

Çalışmada kullanılan ağın büyüklüğüne göre, GA'lar farklı şekillerde modellenmiştir. Şekil 3.38-3.40'da GA akış diyagramları gösterilmiştir.

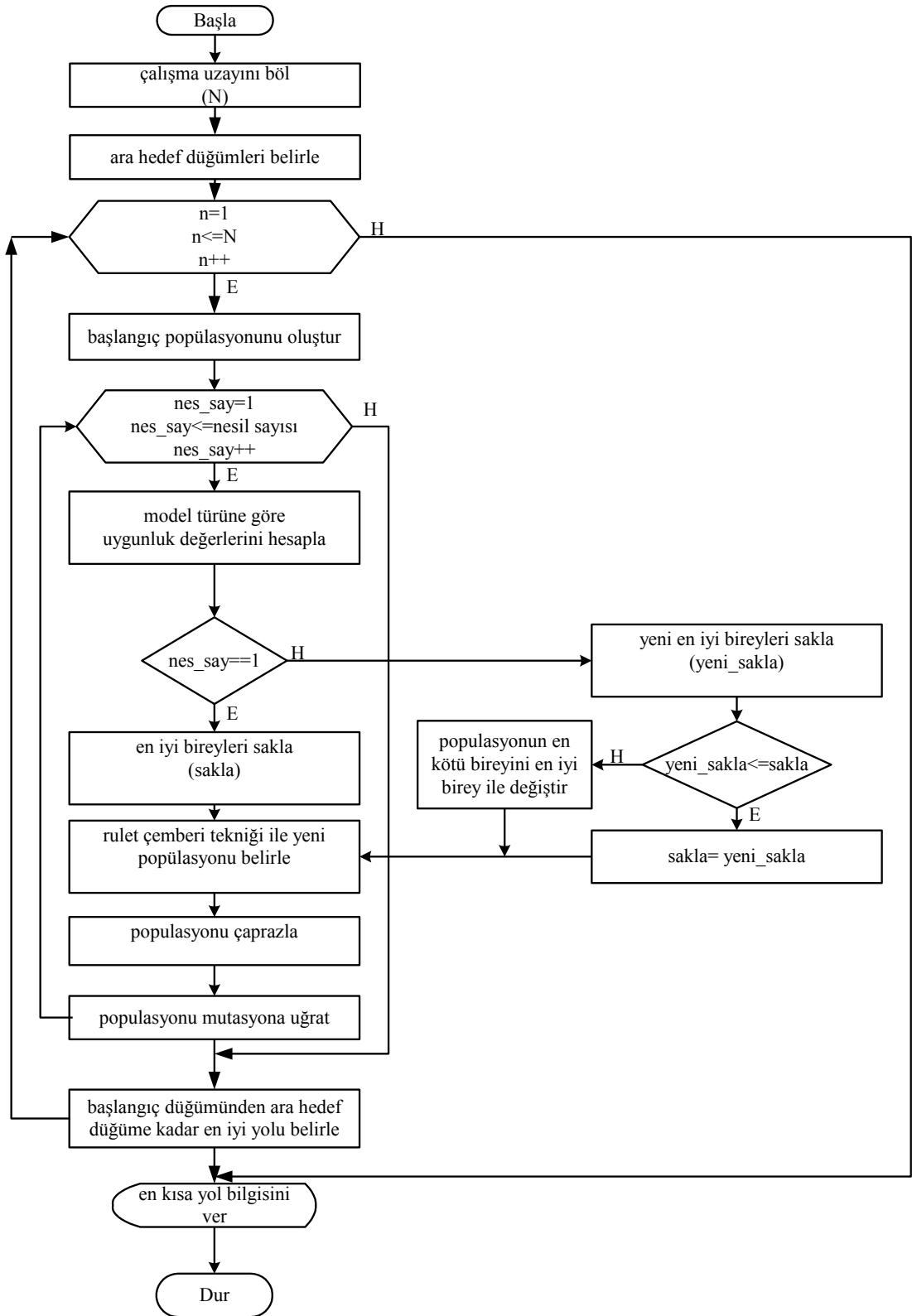


Şekil 3.38. 50 düğümlü ağ için GA akış şeması



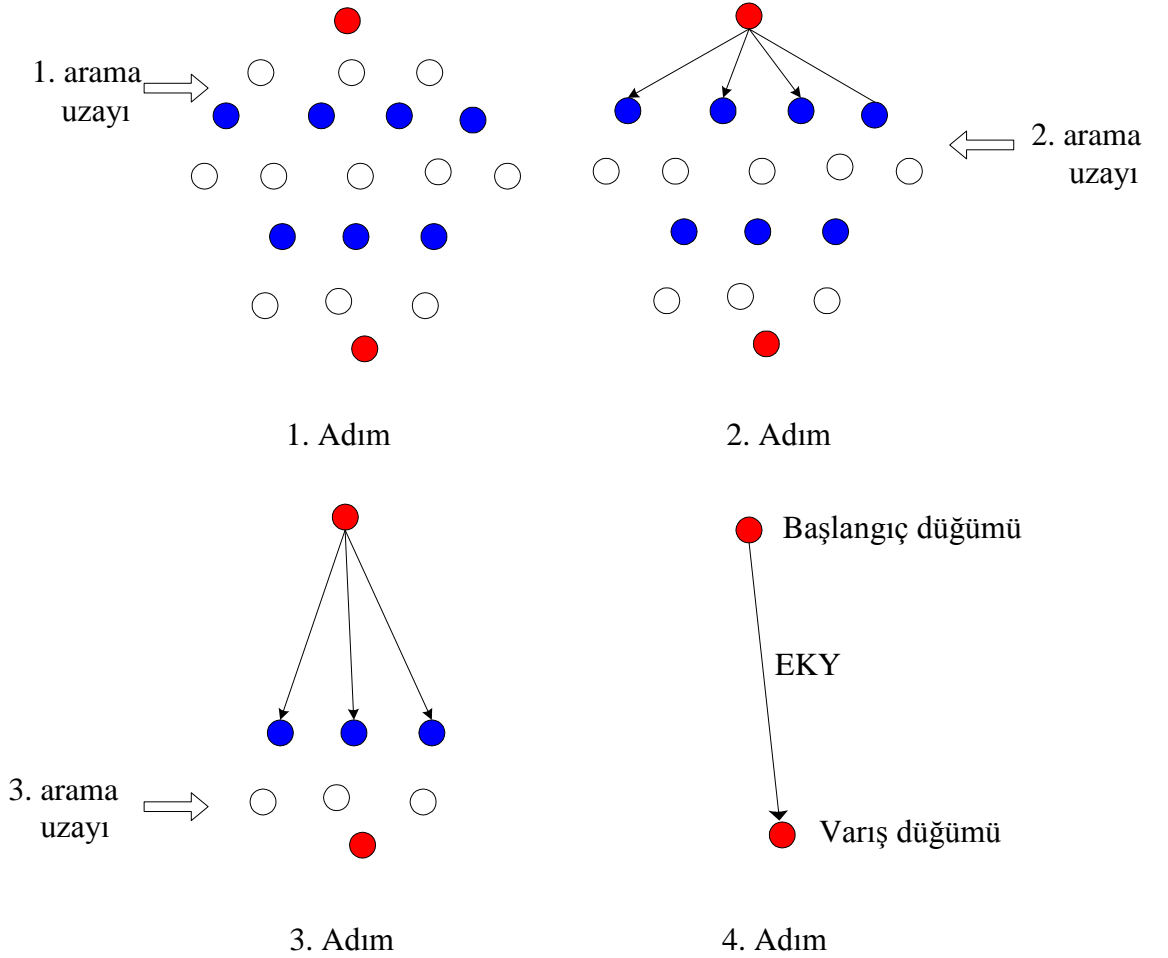
Şekil 3.39. 100 düğümlü ağ için GA akış şeması

Orta büyüklükteki ağlar için diğer algoritmalarından farklı olarak mutasyon planı algoritmalara dahil edilmiştir. Nesilin yarısında ağın 1. bölgesi için mutasyon uygulanmış nesilin geri kalan kısmında ağın 2 bölgesi için mutasyon uygulanmıştır.



Şekil 3.40. Büyük ağlar için GA akış şeması

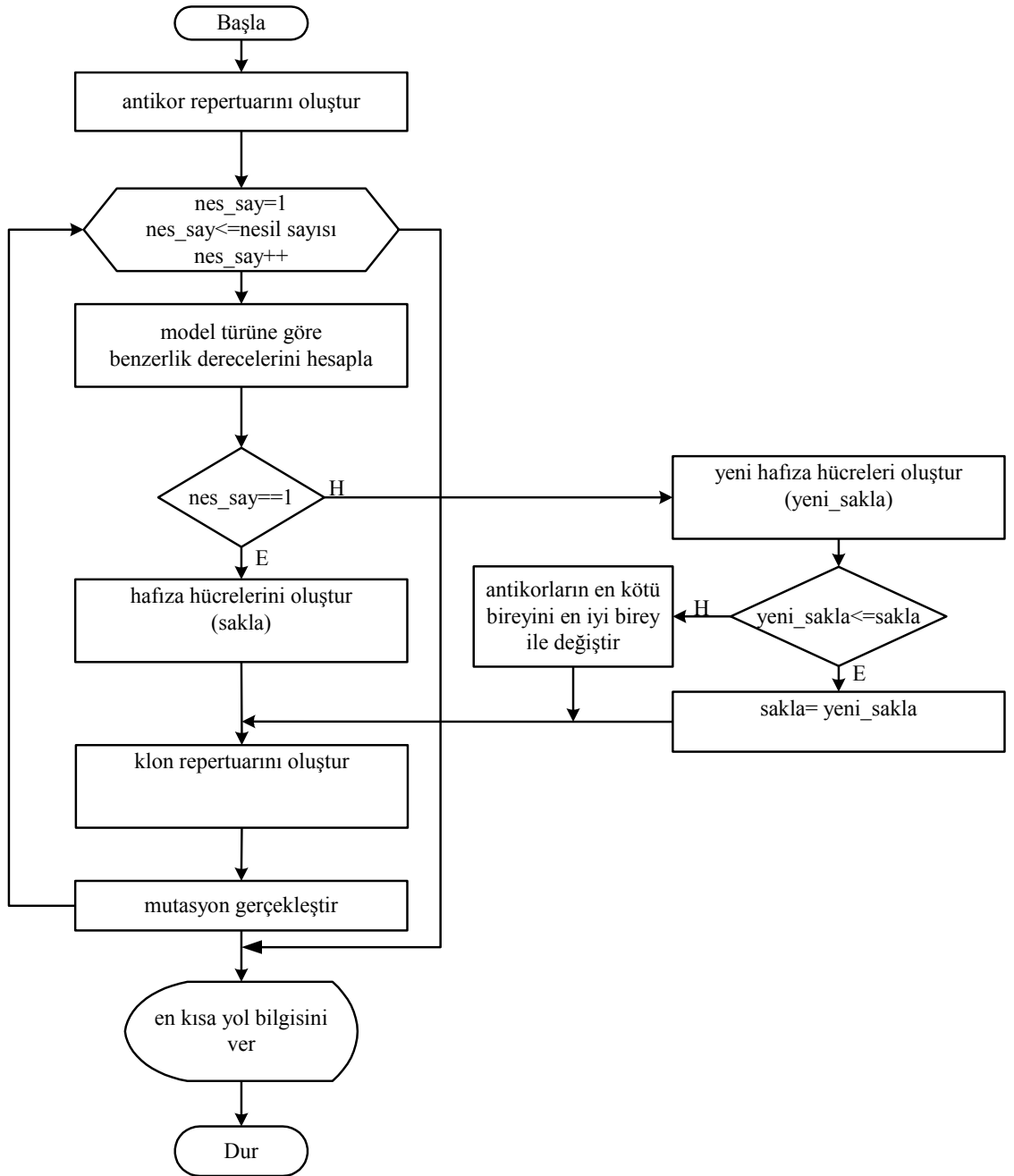
Büyük graflarda araştırma uzayı çok büyük olduğundan dolayı hedef değere istenilen seviyede yakınsayamama problemi ortaya çıktığından bu ağ türü için diğer algoritmalarından farklı olarak ağ içerisinde ara hedef düğümler belirlenip araştırma uzayı küçük parçalara ayrılmıştır.



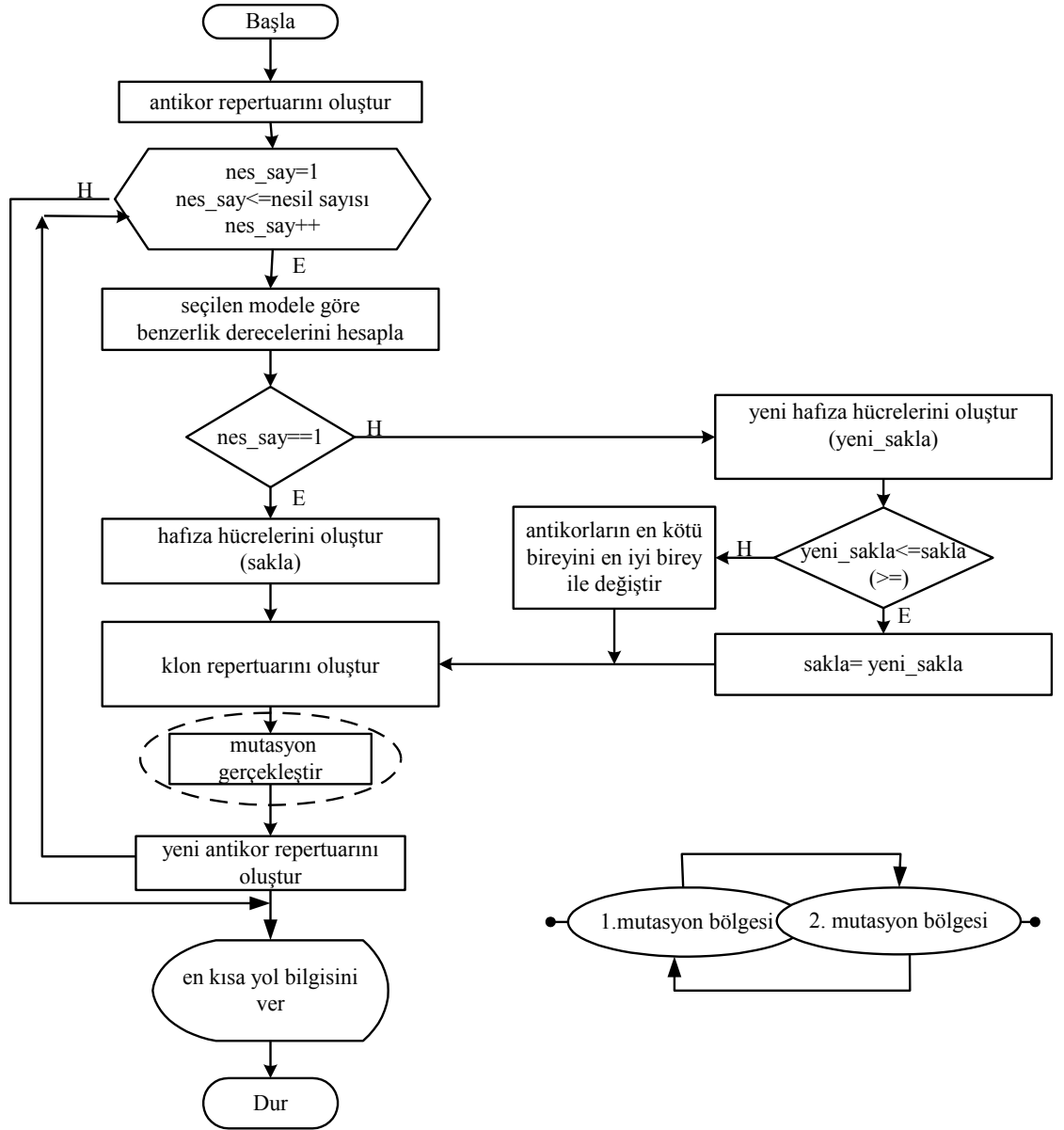
Şekil 3.41. Büyük ağlarda algoritmanın ilerleme adımları

3.3.3.b. Çalışmada kullanılan YBS algoritmaları

Ağın büyüklüğüne göre YBS algoritmaları farklı şekillerde modellenmiştir. Şekil 3.42-3.44'te YBS akış diyagramları gösterilmiştir.

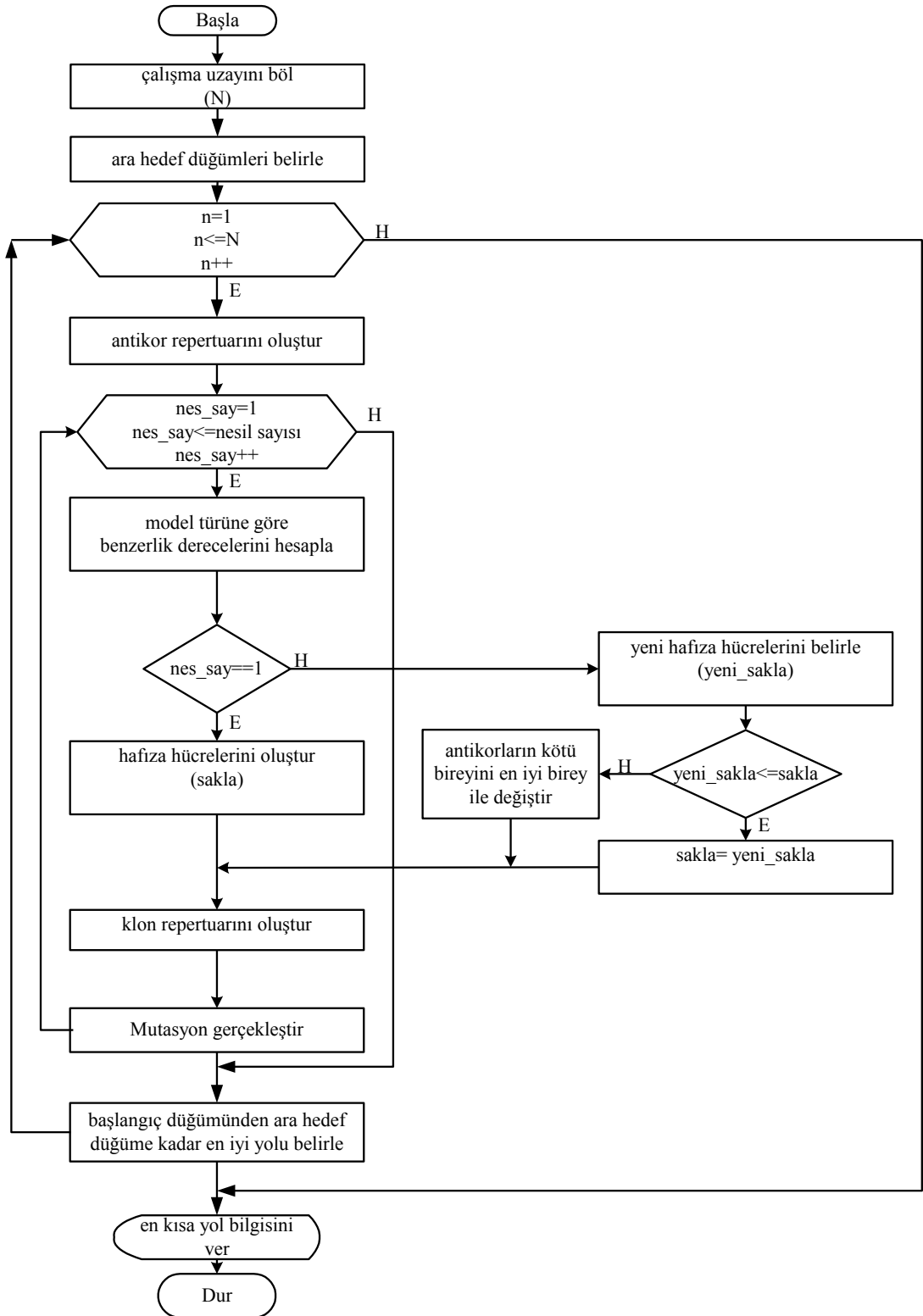


Şekil 3.42. 50 düğümlü ağ için YBS akış şeması



Şekil 3.43. 100 düğümlü ağ için YBS akış şeması

Nesilin %20'si ile %50 ve %70'i arasında ağın 1. bölgesi için mutasyon uygulanmış, nesilin geri kalan kısımlarında ağın 2 bölgesi için mutasyon uygulanmıştır.



Şekil 3.44. Büyük ağlar için YBS akış şeması

GA'da olduğu gibi büyük ağlarda ara hedef düğümler belirlenerek EKY çözülmüştür.

3.3.3.c. GA ve YBS’de kullanılan deęişkenler

Maliyet bilgisinin deterministik ya da stokastik olma durumuna göre algoritmalar farklı şekillerde modellenmiştir. Maliyet bilgisinin türüne ve kullanılan yöntemle göre algoritmalarda kullanılan bazı girdiler aynı olmasına rağmen bazı girdi ve çıktı bilgilerinde farklılıklar bulunmaktadır.

Girdiler

- Ağ komşuluk matrisi
- Başlangıç düğümü
- Varış düğümü
- Kromozom uzunluğu
- Populasyon sayısı
- Nesil sayısı
- Mutasyon olasılığı
- Modelin türüne göre
 - ❖ Stokastik simülasyon sayısı
 - ❖ Hedef maliyet (stokastik model 1)
 - ❖ Güven seviyesi (stokastik model 2)
- Algoritmanın türüne göre
 - ❖ Çaprazlama olasılığı (GA)
 - ❖ Klonlama oranı (YBS)
- Grafın büyüklüğüne göre
 - ❖ Mutasyon bölüm oranı (orta graflar için)
 - ❖ Graf bölüm sayısı (büyük graflar için)

Çıktılar

- Modelin türüne göre
 - ❖ En az maliyet (deterministik)
 - ❖ Hedef maliyetin altında en büyük gitme ihtimali (model1 için)
 - ❖ Güven seviyesinde gidilebilecek en az maliyet (model 2 için)
- En kısa yolun rotası

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Bu çalışmada 50, 100, 250, 500 ve 1000 düğümlü ağ modelleri oluşturulmuştur. Bu ağ modellerinde uzunluk, zaman ve yakıt dağılımı verileri incelenmiştir. Ağdaki her bir kenarın kat edilme süresinin normal, üçgen ve üniform dağılıma; kenarlardaki yakıt tüketiminin ise normal dağılıma uygun olduğu kabul edilmiştir. Gerçekleştirilen stokastik simülasyonla bu dağılımlara bağlı olarak her bir kenar için 100 farklı zaman ve yakıt değeri üretilmiştir.

GA ve YBS algoritmaları, Intel Core 2 Duo CPU 2.00 GHz işlemci, 2 GB bellek ve 32 bit işletim sistemine sahip Windows Vista kurulu bilgisayarda MATLAB ortamında modellenmiştir.

GA ve YBS algoritmalarında kullanılan parametrelerin optimum değerlerini elde etmek için Taguchi optimizasyonu uygulanmıştır. Optimizasyondan elde edilen optimum parametre değerleri ile EKY problemi öncelikle her bir ağ için GA ve YBS yöntemleri ile deterministik olarak modellenmiş ve klasik algoritmalar olan Floyd ve Dijkstra algoritmaları ile karşılaştırılmıştır. Daha sonra stokastik modeller kurularak GA ve YBS ile EKY problemi çözülmüş ve sonuçlar birbirleri ile karşılaştırılmıştır. Belirlenen üç problem türü için GA ve YBS ile çok amaçlı optimizasyon gerçekleştirilmiştir. Elde edilen pareto çözümleri için maliyet değerlendirilmesi yapılmıştır. GA ve YBS ile elde edilen sonuçlar performans ve bilgisayar işlem zamanı yönüyle birbiriyle karşılaştırılmıştır.

4.1. Taguchi Optimizasyonu Sonuçları

GA ve YBS için belirlenen faktör seviyelerine ve L_9 ortogonal dizisine göre deterministik EKY değerleri elde edilerek optimum parametre seviyeleri tespit edilmiştir.

4.1.1. GA parametreleri için Taguchi optimizasyonu

Üzerinde çalışılacak her bir ağ için belirlenen GA parametreleri kullanılarak elde edilen EKY değerleri ile S/N oranı hesap edilerek Çizelge 4.1 ve 4.2’de verilmiştir.

Çizelge 4.1. GA için deterministik EKY değerleri

Sıra no	GA ile elde edilen EKY uzunlukları (km)									
	50 düğüm		100 düğüm		250 düğüm		500 düğüm		1000 düğüm	
1	84	85	244	240	590	597	1205	1144	3031	3063
2	78	79	211	208	542	545	1058	1065	2753	2743
3	72	71	206	206	541	541	1055	1054	2735	2735
4	79	82	209	206	541	542	1054	1054	2735	2735
5	72	79	217	218	547	546	1084	1084	2787	2820
6	83	92	209	213	543	541	1063	1064	2747	2750
7	71	71	207	210	541	541	1054	1061	2735	2745
8	71	71	209	207	542	541	1056	1059	2762	2738
9	78	71	214	218	551	545	1093	1084	2798	2804

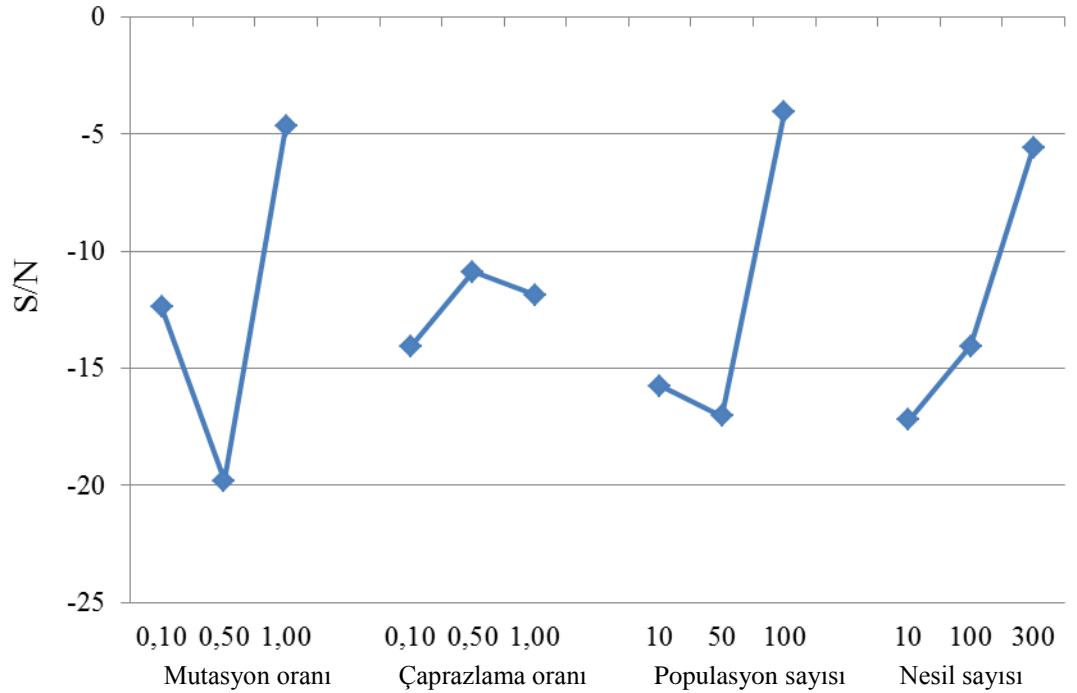
Çizelge 4.2. GA için S/N değerleri

Sıra no	GA için S/N değerleri				
	50 düğüm	100 düğüm	250 düğüm	500 düğüm	1000 düğüm
1	-22,61	-31,14	-34,42	-41,89	-49,89
2	-17,52	-11,61	-9,29	-18,36	-22,88
3	3,01	0,00	0,00	3,01	0,00
4	-19,66	-6,53	3,01	0,00	0,00
5	-15,12	-21,22	-14,84	-29,54	-36,96
6	-24,66	-14,62	-3,01	-19,57	-22,66
7	0,00	-9,29	0,00	-13,89	-16,99
8	0,00	-6,99	3,01	-11,61	-25,67
9	-13,89	-20,17	-17,63	-30,83	-36,40
Ortalama	-12,27	-13,51	-8,13	-18,08	-23,49

Her bir ağ için ortalama S/N değerleri bulunup maksimum olanları tespit edilerek optimum parametre seviyeleri Çizelge 4.3-4.7’de belirlenmiş ve Şekil 4.1-4.5’te gösterilmiştir.

Çizelge 4.3. 50 düğümlük ağda GA parametreleri için optimum seviyeler

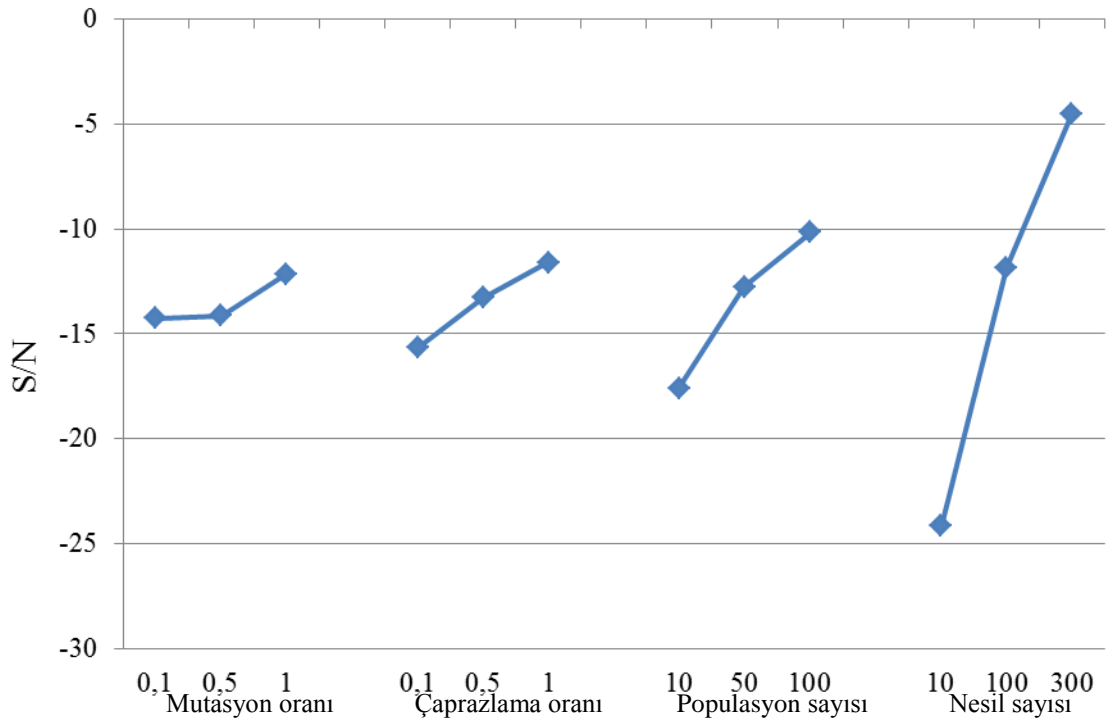
Seviye No	Mutasyon Oranı	Çaprazlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-12,37	-14,09	-15,76	-17,21
2. seviye	-19,81	-10,88	-17,02	-14,06
3. seviye	-4,63	-11,85	-4,04	-5,55
Optimum Seviye	3	2	3	3



Şekil 4.1. 50 düğümlük ağda GA parametrelerinin ortalama S/N etkileri

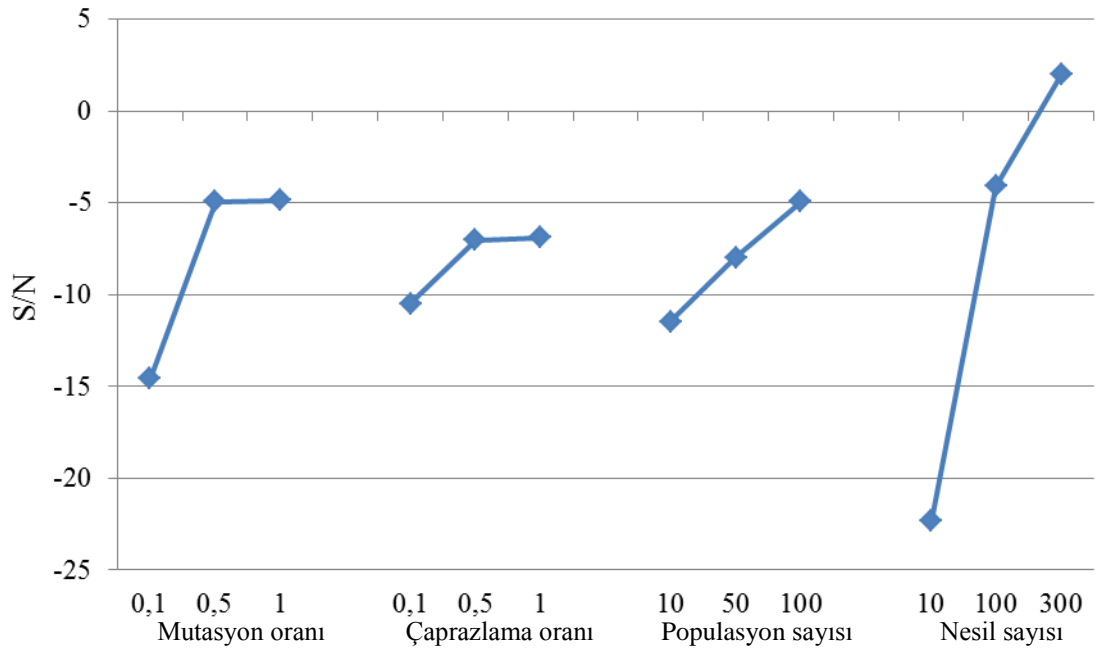
Çizelge 4.4. 100 düğümlük ağda GA parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Çaprazlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-14,25	-15,66	-17,58	-24,18
2. seviye	-14,13	-13,28	-12,77	-11,84
3. seviye	-12,15	-11,60	-10,17	-4,51
Optimum Seviye	3	3	3	3

**Şekil 4.2.** 100 düğümlük ağda GA parametrelerinin ortalama S/N etkileri

Çizelge 4.5. 250 düğümlük ağda GA parametreleri için optimum seviyeler

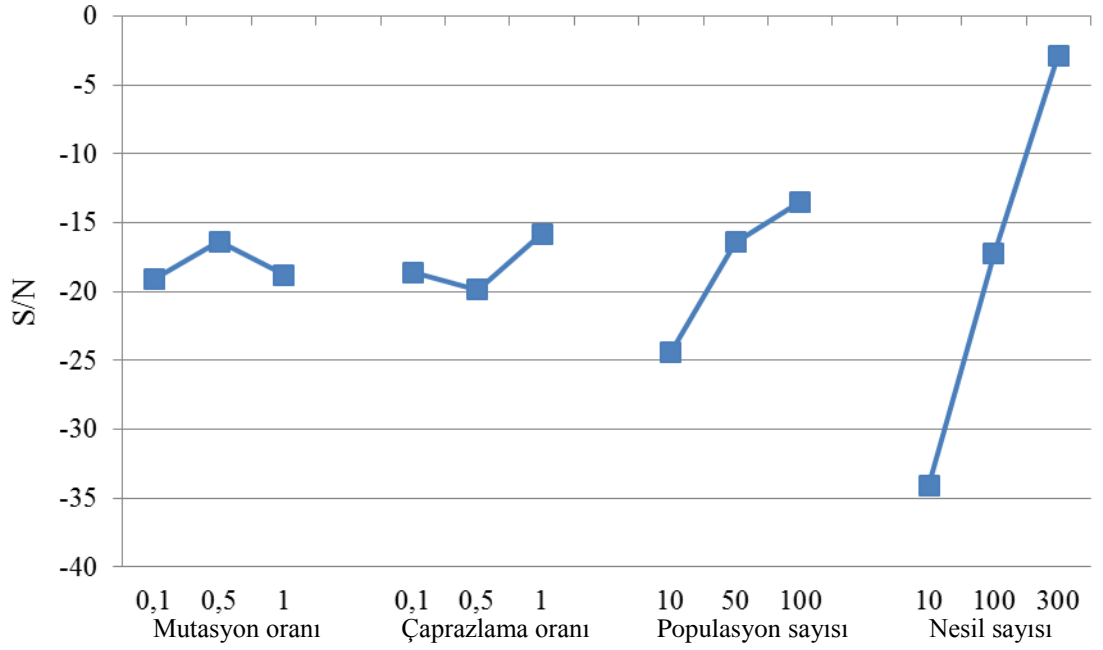
Seviye No	Mutasyon Oranı	Çaprazlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-14,57	-10,47	-11,47	-22,30
2. seviye	-4,95	-7,04	-7,97	-4,10
3. seviye	-4,87	-6,88	-4,95	2,01
Optimum Seviye	3	3	3	3



Şekil 4.3. 250 düğümlük ağda GA parametrelerinin ortalama S/N etkileri

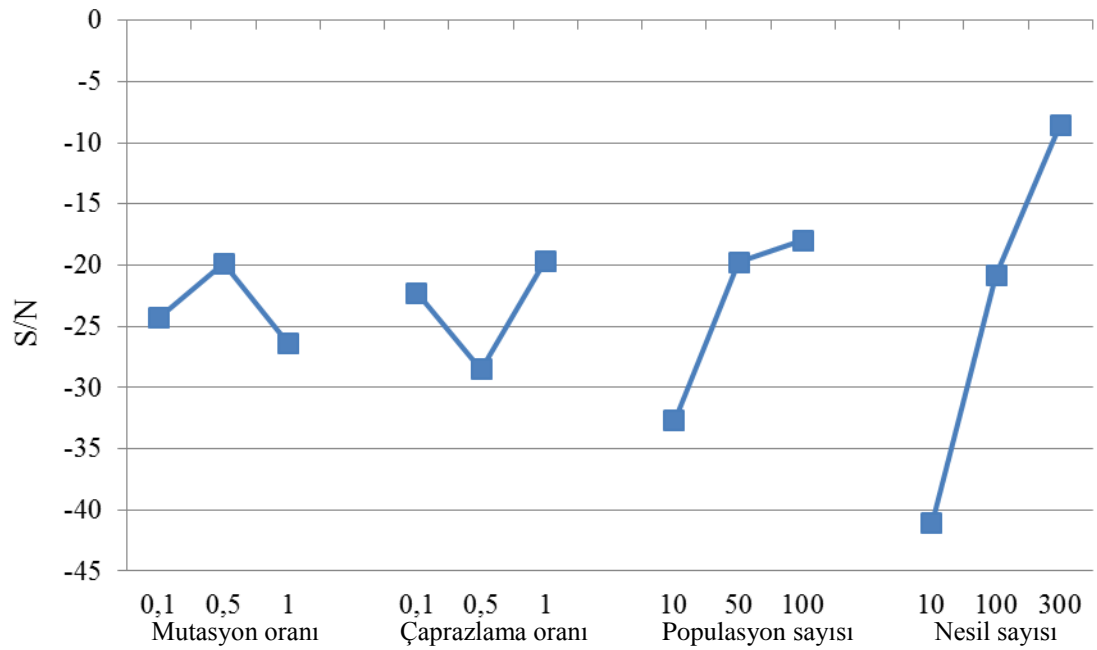
Çizelge 4.6. 500 düğümlük ağda GA parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Çaprazlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-19,08	-18,59	-24,36	-34,09
2. seviye	-16,37	-19,84	-16,40	-17,27
3. seviye	-18,78	-15,80	-13,47	-2,87
Optimum Seviye	2	3	3	3

**Şekil 4.4.** 500 düğümlük ağda GA parametrelerinin ortalama S/N etkileri

Çizelge 4.7. 1000 düğümlük ağda GA parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Çaprazlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-24,26	-22,29	-32,74	-41,08
2. seviye	-19,87	-28,50	-19,76	-20,84
3. seviye	-26,35	-19,69	-17,98	-8,56
Optimum Seviye	2	3	3	3

**Şekil 4.5.** 1000 düğümlük ağda GA parametrelerinin ortalama S/N etkileri**4.1.2. YBS parametreleri için Taguchi optimizasyonu**

Üzerinde çalışılacak her bir ağ için belirlenen YBS parametreleri kullanılarak elde edilen EKY değerleri ile S/N oranı hesap edilerek Çizelge 4.8 ve 4.9’da verilmiştir.

Çizelge 4.8. YBS için deterministik EKY değerleri

Sıra No	YBS ile elde edilen EKY uzunlukları (km)									
	50 düğüm		100 düğüm		250 düğüm		500 düğüm		1000 düğüm	
1	85	82	267	243	591	590	1155	1152	2859	2872
2	76	72	214	212	541	544	1087	1084	2794	2767
3	71	71	210	209	543	541	1059	1058	2740	2755
4	71	71	208	209	541	542	1061	1056	2741	2744
5	76	76	217	209	543	541	1071	1069	2772	2758
6	78	76	213	217	546	551	1077	1092	2782	2772
7	71	71	207	207	541	542	1054	1054	2740	2739
8	71	76	210	209	543	541	1059	1054	2735	2743
9	78	76	213	218	545	542	1075	1077	2798	2794

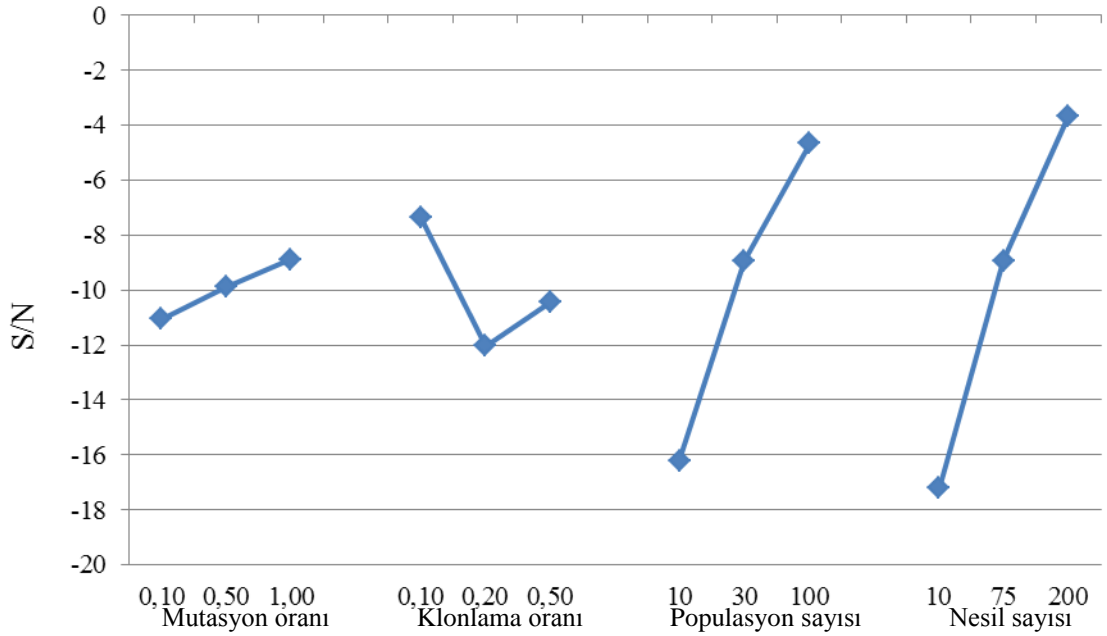
Çizelge 4.9. YBS için S/N değerleri

Sıra No	YBS için S/N değerleri				
	50 düğüm	100 düğüm	250 düğüm	500 düğüm	1000 düğüm
1	-22,00	-34,06	-33,89	-39,96	-42,32
2	-11,14	-16,99	-6,53	-29,98	-33,53
3	0,00	-10,97	-3,01	-13,12	-23,27
4	0,00	-8,13	3,01	-14,23	-17,67
5	-13,98	-18,13	-3,01	-24,10	-29,77
6	-15,68	-19,29	-17,96	-29,94	-32,53
7	0,00	0,00	3,01	0,00	-13,12
8	-10,97	-10,97	-3,01	-10,97	-15,05
9	-15,68	-19,85	-9,29	-26,86	-35,71
Ortalama	-9,94	-15,38	-7,85	-21,02	-27,00

Her bir ağ türü için ortalama S/N değerleri bulunup maksimum olanları tespit edilerek optimum parametre seviyeleri Çizelge 4.10-4.14'te belirlenmiş ve Şekil 4.6-4.10'da gösterilmiştir.

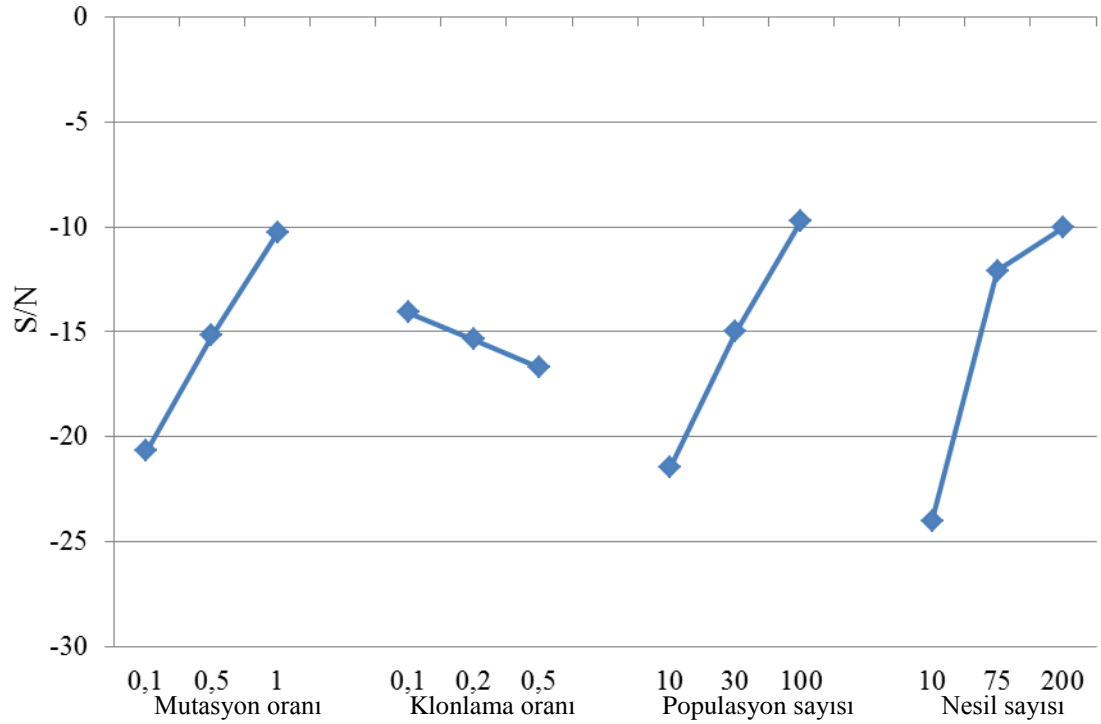
Çizelge 4.10. 50 düğümlük ağda YBS parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Klonlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-11,05	-7,33	-16,22	-17,22
2. seviye	-9,89	-12,03	-8,94	-8,94
3. seviye	-8,88	-10,45	-4,66	-3,66
Optimum Seviye	3	1	3	3

**Şekil 4.6.** 50 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri

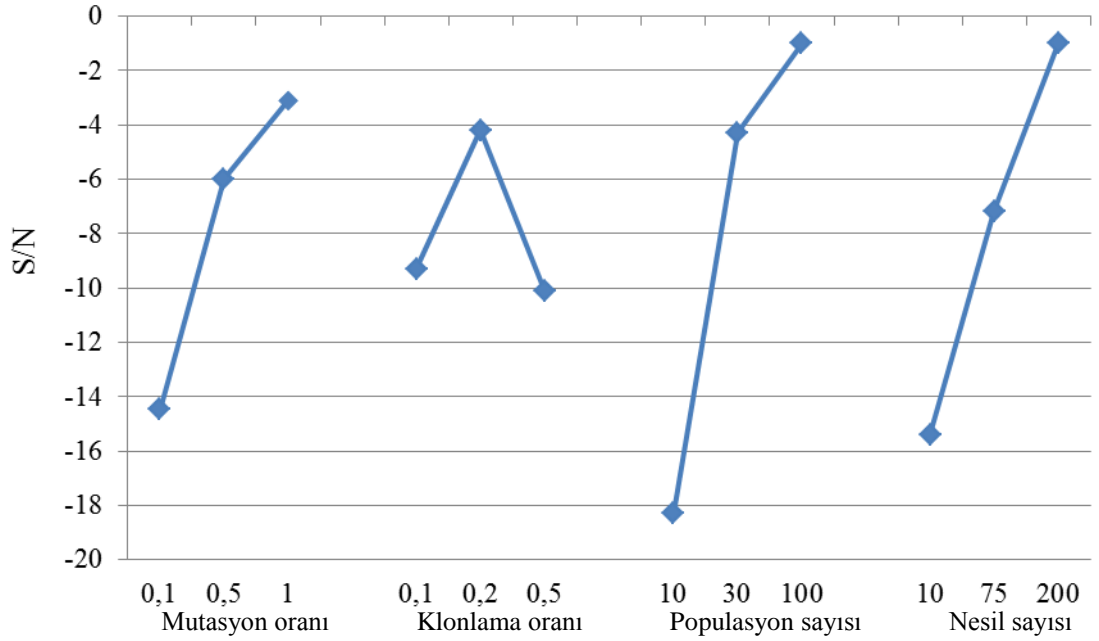
Çizelge 4.11. 100 düğümlük ağda YBS parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Klonlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-20,67	-14,06	-21,44	-24,01
2. seviye	-15,18	-15,36	-14,99	-12,09
3. seviye	-10,27	-16,70	-9,70	-10,02
Optimum Seviye	3	1	3	3

**Şekil 4.7.** 100 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri

Çizelge 4.12. 250 düğümlük ağda YBS parametreleri için optimum seviyeler

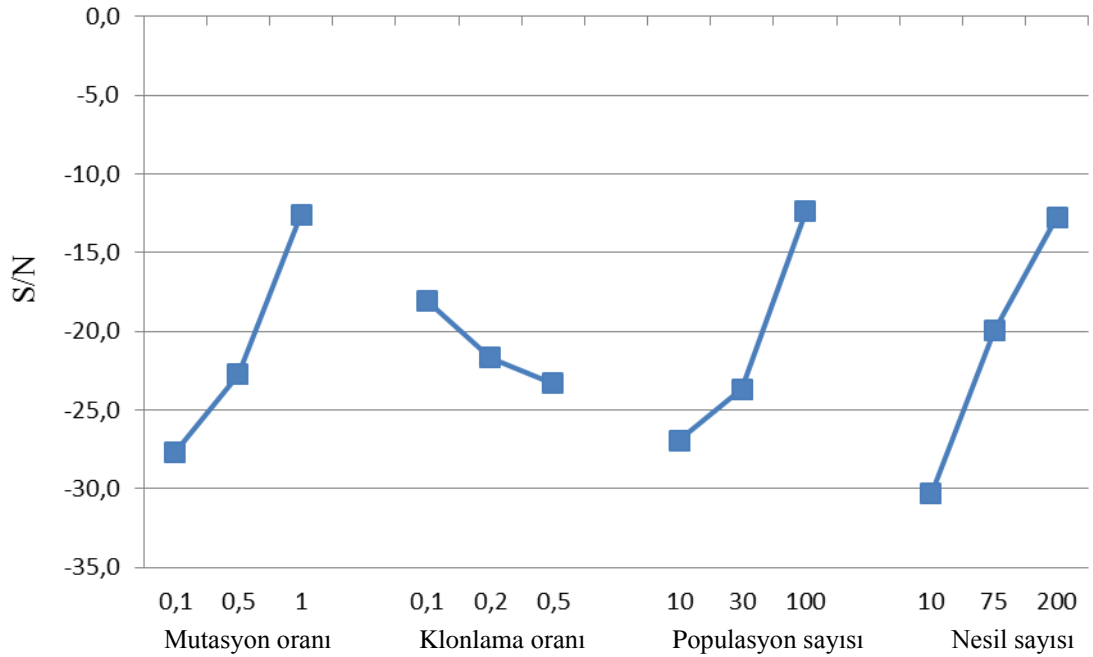
Seviye No.	Mutasyon Oranı	Klonlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-14,48	-9,29	-18,29	-15,40
2. seviye	-5,99	-4,18	-4,27	-7,16
3. seviye	-3,10	-10,09	-1,00	-1,00
Optimum Seviye	3	2	3	3



Şekil 4.8. 250 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri

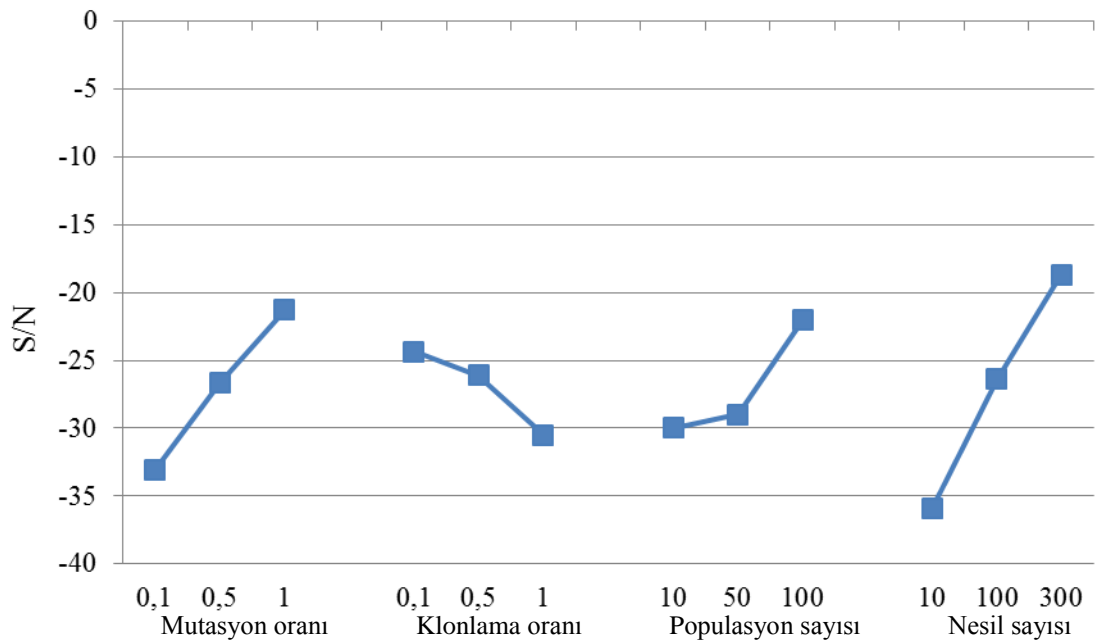
Çizelge 4.13. 500 düğümlük ağda YBS parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Klonlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-27,68	-18,06	-26,96	-30,30
2. seviye	-22,76	-21,68	-23,69	-19,97
3. seviye	-12,61	-23,31	-12,41	-12,77
Optimum Seviye	3	1	3	3

**Şekil 4.9.** 500 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri

Çizelge 4.14. 1000 düğümlük ağda YBS parametreleri için optimum seviyeler

Seviye No	Mutasyon Oranı	Klonlama Oranı	Populasyon Sayısı	Nesil Sayısı
1. seviye	-33,04	-24,37	-29,97	-35,94
2. seviye	-26,66	-26,12	-28,97	-26,39
3. seviye	-21,29	-30,50	-22,05	-18,67
Optimum Seviye	3	1	3	3

**Şekil 4.10.** 1000 düğümlük ağda YBS parametrelerinin ortalama S/N etkileri

4.1.3. Taguchi optimizasyonu doğrulama deneyleri

Yapılan optimizasyon deneylerinde GA için mutasyon oranı, çaprazlama oranı, populasyon sayısı ve nesil sayısı parametreleri 50, 100, 250, 500 ve 1000 düğümlü ağlar için sırasıyla 3-2-3-3, 3-3-3-3, 3-3-3-3, 2-3-3-3 ve 2-3-3-3 optimum seviyeleri belirlenmiştir. YBS'nin mutasyon oranı, klonlama oranı, populasyon sayısı ve nesil

sayısı parametreleri için ağ büyüklüğüne göre sırasıyla 3-1-3-3, 3-1-3-3, 3-2-3-3, 3-1-3-3 ve 3-1-3-3 optimum seviyeleri tespit edilmiştir.

Optimum seviyelere karşılık gelen EKY değerleri L_9 ortogonal dizisinde belirtilen çalışma planına göre elde edilen değerlerin içinde bulunmayabilir (Çizelge 4.1 ve 4.8). Bu durumda parametre değerlerinin EKY'ü hesaplamadaki performansını görmek için doğrulama deneyleri yapılarak %95 güven seviyesinde kontrolü gerçekleştirilmiştir.

Çizelge 4.15. GA için doğrulama deneyleri

Ağ Türleri	GA		
	Doğrulama Sonucu		Güven Aralığı
	EKY 1	EKY2	
50 düğüm	71	71	71±0
100 düğüm	206	206	206±0
250 düğüm	541	541	541±0
500 düğüm	1054	1054	1054±0
1000 düğüm	2735	2737	2735,93±0,93

Çizelge 4.16. YBS için doğrulama deneyleri

Ağ Türleri	YBS		
	Doğrulama Sonucu		Güven Aralığı
	EKY 1	EKY2	
50 düğüm	71	71	71±0
100 düğüm	206	206	206±0
250 düğüm	541	541	541±0,19
500 düğüm	1054	1054	1054±0
1000 düğüm	2735	2735	2734,57±1,43

Sonraki aşamalarda EKY hesap edilirken GA ve YBS algoritmalarında belirlenen optimum parametre seviyeleri kullanılmıştır.

4.2. EKY Değerlerinin Elde Edilmesi

4.2.1. 50 Döğümlü ađ

4.2.1.a. 50 döğümlü ađ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi

Kurulan model öncelikle deterministik kenar bilgisine sahip bir ađ üzerinde çalıştırılmış, elde edilen sonuçlar, klasik EKY algoritmalarından olan Dijkstra ve Floyd algoritması ile karşılaştırılmıştır (Çizelge 4.17). Modelin etkili sonuçlar verdiği bu karşılaştırmalar sonucunda görüldükten sonra stokastik ađ bilgisine sahip yollarda EKY bilgisi elde edilmiş, Çizelge 4.18’de gösterilmiştir.

Çizelge 4.17. 50 döğümlü ađda deterministik EKY değerleri

Problem	Algoritmalar				Rota İndeks
	GA	YBS	Dijkstra	Floyd	
En kısa mesafeli yol (km)	71	71	71	71	Rota 1
Ort. en kısa seyahat süreli yol (dk)	87,13	87,13	87,13	87,13	Rota 2
Ort. en az yakıt tüketilen yol (l)	4,63	4,63	4,63	4,63	Rota 1

Çizelge 4.18. 50 döğümlü ađda stokastik EKY değerleri

Problem	Algoritmalar		Rota İndeks
	GA	YBS	
90 dk. altında gitme ihtimali en yüksek yol (%)	98	98	Rota 2
4,65 l’in altında yakıt tüketme ihtimali en yüksek yol (%)	99	99	Rota 1
%95 ihtimalle en az seyahat süreli yol (dk)	89,53	89,53	Rota 2

%95 ihtimalle en az yakıt tüketilen yol (l)	4,65	4,65	Rota 1
---	------	------	--------

Deterministik ve stokastik EKY'leri sağlayan rotalar Çizelge 4.19'da gösterilmiştir.

Çizelge 4.19. 50 düğümlü ağda EKY rotaları

Rota indeks	Düğüm numaraları								
Rota 1	1	2	8	14	22	28	36	44	50
Rota 2	1	7	13	20	26	32	39	46	50

50 düğümlü ağda yapılan çalışmada en az maliyetli rotalar için kenarlardaki maliyet değerleri Çizelge 4.20 ve 4.21'de gösterilmiştir.

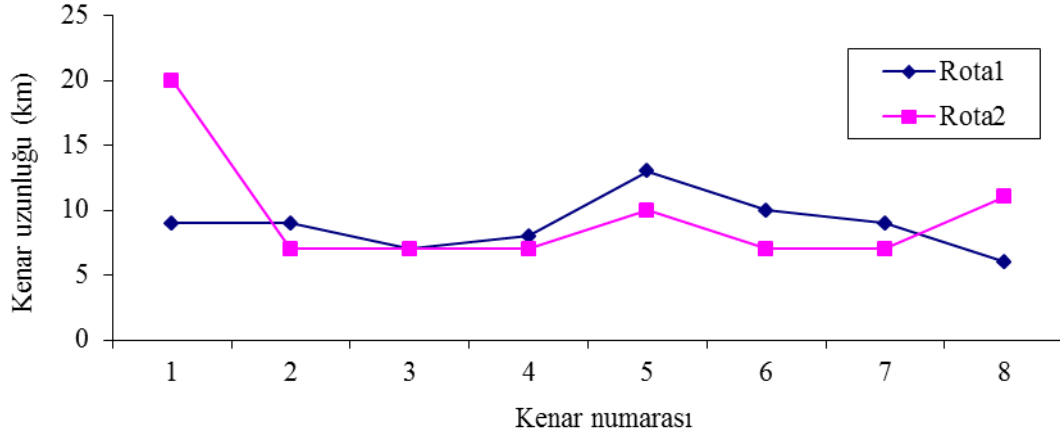
Çizelge 4.20. 50 düğümlü ağda Rota1 için kenar maliyet değerleri

Rota 1	1	2	8	14	22	28	36	44	50	Toplam Yol
Kenar mesafe (km)	9	9	7	8	13	10	9	6	71	
Kenar zaman (dk)	21,38	11,54	14,01	12,07	18,12	15,04	14,95	10,18	117,30	
Kenar yakıt (lt)	0,62	0,58	0,47	0,52	0,84	0,64	0,59	0,39	4,63	
Kenar hız (km/sa)	25,25	46,79	29,97	39,76	43,05	39,89	36,12	35,38	ort 36,32	

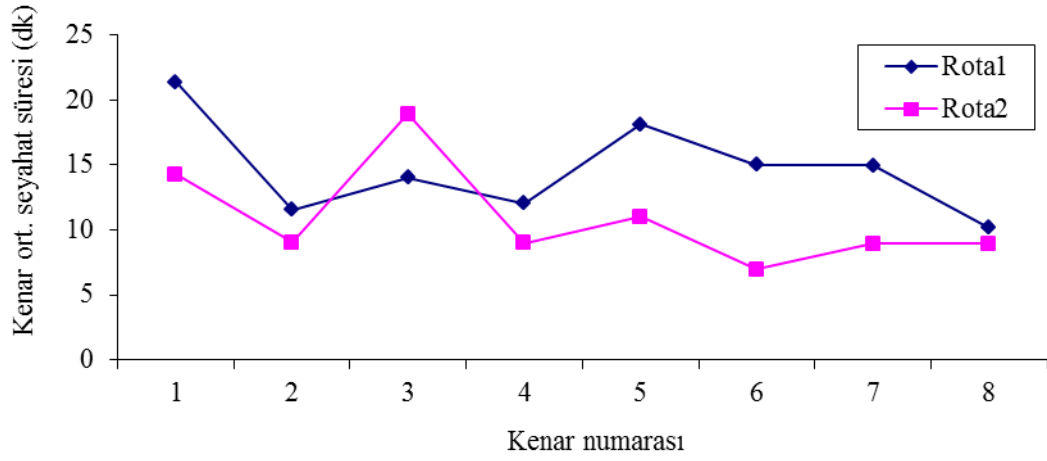
Çizelge 4.21. 50 düğümlü ağda Rota2 için kenar maliyet değerleri

Rota 2	1	7	13	20	26	32	39	46	50	Toplam Yol
Kenar mesafe (km)	20	7	7	7	10	7	7	11	76	
Kenar zaman (dk)	14,3	9,06	18,93	9,01	11	6,97	8,94	8,93	87,13	
Kenar yakıt (lt)	1,77	0,45	0,49	0,45	0,66	0,49	0,45	0,85	5,61	
Kenar hız (km/sa)	83,94	46,37	22,18	46,64	54,57	60,22	46,99	73,89	ort 52,34	

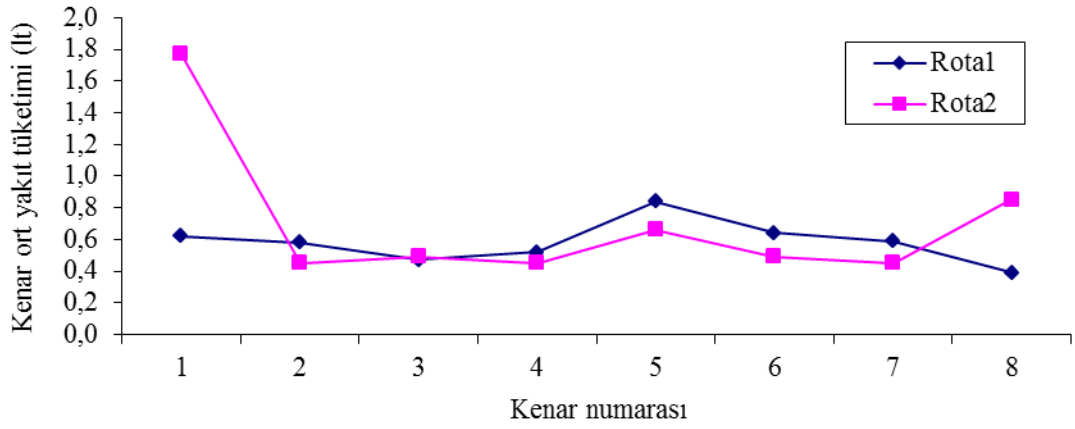
En az maliyetli rotaların mesafe, zaman, yakıt ve hız bakımından karşılaştırılması sırasıyla Şekil 4.11-4.14’de gösterilmiştir.



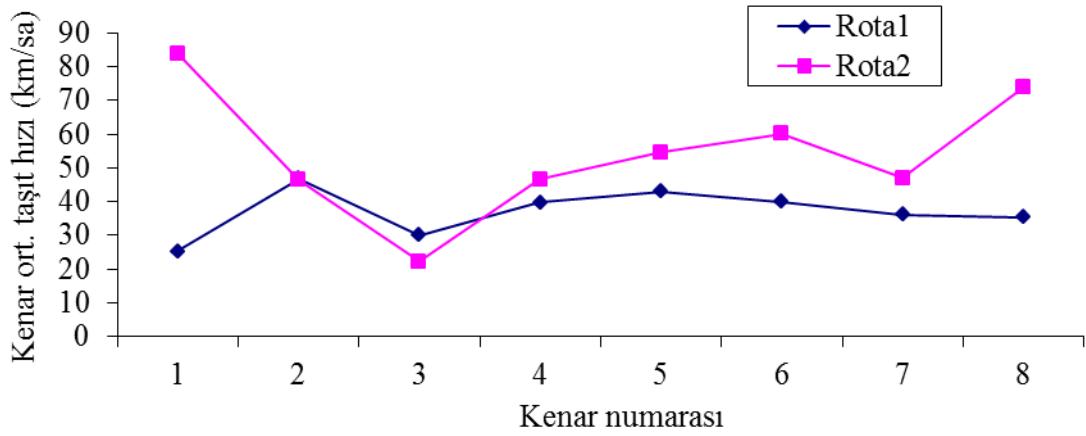
Şekil 4.11. 50 düğümlü ağda rotaların kenar uzunluklarının karşılaştırması



Şekil 4.12. 50 düğümlü ağda rotaların kenar seyahat sürelerinin karşılaştırılması

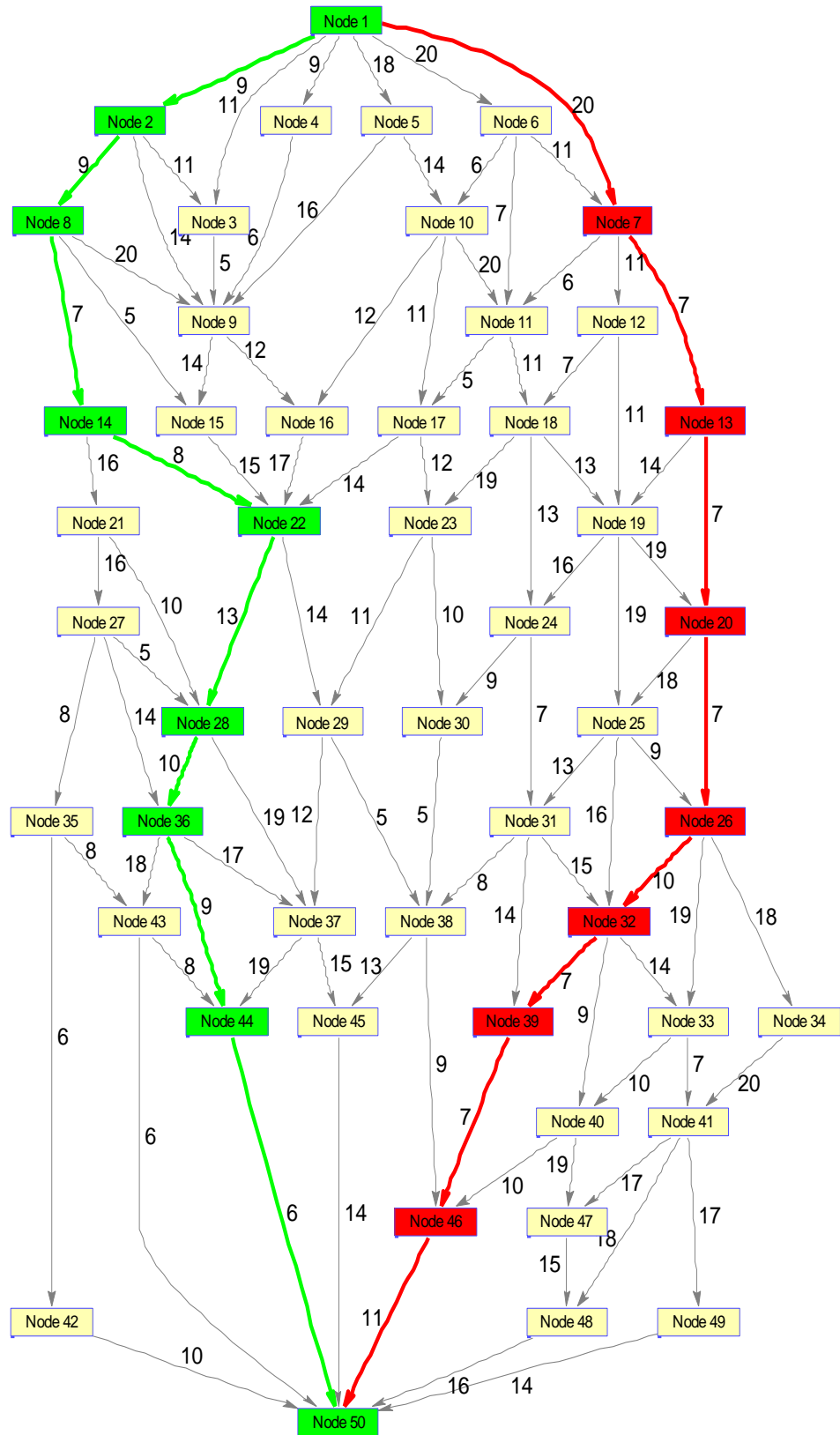


Şekil 4.13. 50 düğümlü ağda kenarlardaki yakıt tüketim miktarlarının karşılaştırılması



Şekil 4.14. 50 düğümlü ağda kenarlardaki ortalama taşıt hızlarının karşılaştırılması

50 düğümlü ağ için EKY'yi sağlayan rotalar Şekil 4.15'de gösterilmiştir.



Şekil 4.15. 50 düğümlü graf üzerinde rotaların gösterilmesi

4.2.1.b. 50 düğümlü ağda Pareto çözümlerinin elde edilmesi

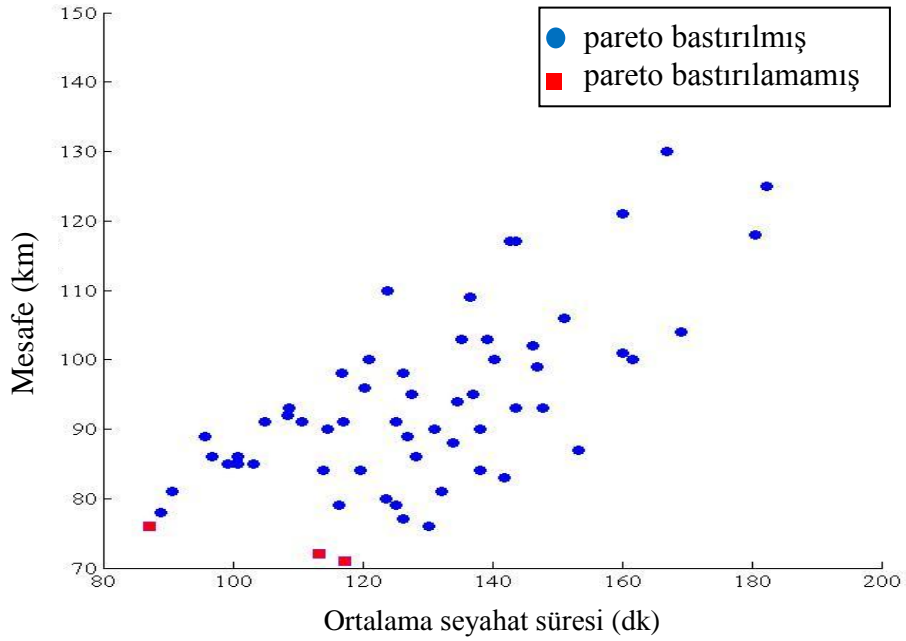
50 düğümlü ağda 3 farklı problem türü üzerinde çalışıldı. Problemlerde EKY için Pareto çözümleri hesap edilmiş, daha sonra elde edilen alternatif çözümlerin maliyet değerleri hesaplanmıştır.

Problem 1 için pareto çözümleri;

Problem 1 için, 50 düğümlü ağda elde edilen pareto çözümleri Çizelge 4.22'de gösterilmiştir.

Çizelge 4.22. 50 düğümlü ağda problem 1 için Pareto çözümleri

	Rota No.	Pareto uygunluğu	Min. zaman (dk)	Min. mesafe (km)	Rota
GA	1	1	113,33	72	1 2 8 14 22 29 38 46 50
	2	1	87,13	76	1 7 13 20 26 32 39 46 50
	3	1	117,30	71	1 2 8 14 22 28 36 44 50
YBS	1	1	113,33	72	1 2 8 14 22 29 38 46 50
	2	1	87,13	76	1 7 13 20 26 32 39 46 50
	3	1	117,30	71	1 2 8 14 22 28 36 44 50



Şekil 4.16. 50 düğümlü ağda problem 1 için Pareto çözümleri grafiği

Kullanıcılara kendi önceliklerine göre tercih edeceği çözümler sunmakla beraber, yol göstermesi bakımından her iki hedefin maliyetleri hesaplanarak elde edilen pareto çözümleri tek bir çözüme indirgenerek çözümlerden minimum maliyete sahip olan rota da kullanıcıya sunulmuştur. Otomobil kullanıcıları için zaman maliyeti=19,35 TL/saat ve taşıt işletme maliyeti=0,36 TL/km olarak tespit edilmiştir (Tulum ve Terzi 2010, VTPİ 2010). Bu birim maliyet değerleri kullanılarak Çizelge 4.22'deki değerlerin toplam maliyeti hesaplanmış elde edilen değerler Çizelge 4.23'de gösterilmiştir.

Çizelge 4.23. 50 düğümlü ağda problem 1 in Pareto çözümlerinin maliyet hesabı

	Rota No	Zaman Maliyeti (TL)	Taşıt İşletme Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	1	36,55	25,92	62,47
	2	28,10	27,36	55,46
	3	37,83	25,56	63,39

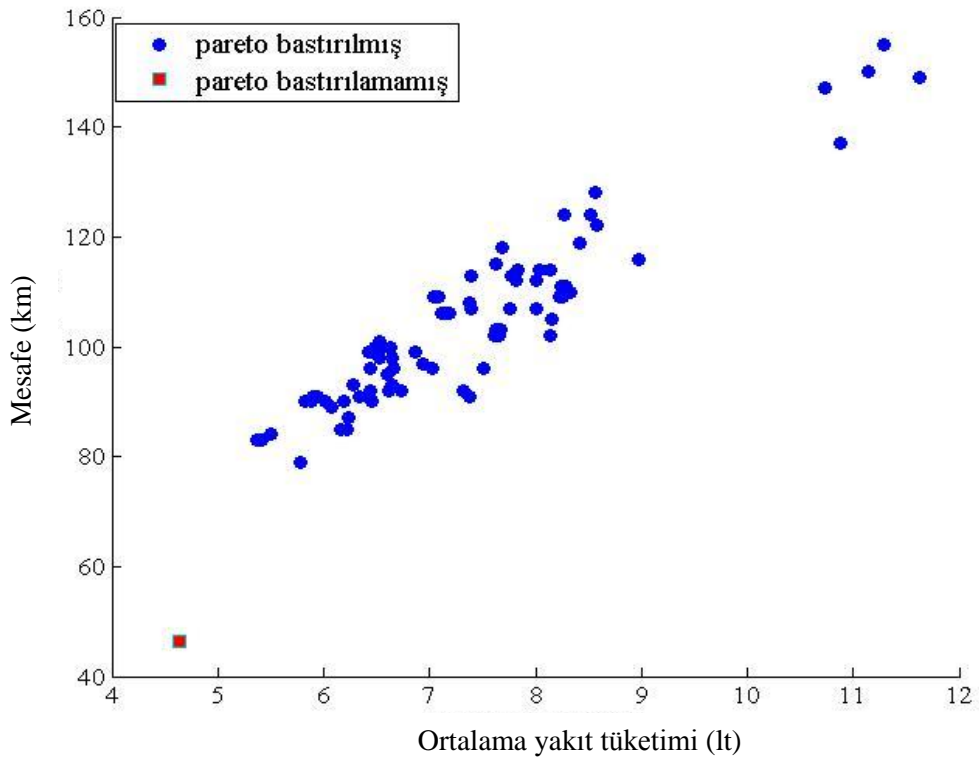
Pareto çözümleri için, kullanıcı kendisine sunulan farklı rotalardan kendi önceliğine göre seçim yapabileceği gibi, yapılan maliyet hesabından, kullanıcıya tek bir rota tavsiye edilebilir.

Çizelge 4. 23'ten görüleceği gibi 2 numaralı rota en az maliyete sahip olduğundan EKY olarak kullanıcıya 1 numaralı düğümden 50 numaralı düğüme gitmesi için izleyeceği rota olarak 1-7-13-20-26-32-39-46-50 rotası tavsiye edilmektedir.

Problem 2 için pareto çözümleri;

Çizelge 4.24. 50 düğümlü ağda problem 2 için Pareto çözümleri

	Pareto uygunluğu	min yakıt (lt)	min mesafe (km)	Rota								
GA	1	4,63	71	1	2	8	14	22	28	36	44	50
YBS	1	4,63	71	1	2	8	14	22	28	36	44	50



Şekil 4.17. 50 düğümlü ağda problem 2 için Pareto çözümleri grafiği

Yakıt maliyeti=3,6 TL /lt olarak alınması durumunda Çizelge 4.23'deki değerlerin toplam maliyeti hesaplanmış elde edilen değerler Çizelge 4.25'de gösterilmiştir.

Çizelge 4.25. 50 düğümlü ağda problem 2'nin Pareto çözümlerinin maliyet hesabı

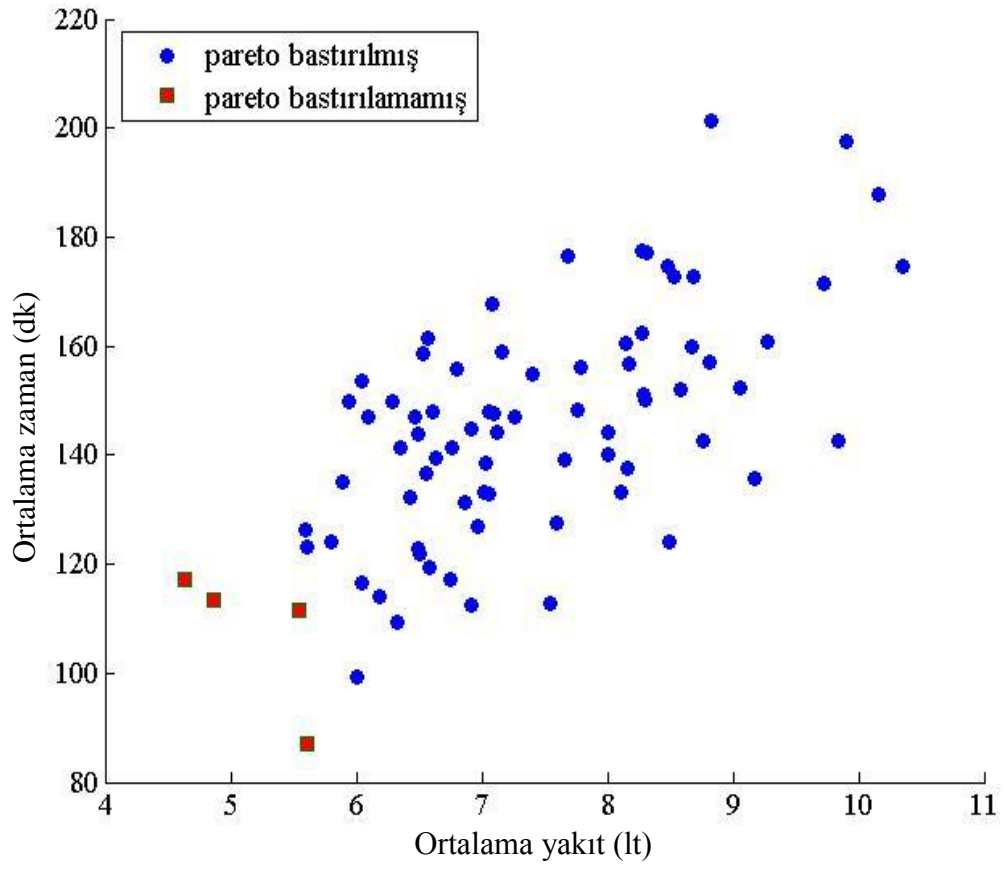
	Rota No	Yakıt Maliyeti (TL)	Taşıt İşletme Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	1	16,67	25,56	42,23

Çizelge 25'den görüleceği gibi pareto çözüm sayısı tek olduğundan EKY olarak kullanıcıya 1 numaralı düğümden 50 numaralı düğüme gitmesi için izleyeceği rota olarak 1-2-8-14-22-28-36-44-50 rotası tavsiye edilmektedir.

Problem 3 için pareto çözümleri;

Çizelge 4.26. 50 düğümlü ağda problem 3 için Pareto çözümleri

	Pareto uygunluğu	Min. yakıt (lt)	Min. zaman (dk)	Rota
GA	1	5,55	111,66	1 4 9 16 22 29 38 46 50
	1	4,87	113,33	1 2 8 14 22 29 38 46 50
	1	5,61	87,13	1 7 13 20 26 32 39 46 50
	1	4,63	117,30	1 2 8 14 22 28 36 44 50
YBS	1	5,55	111,66	1 4 9 16 22 29 38 46 50
	1	4,87	113,33	1 2 8 14 22 29 38 46 50
	1	5,61	87,13	1 7 13 20 26 32 39 46 50
	1	4,63	117,30	1 2 8 14 22 28 36 44 50



Şekil 4.18. 50 düğümlü ağda problem 3 için Pareto çözümleri grafiği

Çizelge 4.26'daki değerlerin toplam maliyeti hesaplanmış elde edilen değerler Çizelge 4.27'de gösterilmiştir.

Çizelge 4.27. 50 düğümlü ağda problem3'ün Pareto çözümlerinin maliyet hesabı

	Rota No	Yakıt Maliyeti (TL)	Zaman Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	1	19,98	36,01	55,99
	2	17,53	36,55	54,08
	3	20,20	28,10	48,30
	4	16,67	37,83	54,50

Çizelge 4.27'den görüleceği üzere 3 numaralı rota en az maliyete sahip olduğundan EKY olarak kullanıcıya 1 numaralı düğümden 50 numaralı düğüme gitmesi için izleyeceği rota olarak 1-7-13-20-26-32-39-46-50 rotası tavsiye edilmektedir.

4.2.2. 100 Düğümlü ağlar

4.2.2.a. 100 Düğümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi

100 düğümlü ağ için kurulan model öncelikle deterministik kenar bilgisine sahip bir ağ üzerinde çalıştırılmış, elde edilen sonuçlar klasik EKY algoritmalarından olan Dijkstra ve Floyd algoritması ile karşılaştırılmıştır (Çizelge 4.28). Modelin etkili sonuçlar verdiği bu karşılaştırmalar sonucunda görüldükten sonra stokastik ağ bilgisine sahip yollarda EKY bilgisi elde edilmiş Çizelge 4.29'da gösterilmiştir.

Çizelge 4.28. 100 düğümlü ağda deterministik EKY değerleri

Problem	Algoritmalar				Rota İndeks
	GA	YBS	Dijkstra	Floyd	
En kısa mesafeli yol (km)	206	206	206	206	Rota 1
Ort. en kısa seyahat süreli yol (dk)	224,61	224,61	224,61	224,61	Rota 2
Ort. en az yakıt tüketilen yol (lt)	14,46	14,46	14,46	14,46	Rota 3

Çizelge 4.29. 100 düğümlü ağda stokastik EKY değerleri

Problem	Algoritmalar		Rota İndeks
	GA	YBS	
230 dk altında gitme ihtimali en yüksek yol (%)	92	92	Rota 2
14,5 lt'nin altında yakıt tüketme ihtimali en yüksek yol (%)	58	58	Rota 4
%90 ihtimalle en az seyahat süreli yol (dk)	228,87	228,87	Rota 5
%90 ihtimalle en az yakıt tüketilen yol (lt)	14,66	14,66	Rota 4

100 düğümlü ağda yapılan çalışmada EKY'ü sağlayan rotalar için kenar değerleri Çizelge 4.30-4.34'te gösterilmiştir.

Çizelge 4.30. 100 düğümlü ağda Rota 1 için kenar değerleri

	Rota 1	Kenar mesafe (km)	Kenar zaman (dk)	Kenar yakıt (lt)	Kenar hız (km/s)
Düğüm numarası	1	12	8,59	1,09	83,86
	10	8	12,09	0,52	39,71
	18	5	3,51	0,45	85,47
	25	16	10,34	1,6	92,88
	30	6	8,91	0,39	40,42
	33	20	28,59	1,29	41,97
	37	9	15,21	0,59	35,5
	41	6	4,93	0,47	72,96
	47	18	13,61	1,52	79,38
	51	13	31,19	0,89	25,01
	55	7	9,93	0,45	42,31
	59	6	5	0,46	71,97
	64	16	19,04	1,04	50,41
	66	9	7	0,73	77,15
	69	6	10,93	0,39	32,94
	73	8	9,11	0,53	52,7
	78	5	8	0,32	37,48
	82	14	24,65	0,91	34,08
	88	5	10,95	0,34	27,39
	94	8	8,49	0,54	56,57
99	9	14,96	0,59	36,1	
100					
Toplam		206	265,01	15,1	46,64

Çizelge 4.31. 100 düğümlü ağda Rota 2 için kenar değerleri

	Rota 2	Kenar mesafe (km)	Kenar zaman (dk)	Kenar yakıt (lt)	Kenar hız (km/s)
Düğüm numarası	1	12	8,59	1,09	83,86
	10	8	12,09	0,52	39,71
	18	5	3,51	0,45	85,47
	25	16	10,34	1,6	92,88
	30	9	7,38	0,7	73,2
	35	12	15,93	0,77	45,2
	39	13	12,8	0,9	60,94
	44	20	16,34	1,55	73,44
	49	6	6,07	0,41	59,34
	52	16	20,74	1,04	46,29
	55	7	9,93	0,45	42,31
	59	10	7,02	0,9	85,45
	63	18	15,91	1,32	67,87
	66	5	4,02	0,39	74,61
	68	12	12,9	0,8	55,81
	73	8	9,11	0,53	52,7
	78	5	8	0,32	37,48
	82	6	8,92	0,39	40,34
	87	18	13,09	1,61	82,5
	93	7	6,97	0,48	60,23
99	9	14,96	0,59	36,1	
100					
Toplam		222	224,61	16,81	59,3

Çizelge 4.32. 100 düğümlü ağda Rota 3 için kenar değerleri

	Rota 3	Kenar mesafe (km)	Kenar zaman (dk)	Kenar yakıt (lt)	Kenar hız (km/s)
Düğüm numarası	1	13	16,97	0,84	45,97
	2	6	8,05	0,39	44,74
	12	6	8,03	0,39	44,82
	23	17	20,03	1,11	50,92
	28	11	11,98	0,73	55,08
	31	7	10,03	0,45	41,88
	36	17	13,91	1,32	73,34
	40	11	14,04	0,71	47
	46	18	22,96	1,16	47,04
	50	6	8,92	0,39	40,35
	54	9	6,03	0,87	89,51
	57	10	15,01	0,64	39,99
	63	16	23,17	1,03	41,43
	65	6	7,9	0,39	45,59
	68	12	12,9	0,8	55,81
	73	8	9,11	0,53	52,7
	78	5	8	0,33	37,48
	82	14	24,65	0,91	34,08
	88	5	10,95	0,34	27,39
	94	8	8,49	0,55	56,57
99	9	14,96	0,59	36,1	
100					
Toplam		214	276,08	14,45	46,51

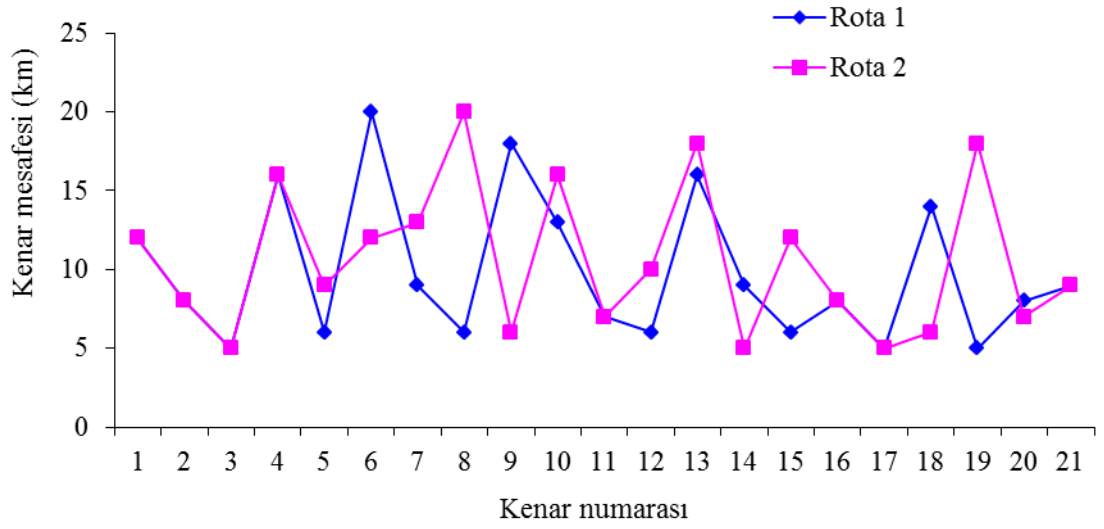
Çizelge 4.33. 100 düğümlü ağda Rota 4 için kenar değerleri

	Rota 4	Kenar mesafe (km)	Kenar zaman (dk)	Kenar yakıt (lt)	Kenar hız (km/s)
Düğüm numarası	1	8	10,06	0,52	47,74
	3	9	8,64	0,63	62,5
	13	13	18,35	0,84	42,5
	24	7	5,03	0,63	83,53
	29	6	5,96	0,41	60,43
	30	6	8,91	0,39	40,41
	33	20	28,59	1,29	41,97
	37	9	15,21	0,59	35,5
	41	6	4,93	0,47	72,96
	47	18	13,61	1,52	79,38
	51	13	31,19	0,89	25,01
	55	7	9,93	0,45	42,31
	59	6	5	0,46	71,97
	64	16	19,04	1,04	50,41
	66	9	7	0,73	77,15
	69	6	10,93	0,39	32,94
	73	8	9,11	0,53	52,7
	78	5	8	0,32	37,48
	82	14	24,65	0,91	34,08
	88	5	10,95	0,34	27,39
94	8	8,49	0,54	56,57	
99	9	14,96	0,59	36,1	
100					
Toplam		208	278,53	14,47	44,81

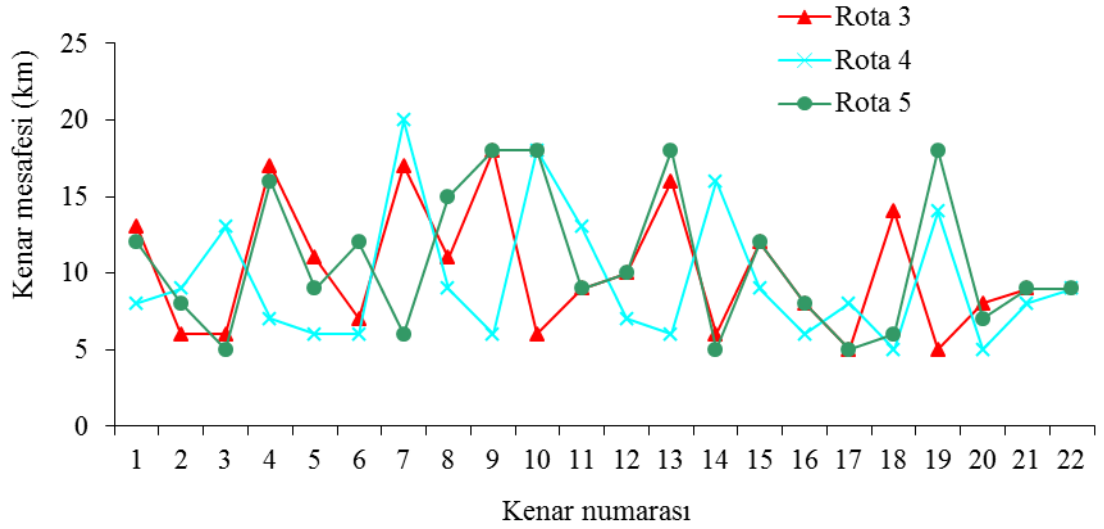
Çizelge 4.34. 100 düğümlü ağda Rota 5 için kenar değerleri

	Rota 5	Kenar mesafe (km)	Kenar zaman (dk)	Kenar yakıt (lt)	Kenar hız (km/s)
Düğüm numarası	1	12	8,59	1,09	83,86
	10	8	12,09	0,52	39,71
	18	5	3,51	0,45	85,47
	25	16	10,34	1,6	92,88
	30	9	7,38	0,7	73,2
	35	12	15,93	0,77	45,2
	39	6	11,89	0,4	30,28
	42	15	15,55	1,01	57,86
	47	18	13,61	1,52	79,38
	51	18	10,99	1,94	98,25
	54	9	6,03	0,87	89,51
	57	10	15,01	0,64	39,99
	63	18	15,91	1,32	67,87
	66	5	4,02	0,39	74,61
	68	12	12,9	0,8	55,81
	73	8	9,11	0,53	52,7
	78	5	8	0,32	37,48
	82	6	8,92	0,39	40,34
	87	18	13,09	1,61	82,5
	93	7	6,97	0,48	60,23
99	9	14,96	0,59	36,1	
100					
Toplam		235	239,75	18,53	58,81

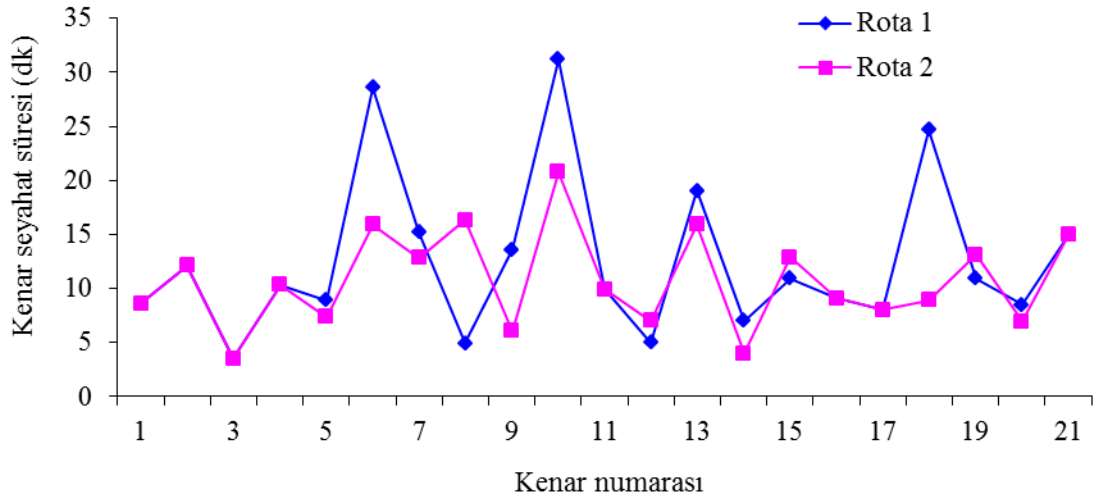
100 düğümlü ağda EKY’u sağlayan rotaların mesafe, zaman, yakıt ve hız bakımından karşılaştırılması sırasıyla Şekil 4.19-4.26’da gösterilmiştir.



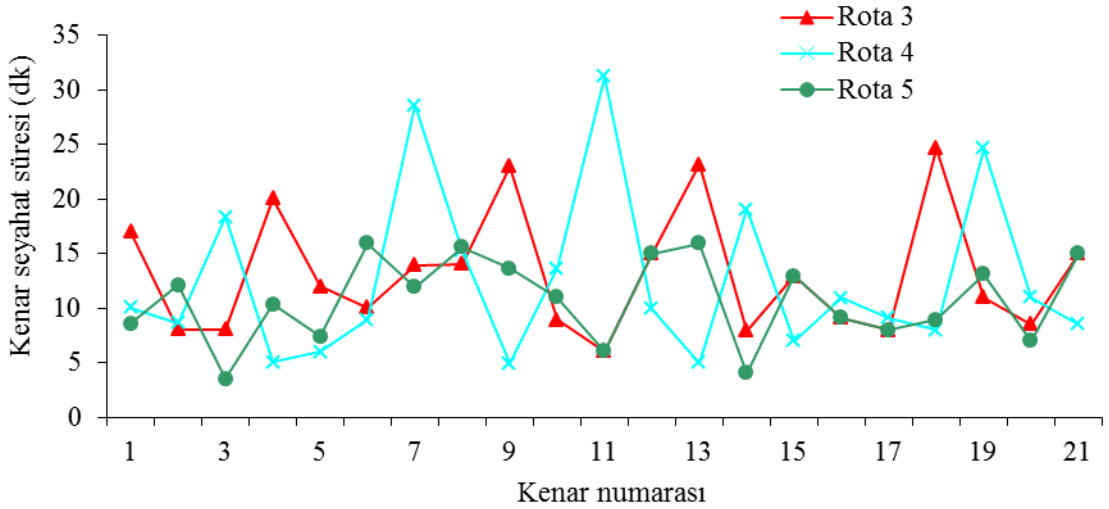
Şekil 4.19. 100 düğümlü ağda rota 1 ve rota 2'nin kenar uzunluklarının karşılaştırılması



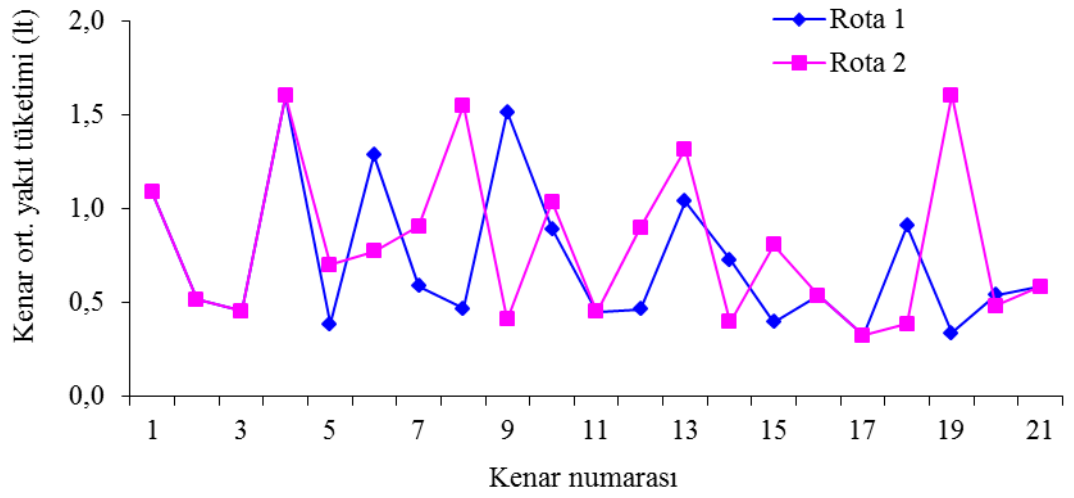
Şekil 4.20. 100 düğümlü ağda rota 3, rota 4 ve rota 5'in kenar uzunluklarının karşılaştırılması



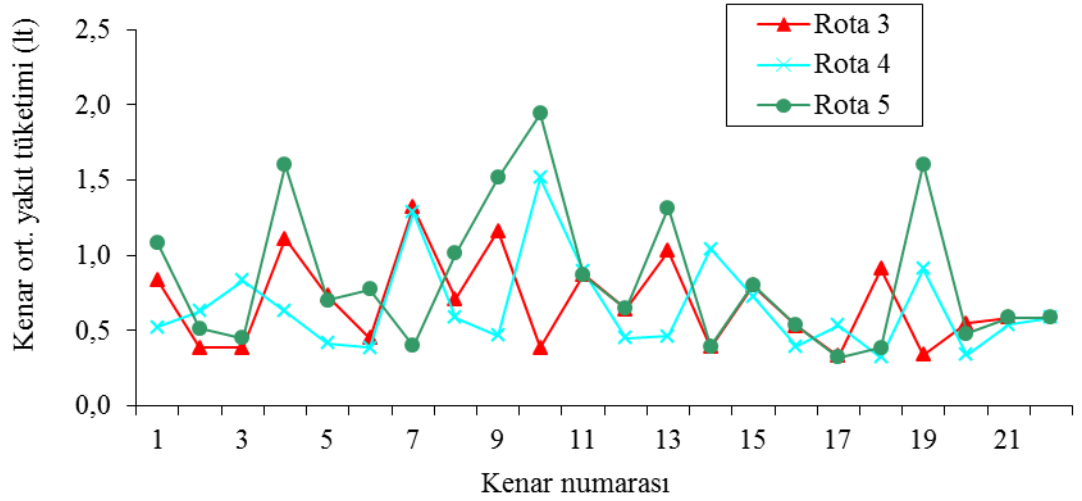
Şekil 4.21. 100 düğümlü ağda rota 1, ve rota 2'nin kenar seyahat sürelerinin karşılaştırılması



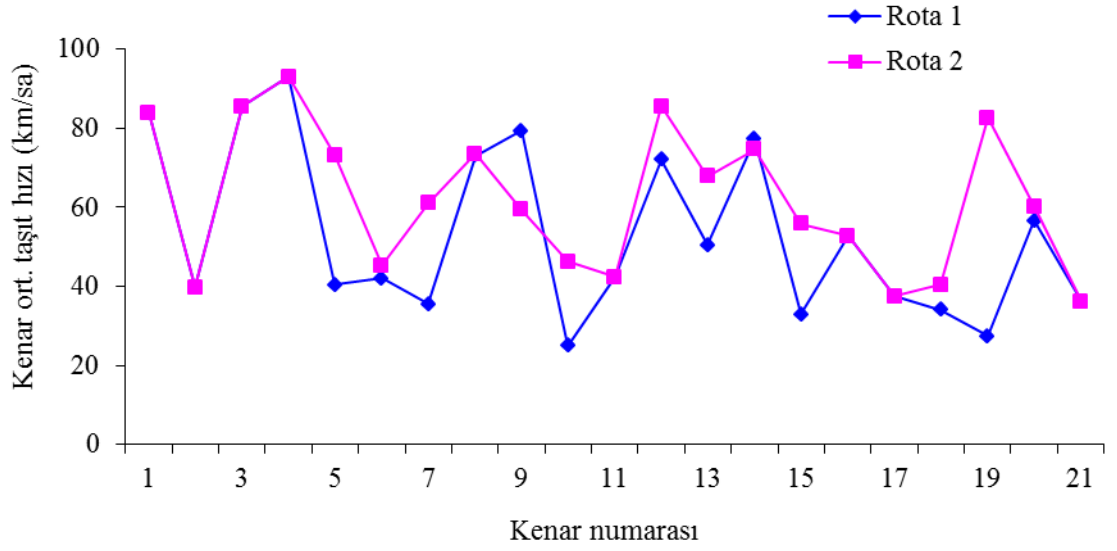
Şekil 4.22. 100 düğümlü ağda rota 3, rota 4, ve rota 5'in kenar seyahat sürelerinin karşılaştırılması



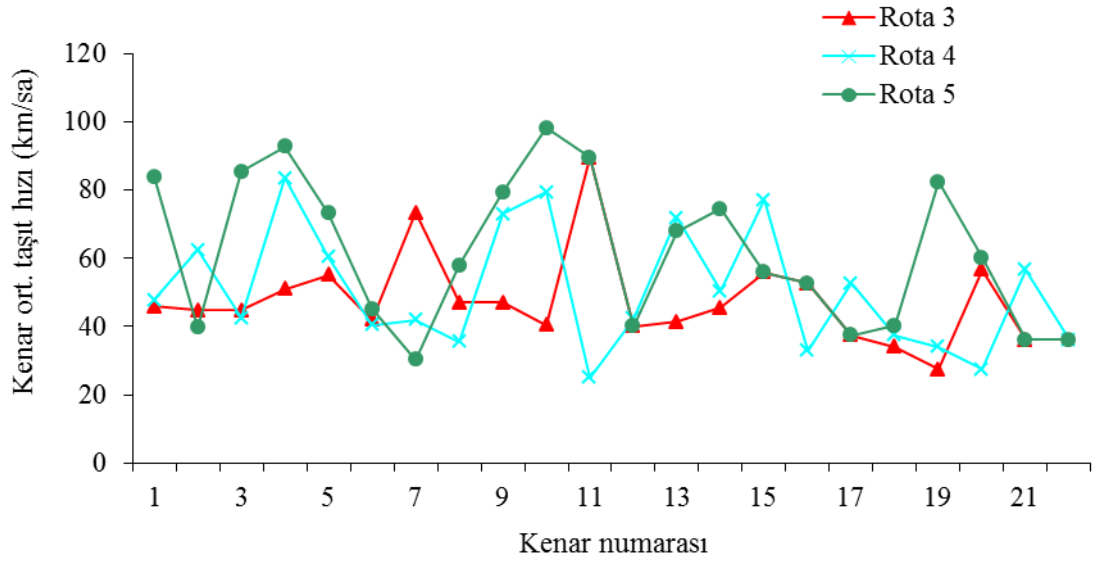
Şekil 4.23. 100 düğümlü ağda rota 1, ve rota 2'nin kenar yakıt tüketimlerinin karşılaştırılması



Şekil 4.24. 100 düğümlü ağda rota 3, rota 4, ve rota 5'in kenar yakıt tüketimlerinin karşılaştırılması



Şekil 4.25. 100 düğümlü ağda rota 1, ve rota 2'nin ortalama taşıt hızlarının karşılaştırılması



Şekil 4.26. 100 düğümlü ağda Rota 3, Rota 4, ve Rota 5'in ortalama taşıt hızlarının karşılaştırılması

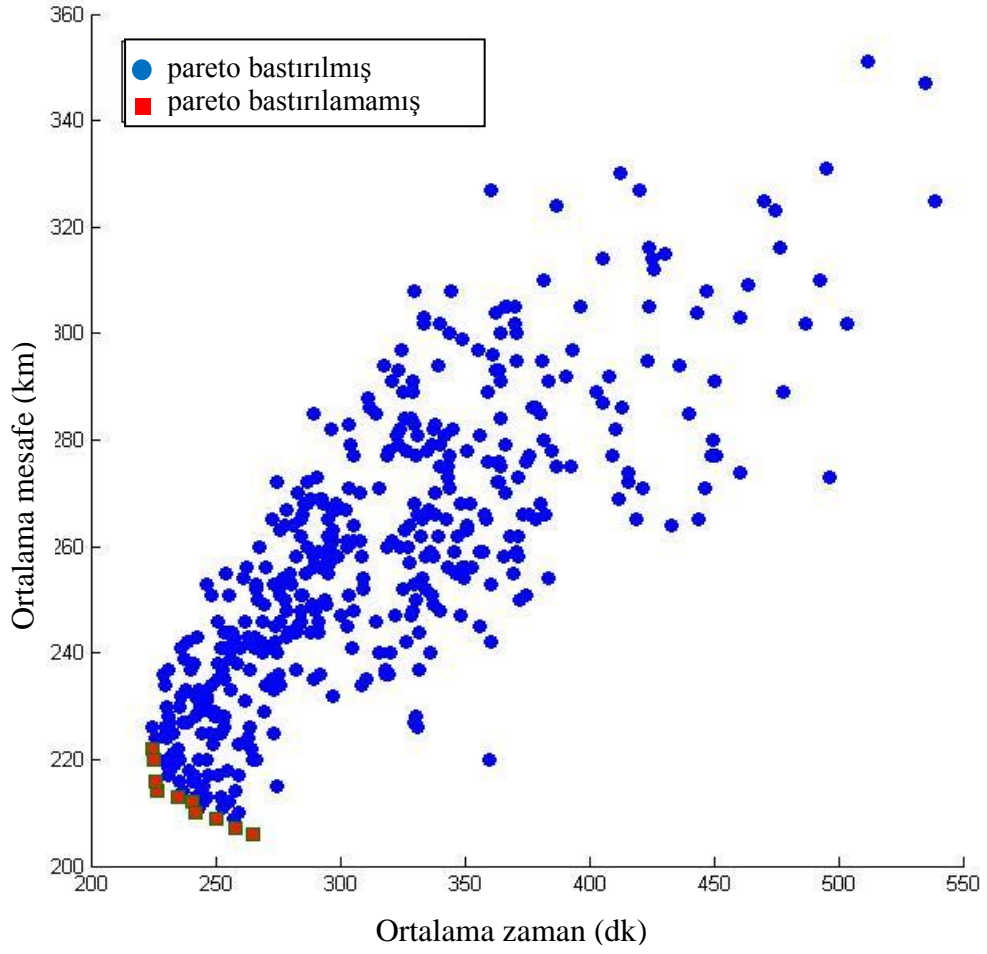
100 düğümlü ağ için EKY'yi sağlayan rotalar **EK 11** 'de gösterilmiştir.

4.2.2.b. 100 düğümlü ağda Pareto çözümlerinin elde edilmesi

Problem 1 için pareto çözümleri;

Çizelge 4.35. 100 düğümlü ağda problem 1 için Pareto çözümleri

	Pareto uygunluğu	min zaman (dk)	min mesafe (km)	Rota							
GA/YBS	1	265,01	206	Rota 1							
	2	224,61	222	Rota 2							
	3	240,82	212	Rota 3							
	4	226,73	214	Rota 4							
	5	225,72	216	Rota 5							
	6	241,82	210	Rota 6							
	7	225,62	220	Rota 7							
	8	235,20	213	Rota 8							
	9	258,12	207	Rota 9							
	10	250,30	209	Rota 10							
ROTALAR											
Rota 1	1	10	18	25	30	33	37	41	47	51	55
	59	64	66	69	73	78	82	88	94	99	100
Rota 2	1	10	18	25	30	35	39	44	49	52	55
	59	63	66	68	73	78	82	87	93	99	100
Rota 3	1	10	18	25	30	35	39	44	49	52	55
	59	64	66	68	73	78	82	88	94	99	100
Rota 4	1	10	18	25	30	35	39	44	49	52	55
	59	64	66	69	73	78	82	87	93	99	100
Rota 5	1	10	18	25	30	35	39	44	49	52	55
	59	64	66	68	73	78	82	87	93	99	100
Rota 6	1	10	18	25	30	35	39	44	49	52	55
	59	64	66	69	73	78	82	88	94	99	100
Rota 7	1	10	18	25	30	35	39	44	49	52	55
	59	63	66	69	73	78	82	87	93	99	100
Rota 8	1	10	18	25	30	33	36	40	46	50	54
	57	63	66	69	73	78	82	87	93	99	100
Rota 9	1	10	18	25	30	35	39	42	47	51	55
	59	64	66	69	73	78	82	88	94	99	100
Rota 10	1	10	18	25	30	33	36	40	46	50	54
	57	63	66	69	73	78	82	88	94	99	100



Şekil 4.27. 100 düğümlü ağda problem 1 için Pareto çözümleri grafiği

Çizelge 4.35'deki değerlerin toplam maliyeti hesaplanmış ve elde edilen sonuçlar Çizelge 4.36'da gösterilmiştir.

Çizelge 4.36. 100 düğümlü ağda problem 1 in Pareto çözümlerinin maliyet hesabı

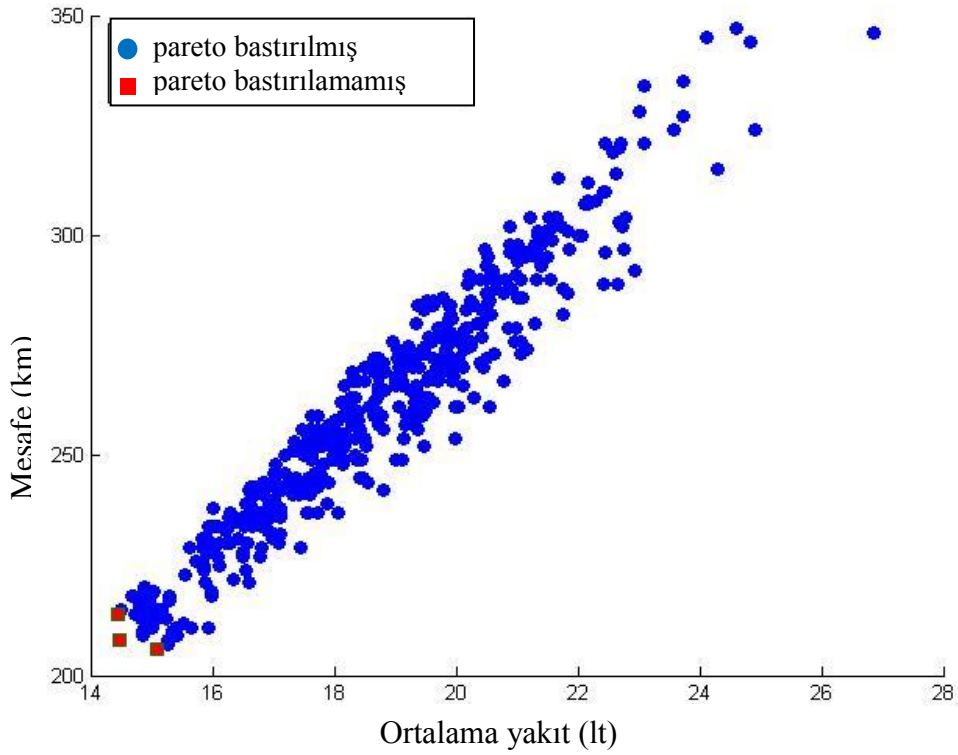
	Rota No	Zaman Maliyeti (TL)	Taşıt işletme Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	1	85,47	74,16	159,63
	2	72,44	79,92	152,36
	3	77,66	76,32	153,98
	4	73,12	77,04	150,16
	5	72,79	77,76	150,55
	6	77,99	75,60	153,59
	7	72,76	79,20	151,96
	8	75,85	76,68	152,53
	9	83,24	74,52	157,76
	10	80,72	75,24	155,96

Çizelge 4.36'dan görüleceği gibi en az maliyete sahip olan Rota 4, EKY olarak kullanıcıya tavsiye edilmektedir.

Problem 2 için pareto çözümleri;

Çizelge 4.37. 100 düğümlü ağda problem 2 için Pareto çözümleri

	Pareto uygunluğu	min yakıt (lt)	min mesafe (km)	Rota
GA/YBS	1	14,48	208	Rota 1
	2	15,10	206	Rota 2
	3	14,46	214	Rota 3
Rota 1	1 3 13 24 29 30 33 37 41 47 51 55 59 64 66 69 73 78 82 88 94 99 100			
Rota 2	1 10 18 25 30 33 37 41 47 51 55 59 64 66 69 73 78 82 88 94 99 100			
Rota 3	1 2 12 23 28 31 36 40 46 50 54 57 63 65 68 73 78 82 88 94 99 100			



Şekil 4.28. 100 düğümlü ağda problem 2 için Pareto çözümleri grafiği

Çizelge 4.38. 100 düğümlü ağda problem 2 nin Pareto çözümlerinin maliyet hesabı

	Rota No	Yakıt Maliyeti (TL)	Taşıt işletme Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	1	52,13	74,88	127,01
	2	54,36	74,16	128,52
	3	52,06	77,04	129,10

Çizelge 4.38'den görüleceği üzere en az maliyete sahip olan Rota 1 EKY olarak kullanıcıya tavsiye edilmektedir.

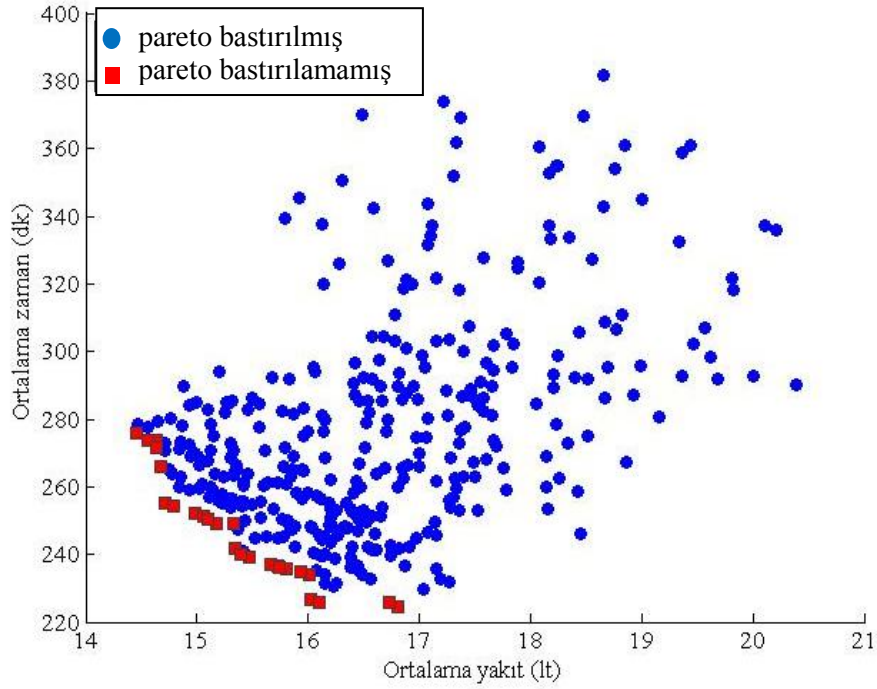
Problem 3 için pareto çözümleri;

Çizelge 4.39. 100 düğümlü ağda problem 3 için Pareto çözümleri

	Pareto uygunluğu	min yakıt (lt)	min zaman (dk)	Rota								
GA/YBS	1	14,45	276,08	Rota 1								
	1	14,56	273,94	Rota 2								
	1	14,63	273,88	Rota 3								
	1	14,63	271,64	Rota 4								
	1	14,67	265,96	Rota 5								
	1	14,72	255,34	Rota 6								
	1	14,79	254,33	Rota 7								
	1	14,98	252,36	Rota 8								
	1	15,06	251,36	Rota 9								
	1	15,10	250,30	Rota 10								
	1	15,18	249,30	Rota 11								
	1	15,33	249,15	Rota 12								
	1	15,34	241,82	Rota 13								
	1	15,39	240,24	Rota 14								
	1	15,47	239,23	Rota 15								
	1	15,66	237,26	Rota 16								
	1	15,73	236,79	Rota 17								
	1	15,74	236,26	Rota 18								
	1	15,80	235,78	Rota 19								
	1	15,93	235,05	Rota 20								
	1	16,08	234,05	Rota 21								
	1	16,02	226,73	Rota 22								
	1	16,10	225,72	Rota 23								
	1	16,73	225,62	Rota 24								
	1	16,81	224,61	Rota 25								
ROTALAR												
Rota 1	1 63	2 65	12 68	23 73	28 78	31 82	36 88	40 94	46 99	50 100	54	57
Rota 2	1 57	3 63	13 65	24 68	29 73	30 78	33 82	36 88	40 94	46 99	50 100	54
Rota 3	1 59	3 64	13 66	24 69	29 73	30 78	34 82	38 88	41 94	47 99	51 100	55
Rota 4	1 59	3 64	13 66	24 69	29 73	30 78	35 82	39 88	42 94	47 99	51 100	55
Rota 5	1 63	2 66	12 69	23 73	28 78	31 82	36 88	40 94	46 99	50 100	54	57

Çizelge 4.39. (devam)

ROTALAR												
Rota 6	1 59	3 64	13 66	24 69	29 73	30 78	35 82	39 88	44 94	49 99	52 100	55
Rota 7	1 59	3 64	13 66	24 68	29 73	30 78	35 82	39 88	44 94	49 99	52 100	55
Rota 8	1 59	4 64	14 66	24 69	29 73	30 78	35 82	39 88	44 94	49 99	52 100	55
Rota 9	1 59	4 64	14 66	24 68	29 73	30 78	35 82	39 88	44 94	49 99	52 100	55
Rota 10	1 59	3 64	13 66	24 69	29 73	30 78	35 82	39 83	44 88	49 94	52 99	55 100
Rota 11	1 59	3 64	13 66	24 68	29 73	30 78	35 82	39 83	44 88	49 94	52 99	55
Rota 12	1 64	9 66	18 68	25 73	30 78	35 82	39 88	44 94	49 99	52 100	55	59
Rota 13	1 64	10 66	18 69	25 73	30 78	35 82	39 88	44 94	49 99	52 100	55	59
Rota 14	1 59	3 64	13 66	24 69	29 73	30 78	35 82	39 87	44 93	49 99	52 100	55
Rota 15	1 59	3 64	13 66	24 68	29 73	30 78	35 82	39 87	44 93	49 99	52 100	55
Rota 16	1 59	4 64	14 66	24 69	29 73	30 78	35 82	39 87	44 93	49 99	52 100	55
Rota 17	1 64	10 66	18 69	25 73	30 78	35 82	39 83	44 88	49 94	52 99	55 100	59
Rota 18	1 59	4 64	14 66	24 68	29 73	30 78	35 82	39 87	44 93	49 99	52 100	55
Rota 19	1 64	10 66	18 68	25 73	30 78	35 82	39 83	44 88	49 94	52 99	55 100	59
Rota 20	1 64	9 66	18 69	25 73	30 78	35 82	39 87	44 93	49 99	52 100	55	59
Rota 21	1 64	9 66	18 68	25 73	30 78	35 82	39 87	44 93	49 99	52 100	55	59
Rota 22	1 64	10 66	18 69	25 73	30 78	35 82	39 87	44 93	49 99	52 100	55	59
Rota 23	1 64	10 66	18 68	25 73	30 78	35 82	39 87	44 93	49 99	52 100	55	59
Rota 24	1 63	10 66	18 69	25 73	30 78	35 82	39 87	44 93	49 99	52 100	55	59
Rota 25	1 63	10 66	18 68	25 73	30 78	35 82	39 87	44 93	49 99	52 100	55	59



Şekil 4.29. 100 düğümlü ağda problem 3 için Pareto çözümleri grafiği

Çizelge 4.40. 100 düğümlü ağda problem 3'ün Pareto çözümlerinin maliyet hesabı

	Rota No.	Yakıt Maliyeti (TL)	Zaman Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	1	52,02	89,04	141,06
	2	52,42	88,35	140,76
	3	52,67	88,33	140,99
	4	52,67	87,60	140,27
	5	52,81	85,77	138,58
	6	52,99	82,35	135,34
	7	53,24	82,02	135,27
	8	53,93	81,39	135,31
	9	54,22	81,06	135,28
	10	54,36	80,72	135,08
	11	54,65	80,40	135,05
	12	55,19	80,35	135,54
	13	55,22	77,99	133,21
	14	55,40	77,48	132,88
	15	55,69	77,15	132,84
	16	56,38	76,52	132,89

Çizelge 4.40. (devam)

	Rota No.	Yakıt Maliyeti (TL)	Zaman Maliyeti (TL)	Toplam Maliyet (TL)
GA/YBS	17	56,63	76,36	132,99
	18	56,66	76,19	132,86
	19	56,88	76,04	132,92
	20	57,35	75,80	133,15
	21	57,89	75,48	133,37
	22	57,67	73,12	130,79
	23	57,96	72,79	130,75
	24	60,23	72,76	132,99
	25	60,52	72,44	132,95

Çizelge 4.40'dan görüleceği gibi en az maliyete sahip olan 23 numaralı rota, EKY olarak kullanıcıya tavsiye edilmektedir.

4.2.3. 250 Düğümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi

Kurulan model öncelikle deterministik kenar bilgisine sahip bir ağ üzerinde çalıştırılmış, elde edilen sonuçlar klasik EKY algoritmalarından olan Dijkstra ve Floyd algoritması ile karşılaştırılmıştır (Çizelge 4.41). Modelin etkili sonuçlar verdiği bu karşılaştırmalar sonucunda görüldükten sonra stokastik ağ bilgisine sahip yollarda EKY bilgisi elde edilmiş Çizelge 4.42'de gösterilmiştir.

Çizelge 4.41. 250 düğümlü ağda deterministik EKY değerleri

Problem	Algoritmalar				Rota İndeks
	GA	YBS	Dijkstra	Floyd	
En kısa mesafeli yol (km)	541	541	541	541	Rota 1
Ort. en kısa seyahat süreli yol (dk)	525,79	525,79	525,79	525,79	Rota 2
Ort. en az yakıt tüketilen yol (lt)	38,71	38,71	38,71	38,71	Rota 3

Çizelge 4.42. 250 düğümlü ağda stokastik EKY değerleri

Problem	Algoritmalar		Rota İndeks
	GA	YBS	
535 dk altında gitme ihtimali en yüksek yol (%)	98	98	Rota 4
38,5 lt'nin altında yakıt tüketme ihtimali en yüksek yol (%)	18	18	Rota 5
%95 ihtimalle en az seyahat süreli yol (dk)	532,32	532,32	Rota 6
%95 ihtimalle en az yakıt tüketilen yol (lt)	39,07	39,07	Rota 7

250 düğümlü ağ için EKY'yi sağlayan rotalar **EK 12**'de gösterilmiştir.

4.2.4. 500 Düğümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi

Kurulan model öncelikle deterministik kenar bilgisine sahip bir ağ üzerinde çalıştırılmış, elde edilen sonuçlar klasik EKY algoritmalarından olan Dijkstra ve Floyd algoritması ile karşılaştırılmıştır (Çizelge 4.43). Modelin etkili sonuçlar verdiği bu karşılaştırmalar sonucunda görüldükten sonra stokastik ağ bilgisine sahip yollarda EKY bilgisi elde edilmiş Çizelge 4.44'de gösterilmiştir.

Çizelge 4.43. 500 düğümlü ağda deterministik EKY değerleri

Problem	Algoritmalar				Rota İndeks
	GA	YBS	Dijkstra	Floyd	
En kısa mesafeli yol (km)	1054	1054	1054	1054	Rota 1
Ort. en kısa seyahat süreli yol (dk)	1289,8	1289,8	1289,8	1289,8	Rota 2
Ort. en az yakıt tüketilen yol (lt)	72,66	72,52	72,52	72,52	Rota 3

Çizelge 4.44. 500 düğümlü ağda stokastik EKY değerleri

Problem	Algoritmalar		Rota İndeks
	GA	YBS	
1300 dk. altında gitme ihtimali en yüksek yol (%)	87	87	Rota 4
73 lt'nin altında yakıt tüketme ihtimali en yüksek yol (%)	95	95	Rota 5
%50 ihtimalle en az seyahat süreli yol (dk)	1290,6	1290,6	Rota 6
%50 ihtimalle en az yakıt tüketilen yol (lt)	72,65	72,65	Rota 7

500 düğümlü ağ için EKY'yi sağlayan rotalar **EK 13**'te gösterilmiştir

4.2.5. 1000 Düğümlü ağ için GA ve YBS yöntemi ile EKY değerlerinin elde edilmesi

Kurulan model öncelikle deterministik kenar bilgisine sahip bir ağ üzerinde çalıştırılmış, elde edilen sonuçlar klasik EKY algoritmalarından olan Dijkstra ve Floyd algoritması ile karşılaştırılmıştır (Çizelge 4.45). Modelin etkili sonuçlar verdiği bu karşılaştırmalar sonucunda görüldükten sonra stokastik ağ bilgisine sahip yollarda EKY bilgisi elde edilmiş Çizelge 4.46'da gösterilmiştir.

Çizelge 4.45. 1000 düğümlü ağda deterministik EKY değerleri

Problem	Algoritmalar				Rota İndeks
	GA	YBS	Dijkstra	Floyd	
En kısa mesafeli yol (km)	2735	2735	2735	2735	Rota 1
Ort. en kısa seyahat süreli yol (dk)	3489,7	3489,7	3489,7	3489,7	Rota 2
Ort. en az yakıt tüketilen yol (lt)	187,48	187,3	187,3	187,3	Rota 3

Çizelge 4.46. 1000 düğümlü ağda stokastik EKY değerleri

Problem	Algoritmalar		Rota İndeks
	GA	YBS	
3500 dk altında gitme ihtimali en yüksek yol (%)	80	80	Rota 2
187,5 lt'nin altında yakıt tüketme ihtimali en yüksek yol (%)	76	76	Rota 4
%50 ihtimalle en az seyahat süreli yol (dk)	3491,5	3491,5	Rota 5
%50 ihtimalle en az yakıt tüketilen yol (lt)	187,27	187,27	Rota 6

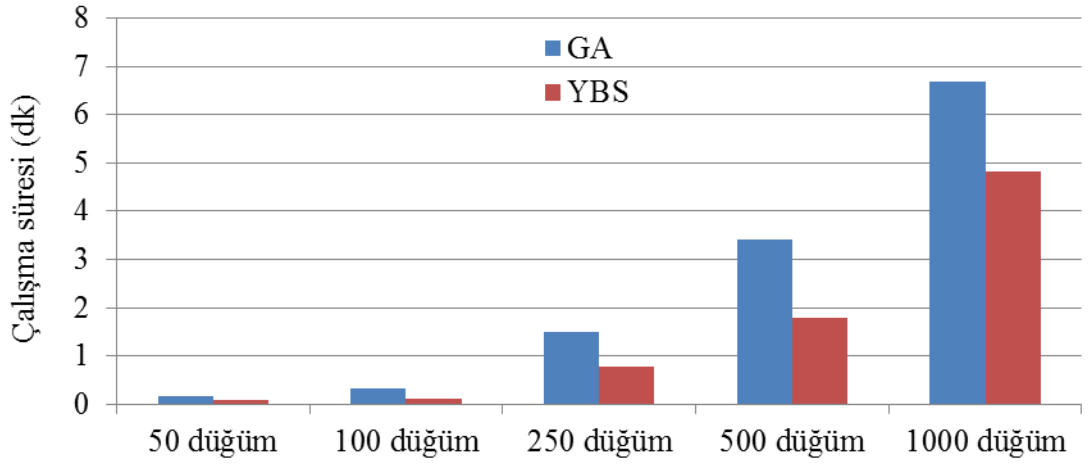
4.3. Çalışma Süresinin Değerlendirilmesi

4.3.1. GA ve YBS algoritmalarının çalışma zamanı açısından karşılaştırılması

Taguchi optimizasyonundan elde edilen parametreler kullanılarak öncelikle deterministik model için GA ve YBS algoritmaları çalıştırılmış, elde edilen sonuçlar Çizelge 4.47'de, algoritmaların çalışma sürelerinin karşılaştırılması ise Şekil 4. 30'da gösterilmiştir.

Çizelge 4.47. Deterministik GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması

Düğüm Sayısı	Deterministik Model				
	GA		YBS		YBS' nin zaman kazancı %
	EKY değeri	İşlem süresi (dk)	EKY değeri	İşlem süresi (dk)	
	Min. mesafe (km)		Min. mesafe (km)		
50	71	0,18	71	0,08	55
100	206	0,32	206	0,11	66
250	610	1,50	610	0,79	47
500	1123	3,42	1123	1,80	47
1000	2387	6,68	2387	4,81	28

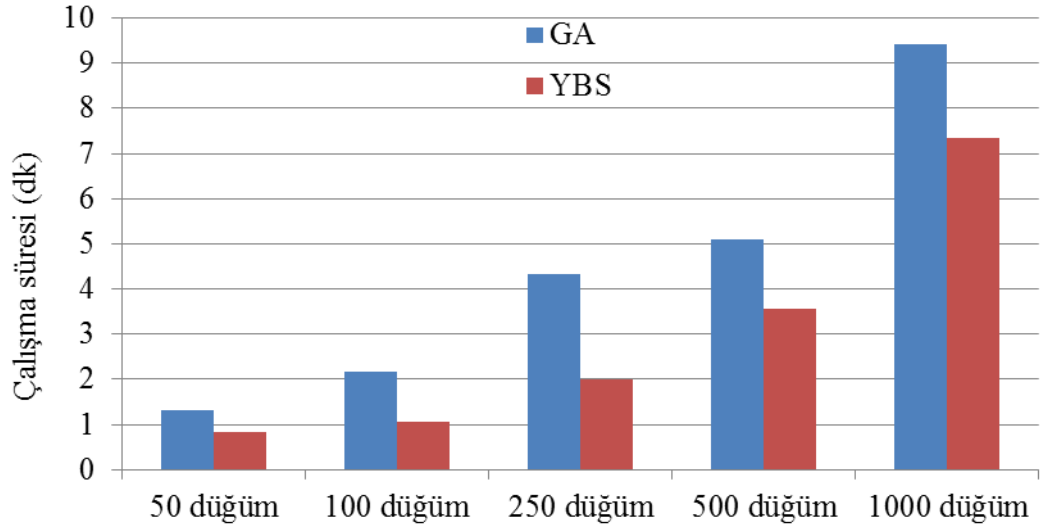


Şekil 4.30. Deterministik GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması

Verilen başlangıç T_0 değerinden daha az değerde gidilme ihtimali en yüksek yol olarak tanımlanan Stokastik Model-1 için GA ve YBS algoritmaları çalıştırılmış elde edilen sonuçlar Çizelge 4.48’de, algoritmaların çalışma sürelerinin karşılaştırılması ise Şekil 4.31’de gösterilmiştir.

Çizelge 4.48. Stokastik Model-1 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması

Düğüm Sayısı	Zaman kriteri (dk)	Stokastik Model-1				YBS’ nin zaman kazancı %
		GA		YBS		
		EKY değeri	İşlem süresi (dk)	EKY değeri	İşlem süresi (dk)	
		Mak. İhtimal (%)		Mak. İhtimal (%)		
50	$T \leq 85$	7	1,30	7	0,83	36
100	$T \leq 230$	92	2,18	92	1,07	51
250	$T \leq 690$	85	4,33	85	2,00	54
500	$T \leq 1470$	97	5,1	97	3,56	30
1000	$T \leq 3800$	93	9,43	93	7,34	22

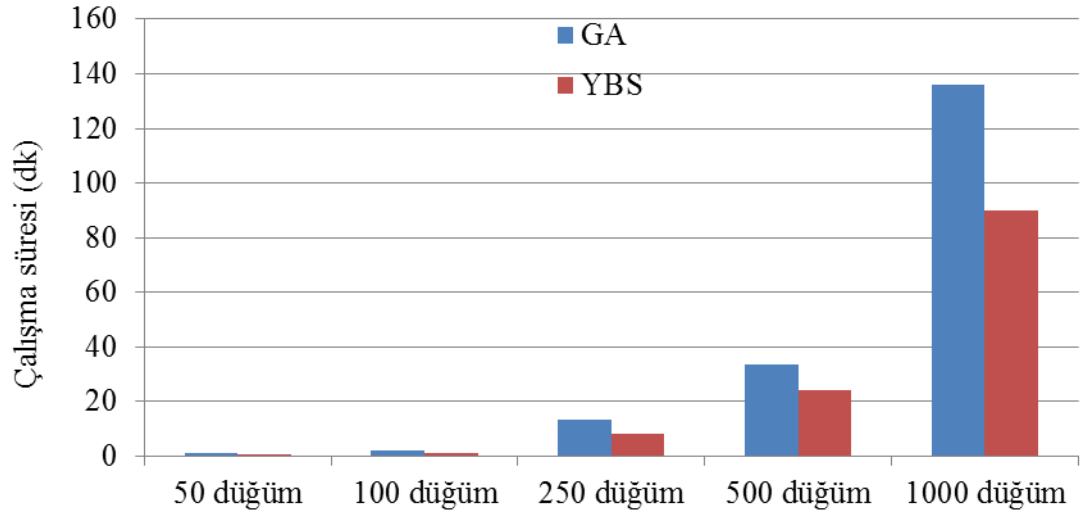


Şekil 4.31. Stokastik Model-1 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması

Verilen α ihtimalle en az değerle gidilen yol olarak tanımlanan Stokastik Model-2 için GA ve YBS algoritmaları çalıştırılmış, elde edilen sonuçlar Çizelge 4.49’da, algoritmaların çalışma sürelerinin karşılaştırılması ise Şekil 4.32’de gösterilmiştir.

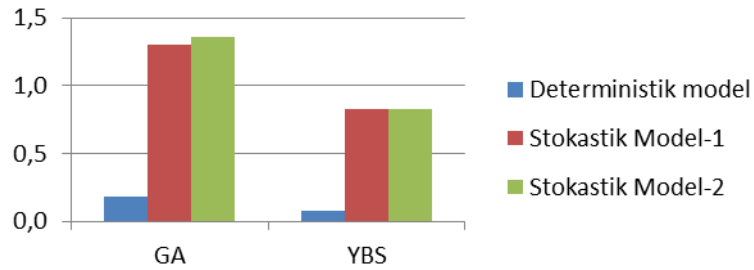
Çizelge 4.49. Stokastik Model-2 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması

Düğüm Sayısı	Stokastik Model-2				
	GA		YBS		YBS’ nin zaman kazancı %
	EKY değeri	Çalışma süresi (dk)	EKY değeri	Çalışma süresi (dk)	
	%95 iht. Min T (dk)		%95 iht. Min T (dk)		
50	88,80	1,36	88,80	0,83	
100	230,09	2,15	230,09	1,08	50
250	692,73	13,38	692,73	8,37	37
500	1465,8	33,65	1466,5	24	28
1000	3809,7	136	3810	90,05	34



Şekil 4.32. Stokastik Model-2 için GA ve YBS algoritmalarının çalışma sürelerinin karşılaştırılması

Kurulan modellerin GA ve YBS'ye göre işlem süreleri Şekil4.33'te gösterilmiştir.



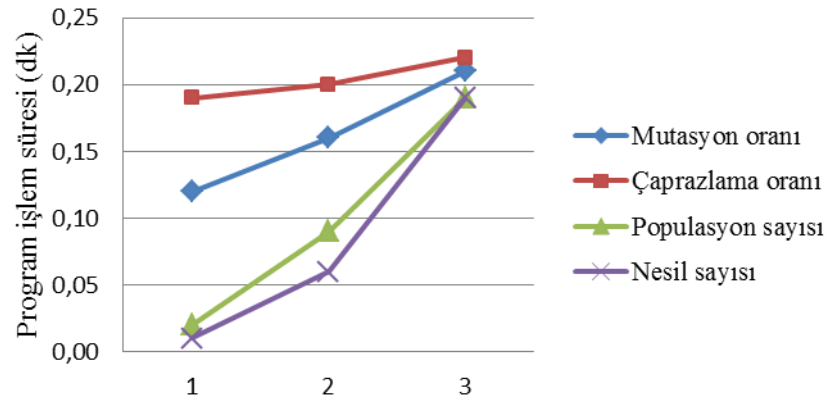
Şekil 4.33. 50 düğümlü ağ için GA ve YBS de modellerin işlem süresi açısından karşılaştırılması

4.3.2. GA ve YBS algoritma parametrelerinin işlem süresine etkileri

GA'da popülasyon sayısı, nesil sayısı, çaprazlama oranı, mutasyon oranı ve klonlama oranının işlem süreleri üzerindeki etkileri Çizelge 4.50'de gösterilmiştir.

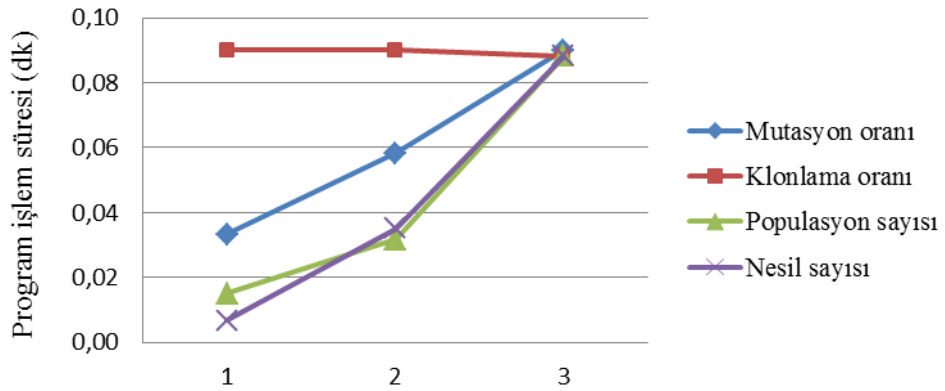
Çizelge 4.50. GA parametrelerinin işlem süresine etkisi

Deterministik GA						
Parametre	Parametre değeri	Düğümler				
		50	100	250	500	1000
		İşlem süresi(dk)	İşlem süresi(dk)	İşlem süresi(dk)	İşlem süresi(dk)	İşlem süresi(dk)
Mutasyon oranı	0,1	0,12	0,16	0,84	1,82	5,5
	0,5	0,16	0,22	1,17	2,46	6,99
	1	0,21	0,3	1,56	3,27	8,96
Klonlama oranı	0,1	0,19	0,19	1,31	1,97	7,71
	0,5	0,2	0,25	1,42	2,18	8,33
	1	0,22	0,31	1,58	2,48	8,98
Populasyon sayısı	10	0,02	0,03	0,33	0,63	1,78
	50	0,09	0,14	0,89	1,45	4,79
	100	0,19	0,31	1,58	2,50	8,85
Nesil sayısı	10	0,01	0,01	0,06	0,10	0,33
	100	0,06	0,11	0,53	0,84	2,97
	300	0,19	0,32	1,58	2,46	8,80

**Şekil 4.34.** 50 düğümlü ağda GA parametrelerinin çalışma süresi üzerine etkileri

Çizelge 4.51. YBS parametrelerinin program çalışma süresine etkisi

Deterministik YBS						
Parametre	Parametre değeri	Düğümler				
		50	100	250	500	1000
		İşlem süresi(dk)	İşlem süresi(dk)	İşlem süresi(dk)	İşlem süresi(dk)	İşlem süresi(dk)
Mutasyon oranı	0,1	0,03	0,03	0,33	0,75	2,52
	0,5	0,06	0,07	0,53	1,18	3,53
	1	0,09	0,12	0,83	1,72	4,85
Klonlama oranı	0,1	0,09	0,11	0,82	1,73	4,85
	0,2	0,09	0,11	0,82	1,73	4,83
	0,5	0,09	0,11	0,82	1,72	4,85
Populasyon sayısı	10	0,02	0,02	0,18	0,43	1,07
	30	0,03	0,04	0,35	0,72	1,85
	100	0,09	0,11	0,80	1,73	4,90
Nesil sayısı	10	0,01	0,01	0,05	0,10	0,27
	75	0,04	0,04	0,32	0,67	1,85
	200	0,09	0,11	0,82	1,72	4,87

**Şekil 4.35.** 50 düğümlü ağda YBS parametrelerinin işlem süresi üzerine etkileri

4.4. Tartışma

Şekil 4.1-4.10'da görüldüğü gibi GA ve YBS'de popülasyon ve nesil sayısının büyük alınması performans istatistiği değerini artırmış dolayısıyla kurulan modellerden daha doğru sonuçlar alınması sağlanmıştır. Fakat her iki parametrenin artması Şekil 4.34 ve 4.35'de görüldüğü gibi çalışma süresinin artmasına sebep olmuştur.

Mutasyon oranının GA'da 0,5 ten ve YBS'de 1den küçük alınması durumunda Şekil 4.6-4.10'da görüleceği üzere performans istatistiği değeri azalmış dolayısıyla algoritmanın hedef değere iyi bir şekilde yakınsayamadığı görülmüştür. Mutasyon oranının GA'da 0,5 ve 1 ve YBS'de 1 olması durumunda optimum EKY değerlerine ulaşılmıştır (Çizelge 4.15-4.16). Mutasyon oranının büyük olması, popülasyonda daha fazla çeşitlilik sağlanması ve araştırma uzayının farklı noktalarında çalışılması anlamına geldiğinden daha iyi sonuçlar alınmasına sebep olmuştur. Şen (2004), mutasyon oranının 1 alınmasının algoritmayı rastgele arama yapan bir algoritmaya dönüştürdüğü ifade edilse de araştırma uzayının büyük olduğu bu tip problemlerde mutasyonun büyük alınması avantaj sağlamıştır. Bu konuda ayrıca Cicirello ve Smith (2000), GA ile EKY problemlerinin çözümünde, optimum veya optimuma yakın çözümlere ulaşmak için parametrelerin seçimi ile ilgili herhangi bir kural mevcut olmadığını ve optimuma yakın çözüm veren bir kontrol parametresi verisinin, başka bir GA uygulaması için genelleştirilemeyeceğini ifade etmişlerdir.

Çaprazlama oranının GA da 0,5 ve 1 değerinde alınması Şekil 4.1-4.5'de görüldüğü gibi S/N oranını artırma yönünde etki etmiştir. İşlem süresi bakımından çaprazlama oranının artışının, GA'nın diğer parametreleriyle karşılaştırıldığında en az etkiye sahip olduğu gözlenmiştir (Şekil 4.34).

Klonlama oranı Şekil 4.6-4.10'da görüldüğü gibi 0,1 ve 0,2 değerinde başarılı sonuçlar vermiştir. Bu sonuç en başarılı kromozomlardan çok sayıda oluşturulmuş klon popülasyonu ile çalışmanın, EKY'ye ulaşmada etkili olacağını göstermiştir. Şekil 4.35'de klonlama oranının işlem süresi üzerinde etkili olmadığı görülmüştür.

50 düğümlü ağlarda Şekil 4.16-4.18 ve 100 düğümlü ağlarda Şekil 4.27-4.29'da görüldüğü gibi, Pareto çözümlerinin sayısının problemin türüne göre değiştiği gözlenmiştir.

AUS kullanıcılarına yol göstermesi bakımından Pareto çözümlerinden hangisinin seçileceği konusunda yapılan maliyet değerlendirmesi 50 düğümlü ağ için Çizelge 4.23, 4.25 ve 4.27'de 100 düğümlü ağ için Çizelge 4.36, 4.38 ve 4.40'da verilmiştir.

50 ve 100 düğümlü ağlarda tek amaçlı GA ve YBS algoritmalarında kullanılan popülasyon sayısı ve nesil sayısı, pareto çözümlerin elde edilmesinde yetersiz kalmış bu sebeple daha büyük popülasyon ve nesil sayılarıyla çalışılmıştır.

GA ve YBS'nin performansı çalışma süreleri açısından karşılaştırıldığında genel olarak YBS'nin GA'ya göre işlem zamanı bakımından deterministik problemlerde Çizelge 4.47'de görüldüğü gibi %28 ile %66 arasında, Stokastik model-1 için Çizelge 4.48'den görüleceği üzere %22 ile % 54 arasında ve Stokastik model-2 için Çizelge 4.49'dan görüleceği üzere %28 ile % 50 arasında üstünlük sağladığı görülmüştür. Genel olarak düğüm sayısının artması çalışma süresini artırmıştır (Şekil 4.30-4.32). Bunun temelinde, büyük düğümlü ağların ara hedef düğümlerine bölünerek araştırma uzayının detaylı bir şekilde aranmasının sebep olduğu tespit edilmiştir.

Şekil 4.33'den görüldüğü gibi seçilen modelin türünün çalışma süresi üzerinde etkili olduğu belirlenmiştir. Deterministik model ile Stokastik model-2, çalışma süreleri açısından karşılaştırıldığında GA için genel olarak %85 ile %95 arasında işlem süresi artarken YBS için % 90 ile %95 arasında işlem süresi artmıştır (Çizelge 4.34 ve 4.36). Bu duruma stokastik model-2'de bulunan stokastik simülasyon operatörünün varlığı sebep olmuştur.

EKY problemlerinin GA ve YBS ile çözüm performansının artırılması için en etkin operatör seviyelerinin belirlenmesi amacıyla Taguchi optimizasyonunun uygulanması

özüm süresini kısaltmak ve optimum çözümler elde edilebilmesi açısından önem taşımaktadır.

5. SONUÇ ve ÖNERİLER

5.1. Sonuçlar

Bu çalışmada maliyetli ve yönlü 50, 100, 250, 500 ve 1000 düğümlü olmak üzere toplam beş tür ağ üzerinde üç farklı GA ve YBS algoritmaları ile EKY problemi çözülmüştür (Çizelge 4.17-4.46). Bu ağlar, Matlab ortamında rastgele ağ oluşturma algortimasıyla elde edilmiştir.

Taguchi optimizasyon tekniği ile GA için mutasyon oranı, çaprazlama oranı, populasyon sayısı ve nesil sayısı ve YBS için mutasyon oranı, klonlama oranı, populasyon sayısı ve nesil sayısı parametrelerinin EKY hesaplamada etkileri araştırılmış ve her bir ağ türü için optimum seviyeleri belirlenmiştir. Belirlenen optimum seviyelerde EKY problemi çözülerek doğrulama deneyleri yapılmış ve sonuçların %95 güven aralığında elde edildiği gösterilerek optimum seviyelerin doğruluğu ispatlanmıştır (Çizelge 4.1-4.16).

Parametreler için optimum seviyeler kullanılarak GA ve YBS ile deterministik EKY problemlerinden olan en kısa mesafeli, en kısa ortalama seyahat süreli ve en az ortalama yakıt maliyetli problemler çözülmüş, elde edilen sonuçlar klasik en kısa yol algoritmalarından olan Dijkstra ve Floyd algoritması ile karşılaştırılarak kurulan algoritmaların doğruluğu ispatlanmıştır (Çizelge 4.17, 4.28, 4.41, 4.43 ve 4.45).

Ağ kenar bilgisi olarak mesafe, deterministik bilgi olarak kullanılırken, taşıtların kenar seyahat süresi dağılımı ve yakıt tüketimi dağılım fonksiyonları stokastik bilgi olarak kullanılmıştır. Kenarlardaki seyahat süresi için normal dağılım, üçgen dağılım ve uniform dağılım; yakıt tüketimi için ise normal dağılıma uygun veriler kullanılarak stokastik bir ortam oluşturulmuştur.

Stokastik modeller için GA ve YBS ile EKY problemleri çözülerek sonuçlar birbirleri ile karşılaştırılmıştır. Her iki algoritma yaklaşık aynı sonuçları vermiştir (Çizelge 4.18, 4.29, 4.42, 4.44 ve 4.46).

50 ve 100 düğümlü ağ için iki amaçlı üç farklı problem türü için pareto optimum çözümleri matris yöntemi ile elde edilmiş ve her iki algoritmanın benzer sonuçlar verdiği gözlenmiştir (Çizelge 4.22, 4.24, 4.26, 4.35, 4.37 ve 4.39). AUS kullanıcılarına Pareto optimum çözümlerinden hangisinin kullanılacağına karar vermeleri bakımından çözümlerin maliyet karşılaştırması yapılarak EKY belirlenmiştir (Çizelge 4.23, 4.25, 4.27, 4.36, 4.38 ve 4.40).

250, 500 ve 1000 düğümlü ağların araştırma uzayı büyük olduğundan etkili sonuçlar alınabilmesi için ara hedef düğümler belirlenerek araştırma uzayı küçük parçalara ayrılmıştır. Ara hedef düğümler arasında EKY seçilirken klasik algoritmaların ilerleme tekniğinden yararlanılarak melez bir metot oluşturulmuştur. Ara hedef düğümler arasında mevcut bulunan yolların doğru şekilde bulunması amacıyla ağ budama algoritması oluşturularak ara hedef düğümlerle ilgisi olmayan kenarlar ağdan çıkarılmıştır. Böylece tamir algoritmaları kullanma zorunluluğu ortadan kaldırılmıştır.

100 düğümlü ağda mutasyon planı oluşturularak belirli bir nesil sayısında farklı mutasyon bölgesinde çalışması gerçekleştirilerek bir yolun küçük salınımlar yapması sağlanarak yakınındaki en iyi yolların aranması stratejisi oluşturulmuştur.

GA ve YBS çalışma zamanı açısından değerlendirildiğinde YBS'nin çözümleri daha kısa sürede bularak GA ya üstünlük sağladığı görülmüştür (Şekil 4.30-4.32).

Ağ büyüklüğüne göre algoritmaların çalışma zamanı değerlendirildiğinde düğüm sayısı arttıkça işlem yükü artışında GA ve YBS algoritmalarının sonuç alma süresi önemli ölçüde artmıştır (Şekil 4.30-4.32).

GA ve YBS parametrelerinden olan populasyon sayısı ve nesil sayısının artması algoritmanın daha doğru sonuçlar almasını sağlarken çalışma zamanını artırdığı gözlenmiştir (Şekil 4.34 ve 4.35).

Algoritmalarda kullanılan stokastik simülasyon işlem süresini artırmıştır. 50 ve 100 düğümlü ağlarda GA için %85 ve YBS için %90 oranında işlem süresinin uzamasına sebep olurken 250, 500 ve 1000 düğümlü ağlarda GA ve YBS için %90-95 oranında işlem süresinin uzadığı gözlenmiştir (Çizelge 4.47-4.49).

5.2. Öneriler

- Kurulan algoritmalar CBS ile entegre edilip gerçek bir ulaşım ağı üzerinde çalışması sağlanarak kenarların ihtimal dağılımlarının zamana göre değiştiği stokastik ve dinamik programlama gerçekleştirilebilir.
- GA ve YBS algoritmalarının işlem sürelerinin kısaltılması konusunda algoritmalar üzerinde çeşitli bilgisayar bilimi tekniklerden yararlanılabilir.
- Kurulan modeller AUS içinde kullanılarak gerçek zamanlı bilgi ile çalışması sağlanabilir. Böylece kullanıcıya, EKY bilgisi sağlanarak zaman ve yakıt tasarrufunda bulunması sağlanırken gecikmelerini de minimize etmesinde yardımcı olur.

KAYNAKLAR

- Ahn, C.W., and Ramakrishna, R.S., 2002. A Genetic Algorithm for Shortest Path Routing Problem and Sizing of Populations. *IEEE Transaction on Evolutionary Computation*, Vol 6 No: 6 pp 566-578.
- Badawy, G., 2010. Resource Management in Solar Powered Wireless Mesh Networks. Ph. D. Dissertation. McMaster University, Canada.
- Bakhouya M. and Gaber J., 2007. An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salesman Problem *Advanced Modeling and Optimization*, Volume 9, Number 1
- Bakırcı, E., 2005. Taşıt İşletme Maliyetleri Bileşenlerinin İrdelenmesi, Tasarruf Bakışının Yerleştirilmesi. 6. Ulaştırma Kongresi Bildiriler Kitabı
- Berbert, P.C. , Filho L.J.R., Almeida T. A. , Carvalho M.B. , and Yamakami A., (2007) Artificial Immune System to Find a Set of k -Spanning Trees with Low Costs and Distinct Topologies. *ICARIS 2007, LNCS 4628*, pp. 395–406. Springer-Verlag
- Boonsuwan, P., 2009. An Evolutionary Algorithm Approach to Complex Network Optimization. Ph. Dissertation. University of California, Berkeley. California
- Brownlee, J., 2007. Clonal Selection Algorithms. CIS Technical Report 070209A. pp. 1-13
- Castro, L.N. and Zuben, F.J.V., 1999. Artificial Immune System: Part 1- Basic Theory and Applications. Technical Report, TR-DCA 01/99.
- Castro, L.N. and Zuben, F.J.V., 2001. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*, Special Issue on Artificial Immune Systems
- Castro, L.N. and Timmis, J., 2002. *Artificial Immune System A New Computational Intelligence Approach*. Springer
- Castro L.N. and Timmis, J., 2002. Artificial Immune Systems: A Novel Paradigm to Pattern Recognition In *Artificial Neural Networks in Pattern Recognition* , J. M. Corchado, L. Alonso, and C. Fyfe (eds.), *SOCO-2002*, University of Paisley, UK, pp. 67-84
- Castro, L.N. and Zuben, V.F. J., 2000. “The Clonal Selection Algorithm with Engineering Applications”, *Workshop proceedings of GECCO’00*.
- Cavit, A., 1998. Efficient Algorithms for Navigation on Large Transportation Graphs. Ph.D., University of Southern California. United States – California
- Chuang, T.N. and Kung J.Y., 2005. The Fuzzy Shortest Path Length and the Corresponding Shortest Path in a Network. *Computers & Operations Research*, 32, 1409–1428
- Cicirello, V.A. and S.F. Smith. 2000. "Modeling GA Performance for Control Parameter Optimization. *Proceedings of the Genetic and Evolutionary Computation Conference*. Las Vegas. pp. 235-242.
- Çölkesen, R., 2004. *Veri Yapıları ve Algoritmalar*. Papatya Yayıncılık Eğitim, 3. Baskı
- Dasgupta, D., 1998. *Artificial Immune System and Their Applications*. Springer

- Deliktaş, B., Bikçe, M., Çoşkun, H., Türker, T.,H., 2009. Betonarme Kirişlerin optimum tasarımında Genetik Algoritma parametrelerinin Etkisinin Belirlenmesi.Fırat Üniversitesi Mühendislik Bilimleri dergisi. 21(2), 125-132
- Frechi, F. and Repetto, M., Multiobjective Optimization by a Modified Artificial Immune Systems Algorithm. Lecture Notes in Computer Science, 1-14
- Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 141–153. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).
- Fu, L., Sun, D. and Rilett, L.R., 2006. Heuristic Shortest Path Algorithms for Transportation Applications. Computers & Operations Research, 33, 3324–3343
- Gandibleux, X., Beugnies, F. and Randriamasy, S., 2005. Martins' algorithm revisited for multi-objective shortest path problems with a MaxMin cost function. A Quarterly Journal of Operations Reserach. 47-59
- Ghoseiri, K. and Nadjari, B., 2009. An ant colony optimization algorithm for the bi-objective shortest path problem Applied Soft Computing.
- Giannakoglou et al., editors, Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), pages 95–100.
- Gong, M., Jiao, L. and Zhang, X., 2008. A Population-based Artificial Immune System for Numerical Optimization. Neurocomputing, 1-13
- Goldberg, D.E. 1997. Genetic Algorithms in Search, Optimization, and Machine Learning.
- Gomez-Albarran, M., Fernandez-Pampillon-Cesteros, M.F.P. and Sancez-Perez, J.M., 1997. A Routing Strategy Based on Genetic Algorithms. Microelectronics Journal, 28, 641–656
- Gomez-Sanchez, M.A., 2001. Genetic Algorithms and Simulation Applied to Optimization: The Stochastic Shortest Path Model. Ph.D., Mexico State University, United States, New Mexico
- Grosan, C., Abraham, A., 2007. Solving Shortest Capacitated Path Problem Using a Bi-Objective Heuristic Approach. Proceedings of the First Asia International Conference on Modelling & Simulation. IEEE Computer Society. ISBN: 0-7695-2845-7. s 427-433
- Gürsu, B., İnce, M.Düzenli Dizaynılı Genetik Algoritmalar ile Çok Amaçlı Programlama. 5. Uluslararası İleri Teknolojiler Sempozyumu (İATS'09), 13-15 Mayıs 2009, Karabük, Türkiye
- Hamacher, H.W., Ruzika, S. And Tijandra, S.A., 2006. Algorithms for time-dependent bicriteria shortest path problems. Discrete Optimization, 3, 238–254
- Hamdan, M., A., 2004. Genetic Algorithm Approach for Constrained Multicast Routing in Computer Networks. PhD. Dalhousie University, Canada.
- He, R.R., Kornhauser, A.L. and Ran B., 2005. Essentially Best Routes in Dynamic and Stochastic Network. International Journal of Vehicle Information and Communication Systems Vol. 1, No.1/2 pp. 1 - 14
- He, F., Qi, H. And Fan, Q., 2007. Evolutionary Algorithm for the Multi-objective Shortest Path Problem.

- Hernandes, F., Lamata, M. T., Verdegay, J.L. and Yamakami, A., 2007. The Shortest Path Problem on Networks with Fuzzy Parameters. *Fuzzy Sets and Systems* 158. pp 1561-1570
- Hınısliođlu, S. and Bayrak, O. Ü., 2004. Optimization Of Early Flexural Strength Of Pavement Concrete With Silica Fume and Fly Ash By The Taguchi Method. *Civil Engineering and Environmental Systems*, Vol. 21, No. 2, June 2004, pp. 79–90
- Jang J. S. R., 1997. *Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence*, Chapter 7: Derivative-Free Optimization, Prentice-Hall, USA, s. 173-196.
- Ji, X., 2005. Models and Algorithm for Stochastic Shortest Path Problem. *Applied Mathematics and Computation*, 170, 503–514
- Jung, S., 2000. A Genetic Algorithm for The Vehicle Routing Problem With Time-Dependent Travel Times. Ph. D., University of Maryland College Park, United States, Maryland
- Kang, S., 2008. Concierge Service Problem for Location-Based Services: Combined-Cost and Multi Objective Approaches. Ph Dissertation. University of Illinois at Urbana- Champaign, United States Illinois.
- Keko, H., Skok, M. and Skrlec, D., 2003. Artificial Immune System in Solving Routing Problems. *Eurocon*, Ljubljana, Slovenia, 62–66
- Keko, H., Skok, M. and Skrlec, D., 2004. Solving the Distribution Network Routing Problem with Artificial Immune Systems. *IEEE Melecon*, May, 12–15
- Kim, S., Lewis, M.E. and White, C.C., 2005. Optimal Vehicle Routing With Real-Time Traffic Information. *IEEE Transactions on Intelligent Transportation Systems*, 6/2, 178–188
- Kumar A.J.S., Arunedeivi, J. and Mohan, V., 2009 Intelligent Transport Route Planning Using Genetic Algorithms in Path Computation Algorithms. *European Journal of Scientific Research*. ISSN 1450-216 X Vol.25 No.3 pp. 463
- Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In H.-P. Schwefel and R. Männer (Eds.), *Parallel Problem Solving from Nature — Proc. 1st Workshop PPSN*, Berlin, pp. 193–197. Springer.
- Kurt, M., Semetay, C., 2001. Genetik Algoritma ve Uygulama Alanları. *Mühendis ve Makine*, cilt 42 sayı 501
- Liang, Z., Jianmin, X. and Lingxiang, 2007. Application of Genetic Algorithm in Dynamic Route Guidance System. *Journal of Transportation System Engineering and Information Technology*, 7(3), 45–48
- Lin, M., 2008. A Regional Traffic Control System Using Emerging Intelligent Transportation Systems (ITS) Technologies. Dissertation. The faculty of the School of Engineering and Applied Science University of Virginia. 230s
- Lin, C., H., Yu J., L., Liu, J., C. And Lee, C., J, 2008. Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems. 2008 International Conference on Multimedia and Ubiquitous Engineering. IEEE Computer Society. p. (402-406).
- Lin, C., H., Yu J., L., Lai, W., S. And Ho, C., H, 2009. Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems. *International Journal of Hybrid Information Technology*. Vol. 2, No. 1, p. (21-30).

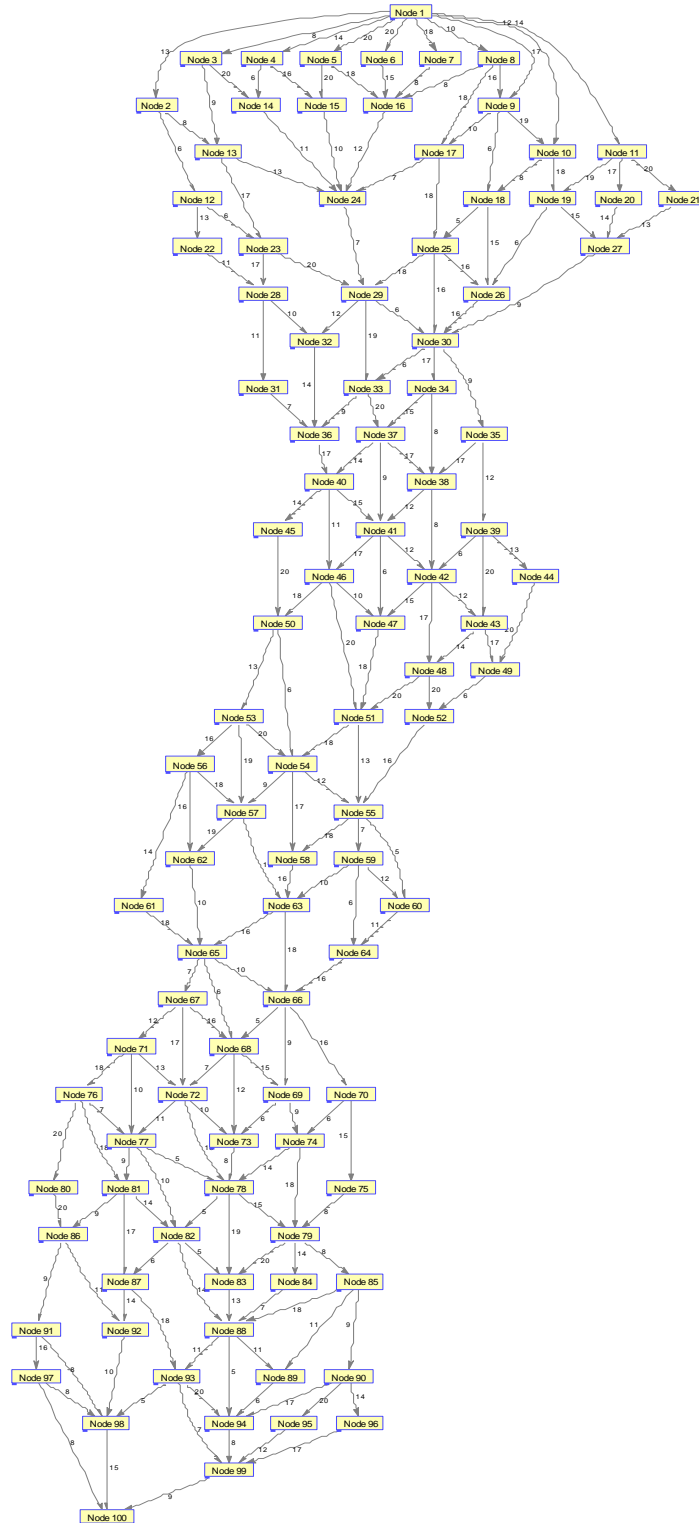
- Liu F., Liu Y., Chen X., and Wang J.Ş., 2006. Multi-Agent Immune Clonal Selection Algorithm Based Multicast Routing. (Eds.): ICNC 2006, Part II, LNCS 4222, pp. 319 – 327, 2006. Springer-Verlag Berlin Heidelberg
- McGregor, R., V., 2003. Development of an Intelligent Transportation System Architecture for a Medium Sized City. A master thesis. Department of Civil engineering and. University of Calgary.216s
- Michalewicz, Z., 1992. Genetic Algorithms + Data Structures = Evolution Programs. Spinger Series Artificial Intelligence
- Miller, J., 2009. Dynamically Computing Fastest Paths for Intelligent Transportation Systems. IEEE Intelligent Transportation Systems Magazine 20 Spring
- Pangilinan, J.M.A. and Janssens, 2007. Evolutionary Algorithms for the Multiobjective Shortest Path Problem. International Journal of Computer and Information Science and Engineering, 1, 54–59
- Pangilinan, J. And Janssens, G., 2007. Evolutionary Algorithms for the Multiobjective Shortest Path Problem. World Academy of Science, Engineering and Technology. pages 205-210
- Pattanamekar, P., Park, D. Rilett, L.R., Lee, J. and Lee, C., 2003. Dynamic and Stochastic Shortest Path Transportation Networks with Two Components of Travel Time Uncertainty. Transportation Research Part C, 11, 331–354
- Pınar, M.,A., Güllü,A., 2010. Sayısal Denetimli Hidrolik Pozisyonlama Sisteminin Taguchi Metodu ile Optimizasyonu. Gazi Üniv. Müh. Mim. Fak. Der. Cilt 25, No:1s 93-100
- Sağ, T., Çunkaş, M., 2009. Çok amaçlı Genetik Algoritmalar için Bir Çevrimdışı Performans Değerlendirmesi. 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), 3-15 Mayıs 2009, Karabük, Türkiye
- Savaşkan M., Taptık Y., Ürgen, M., 2004. Deney Tasarımı Yöntemi ile matkap uçlarında performans optimizasyonu. itü dergisi/d mühendislik cilt 3 sayı6 117-128 Aralık 2004
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 93–100. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).
- Shimizu, H. and Kobayashi, M., 1994. A Route Guidance Algorithm of Traffic Network. Vehicle Navigation & Information Systems Conference Proceedings, 327–33
- Soltani, A.R., Tawfik, H., Goulermas J.Y. and Fernando, T., 2003. Path Planning Construction Sites: Performance Evaluation of The Dijkstra, A*, and GA Search Algorithms. Advanced Engineering Informatics 16. pp 291-303.
- Subramanian, S., 1999. Routing Algorithms for Dynamic, Intelligent Transportation Networks.Ph.D., Virginia Polytechnic Institute and State University. United States – Virginia.
- Şahan S. (2004) Yapay Bağışıklıkta Yeni algoritmalar. Selçuk Üniversitesi Elektrik Elektronik Mühendisliği A.B.D. Yüksek Lisans Tezi.
- Şen, Z., 2004. Mühendislikte Bulanık (Fuzzy) Mantık ile Modelleme Prensipleri. Su Vakfı Yayınları, 2. Baskı

- Taniguchi, E. and Shimamoto, H., 2004. Intelligent Transportation System Based Dynamic Vehicle Routing and Scheduling with Variable Travel Times. *Transportation Research Part C*. Vol. 12, pp 235-250
- Timmis, J., Neal, M. and Hunt, J., 2000. An Artificial Immune System for Data Analysis. *BioSystems*, 55, 143–150
- Tsaggoruis, G., and Zaroliagis, C., 2007. Multiobjective Optimization: Improved FPTAS for Shortest Paths and Non-Linear Objectives with Applications. *Theory Comput Syst*, pages 162–186.
- Tulum, G.,S. and Terzi, S., 2010. Karayollarında Seyahat Edenlerin Karayolu Servis Yeteneğine Göre Zaman Kaybının Maliyetinin Belirlenmesi. Volume 2 No 1 February 2010 pp 55-64
- Vanderschuren, M.J.V.A., 2003. The Benefits of Intelligent Transport Systems: Modelling The Effects of Different ITS Sytems. *Proceedings of the 22nd. Southern African Transport Conference (SATC2003)*. ISBN number: 0-958-46096-5.
- Velega, N., R., Quddus, M.,A. And Bristow, L.,A., 2009. Developing An Enhanced Weight-Based Topological Map-Matching Algorithm for Intelligent Transport Systems. *Transportation Research Part C* 17, 672–683
- Victoria Transport Policy Institute. 2009 Transportation Cost and Benefit Analysis II – Travel Time Costs. www.vtpi.org/tca/tca0501.pdf (09.02.2009)
- Victoria Transport Policy Institute. 2010 Transportation Cost and Benefit Analysis II – Vehicle Costs. www.vtpi.org/tca/tca0502.pdf (04.06.2010)
- Vijayalakshmi, K. and Radhakrishnan, S., 2008. Artificial Immune Based Hybrid GA for QoS Based Multicast Routing in Scale Networks. *Computer Communications*, 31, 3984–3994
- Wang X., Gao X. Z., and Ovaska J., 2004. Artificial Immune Optimization Methods and Applications- A Survey. *IEEE International Conference on Systems, Man and Cybernetics*.
- Wanichworanant, N., 2003. A Traffic Engineering Approach Employing Genetic Algorithms Over MPLS Networks. Ph.D., Wichita State University, United States, Kansas.
- Xu, M.H., Lui, Y.Q., Huang, Q.L., Zhang, Y.X. and Luan, G.F., 2007. An Improved Dijkstra's Shortest Path Algorithm for Sparse Network. *Applied Mathematics and Computation*, 185, 247–254
- Zhang, Y., Jun, Y., Wei, G. and Wu L., 2009. Find Multi-objective paths in Stochastic Networks via Chaotic Immune PSO. *Expert Systems with Applications*. Pages
- Ziliaskopoulos, A., Kotzinos, D. and Mahmassani, H., 1997. Design and Implementations of Parallel Time Dependent Least Time Path Algorithms for Intelligent Transportation Systems Applications. *Transportation Research part C*. Vol. 5, No.2, pp. 95-107.
- Zitzler, E., 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, ETH Zurich, Switzerland.
- Zitzler, E., Laumanns, M. and Thiele, L., 2002. Spea2: Improving The Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K.C.

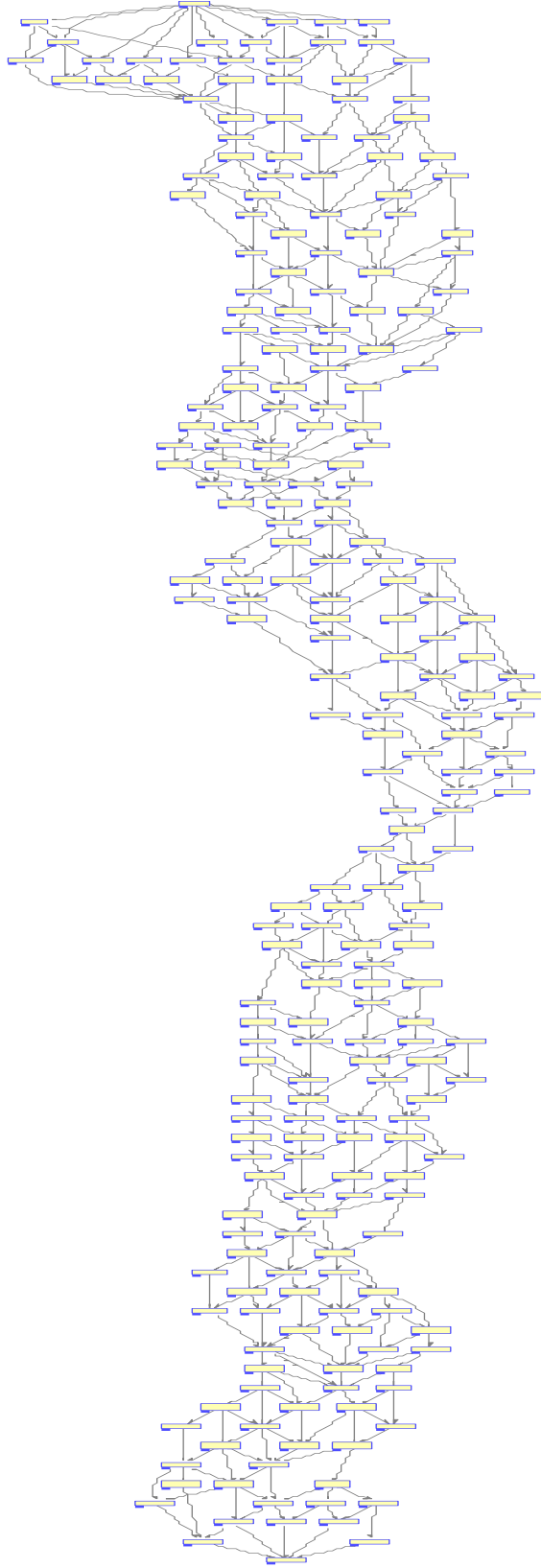
- Zitzler, E., Laumanns, M., and Bleuler, S., 2004. A Tutorial on Evolutionary Multiobjective Optimization. In X. Gandibleux et al., editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*. Springer.
- Zhu, W., 2010. Design and development of novel routing methodologies for dynamic roadway navigation systems. Ph.D. dissertation, University of California, Riverside, United States - California.

EKLER

EK 1. 100 düğümlü ağ

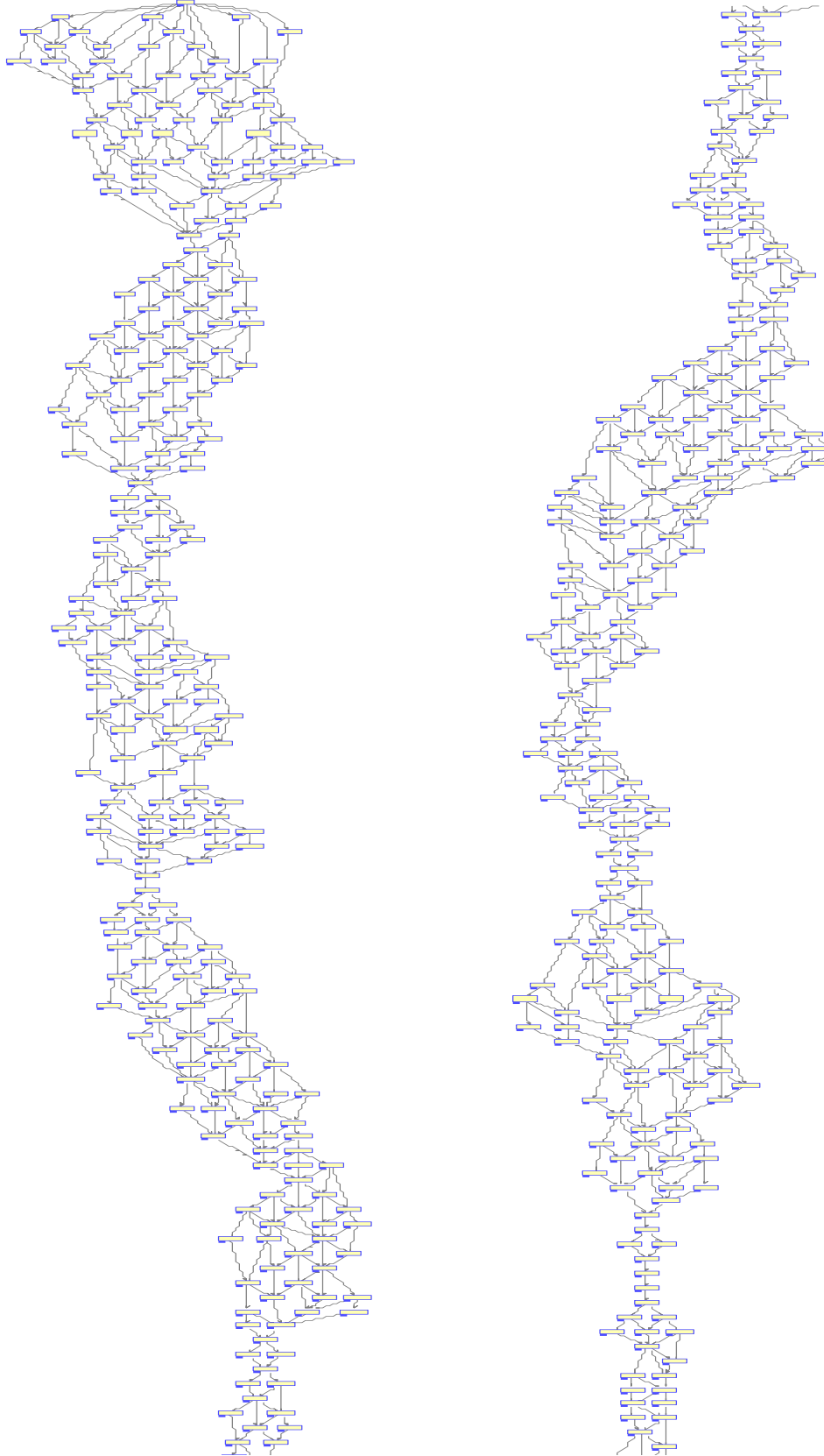


EK 2. 250 düğümlü ağ

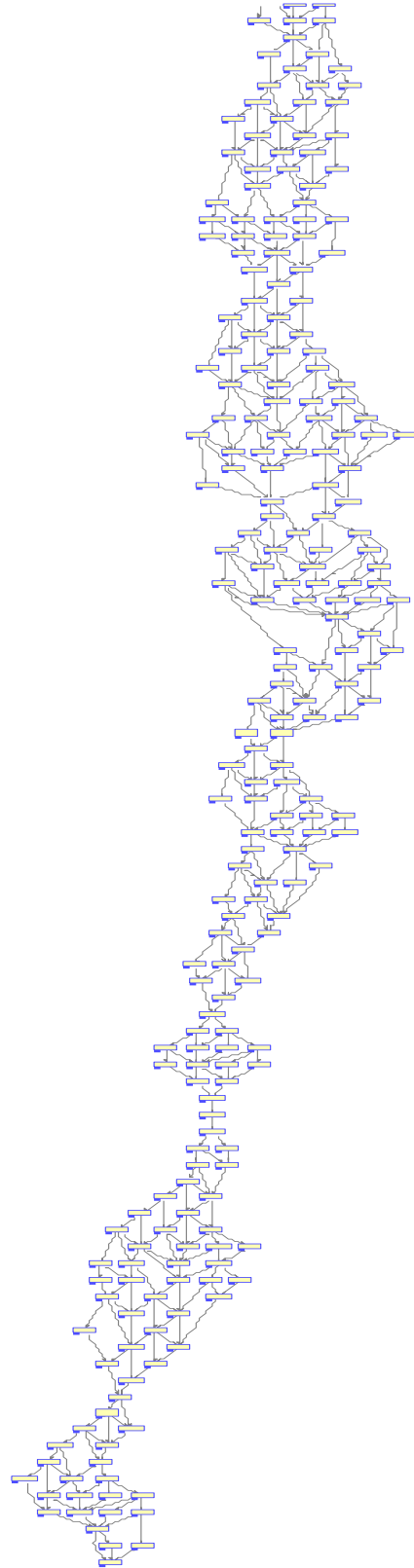
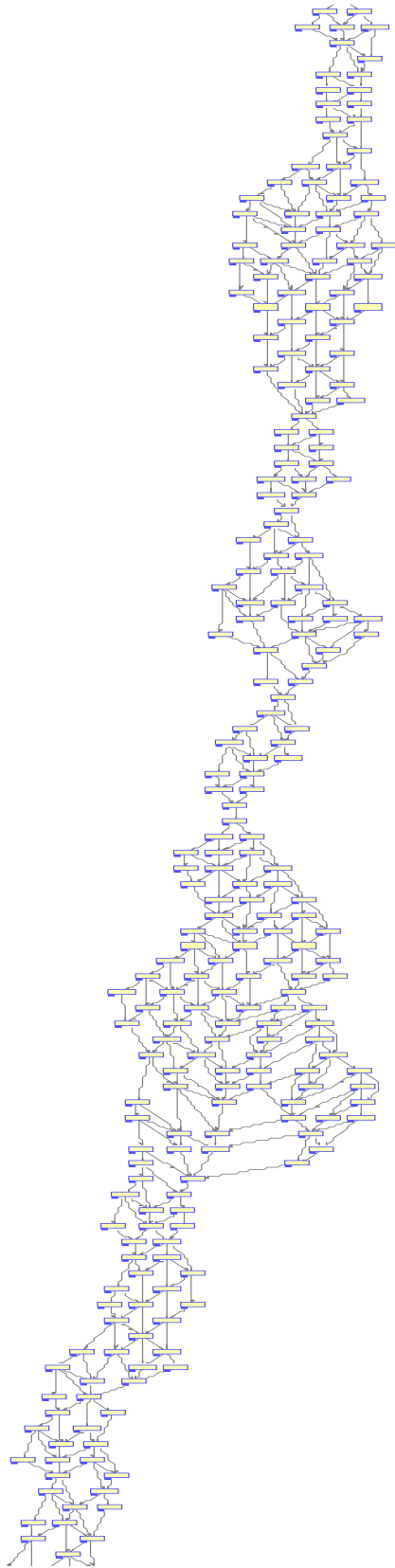


EK 3. 500 düğümlü ağ



EK 4. 1000 düğümlü ağ

EK 4. (devam)



EK 5. 50 düğümlük ağıın kenar mesafeleri(km)

Kenar mesafe (km)					
(1,2)	9	(19,20)	19	(29,37)	12
(1,3)	11	(14,21)	16	(36,37)	17
(2,3)	11	(14,22)	8	(29,38)	5
(1,4)	9	(15,22)	15	(30,38)	5
(1,5)	18	(16,22)	17	(31,38)	8
(1,6)	20	(17,22)	14	(31,39)	14
(1,7)	20	(17,23)	12	(32,39)	7
(6,7)	11	(18,23)	19	(32,40)	9
(2,8)	9	(18,24)	13	(33,40)	10
(2,9)	14	(19,24)	16	(33,41)	7
(3,9)	5	(19,25)	19	(34,41)	20
(4,9)	6	(20,25)	18	(35,42)	6
(5,9)	16	(20,26)	7	(35,43)	8
(8,9)	20	(25,26)	9	(36,43)	18
(5,10)	14	(21,27)	16	(36,44)	9
(6,10)	6	(21,28)	10	(37,44)	19
(6,11)	7	(22,28)	13	(43,44)	8
(7,11)	6	(27,28)	5	(37,45)	15
(10,11)	20	(22,29)	14	(38,45)	13
(7,12)	11	(23,29)	11	(38,46)	9
(7,13)	7	(23,30)	10	(39,46)	7
(8,14)	7	(24,30)	9	(40,46)	10
(8,15)	5	(24,31)	7	(40,47)	19
(9,15)	14	(25,31)	13	(41,47)	17
(9,16)	12	(25,32)	16	(41,48)	18
(10,16)	12	(26,32)	10	(47,48)	15
(10,17)	11	(31,32)	15	(41,49)	17
(11,17)	5	(26,33)	19	(42,50)	10
(11,18)	11	(32,33)	14	(43,50)	6
(12,18)	7	(26,34)	18	(44,50)	6
(12,19)	11	(27,35)	8	(45,50)	14
(13,19)	14	(27,36)	14	(46,50)	11
(18,19)	13	(28,36)	10	(48,50)	16
(13,20)	7	(28,37)	19	(49,50)	14

EK 6. 50 düğümlük ağı kenar zaman dağılım bilgileri (dk)

Kenar	Zaman (dk)	Kenar	Zaman (dk)	Kenar	Zaman (dk)
(1,2)	ucgen(9,11,10)	(14,21)	unif(19,22)	(29,38)	normrnd(6,0.5)
(1,3)	ucgen(20,22,21)	(14,22)	ucgen(5,7,6)	(30,38)	normrnd(6,0.5)
(2,3)	unif(10,13)	(15,22)	ucgen(30,32,31)	(31,38)	ucgen(5,8,7)
(1,4)	ucgen(8,11,10)	(16,22)	normrnd(24,2)	(31,39)	unif(16,17)
(1,5)	normrnd(22,2)	(17,22)	ucgen(32,34,33)	(32,39)	unif(6,8)
(1,6)	unif(15,18)	(17,23)	unif(8,11)	(32,40)	unif(7,8)
(1,7)	ucgen(13,16,14)	(18,23)	normrnd(25,2)	(33,40)	unif(40,42)
(6,7)	normrnd(13,1)	(18,24)	normrnd(18,1)	(33,41)	unif(11,13)
(2,8)	unif(11,12)	(19,24)	ucgen(23,26,24)	(34,41)	ucgen(35,37,36)
(2,9)	ucgen(10,13,12)	(19,25)	normrnd(25,2)	(35,42)	unif(4,5)
(3,9)	normrnd(8,0.5)	(20,25)	unif(26,28)	(35,43)	ucgen(8,11,10)
(4,9)	normrnd(8,0.5)	(20,26)	ucgen(8,10,9)	(36,43)	ucgen(11,13,12)
(5,9)	ucgen(24,26,25)	(25,26)	unif(9,11)	(36,44)	normrnd(15,1)
(8,9)	normrnd(24,2)	(21,27)	ucgen(36,38,37)	(37,44)	ucgen(11,13,12)
(5,10)	normrnd(22,2)	(21,28)	unif(18,19)	(43,44)	unif(20,21)
(6,10)	ucgen(10,13,11)	(22,28)	normrnd(18,1)	(37,45)	normrnd(23,2)
(6,11)	ucgen(5,8,6)	(27,28)	normrnd(7,0.5)	(38,45)	normrnd(21,2)
(7,11)	normrnd(8,0.5)	(22,29)	normrnd(20,2)	(38,46)	unif(19,20)
(10,11)	normrnd(26,2)	(23,29)	ucgen(39,41,40)	(39,46)	normrnd(9,0.5)
(7,12)	ucgen(15,18,16)	(23,30)	ucgen(11,14,13)	(40,46)	ucgen(11,13,12)
(7,13)	normrnd(9,0.5)	(24,30)	normrnd(13,1)	(40,47)	unif(14,17)
(8,14)	ucgen(5,7,6)	(24,31)	ucgen(9,11,10)	(41,47)	ucgen(13,15,14)
(8,15)	normrnd(8,0.5)	(25,31)	normrnd(22,2)	(41,48)	unif(23,26)
(9,15)	ucgen(22,25,23)	(25,32)	unif(48,51)	(47,48)	unif(26,28)
(9,16)	normrnd(16,1)	(26,32)	ucgen(10,12,11)	(41,49)	ucgen(20,22,21)
(10,16)	normrnd(17,1)	(31,32)	normrnd(23,2)	(42,50)	normrnd(15,1)
(10,17)	ucgen(12,14,13)	(26,33)	unif(11,13)	(43,50)	unif(19,20)
(11,17)	ucgen(9,11,10)	(32,33)	unif(24,26)	(44,50)	normrnd(10,1)
(11,18)	unif(10,12)	(26,34)	ucgen(18,20,19)	(45,50)	normrnd(19,1)
(12,18)	ucgen(6,8,7)	(27,35)	unif(7,10)	(46,50)	ucgen(8,10,9)
(12,19)	normrnd(17,1)	(27,36)	normrnd(20,2)	(48,50)	unif(10,13)
(13,19)	ucgen(38,40,39)	(28,36)	ucgen(6,8,7)	(49,50)	ucgen(10,13,12)
(18,19)	normrnd(16,1)	(28,37)	ucgen(14,17,15)		

(13,20)	ucgen(18,20,19)	(29,37)	unif(12,14)		
(19,20)	ucgen(11,13,12)	(36,37)	ucgen(11,13,12)		

EK 7. 50 düğümlük ağır kenar yakıt bilgileri (desilitre)

Kenar	Normal dağılım parametreleri		Kenar	Normal dağılım parametreleri		Kenar	Normal dağılım parametreleri	
	Ortalama	St.sapma		Ortalama	St.sapma		Ortalama	St.sapma
(1,2)	6.1772	0.0328	(14,21)	10.315	0.0322	(29,38)	3.2535	0.0406
(1,3)	7.2736	0.0193	(14,22)	5.1564	0.02	(30,38)	3.2705	0.0586
(2,3)	7.4102	0.1982	(15,22)	10.0428	0.0232	(31,38)	6.263	0.5721
(1,4)	6.0339	0.143	(16,22)	10.9546	0.0493	(31,39)	9.1262	0.0267
(1,5)	11.7131	0.2076	(17,22)	9.5937	0.0244	(32,39)	4.832	0.1826
(1,6)	15.4877	0.6855	(17,23)	9.6418	0.8008	(32,40)	6.908	0.2097
(1,7)	17.685	0.823	(18,23)	12.2825	0.1318	(33,40)	7.5543	0.0163
(6,7)	7.196	0.1187	(18,24)	8.3578	0.0171	(33,41)	4.5582	0.0273
(2,8)	5.8013	0.0127	(19,24)	10.3044	0.0158	(34,41)	13.1051	0.0199
(2,9)	10.7435	0.5007	(19,25)	12.2814	0.1182	(35,42)	5.0845	0.3553
(3,9)	3.2348	0.0169	(20,25)	11.5838	0.0122	(35,43)	5.1997	0.056
(4,9)	3.8645	0.0168	(20,26)	4.5145	0.0184	(36,43)	17.3193	0.7362
(5,9)	10.3244	0.0119	(25,26)	5.9428	0.0915	(36,44)	5.843	0.0361
(8,9)	13.0337	0.1848	(21,27)	10.9235	0.0236	(37,44)	19.721	0.9736
(5,10)	9.0625	0.0638	(21,28)	6.5783	0.0145	(43,44)	5.5724	0.0151
(6,10)	3.9614	0.0325	(22,28)	8.3574	0.0128	(37,45)	9.6868	0.0608
(6,11)	5.1382	0.3214	(27,28)	3.2178	0.0101	(38,45)	8.4304	0.0731
(7,11)	3.864	0.0139	(22,29)	9.0344	0.0689	(38,46)	6.0757	0.0158
(10,11)	12.9065	0.0884	(23,29)	8.1541	0.0137	(39,46)	4.5203	0.0236
(7,12)	7.0803	0.0157	(23,30)	6.4593	0.0362	(40,46)	6.4972	0.0344
(7,13)	4.5158	0.0276	(24,30)	5.7911	0.0142	(40,47)	14.7014	0.6153
(8,14)	4.6616	0.0219	(24,31)	4.4994	0.0047	(41,47)	13.1389	0.3096
(8,15)	3.2353	0.0163	(25,31)	8.4527	0.076	(41,48)	11.5712	0.0119
(9,15)	9.0909	0.0265	(25,32)	11.5239	0.0376	(47,48)	9.8357	0.0278
(9,16)	7.7304	0.0293	(26,32)	6.6249	0.0761	(41,49)	10.995	0.0232
(10,16)	7.7195	0.0161	(31,32)	9.6851	0.052	(42,50)	6.443	0.0227
(10,17)	7.1703	0.0408	(26,33)	19.7206	1.3004	(43,50)	4.3596	0.0112
(11,17)	3.3303	0.0263	(32,33)	9.1595	0.0282	(44,50)	3.9093	0.0459
(11,18)	7.532	0.177	(26,34)	12.0788	0.0932	(45,50)	9.0062	0.0225
(12,18)	4.8245	0.1338	(27,35)	5.4462	0.2132	(46,50)	8.5913	0.3156
(12,19)	7.0967	0.0271	(27,36)	9.0337	0.065	(48,50)	14.307	1.1691
(13,19)	9.8992	0.0207	(28,36)	6.4368	0.0101	(49,50)	10.7385	0.5257
(18,19)	8.429	0.0698	(28,37)	14.9211	0.5157			
(13,20)	4.9213	0.0201	(29,37)	7.9762	0.1116			
(19,20)	19.6265	0.8444	(36,37)	15.2351	0.5631			

EK 8. 100 düğümlük ağın kenar mesafeleri (km)

Kenar mesafe (km)									
(1,2)	13	(18,26)	15	(39,44)	13	(60,64)	11	(78,83)	19
(1,3)	8	(19,26)	6	(40,45)	14	(61,65)	18	(79,83)	20
(1,4)	14	(25,26)	16	(40,46)	11	(62,65)	10	(82,83)	5
(1,5)	20	(19,27)	15	(41,46)	17	(63,65)	16	(79,84)	14
(1,6)	20	(20,27)	14	(41,47)	6	(63,66)	18	(79,85)	8
(1,7)	18	(21,27)	13	(42,47)	15	(64,66)	16	(80,86)	20
(1,8)	10	(22,28)	11	(46,47)	10	(65,66)	10	(81,86)	9
(1,9)	17	(23,28)	17	(42,48)	17	(65,67)	7	(81,87)	17
(8,9)	16	(23,29)	20	(43,48)	14	(65,68)	6	(82,87)	6
(1,10)	12	(24,29)	7	(43,49)	17	(66,68)	5	(82,88)	14
(9,10)	19	(25,29)	18	(44,49)	20	(67,68)	16	(83,88)	13
(1,11)	14	(25,30)	16	(45,50)	20	(66,69)	9	(84,88)	7
(2,12)	6	(26,30)	16	(46,50)	18	(68,69)	15	(85,88)	18
(2,13)	8	(27,30)	9	(46,51)	20	(66,70)	16	(85,89)	11
(3,13)	9	(29,30)	6	(47,51)	18	(67,71)	12	(88,89)	11
(3,14)	20	(28,31)	11	(48,51)	20	(67,72)	17	(85,90)	9
(4,14)	6	(28,32)	10	(48,52)	20	(68,72)	7	(86,91)	9
(4,15)	16	(29,32)	12	(49,52)	6	(71,72)	13	(86,92)	11
(5,15)	20	(29,33)	19	(50,53)	13	(68,73)	12	(87,92)	14
(5,16)	18	(30,33)	6	(50,54)	6	(69,73)	6	(87,93)	18
(6,16)	15	(30,34)	17	(51,54)	18	(72,73)	10	(88,93)	11
(7,16)	8	(30,35)	9	(53,54)	20	(69,74)	9	(88,94)	5
(8,16)	8	(31,36)	7	(51,55)	13	(70,74)	6	(89,94)	6
(8,17)	18	(32,36)	14	(52,55)	16	(70,75)	15	(90,94)	17
(9,17)	10	(33,36)	9	(54,55)	12	(71,76)	18	(93,94)	20
(9,18)	6	(33,37)	20	(53,56)	16	(71,77)	10	(90,95)	20
(10,18)	8	(34,37)	15	(53,57)	19	(72,77)	11	(90,96)	14
(10,19)	18	(34,38)	8	(54,57)	9	(76,77)	7	(91,97)	16
(11,19)	19	(35,38)	17	(56,57)	18	(72,78)	16	(91,98)	8
(11,20)	17	(37,38)	17	(54,58)	17	(73,78)	8	(92,98)	10
(11,21)	20	(35,39)	12	(55,58)	18	(74,78)	14	(93,98)	5
(12,22)	13	(36,40)	17	(55,59)	7	(77,78)	5	(97,98)	8
(12,23)	6	(37,40)	14	(55,60)	5	(74,79)	18	(93,99)	7
(13,23)	17	(37,41)	9	(59,60)	12	(75,79)	8	(94,99)	8
(13,24)	13	(38,41)	12	(56,61)	14	(78,79)	15	(95,99)	12
(14,24)	11	(40,41)	15	(56,62)	16	(76,80)	20	(96,99)	17
(15,24)	10	(38,42)	8	(57,62)	19	(76,81)	18	(97,100)	8
(16,24)	12	(39,42)	6	(57,63)	10	(77,81)	9	(98,100)	15
(17,24)	7	(41,42)	12	(58,63)	16	(77,82)	10	(99,100)	9
(17,25)	18	(39,43)	20	(59,63)	10	(78,82)	5		

(18,25)	5	(42,43)	12	(59,64)	6	(81,82)	14	
---------	---	---------	----	---------	---	---------	----	--

EK 9. 100 düğümlük ağın kenar zaman dağılım bilgileri (dk)

Kenar	Zaman dağılımı	Kenar	Zaman dağılımı	Kenar	Zaman dağılımı
(1,2)	normrnd(17,1)	(19,26)	normrnd(9,0.5)	(40,46)	normrnd(14,1)
(1,3)	normrnd(10,1)	(25,26)	unif(14,18)	(41,46)	ucgen(15,19,17)
(1,4)	normrnd(17,1)	(19,27)	normrnd(17,1)	(41,47)	ucgen(4,6,5)
(1,5)	normrnd(30,2)	(20,27)	unif(19,23)	(42,47)	unif(14,17)
(1,6)	ucgen(24,26,25)	(21,27)	unif(21,25)	(46,47)	normrnd(15,1)
(1,7)	ucgen(10,12,11)	(22,28)	normrnd(15,1)	(42,48)	ucgen(24,28,27)
(1,8)	ucgen(11,14,13)	(23,28)	normrnd(20,1)	(43,48)	unif(20,24)
(1,9)	ucgen(22,24,23)	(23,29)	ucgen(12,14,13)	(43,49)	normrnd(23,2)
(8,9)	ucgen(23,27,25)	(24,29)	ucgen(4,6,5)	(44,49)	ucgen(15,18,16)
(1,10)	unif(7,10)	(25,29)	unif(11,13)	(45,50)	normrnd(26,2)
(9,10)	normrnd(30,2)	(25,30)	ucgen(9,12,10)	(46,50)	normrnd(23,1)
(1,11)	unif(19,21)	(26,30)	normrnd(25,2)	(46,51)	ucgen(12,14,13)
(2,12)	normrnd(8,0.5)	(27,30)	ucgen(6,8,7)	(47,51)	unif(12,16)
(2,13)	normrnd(11,1)	(29,30)	ucgen(5,7,6)	(48,51)	unif(92,95)
(3,13)	ucgen(7,10,9)	(28,31)	unif(11,13)	(48,52)	ucgen(12,14,13)
(3,14)	normrnd(28,2)	(28,32)	normrnd(15,1)	(49,52)	ucgen(5,7,6)
(4,14)	normrnd(9,0.5)	(29,32)	unif(8,10)	(50,53)	ucgen(8,10,9)
(4,15)	normrnd(19,1)	(29,33)	normrnd(26,2)	(50,54)	normrnd(9,0.5)
(5,15)	unif(20,24)	(30,33)	normrnd(9,0.5)	(51,54)	ucgen(10,12,11)
(5,16)	unif(23,26)	(30,34)	normrnd(24,2)	(53,54)	unif(63,66)
(6,16)	normrnd(20,2)	(30,35)	ucgen(6,9,7)	(51,55)	unif(29,33)
(7,16)	ucgen(7,9,8)	(31,36)	normrnd(10,0.5)	(52,55)	normrnd(21,2)
(8,16)	unif(5,9)	(32,36)	normrnd(18,1)	(54,55)	ucgen(7,10,8)
(8,17)	ucgen(14,16,15)	(33,36)	normrnd(16,1)	(53,56)	unif(37,40)
(9,17)	unif(8,12)	(33,37)	unif(27,30)	(53,57)	unif(39,42)
(9,18)	unif(5,7)	(34,37)	normrnd(19,1)	(54,57)	ucgen(5,7,6)
(10,18)	normrnd(12,0.7)	(34,38)	ucgen(7,9,8)	(56,57)	normrnd(19,1)
(10,19)	normrnd(31,2)	(35,38)	normrnd(26,2)	(54,58)	normrnd(19,1)
(11,19)	unif(14,16)	(37,38)	normrnd(20,1)	(55,58)	unif(11,15)
(11,20)	normrnd(21,2)	(35,39)	normrnd(16,1)	(55,59)	unif(8,12)
(11,21)	ucgen(15,17,16)	(36,40)	unif(12,16)	(55,60)	ucgen(4,7,6)
(12,22)	normrnd(17,1)	(37,40)	normrnd(20,2)	(59,60)	ucgen(7,9,8)
(12,23)	normrnd(8,0.5)	(37,41)	ucgen(13,17,16)	(56,61)	ucgen(11,15,12)
(13,23)	unif(12,16)	(38,41)	normrnd(16,1)	(56,62)	normrnd(19,1)
(13,24)	ucgen(17,20,18)	(40,41)	normrnd(19,1)	(57,62)	normrnd(25,2)
(14,24)	ucgen(7,9,8)	(38,42)	normrnd(10,1)	(57,63)	normrnd(15,1)
(15,24)	normrnd(15,1)	(39,42)	normrnd(12,1)	(58,63)	ucgen(14,17,16)
(16,24)	unif(65,68)	(41,42)	normrnd(16,1)	(59,63)	ucgen(6,8,7)
(17,24)	unif(6,9)	(39,43)	unif(13,16)	(59,64)	ucgen(4,6,5)
(17,25)	ucgen(21,25,24)	(42,43)	ucgen(7,11,10)	(60,64)	normrnd(14,1)
(18,25)	unif(3,4)	(39,44)	unif(11,15)	(61,65)	ucgen(11,13,12)
(18,26)	normrnd(31,2)	(40,45)	normrnd(18,1)	(62,65)	normrnd(16,1)

EK 9. (devam)

Kenar	Zaman dağılımı	Kenar	Zaman dağılımı
(63,65)	normrnd(23,2)	(79,83)	unif(29,33)
(63,66)	unif(15,17)	(82,83)	unif(3,4)
(64,66)	normrnd(19,1)	(79,84)	unif(76,79)
(65,66)	unif(9,12)	(79,85)	ucgen(5,9,8)
(65,67)	unif(4,5)	(80,86)	ucgen(19,23,22)
(65,68)	unif(7,9)	(81,86)	unif(8,12)
(66,68)	ucgen(3,5,4)	(81,87)	unif(11,14)
(67,68)	normrnd(18,1)	(82,87)	normrnd(9,0.5)
(66,69)	ucgen(6,8,7)	(82,88)	ucgen(23,26,25)
(68,69)	ucgen(11,15,13)	(83,88)	normrnd(16,1)
(66,70)	ucgen(10,14,12)	(84,88)	normrnd(10,0.5)
(67,71)	normrnd(15,1)	(85,88)	ucgen(12,15,14)
(67,72)	unif(78,82)	(85,89)	unif(7,11)
(68,72)	unif(16,20)	(88,89)	ucgen(7,9,8)
(71,72)	ucgen(8,10,9)	(85,90)	unif(34,37)
(68,73)	unif(11,15)	(86,91)	normrnd(13,1)
(69,73)	ucgen(10,12,11)	(86,92)	normrnd(15,1)
(72,73)	unif(9,11)	(87,92)	unif(8,9)
(69,74)	normrnd(13,1)	(87,93)	unif(11,15)
(70,74)	normrnd(10,1)	(88,93)	unif(15,18)
(70,75)	ucgen(14,16,15)	(88,94)	unif(9,13)
(71,76)	normrnd(23,2)	(89,94)	normrnd(9,0.5)
(71,77)	unif(33,37)	(90,94)	normrnd(20,1)
(72,77)	unif(14,16)	(93,94)	ucgen(25,29,27)
(76,77)	normrnd(12,1)	(90,95)	ucgen(22,26,25)
(72,78)	unif(12,15)	(90,96)	normrnd(18,1)
(73,78)	unif(7,11)	(91,97)	normrnd(19,1)
(74,78)	normrnd(18,1)	(91,98)	normrnd(13,1)
(77,78)	ucgen(7,9,8)	(92,98)	normrnd(15,1)
(74,79)	normrnd(20,1)	(93,98)	normrnd(8,0.5)
(75,79)	normrnd(13,1)	(97,98)	unif(6,9)
(78,79)	normrnd(19,1)	(93,99)	ucgen(6,8,7)
(76,80)	unif(33,36)	(94,99)	unif(7,10)
(76,81)	normrnd(26,2)	(95,99)	normrnd(16,1)
(77,81)	unif(6,9)	(96,99)	unif(14,18)
(77,82)	normrnd(14,1)	(97,100)	normrnd(10,0.7)
(78,82)	normrnd(8,0.5)	(98,100)	normrnd(19,1)
(81,82)	ucgen(9,12,10)	(99,100)	normrnd(15,1)]
(78,83)	normrnd(24,2)		

EK 10. 100 düğümlük ağı kenar yakıt bilgileri (desilitre)

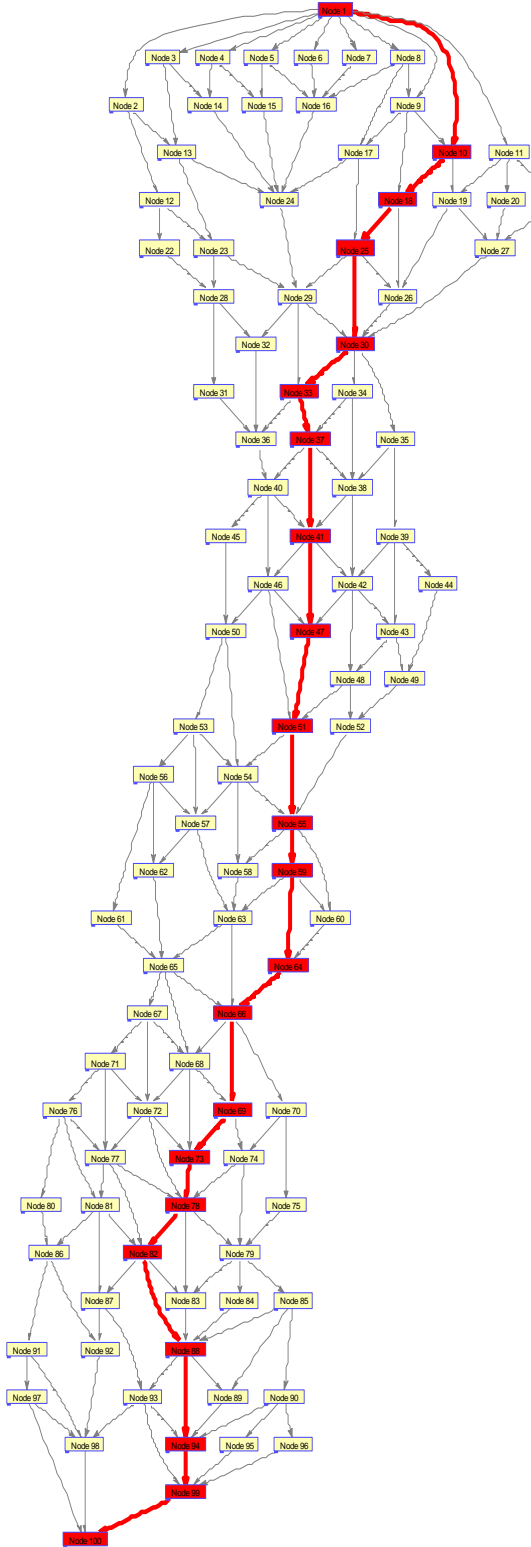
Kenar	Normal dağılım parametreleri		Kenar	Normal dağılım parametreleri		Kenar	Normal dağılım parametreleri	
	Ortalama	St. sapma		Ortalama	St. sapma		Ortalama	St. sapma
(1,2)	8.3787	0.0346	(15,24)	6.4402	0.0246	(36,40)	13.3238	0.9574
(1,3)	5.1992	0.0889	(16,24)	9.4565	0.0149	(37,40)	9.0286	0.0442
(1,4)	9.0839	0.0698	(17,24)	4.757	0.2114	(37,41)	5.8532	0.0364
(1,5)	12.8888	0.0533	(17,25)	11.5957	0.0336	(38,41)	7.7246	0.0311
(1,6)	12.9197	0.0241	(18,25)	4.5523	0.4087	(40,41)	9.6983	0.0692
(1,7)	19.347	1.005	(18,26)	10.0327	0.1201	(38,42)	5.2082	0.0809
(1,8)	6.4577	0.0317	(19,26)	3.8644	0.0118	(39,42)	3.9902	0.0535
(1,9)	10.9239	0.0054	(25,26)	11.0046	0.3639	(41,42)	7.7272	0.029
(8,9)	10.3251	0.028	(19,27)	9.8613	0.1502	(39,43)	17.3537	1.1251
(1,10)	10.8205	1.3207	(20,27)	9.0153	0.0247	(42,43)	9.9779	1.0332
(9,10)	12.2741	0.0647	(21,27)	8.506	0.0537	(39,44)	9.0467	0.4084
(1,11)	8.9955	0.0051	(22,28)	7.0791	0.0219	(40,45)	9.0329	0.0397
(2,12)	3.8635	0.0135	(23,28)	11.0938	0.1143	(40,46)	7.1129	0.0611
(2,13)	5.1571	0.0276	(23,29)	19.8022	0.7919	(41,46)	11.6721	0.2431
(3,13)	6.3148	0.2507	(24,29)	6.2237	0.5807	(41,47)	4.6855	0.3479
(3,14)	12.8665	0.0325	(25,29)	17.141	1.0248	(42,47)	10.142	0.2155
(4,14)	3.8638	0.0102	(25,30)	16.0843	1.2482	(46,47)	6.4416	0.0235
(4,15)	10.4384	0.1132	(26,30)	10.3431	0.0582	(42,48)	10.9604	0.0261
(5,15)	13.3032	0.1977	(27,30)	7.3577	0.3624	(43,48)	9.0431	0.0381
(5,16)	11.5714	0.0126	(29,30)	4.1382	0.1319	(43,49)	10.9561	0.0589
(6,16)	9.6876	0.1172	(28,31)	7.3108	0.103	(44,49)	15.5308	0.5077
(7,16)	5.5136	0.1361	(28,32)	6.446	0.0251	(45,50)	12.9136	0.1098
(8,16)	6.133	0.7747	(29,32)	10.0882	0.6697	(46,50)	11.6154	0.0457
(8,17)	13.7294	0.2872	(29,33)	12.2421	0.0519	(46,51)	19.8824	0.873
(9,17)	6.9595	0.3798	(30,33)	3.8633	0.01	(47,51)	15.1465	1.1561
(9,18)	4.1533	0.1881	(30,34)	10.9462	0.0342	(48,51)	15.407	0.019
(10,18)	5.155	0.0163	(30,35)	7.0448	0.5297	(48,52)	19.756	0.7678
(10,19)	11.7259	0.0831	(31,36)	4.4996	0.0046	(49,52)	4.1008	0.1083
(11,19)	15.1457	0.5226	(32,36)	9.0384	0.0548	(50,53)	12.0212	0.7363
(11,20)	11.0681	0.2399	(33,36)	5.8895	0.0464	(50,54)	3.8636	0.0101
(11,21)	15.832	0.3357	(33,37)	12.8515	0.0069	(51,54)	19.4389	1.0017
(12,22)	8.3841	0.0442	(34,37)	9.6953	0.0586	(53,54)	14.5177	0.0341
(12,23)	3.8641	0.0162	(34,38)	5.5025	0.1424	(51,55)	8.9387	0.0567
(13,23)	13.3092	0.8629	(35,38)	10.9733	0.0544	(52,55)	10.362	0.1482
(13,24)	8.3533	0.0058	(37,38)	11.0964	0.106	(54,55)	11.0556	0.9221

(14,24)	9.6453	0.4979	(35,39)	7.7292	0.0331	(53,56)	11.0084	0.0471
---------	--------	--------	---------	--------	--------	---------	---------	--------

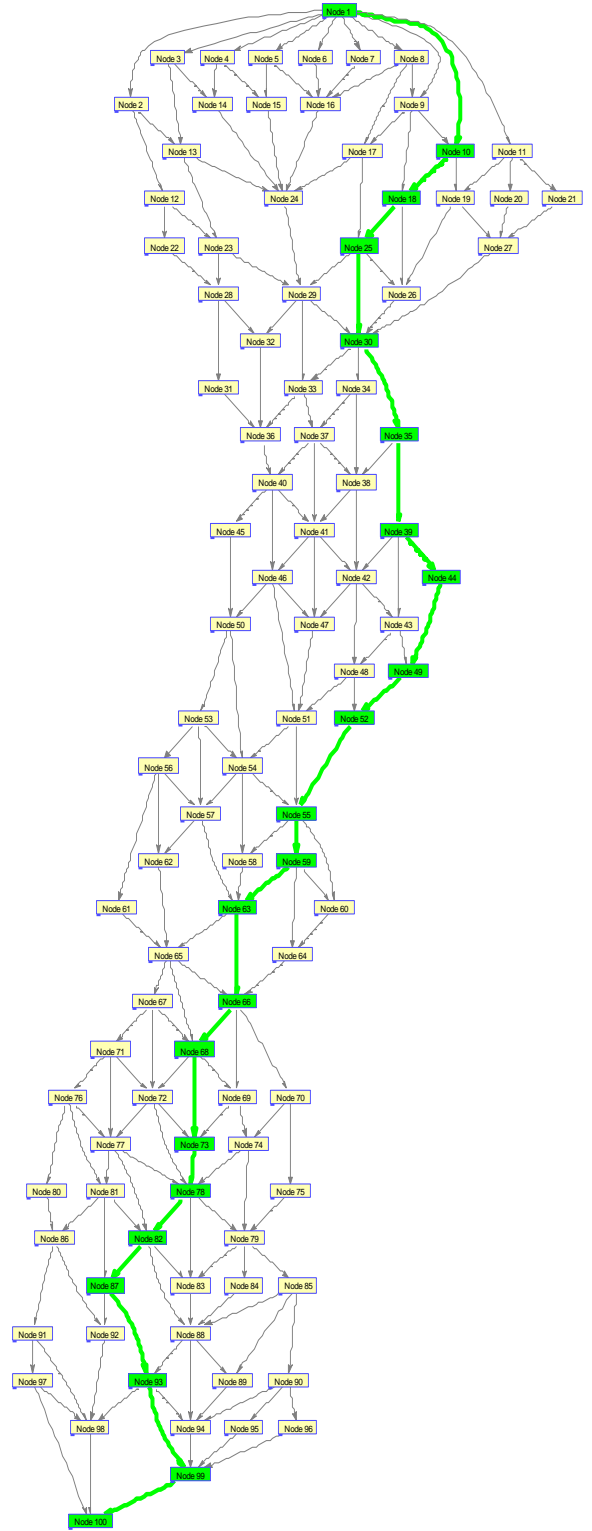
EK 10. (devam)

Kenar	Normal dağılım parametreleri		Kenar	Normal dağılım parametreleri		Kenar	Normal dağılım parametreleri	
	Ortalama	St. sapma		Ortalama	St. sapma		Ortalama	St. sapma
(53,57)	12.7903	0.0513	(70,74)	3.8958	0.0369	(87,92)	15.2663	0.7685
(54,57)	8.6324	0.6797	(70,75)	10.2471	0.1165	(87,93)	15.7935	1.4465
(56,57)	12.0948	0.2117	(71,76)	11.652	0.1161	(88,93)	7.0833	0.0192
(54,58)	11.2599	0.1748	(71,77)	7.3671	0.0444	(88,94)	3.3811	0.0665
(55,58)	15.93	1.6516	(72,77)	7.0707	0.0073	(89,94)	3.8649	0.0132
(55,59)	4.5232	0.0252	(76,77)	4.5639	0.0467	(90,94)	11.0941	0.1048
(55,60)	3.3264	0.121	(72,78)	12.2663	0.6145	(93,94)	12.8544	0.0125
(59,60)	11.5388	0.7344	(73,78)	5.3276	0.2121	(90,95)	12.9733	0.0726
(56,61)	10.162	0.4221	(74,78)	9.0407	0.0534	(90,96)	9.0316	0.0506
(56,62)	10.4233	0.1005	(77,78)	3.2352	0.0145	(91,97)	10.4226	0.0945
(57,62)	12.2556	0.0728	(74,79)	11.8667	0.135	(91,98)	5.1801	0.0334
(57,63)	6.4441	0.0237	(75,79)	5.1868	0.0312	(92,98)	6.4409	0.0246
(58,63)	11.0362	0.2045	(78,79)	9.6798	0.0423	(93,98)	3.2351	0.0154
(59,63)	9.0667	0.6595	(76,80)	13.0344	0.0388	(97,98)	5.7571	0.4267
(59,64)	4.6213	0.3073	(76,81)	11.5939	0.0428	(93,99)	4.8154	0.1347
(60,64)	7.1182	0.0625	(77,81)	7.018	0.6927	(94,99)	5.4037	0.194
(61,65)	17.3749	0.7121	(77,82)	6.4338	0.0171	(95,99)	7.7278	0.0357
(62,65)	6.4757	0.0358	(78,82)	3.2344	0.0164	(96,99)	12.0501	0.4873
(63,65)	10.3087	0.042	(81,82)	12.0008	0.7569	(97,100)	5.1861	0.0546
(63,66)	13.1773	0.3238	(78,83)	12.3385	0.17	(98,100)	9.6999	0.0638
(64,66)	10.4198	0.0856	(79,83)	12.9007	0.0345	(99,100)	5.844	0.0396
(65,66)	6.7859	0.2133	(82,83)	4.4676	0.4217			
(65,67)	7.0504	0.5612	(79,84)	11.0286	0.0164			
(65,68)	3.8703	0.0173	(79,85)	5.7727	0.5165			
(66,68)	4.0124	0.4244	(80,86)	13.3766	0.1775			
(67,68)	10.5476	0.1579	(81,86)	6.0318	0.2462			
(66,69)	7.3179	0.3797	(81,87)	14.737	1.0754			
(68,69)	11.1313	0.4915	(82,87)	3.8626	0.0109			
(66,70)	13.7302	1.0589	(82,88)	9.1466	0.0304			
(67,71)	7.7742	0.0757	(83,88)	8.435	0.0841			
(67,72)	13.1069	0.0253	(84,88)	4.5006	0.0055			
(68,72)	4.8737	0.0618	(85,88)	14.9278	0.6849			
(71,72)	12.0642	0.6826	(85,89)	8.8447	1.0271			
(68,73)	8.0544	0.2439	(88,89)	9.646	0.5208			
(69,73)	3.9382	0.0205	(85,90)	6.763	0.0243			
(72,73)	6.8631	0.1766	(86,91)	5.7962	0.0307			
(69,74)	5.7934	0.0166	(86,92)	7.083	0.0277			

EK 11. 100 düğümlü ağlar üzerinde EKY'ların gösterilmesi

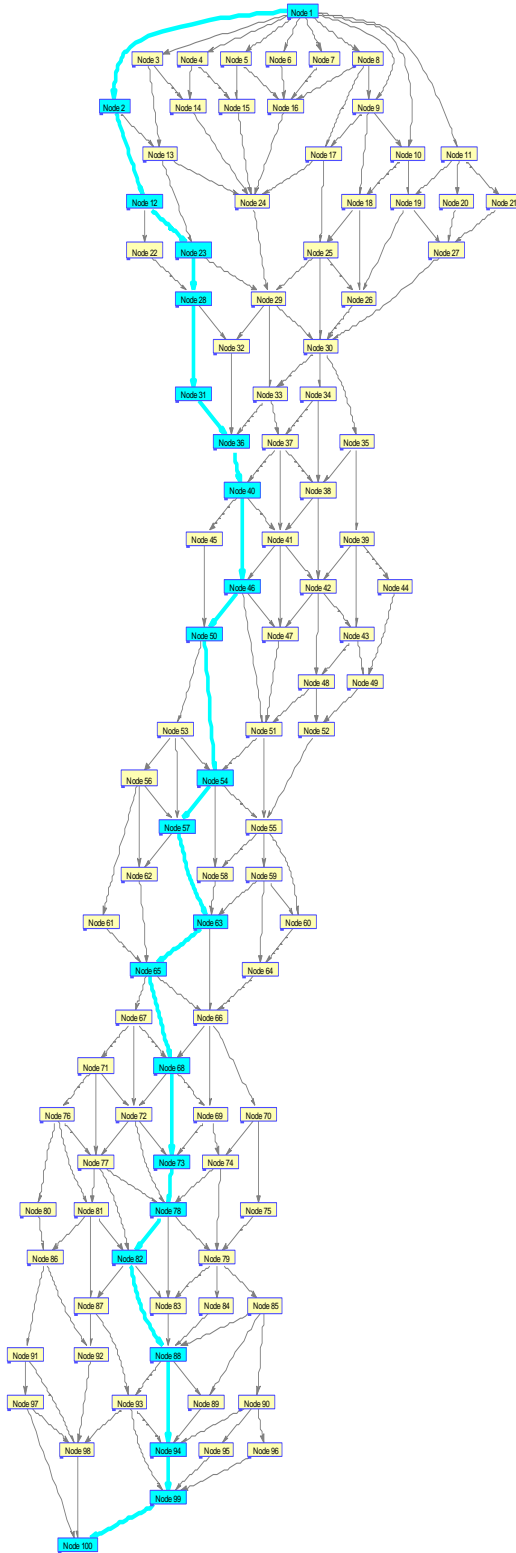


a) En kısa mesafeli yol (Rota 1)

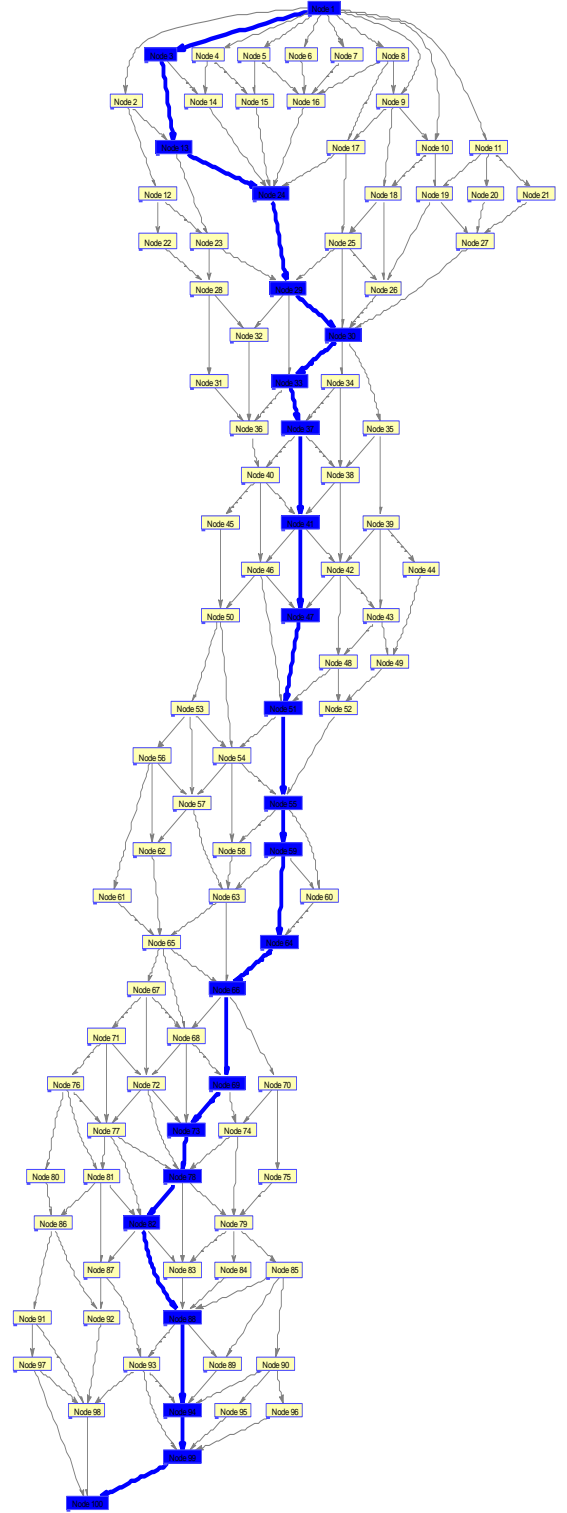


b) Ortalama en kısa seyahat süreli yol ve 230 dk'nın altında gitme ihtimali en yüksek yol (Rota 2)

EK 11. (devam)

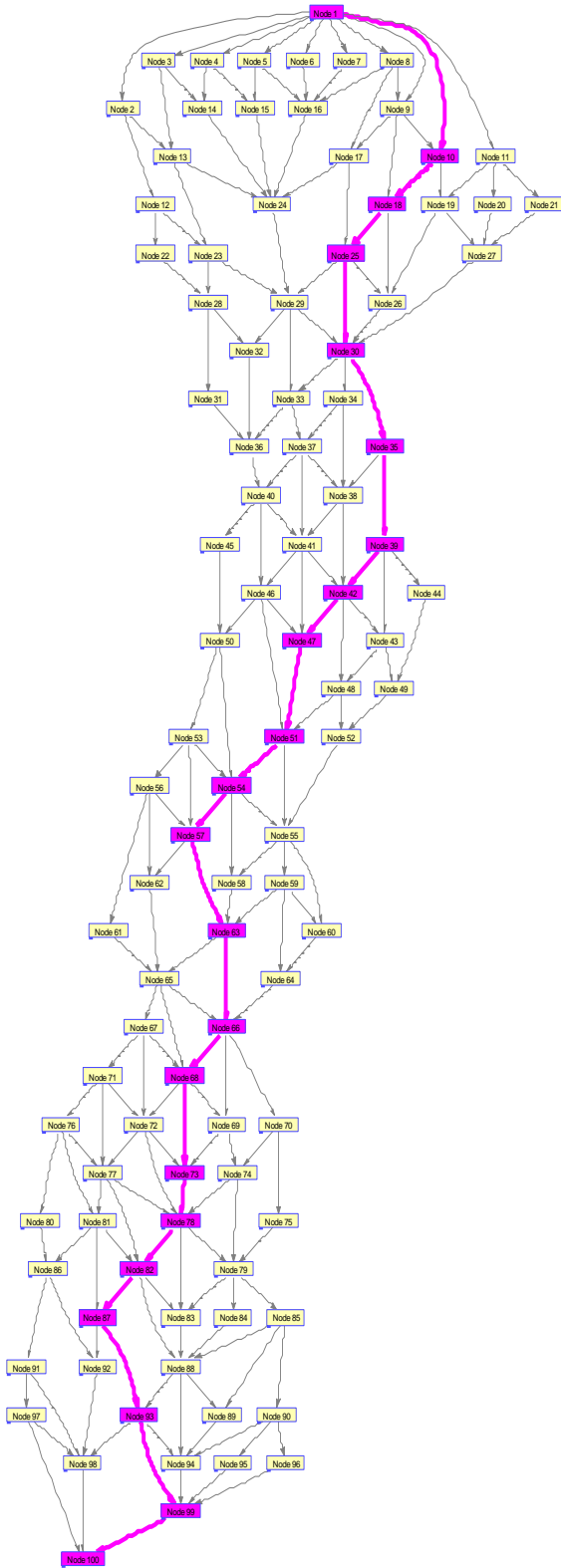


c) Ortalama en az yakıt tüketilen yol (Rota 3)

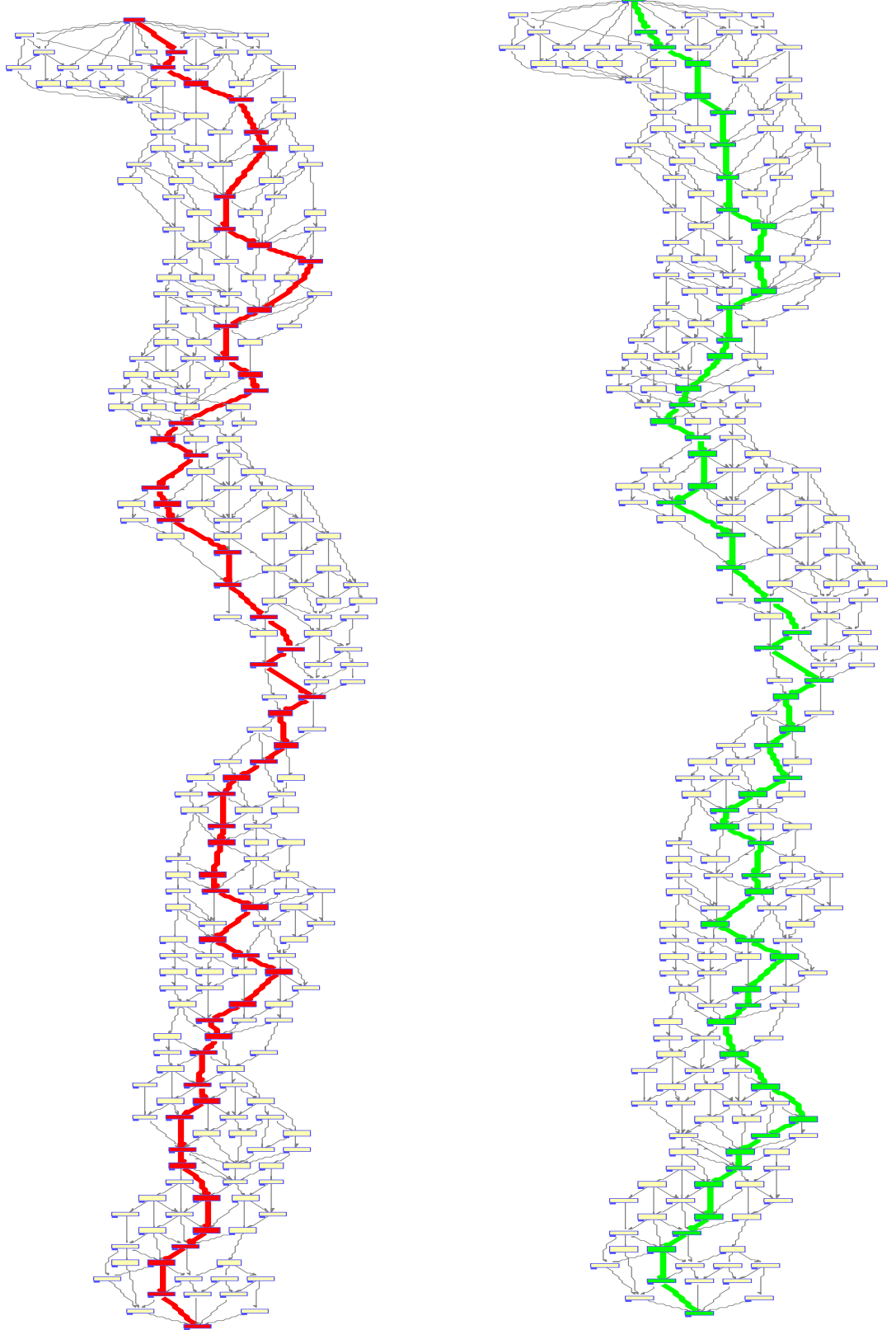


d) 14,5 lt nin altında yakıt tüketme ihtimali en yüksek yol ve %90 ihtimalle en az yakıt tüketilen yol (Rota4)

EK 11. (devam)

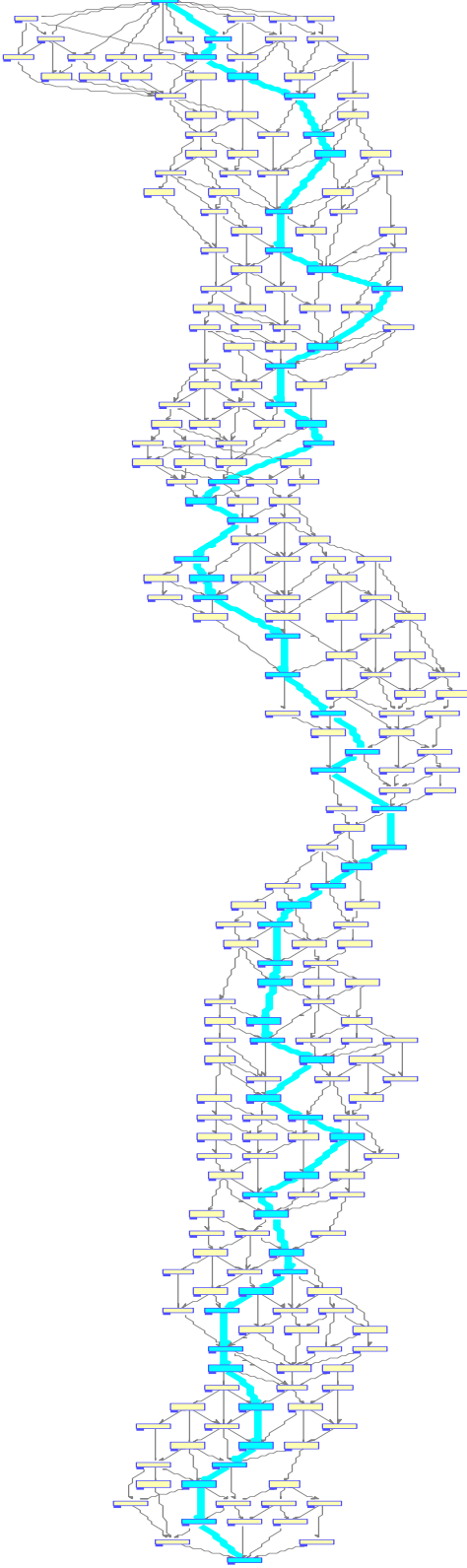


e) %90 ihtimalle en az seyahat süreli yol (Rota 5)

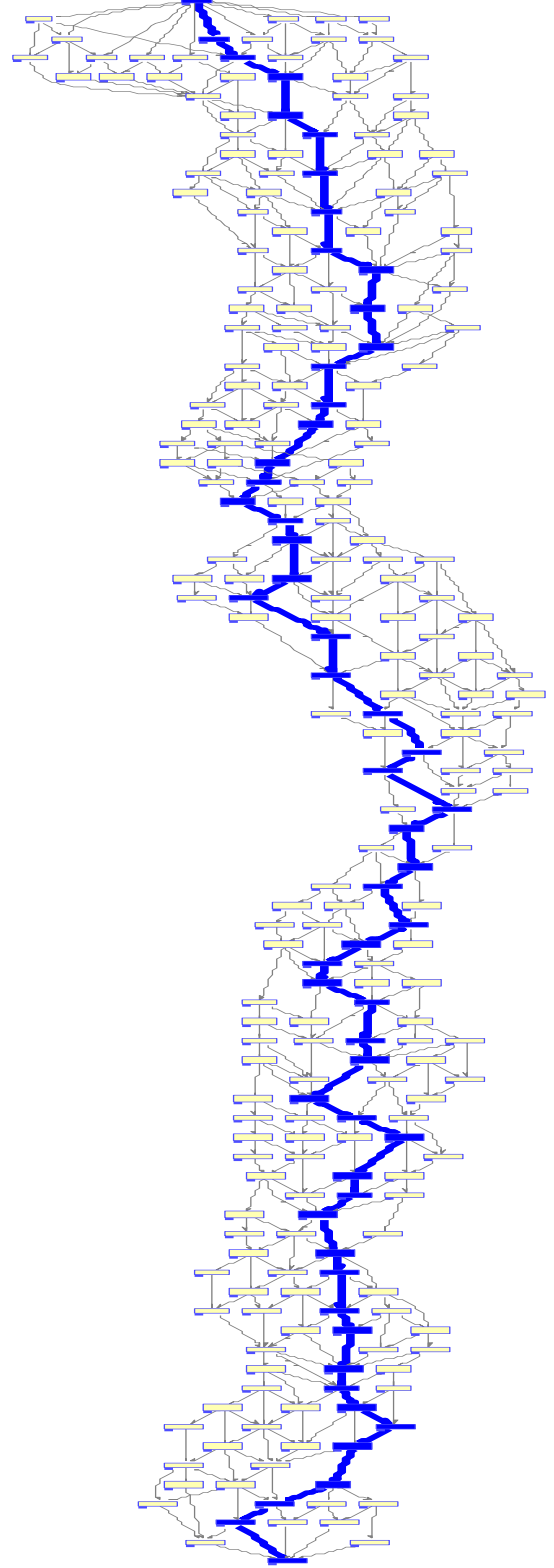
EK 12. 250 düğümlü ağlar üzerinde EKY'ların gösterilmesi

a) En kısa mesafeli yol (Rota 1)

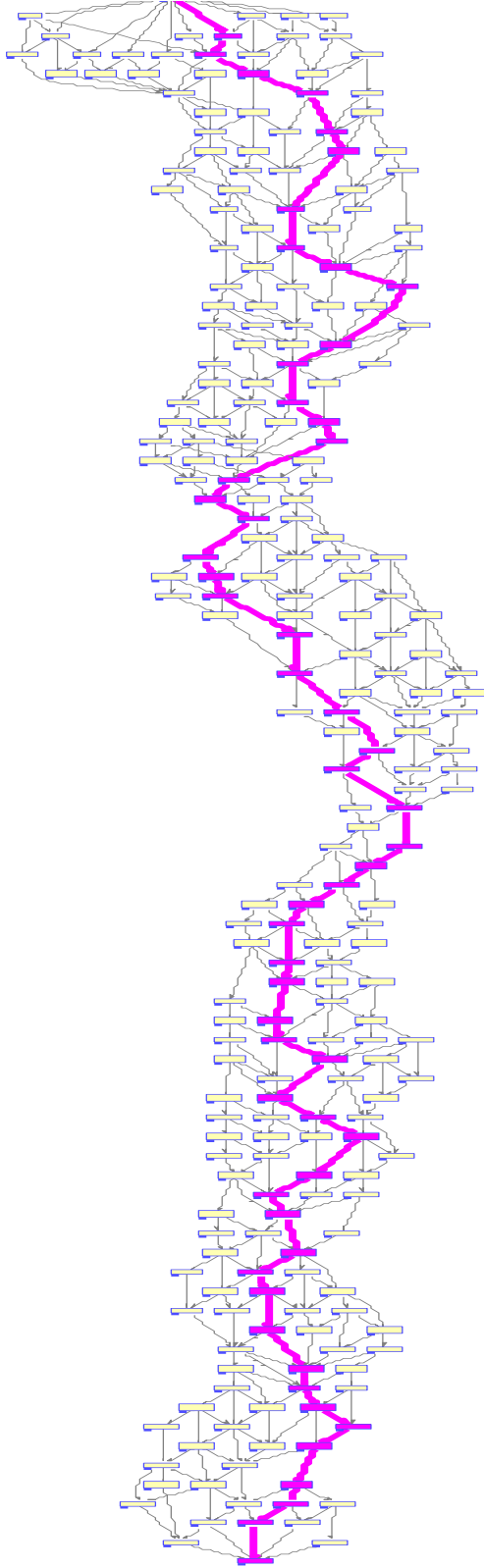
b) Ortalama en kısa seyahat süreli yol (Rota 2)

EK 12. (devam)

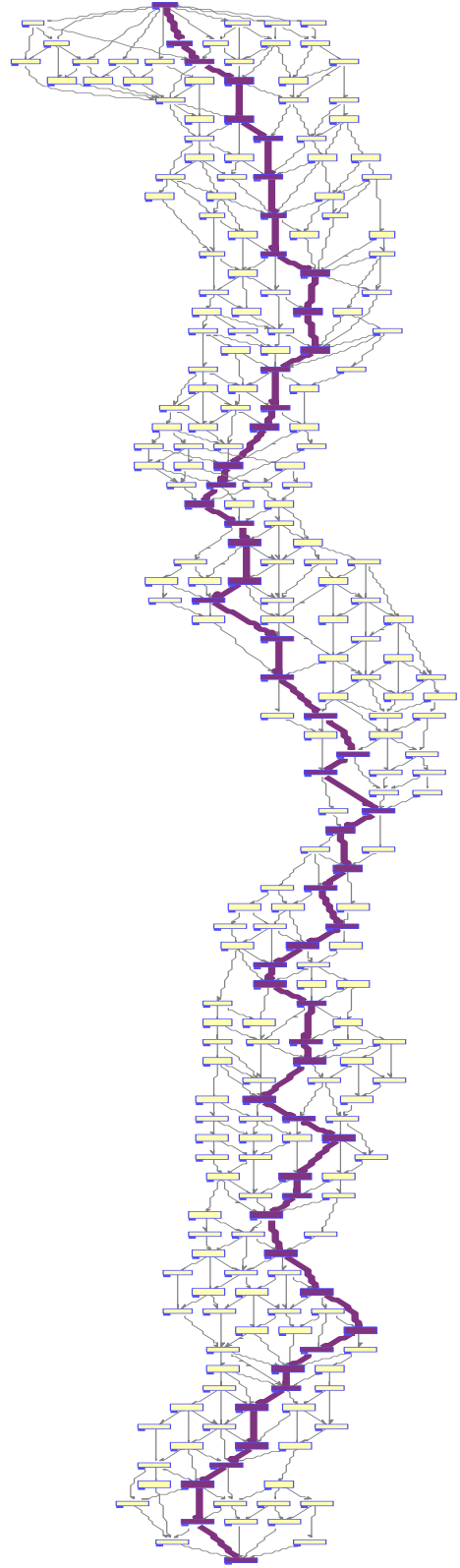
c) Ortalama en az yakıt tüketilen yol (Rota 3)



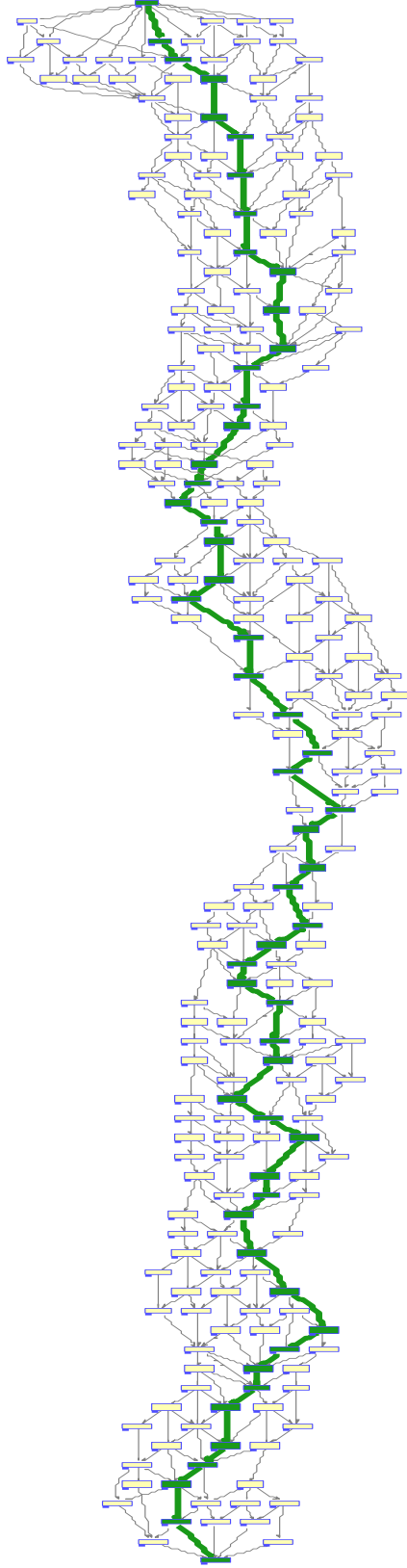
d) 535 dk'nın altında gitme ihtimali en yüksek yol (Rota 4)

EK 12. (devam)

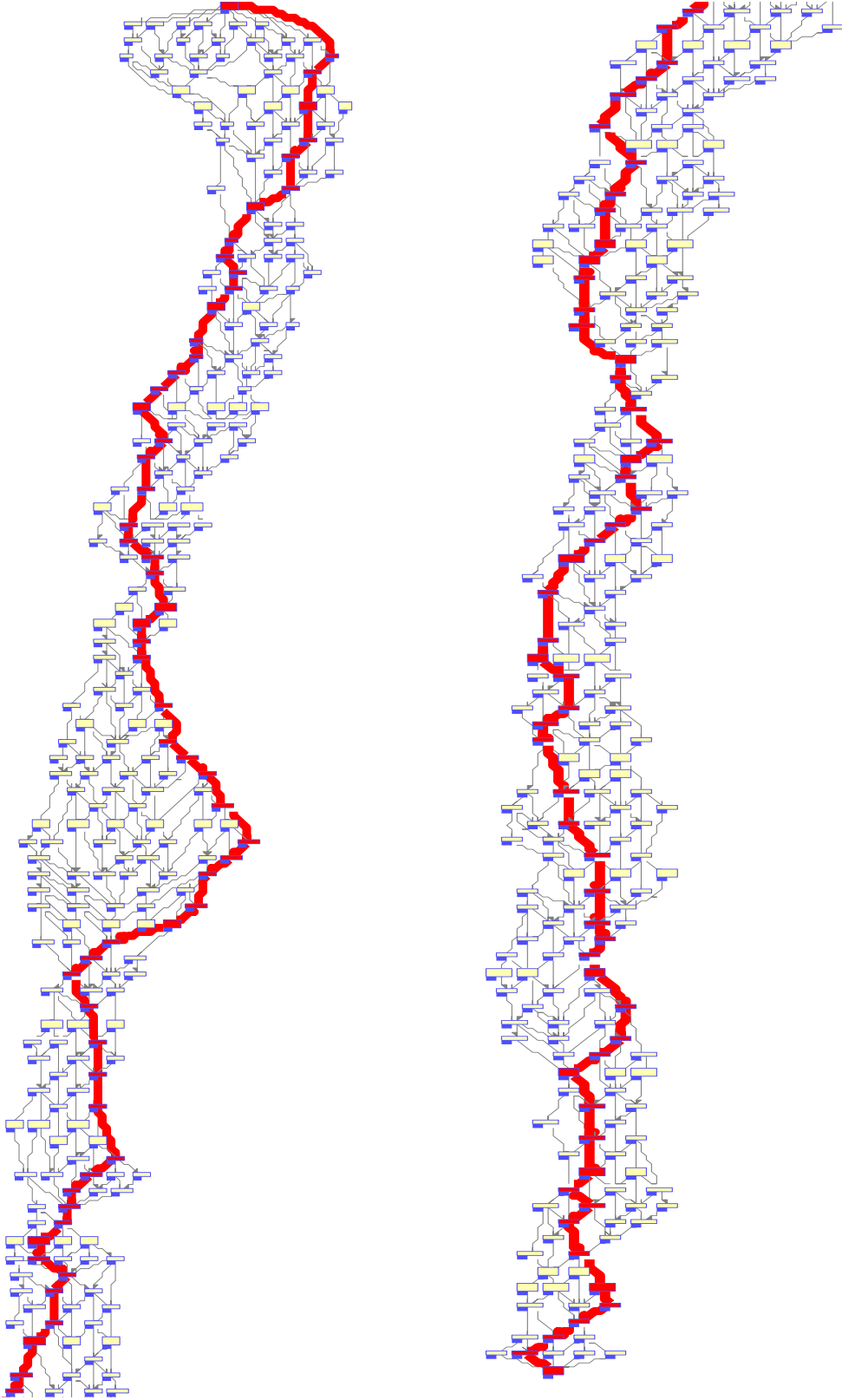
e) 38,5 lt nin altında yakıt tüketme ihtimali en yüksek yol (Rota 5)



f) %95 ihtimalle en az seyahat süreli yol (Rota 6)

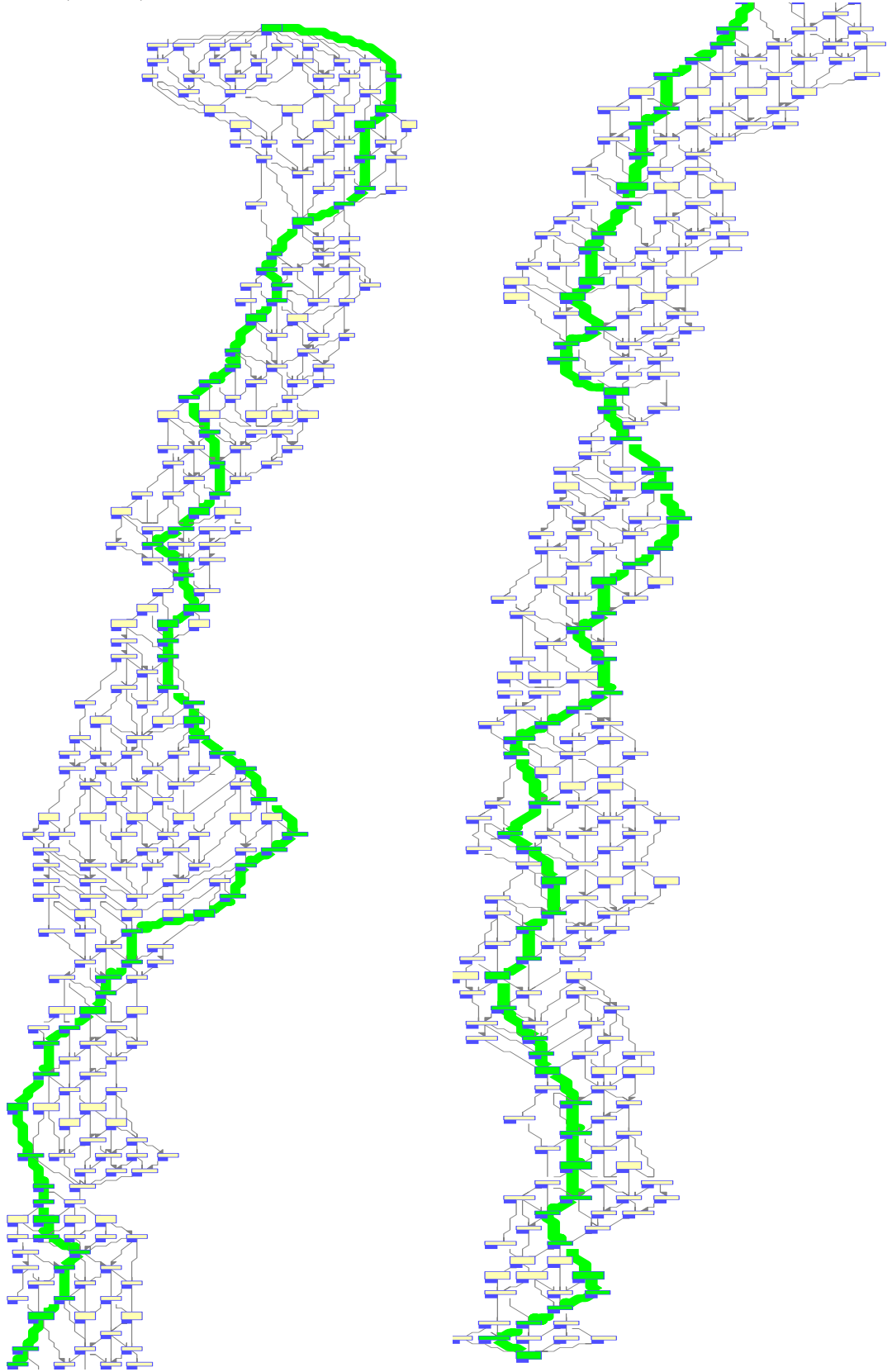
EK 12. (devam)

g) %95 ihtimalle en az yakıt tüketilen yol (Rota 7)

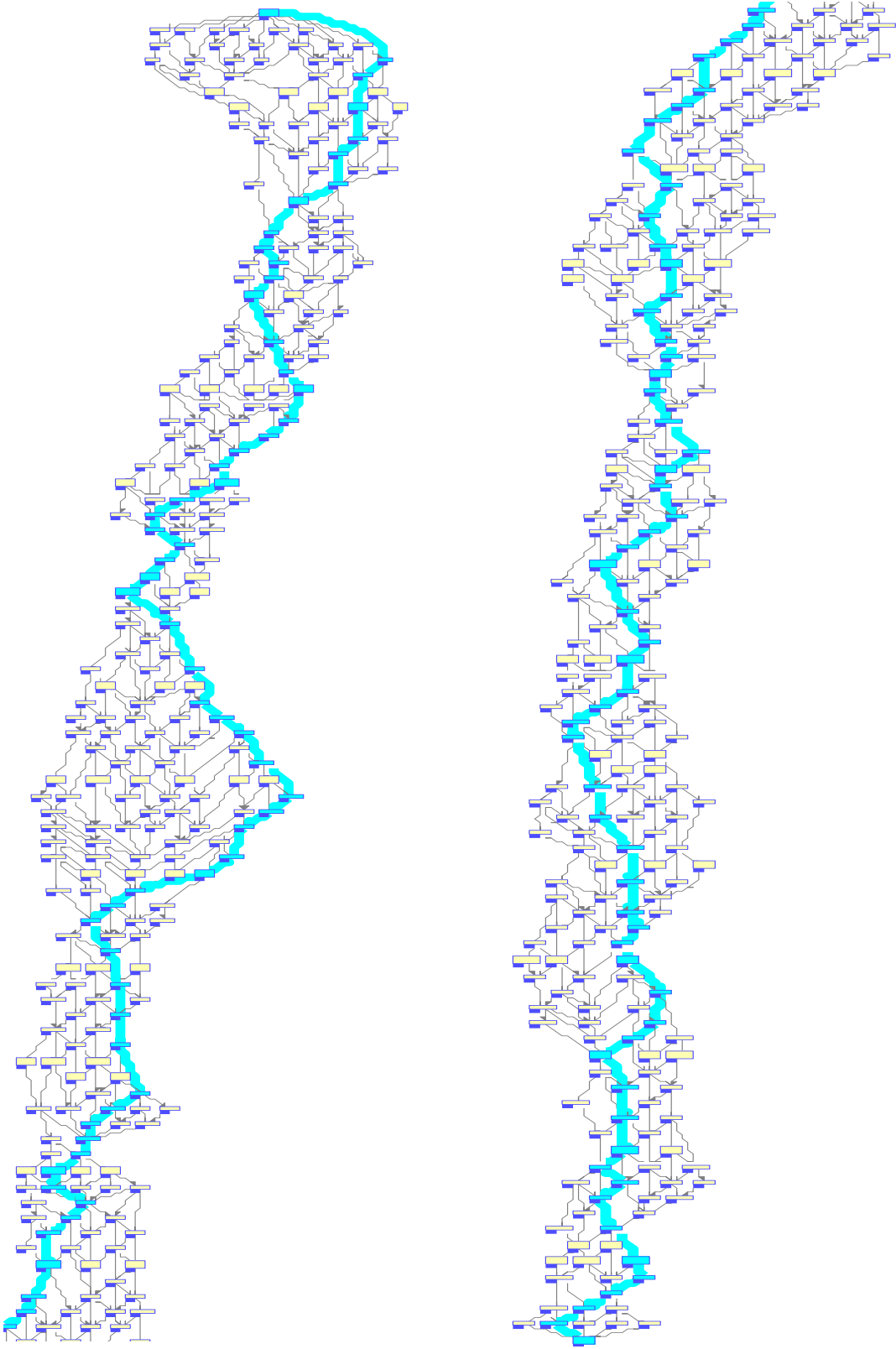
EK 13. 500 düğümlü ağlar üzerinde EKY'ların gösterilmesi

a) En kısa mesafeli yol (Rota 1)

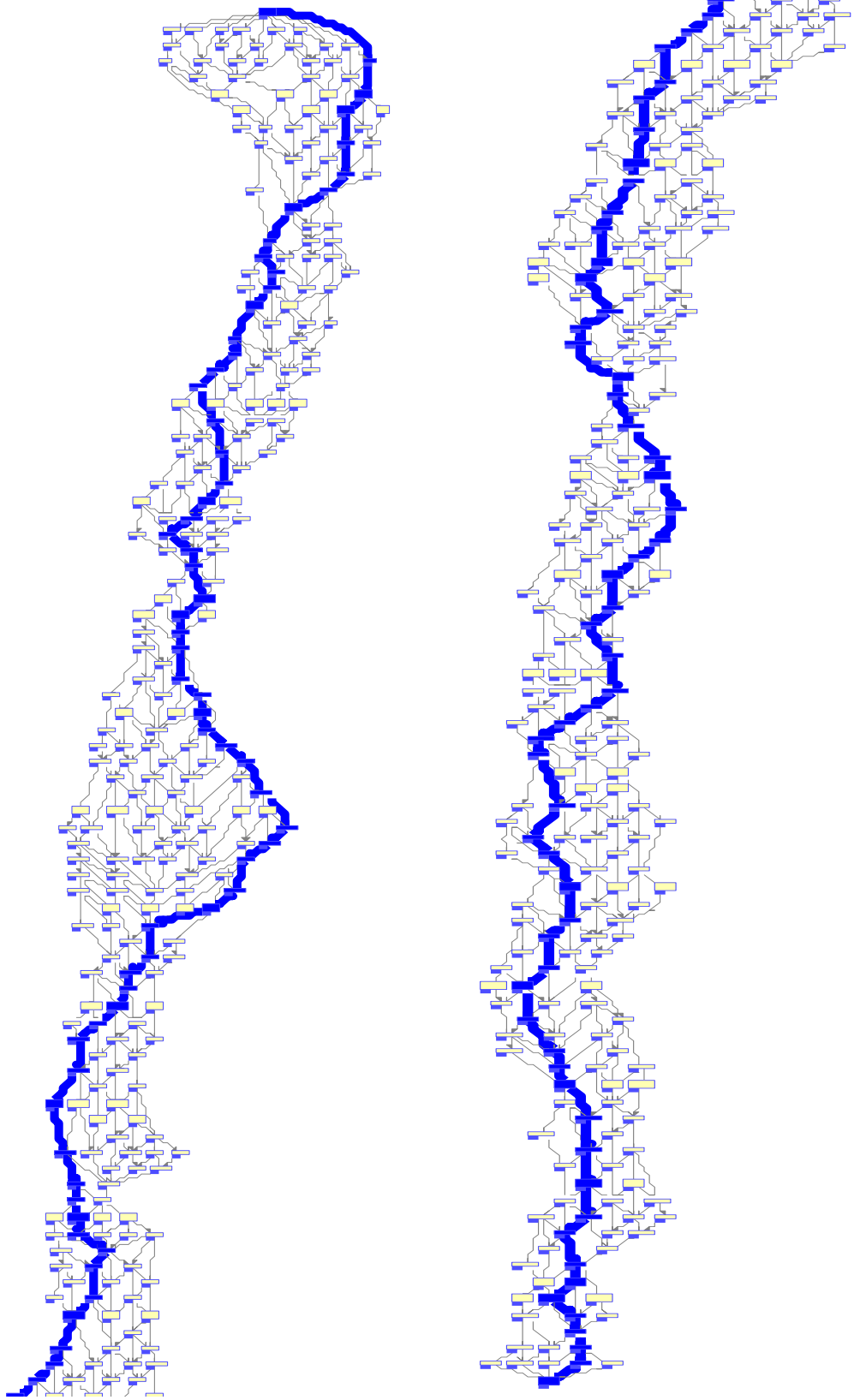
EK 13. (devam)



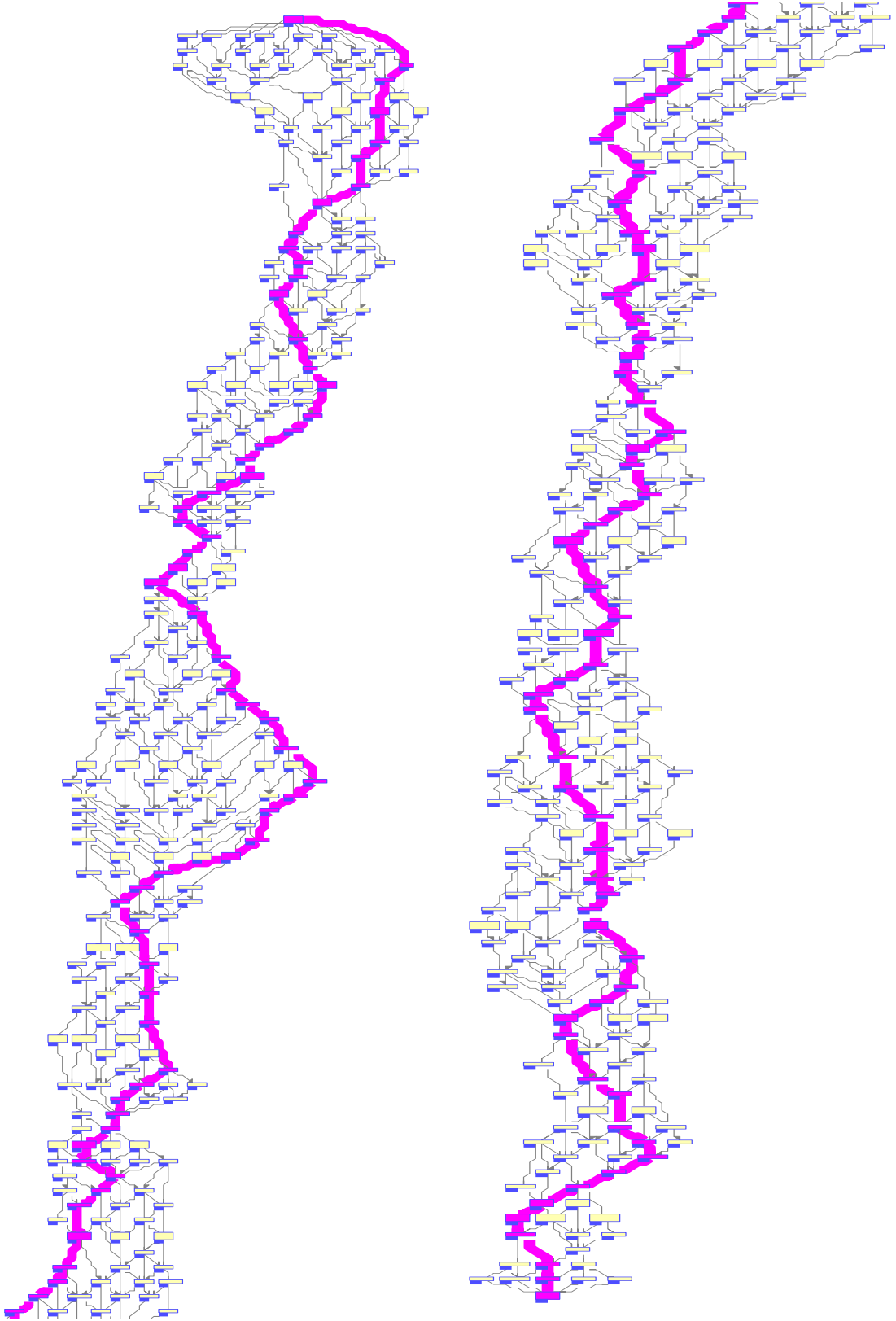
b) Ortalama en kısa seyahat süreli yol (Rota 2)

EK 13. (devam)

c) Ortalama en az yakıt tüketilen yol (Rota 3)

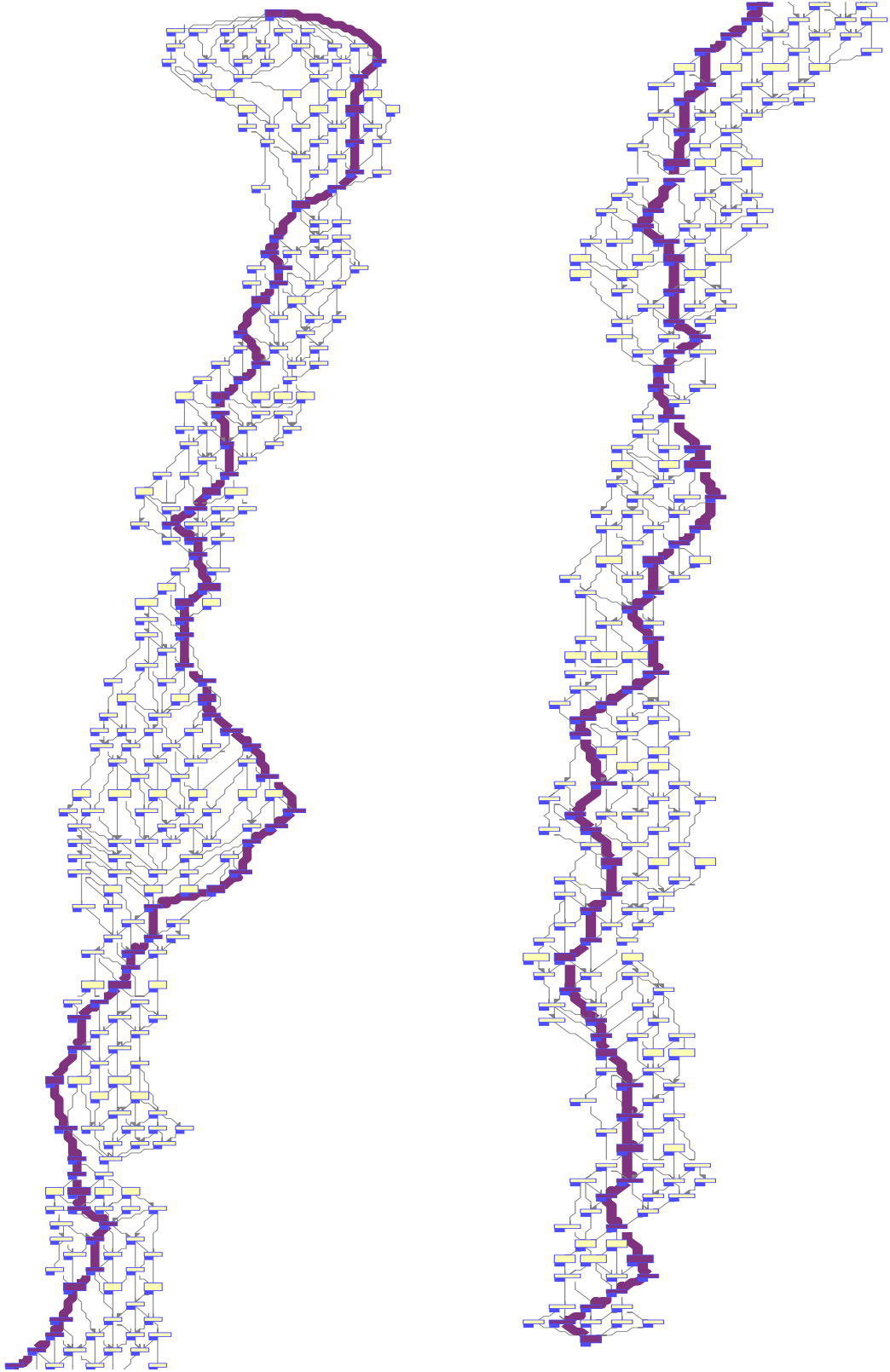
EK 13. (devam)

d) 1300 dk'nın altında gitme ihtimali en yüksek yol (Rota 4)

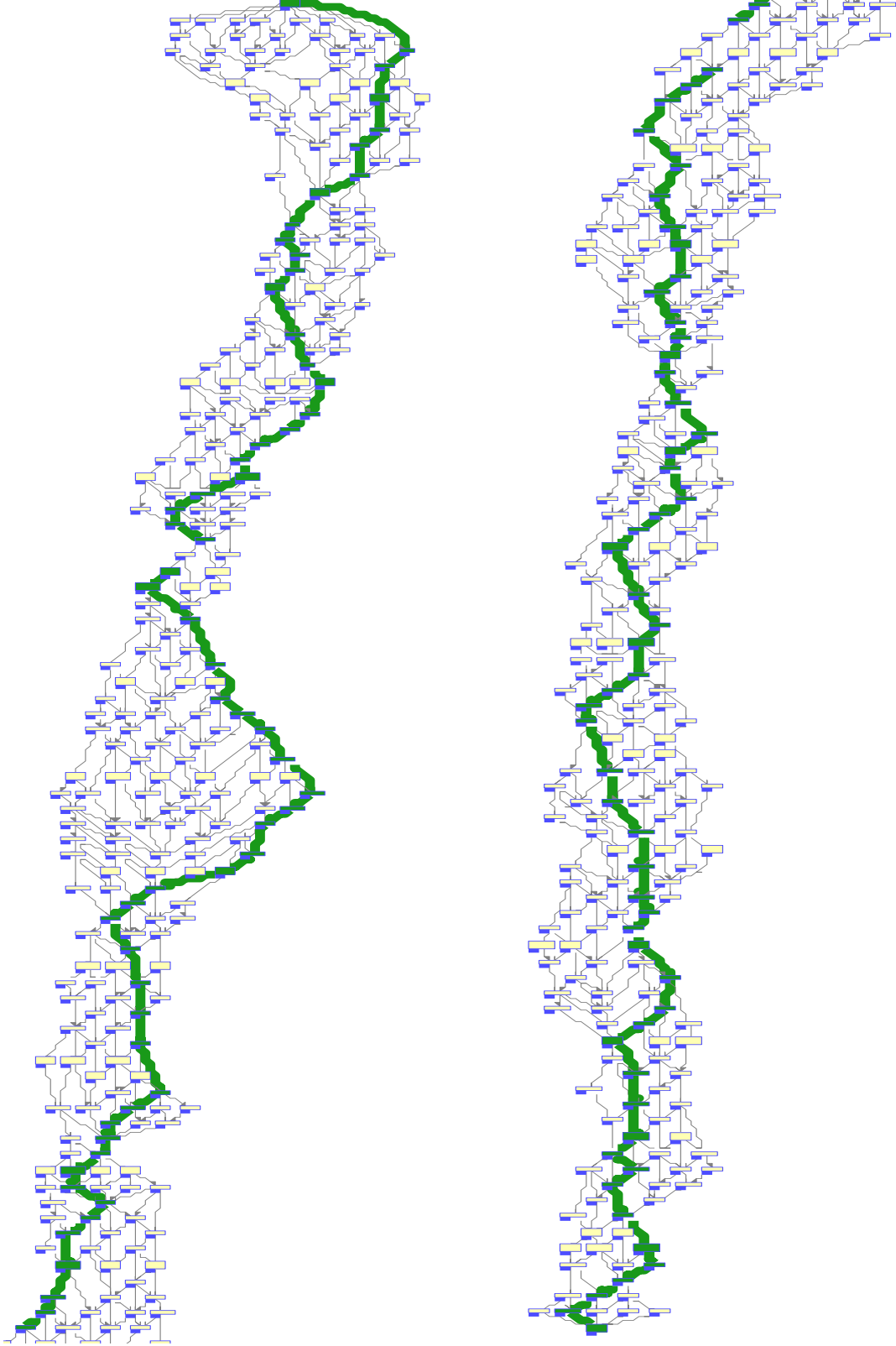
EK 13. (devam)

e) 73 lt'nin altında yakıt tüketme ihtimali en yüksek yol (Rota 5)

EK 13. (devam)



f) %50 ihtimalle en az seyahat süreli yol (Rota 6)

EK 13. (devam)

g) %50 ihtimalle en az yakıt tüketilen yol (Rota 7)

ÖZGEÇMİŞ

1978 yılında Erzurum'un Pasinler ilçesinde doğdu. İlk, orta ve lise tahsilini Erzurum'da tamamladı. 1996 yılında girdiği Atatürk Üniversitesi Mühendislik Fakültesi İnşaat Mühendisliği Bölümünden 2000 yılında mezun oldu. Aynı yıl başladığı, Atatürk Üniversitesi Fen Bilimleri Enstitüsünde Yüksek Lisans programından 2004 yılında mezun oldu. 2004 yılında doktora programına başladı.

Atatürk Üniversitesi Narman Meslek Yüksekokulunda 2003-2008 yılları arasında öğretim görevlisi olarak çalıştı. 2008 yılından beri DSİ 8. Bölge Müdürlüğünde görev yapmaktadır.