

**T.C.
AKDENİZ ÜNİVERSİTESİ**



**TOPLU TAŞIMADA AKIŞ DOĞRULTUSU TABANLI GÜZERGAH
OPTİMİZASYONU**

Kadir AKGÖL

FEN BİLİMLERİ ENSTİTÜSÜ

İNŞAAT MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

EYLÜL 2018

ANTALYA

**T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TOPLU TAŞIMADA AKIŞ DOĞRULTUSU TABANLI GÜZERGAH
OPTİMİZASYONU**

**Kadir AKGÖL
İNŞAAT MÜHENDİSLİĞİ
ANABİLİM DALI
DOKTORA TEZİ**

EYLÜL 2018

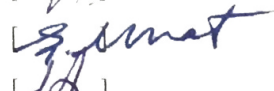
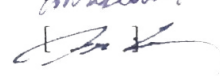
T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TOPLU TAŞIMADA AKIŞ DOĞRULTUSU TABANLI GÜZERGAH
OPTİMİZASYONU

Kadir AKGÖL
İNŞAAT MÜHENDİSLİĞİ
ANABİLİM DALI
DOKTORA TEZİ

Bu tez 18.09.2018 tarihinde jüri tarafından Oybirliği ile kabul edilmiştir.

Dr. Öğr. Üyesi İbrahim AYDOĞDU (Danışman)
Prof. Dr. Yetiş Şazi MURAT
Dr. Öğr. Üyesi Sevil KÖFTECİ
Dr. Öğr. Üyesi Emre DEMİR
Dr. Öğr. Üyesi Ferhat ERDAL

[
[
[
[
[

ÖZET

TOPLU TAŞIMADA AKIŞ DOĞRULTUSU TABANLI GÜZERGAH OPTİMİZASYONU

Kadir AKGÖL

Doktora Tezi, İnşaat Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi İbrahim AYDOĞDU

Eylül 2018, 160 sayfa

Trafik problemlerinin çözümü üzerine önerilen en etkili yöntemlerden biri toplu taşıma kullanımının teşvik edilmesidir. Bu yüzden toplu taşıma güzergahlarının iyi şekilde tasarlanması gerekmektedir. Optimizasyon toplu taşıma güzergahlarını belirlemede kullanılabilecek en etkin araçlardan bir tanesidir. Matematiksel açıdan optimum güzergah tasarımı problemi ayrık fonksiyonlar içerir. Bu yüzden meta-sezgisel optimizasyon yöntemleri bu problem için etkili araçlardır. Çoklu güzergah optimizasyonu için özellikle büyük ölçekli problemlerde meta-sezgisel optimizasyon yöntemlerinin modifiye edilmemiş versiyonları çok başarılı performans sergileyebilmekte ve çözümde erken yakınsama gözlenebilmektedir. Bu nedenle gerçekçi olmayan güzergah tasarımları elde edilebilmektedir. Bu çalışmanın ana amacı toplu taşıma kullanıcıları için maksimum faydayı sağlayacak optimum toplu taşıma güzergahları elde etmektir. Bu amaç doğrultusunda, tüm yolcular açısından faydayı ölçebilmek için akış doğrultusu yöntemi ismiyle belirtilen yeni bir yöntem ve optimum toplu taşıma güzergahları belirleyebilmek için hibrit bir model geliştirilmiştir. Hibrit modelde meta-sezgisel optimizasyon yöntemlerinden olan genetik algoritma ile akış doğrultusu yöntemi beraber kullanılmıştır. Geliştirilen hibrit model, ülkemizin metropoliten kentlerinden olan İstanbul şehrine ait otobüs hatları üzerinde uygulanmıştır. Akış doğrultusu yöntemiyle mevcut güzergahlar, genetik algoritmayla optimize edilen güzergahlar ve geliştirilen hibrit algoritmayla optimize edilen güzergahlar değerlendirilmiş ve kıyaslanmıştır. Sonuç olarak geliştirilen hibrit yöntemin hem genetik algoritmadan hem de mevcut güzergahlardan daha iyi sonuçlar ürettiği görülmüştür.

ANAHTAR KELİMELE: Dolaştırma katsayısı, Genetik algoritma, Güzergah optimizasyonu, Meta-sezgisel optimizasyon yöntemleri, Toplu taşıma güzergahları.

JÜRİ: Dr. Öğr. Üyesi İbrahim AYDOĞDU
Prof. Dr. Yetiş Şazi MURAT
Dr. Öğr. Üyesi Sevil KÖFTECİ
Dr. Öğr. Üyesi Emre DEMİR
Dr. Öğr. Üyesi Ferhat ERDAL

ABSTRACT

FLOW DIRECTION BASED ROUTE OPTIMIZATION IN PUBLIC TRANSPORT

Kadir AKGÖL

PhD Thesis in Civil Engineering

Supervisor: Asst. Prof. Dr. İbrahim AYDOĞDU

September 2018, 160 pages

One of the most effective ways to solve traffic problems is to encourage the use of public transport. This means that public transport routes need to be well designed. Optimization is one of the most useful tools that can be used to identify public transport routes. Mathematically, the optimal route design problem involves discrete functions. Therefore, meta-heuristic optimization methods are practical tools for this problem. For multiple route optimization, the unmodified versions of meta-heuristic optimization methods may not perform very well, especially for large-scale problems. Hence, early convergence can be observed in the solution, and unrealistic route designs can be obtained. The primary purpose of this study is to obtain optimum public transport routes that will provide maximum benefit for public transport users. For the purpose, a hybrid method has been developed to determine the optimum public transportation routes and a new method named by the flow direction method to measure utility for all the passengers. In the hybrid model, the flow direction method is used together with the genetic algorithm which is a well-known meta-heuristic optimization method. The developed hybrid model was applied to the bus lines belonging to Istanbul city which is one of the metropolitan cities of our country. Route directions, routes optimized with a genetic algorithm, routes optimized with developed hybrid algorithm are evaluated and compared. As a result, it is seen that the developed hybrid method produces better results than both genetic algorithm and current routes.

KEYWORDS: Circuitry coefficient, Genetic algorithm, Meta-heuristic optimization methods, Public transport routes, Route optimization.

COMMITTEE: Asst. Prof. Dr. İbrahim AYDOĞDU
Prof. Dr. Yetiş Şazi MURAT
Asst. Prof. Dr. Sevil KÖFTECİ
Asst. Prof. Dr. Emre DEMİR
Asst. Prof. Dr. Ferhat ERDAL

ÖNSÖZ

Ulaşım sorunu, gün içerisinde herkesi strese sokabilecek düzeyde etkileyebilen günümüz problemlerinden biridir. Yeni yapılan yol ve köprül  kavşak gibi büyük maliyet gerektiren yatırımlar ile bu sorunlara geçici olarak çözüm üretilebilmektedir. Fakat bir süre sonra bu çözüm yetersiz kalarak daha büyük bir sorun haline dönüşmektedir. Bu konudaki en etkili çözümlerden bir tanesi toplu taşıma sistemlerine yatırım yapmak ve insanları toplu taşıma kullanmaya teşvik etmektir. Toplu taşıma araçları için oluşturulan güzergahlara bakıldığında ise bu güzergahların insanları daha çok özel araç kullanmaya teşvik ettiği gör lmektedir. Bu durum güzergahların toplu taşıma kullanıcıları için maksimum fayda sağlayacak şekilde optimize edilmemesinden kaynaklanmaktadır. Ayrıca mevcut optimizasyon yöntemleriyle yeterli düzeyde çözüm üreten sonuçlar alınamamaktadır. Dolayısıyla mevcut sonuçlardan daha iyi çözümler sunacak yeni bir model ile yetkililere daha verimli toplu taşıma güzergahları oluşturma fırsatı verebilmek için bu çalışma gerçekleştirilmiştir. Daha verimli çalışan bir toplu taşıma sistemi, toplumdaki stresin azalması ve ekonomik kazançlar gibi bir çok etki yaratacaktır.

Geliştirilen modelde meta-sezgisel yöntemlerden genetik algoritma kullanılmıştır. Fakat bu algoritma bir araç olarak düşün lmelidir. Geliştirilen yöntemde genetik algoritma yerine farklı meta-sezgisel yöntemler de kullanılarak mevcut sonuçlardan daha iyi çözümler üretilebilir. Seçilen konu bir ulaştırma problemi olmasına rağmen çözüm tamamen optimizasyon ve programlama içermektedir. Dolayısıyla farklı alanlarda uzmanlık gerektiren bilgilerin bir arada kullanılması gerekmektedir. Çalışmanın bu yüzden alanında önemli bir yere sahip olduğunu düşünmekteyim.

Bu tezin oluşmasına katkı sağlayan, her türlü bilgi ve tecrübesini benimle paylaşan, uzmanlık alanlarıma özellikle farklı disiplinlerden yeni konular ekleyerek ufku mün genişlemesine büyük katkı sağlayan ve çalışmanın her aşamasında hiçbir desteğini esirgemeyen danışman hocalarıma, tez izleme komitesinde yer alarak tezin şekillenmesine katkı sağlayan hocalarıma, emeği geçen tüm arkadaşlarıma ve aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ	iii
İÇİNDEKİLER	iv
AKADEMİK BEYAN	vi
SİMGELER VE KISALTMALAR DİZİNİ.....	vii
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
2. KAYNAK TARAMASI.....	3
2.1. Toplu Taşıma	3
2.2. Güzergahların Değerlendirilmesi	3
2.3. Optimizasyon Yöntemleri.....	5
2.3.1. Deterministik yöntemler	5
2.3.2. Stokastik / Meta-Sezgisel yöntemler	6
2.3.3. Güzergah optimizasyonu	7
3. MATERYAL VE METOT	9
3.1. Veriler	9
3.2. Optimizasyon Problemi	10
3.2.1. ADP hesabı için örnek bir uygulama	11
3.3. Akış Doğrultusu Yöntemi Kullanılarak Oluşturulan Hibrit Bir Meta-Sezgisel Yöntem	17
3.3.1. Başlangıç işlemleri.....	18
3.3.2. Küme derleme	20
3.3.3. Güzergah kümelerinin ADP hesabı	23
3.3.4. Durdurma kriteri	23
3.3.5. Genetik algoritma	24
3.3.5.1. Rulet tekerleği yöntemi.....	25
3.3.5.2. Çaprazlama.....	26
3.3.5.3. Rastgele seçilen durakların yer değişimi mutasyonu.....	29
3.3.5.4. Güzergahtaki iki parçanın değişimi mutasyonu.....	30
3.3.5.5. Elitizm.....	31

3.3.5.6. Genetik algoritma karşılaştırma örnekleri.....	32
3.3.6. Genetik algoritma sonundaki optimum güzergah kümesinin belirlenmesi.....	34
3.3.7. Durağın en uygun konuma yerleştirilmesi iyileştirilmesi.....	35
3.3.7.1. Güzergah kümesini güzergahlara ayırma.....	37
3.3.7.2. Güzergahlardan durak çıkarma	37
3.3.7.3. ADY ile güzergahları değerlendirme	38
3.3.7.4. Güzergahtaki en kötü konuma sahip durağı belirleme.....	38
3.3.7.5. En kötü konuma sahip durağa en yakın durağı belirleme.....	39
3.3.7.6. Durağın doğru konuma yerleştirilmesi	39
3.3.8. Güzergahlar arası parça değiştirme iyileştirmesi.....	40
3.3.9. Optimum sonuç.....	42
3.3.10. Sonuçların değerlendirilmesi	42
3.4. Kümeleme.....	44
4. BULGULAR VE TARTIŞMA	45
4.1. 18 Duraklı Swain Yol Ağı Örneği.....	45
4.2. İstanbul Şehrine Ait Otobüs Hatları İçin Uygulama	81
4.2.1. Öklid mesafe ile sonuçlar	81
4.2.2. Google Maps ve öklid mesafe karşılaştırması	83
4.2.3. ADP haritaları	88
5. SONUÇLAR	93
6. KAYNAKLAR	95
7. EKLER.....	101
EK-1. Güzergah Kümelerine Ait Hesap Sonuçları	101
EK-2. İstanbul İlçelerine Ait ADP Sonuçları.....	108
ÖZGEÇMİŞ	

AKADEMİK BEYAN

Doktora Tezi olarak sunduğum “Toplu Taşımada Akış Doğrultusu Tabanlı Güzergah Optimizasyonu” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak yazıldığını belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

18.09.2018

Kadir AKGÖL



SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

GK_p	: Güzergah kümesi
D_i	: Durak numarası
M_{ij}	: Duraklar arasındaki yolculuk mesafesi
$rand$	: 0-1 arasında üretilen rastgele bir sayı
P_i	: Seçilme olasılığı

Kısaltmalar

ADY	: Akış Doğrultusu Yöntemi
ADP	: Akış Doğrultusu Performansı
P	: Popülasyon
GS	: Güzergah Sayısı
DS	: Durak Sayısı
İS	: İterasyon Sayısı
M	: Mevcut
H	: Hibrit
GA	: Genetik Algoritma
KM	: Kümülatif Mevcut
KH	: Kümülatif Hibrit
KGA	: Kümülatif Genetik Algoritma
YAK	: Yapay Arı Kolonisi
BTO	: Biyocoğrafya Tabanlı Optimizasyon
BFO	: Beyin Fırtınası Optimizasyonu
HAO	: Harmoni Arama Optimizasyonu
ABA	: Ateş Böceği Algoritması
Std. Sp.	: Standart Sapma
Min.	: Minimum
Maks.	: Maksimum

ŞEKİLLER DİZİNİ

Şekil 3.1. ADP hesabı için örnek bir ağ ve duraklar.....	11
Şekil 3.2. 1 numaralı alternatif plan.....	11
Şekil 3.3. 2 numaralı alternatif plan.....	12
Şekil 3.4. 3 numaralı alternatif plan.....	12
Şekil 3.5. 4 numaralı alternatif plan.....	13
Şekil 3.6. Akış doğrultusu tabanlı optimum çoklu güzergah tasarımı yapan hibrit modelin genel akış şeması	18
Şekil 3.7. Rastgele durak dizilimlerinden oluşan bir popülasyon örneği	19
Şekil 3.8. Güzergah kümelerinin derleme işlemi örneği.....	22
Şekil 3.9. ADY ile güzergah kümelerinin performanslarının değerlendirilmesi akış şeması	23
Şekil 3.10. Genetik algoritma işlem adımları akış şeması	24
Şekil 3.11. Çaprazlama için örnek olarak seçilen $GKCh^1$ ve $GKCh^2$ güzergah kümeleri.....	27
Şekil 3.12. İlk parçaların değişimi	28
Şekil 3.13. $GKCh^1$ güzergah kümesi için ikinci parçadaki düzenleme işlemi	28
Şekil 3.14. $GKCh^1$ ve $GKCh^2$ güzergah kümelerinin çaprazlanmış halleri	29
Şekil 3.15. Rastgele seçilen güzergah kümesinde durak yer değişimi örneği	30
Şekil 3.16. Rastgele seçilen güzergah kümesinde iki parçanın değişimi örneği	31
Şekil 3.17. ADY'ye göre en iyi performansa sahip güzergah kümesi üzerinde yapılan iyileştirme işlemi akış şeması.....	35
Şekil 3.18. ADY'ye göre en iyi performansa sahip güzergah kümesi üzerinde yapılan iyileştirme işlemi için bir örnek.....	36
Şekil 3.19. Rastgele seçilen güzergah kümesinde içinde yer alan güzergahlar arasında parça değişimi örneği.....	41
Şekil 3.20. Farklı değerlendirme yüzdelerine göre ADP değerleri.....	43
Şekil 4.1. Örnek uygulama için seçilen 55 düğüm noktalı yol ağı	46
Şekil 4.2. Yol ağı üzerinde seçilen 18 durak	47

Şekil 4.3. 1 numaralı güzergah kümesi için kesit mesafeleri.....	48
Şekil 4.4. 300. iterasyon için 13 numaralı optimum güzergah kümesine uygulanan durağın en uygun konuma yerleştirilmesi iyileştirmesi.....	67
Şekil 4.5. 5. iterasyondaki optimum çözüm.....	70
Şekil 4.6. 10. iterasyondaki optimum çözüm.....	71
Şekil 4.7. 15. iterasyondaki optimum çözüm.....	72
Şekil 4.8. 20. iterasyondaki optimum çözüm.....	73
Şekil 4.9. 65. iterasyondaki optimum çözüm.....	74
Şekil 4.10. 295. iterasyondaki optimum çözüm.....	75
Şekil 4.11. 300. iterasyondaki optimum çözüm.....	76
Şekil 4.12. 450. iterasyondaki optimum çözüm.....	77
Şekil 4.13. 460. iterasyondaki optimum çözüm.....	78
Şekil 4.14. Ağ üzerinde hesaplanan optimum güzergah kümesi	79
Şekil 4.15. İterasyon sayısına göre mesafe ve ADP değişimleri	80
Şekil 4.16. 184 numaralı güzergah kümesine ait mevcut güzergah.....	83
Şekil 4.17. 1 numaralı ADP fonksiyonu	85
Şekil 4.18. 2 numaralı ADP fonksiyonu	86
Şekil 4.19. 3 numaralı ADP fonksiyonu	87
Şekil 4.20. İstanbul şehrinin mevcut otobüs hatlarına göre ADP haritası	89
Şekil 4.21. İstanbul şehrinin hibrit optimizasyon sonucu otobüs hatlarına göre ADP haritası	90
Şekil 4.22. İstanbul için mevcut durum ile hibrit modelin karşılaştırılması	91
Şekil 7.1. Üsküdar ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu.....	108
Şekil 7.2. Kadıköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu.....	109
Şekil 7.3. Ataşehir ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu.....	110
Şekil 7.4. Maltepe ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	111
Şekil 7.5. Ümraniye ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	112

Şekil 7.6. Çekmeköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	113
Şekil 7.7. Sancaktepe ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	114
Şekil 7.8. Beykoz ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	115
Şekil 7.9. Pendik ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	116
Şekil 7.10. Sultanbeyli ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	117
Şekil 7.11. Kartal ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	118
Şekil 7.12. Tuzla ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	119
Şekil 7.13. Sarıyer ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	120
Şekil 7.14. Beşiktaş ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	121
Şekil 7.15. Beyoğlu ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	122
Şekil 7.16. Arnavutköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	123
Şekil 7.17. Şişli ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	124
Şekil 7.18. Fatih ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	125
Şekil 7.19. Kağıthane ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	126
Şekil 7.20. Zeytinburnu ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	127
Şekil 7.21. Eyüp ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	128
Şekil 7.22. Bayrampaşa ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	129
Şekil 7.23. Bahçelievler ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	130
Şekil 7.24. Bağcılar ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	131
Şekil 7.25. Bakırköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	132
Şekil 7.26. Güngören ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	133
Şekil 7.27. Başakşehir ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	134
Şekil 7.28. Esenler ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	135

Şekil 7.29. Gaziosmanpaşa ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	136
Şekil 7.30. Sultangazi ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	137
Şekil 7.31. Küçükçekmece ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	138
Şekil 7.32. Beylikdüzü ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	139
Şekil 7.33. Avcılar ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	140
Şekil 7.34. Büyükçekmece ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	141
Şekil 7.35. Esenyurt ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	142
Şekil 7.36. Şile ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	143
Şekil 7.37. Silivri ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	144
Şekil 7.38. Çatalca ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu	145

ÇİZELGELER DİZİNİ

Çizelge 3.1. Alternatif planlara ait ADP değerleri.....	14
Çizelge 3.2. Yolculuk sayıları matrisi.....	15
Çizelge 3.3. Yolcu sayılarına göre hesaplanan alternatif planlara ait ADP değerleri.....	17
Çizelge 3.4. Karşılaştırma fonksiyonu özellikleri (Aydogdu vd. 2017).....	32
Çizelge 3.5. Genetik algoritma karşılaştırma örnekleri istatistiksel veriler (n=5).....	33
Çizelge 3.6. Genetik algoritma karşılaştırma örnekleri istatistiksel veriler (n=30).....	33
Çizelge 3.7. Genetik algoritma karşılaştırma örnekleri istatistiksel veriler (n=100).....	34
Çizelge 3.8. Değerlendirme yöntemi için belirlenen en kısa mesafe ve ADP yüzdeleri.....	43
Çizelge 3.9. K-means kümeleme sonucunda oluşturulan kümeler ve içerdikleri hat sayıları.....	44
Çizelge 4.1. Duraklar arasındaki yolculuk mesafesi matrisi.....	45
Çizelge 4.2. 1. iterasyon için başlangıç popülasyonu.....	49
Çizelge 4.3. 1. iterasyon için küme derleme işlemi.....	50
Çizelge 4.4. 1. iterasyon için rulet tekerleği yöntemine göre seçilen bireyler.....	51
Çizelge 4.5. 1. iterasyon için seçilen kümelerin çift haline getirilmesi.....	53
Çizelge 4.6. 1. iterasyon için çaprazlanmış güzergah kümeleri.....	54
Çizelge 4.7. 1. iterasyon için yer değişimi mutasyonu.....	55
Çizelge 4.8. 1. iterasyon için parça değişimi mutasyonu.....	56
Çizelge 4.9. 1. iterasyon için elitizm.....	57
Çizelge 4.10. 1. iterasyon için parça değiştirme iyileştirmesi.....	58
Çizelge 4.11. 300. iterasyon için önceki iterasyondan gelen popülasyon.....	60
Çizelge 4.12. 300. iterasyon için rulet tekerleği yöntemine göre seçilen bireyler.....	61
Çizelge 4.13. 300. iterasyon için seçilen kümelerin çift haline getirilmesi.....	62

Çizelge 4.14. 300. iterasyon için çaprazlanmış güzergah kümeleri.....	63
Çizelge 4.15. 300. iterasyon için yer değişimi mutasyonu	64
Çizelge 4.16. 300. iterasyon için parça değişimi mutasyonu.....	65
Çizelge 4.17. 300. iterasyon için elitizm.....	66
Çizelge 4.18. 300. iterasyon için durak konumu değiştirme iyileştirmesi.....	68
Çizelge 4.19. 300. iterasyon için parça değiştirme iyileştirmesi	69
Çizelge 4.20. 8 grup için mevcut (M ADP) ortalamaları ile hibrit model (H ADP) ve genetik algoritma (GA ADP) sonucu elde edilen ADP ortalamaları	81
Çizelge 4.21. Hibrit modelin mevcut duruma göre verimlilik yüzdeleri.....	82
Çizelge 4.22. Optimizasyon sonucunda daha iyi performansa sahip kümeler	82
Çizelge 4.23. Alternatif ADP hesapları	84
Çizelge 4.24. İstanbul şehri ilçelerine ait ortalama mevcut ve hibrit model ADP değerleri.....	91
Çizelge 7.1. 1-36 numaralı güzergah kümelerine ait hesap sonuçları	101
Çizelge 7.2. 37-76 numaralı güzergah kümelerine ait hesap sonuçları	102
Çizelge 7.3. 77-116 numaralı güzergah kümelerine ait hesap sonuçları	103
Çizelge 7.4. 117-156 numaralı güzergah kümelerine ait hesap sonuçları	104
Çizelge 7.5. 157-196 numaralı güzergah kümelerine ait hesap sonuçları	105
Çizelge 7.6. 197-236 numaralı güzergah kümelerine ait hesap sonuçları	106
Çizelge 7.7. 237-250 numaralı güzergah kümelerine ait hesap sonuçları	107

1. GİRİŞ

Trafik sorunlarına karşı toplu taşıma kullanımının teşvik edilmesi bu konudaki en etkili çözüm önerilerindendir. Fakat toplu taşıma araçlarına ait güzergahlar bazen başlı başına bu ulaşım modunun tercih edilmeme sebebi olabilmektedir. Dolayısıyla toplu taşıma araçlarına ait güzergahlarda, yolcular için maksimum faydayı sağlayacak şekilde düzenlemeler yapılması, özel araç kullanımına olan ilgiyi azaltabileceği gibi bu ulaşım türünden yolcu çekme potansiyeline de sahiptir. Geliştirilecek bir optimizasyon yöntemiyle toplu ulaşım ile ilgili karar alacak yetkililerin daha verimli sonuçlara ulaşmasını sağlayacak bir mekanizma oluşturulması gerekmektedir. Güzergahlarda yapılacak iyileştirmeler toplu taşımanın daha cazip hale gelmesine katkıda bulunacaktır. Toplu ulaşımın cazip hale getirilerek bir mod kayması oluşturulması zincirleme olarak başka faydalar da sağlayacaktır. Örneğin yapılacak altyapı yatırımlarından önce ekstra maliyet gerektirmeyen güzergah düzenlemeleri gibi çözümler, ülke ekonomisine fayda sağlayacak tasarruflar oluşturacaktır.

Toplu ulaşım güzergahlarında bazen daha fazla yolcuya ulaşabilme çabasından dolayı anlamsız uzamalar meydana gelebilmektedir. Bu da toplu ulaşımın cazibesini azaltarak özel araç kullanımına olan ilgiyi arttırabilmektedir. Toplu ulaşımın cazip hale getirilebilmesi için yetkililerin mevcut sıkıntıyı iyi çözümlemesi gerekmektedir. Bunun için toplu ulaşım moduyla ilgili sıkıntılara yönelik kriterler oluşturulmalı ve mevcut durumun kalitesi sayısal olarak ölçülebilmelidir. Toplu ulaşım güzergahlarının belli kriterler göz önüne alınarak kalitesinin değerlendirilmesi ve bu kriterlere uymayan güzergahların daha verimli hale getirilmesi bu ulaşım moduna olan ilgiyi arttıracaktır.

Farklı araştırmacıların, yol ağlarının ve güzergahların geometrik olarak kalitesinin ölçülmesine dair çalışmaları mevcuttur. Fakat toplu taşıma güzergahlarının kalitesini ölçebilmek için bu hatta yapılan yolcuları da ele almak gerekmektedir. Literatürde bir hat içerisindeki tüm alternatif yolculukları dikkate alarak güzergah kalitesinin değerlendirilmesine yönelik bir çalışmaya rastlanmamıştır. Bu değerlendirme yöntemi kullanılarak geliştirilecek bir model ile toplu taşımanın daha verimli güzergahlarla otomobil kullanımına iyi bir alternatif sunması planlanmaktadır.

Ulaşım araçlarına yönelik güzergahlarının hesaplanabilmesi için geliştirilen yöntemlerde genellikle amaç fonksiyonu en kısa mesafe ya da süreyi veren güzergahın tespit edilmesi olarak belirlenmektedir. Oysa ana amaç toplam yolculuklar için minimum mesafe ya da süreyi sağlamak olmalıdır. Yani en kısa mesafenin ya da sürenin hesaplanması toplam yolculuklar için en verimli durumu oluşturamayabilmektedir. Dolayısıyla bu çalışmada, tasarlanan güzergahların mümkün olduğunca özel araçlarla yapılabilecek en kısa yolculuklara yakın sürelerde seyahat imkanı sunmasını istenmektedir. Bu doğrultuda yolculuk bazında maksimum faydanın sağlanabilmesi için, güzergahları değerlendirerek daha kaliteli güzergahlar elde edilmesini sağlayacak akış doğrultusu yöntemi geliştirilmiştir. Bu yöntem ile minimum süre ya da mesafeyi veren güzergahlar elde etmek yerine yolculukların toplamda maksimum verimi sağlayacağı güzergahlar elde edilebilecektir.

Toplu taşıma araçlarına ait güzergahların toplu taşıma kullanıcılarına maksimum faydayı sağlayamaması sonucu gerekli ilgiyi görememekte ve bunun sonucunda özel araç kullanımına olan talep artarak trafik sorunlarına sebep olmaktadır. Toplu taşımanın

daha cazip hale getirilmesi ve özel araç kullanımına karşı güçlü bir alternatif sunması gerekmektedir. Toplu taşıma araçlarıyla yapılacak yolculuk maliyetlerini artırmadan yolculuk sürelerinin azaltılması, bu problemin çözümüne önemli bir katkı sağlayacaktır. Toplu ulaşım güzergahlarının belirlenmesinde genellikle maliyet ve yolculuk süresi gibi durumlar göz önüne alınarak alternatif planlamalar arasında bir seçim yapma yoluna gidilmektedir. Fakat oluşturulan güzergahlar bir optimizasyon sonucu elde edilmediği için sadece belirlenen alternatifler arasında kıyaslama yapılabilmektedir. Dolayısıyla bu çalışmanın amacı, toplam yolculuklar açısından toplu taşıma araçlarının geometrik açıdan daha verimli güzergahlara sahip olmasını sağlayacak yeni bir hibrit meta-sezgisel optimizasyon modeli geliştirmektir. Bu model ile otobüs hatları bireysel olarak değil gruplandırma yöntemiyle toplu olarak incelenmektedir. Ayrıca toplu taşıma hatları tek bir güzergah olarak ele alınmamaktadır. Bir hat içerisinde yapılabilecek tüm yolculuklar akış doğrultusu yöntemi adı altında ayrı bir güzergah olarak değerlendirilmektedir. Dolayısıyla yolculukların tamamını gözeten bir optimizasyon çalışması yürütülmüştür. Bu çalışmada, farklı örneklerde kullanılmış ve başarılı olmuş olan genetik algoritmadan faydalanılmıştır. Amaç fonksiyonunda sadece güzergahların geometrileri dikkate alınmıştır. Fakat geliştirilen yöntem ile amaç fonksiyonu değiştirilerek farklı açılardan optimum güzergahların elde edilebileceği bir model oluşturulmuştur. Geliştirilen yöntem ülkemizin metropoliten kentlerinden olan İstanbul şehrine ait otobüs hatları üzerinde uygulanmıştır. Akış doğrultusu yöntemiyle mevcut durum, sadece genetik algoritmayla yapılan optimizasyon sonuçları ve geliştirilen hibrit algoritmayla yapılan optimizasyon sonuçları kıyaslanmıştır.

2. KAYNAK TARAMASI

2.1. Toplu Taşıma

Toplu taşıma araçlarının güzergahları bazen iki nokta arasındaki en kısa güzergahı verememekte ve daha fazla yolcuya ulaşılması çabasıyla dolaylı güzergahlarda anlamsız uzamalar görülebilmektedir. Yolcular biniş ve iniş duraklarına bağlı olarak bu güzergahlardan farklı derecelerde etkilenmektedir. Bu etkinin boyutu kişilerin ulaşım modu tercihlerini değiştirebilmektedir. Toplu taşıma araçlarına ait güzergahların seçimi, o hattı kullanacak tüm yolcular açısından maksimum faydayı sağlamalıdır. Bu durum, otomobil kullanımından toplu ulaşım moduna olan kaymayı teşvik etmede faydalı olacaktır. Otoriteler özel araç kullanımından toplu taşımaya mod kayması oluşturabilmek için toplu ulaşımında hizmet kalitesinin artırılması gerektiğini savunmaktadır (S. Chowdhury vd. 2014). Toplu taşımanın daha cazip hale getirilmesine yönelik yapılan bir çalışmada, hizmet kalitesinin iyileştirilmesinin sadece toplum için değil bu alanda faaliyet gösteren şirketler için de karlı olabileceği belirtilmiştir (Rojo vd. 2015).

Ulaşım performansının artırılması konusunda genellikle yetkililer ve yolcuların çıkarları çatışmaktadır (Vreeswijk vd. 2015). Ulaşım odaklı politikalarda insanların seyahat davranışları üzerinde yeterince durulmaması ve toplu taşıma rekabet gücünün arttırılamaması gibi durumlar ulaşım sektöründe acil çözülmesi gereken sorunlardandır. (Garcia-Sierra vd. 2015; Raveau vd. 2014; Sun vd. 2016). Yolcu ihtiyaçlarının dikkate alınması ve bu ihtiyaçların karşılanması konusunda standartların oluşturulması, toplu taşıma hizmet kalitesinin arttırılmasına katkı sağlayacaktır (Yildirim 2013). Sürdürülebilir ulaşım sistemleri için toplu taşıma hizmet kalitesini arttıran ve kullanımını teşvik eden etkili çözümler, ulaşım sektörü üzerindeki baskıları azaltmada etkili olacaktır (Davison ve Knowles 2006; Vitale vd. 2014). Bu şekilde toplu taşıma hizmet düzeyleri dizayn edilerek potansiyel toplu taşıma kullanıcılarına ulaşılabilir. (Beirão ve Sarsfield Cabral 2007). Hadas ve Ranjitkar (2012) toplu taşıma bağlantılarının geliştirilmesinin ulaşım planlamasının en önemli görevlerinden biri olduğunu ve kötü bağlantıların toplu taşımanın tercih edilmemesine neden olabileceğini belirtmiştir. Toplu taşıma araçlarına ait güzergahların belli bir hizmet düzeyinde kalacak şekilde tasarlandığı bir çalışmada özel araç kullanımından %52'lik bir mod kayması potansiyelinin olduğu belirlenmiştir (Gunay vd. 2016). Bu da toplu taşıma güzergahlarının düzenlenerek daha etkili ve verimli bir hale getirilmesinin ulaşım politikalarındaki önemini göstermektedir (Buchari ve Ndiaye 2008).

2.2. Güzergahların Değerlendirilmesi

Son yıllarda toplu taşıma ağ yapıları ve topolojisine odaklanan çalışmalar artmıştır (Lin ve Ban 2013). Yan ve Wang (2009), statik toplu taşıma ağlarında topolojik özellikleri incelemek için bazı ampirik analizler gerçekleştirmiştir. Çalışmada toplu taşıma ağlarının evrimsel gelişimini görmek için hızlandırılmış büyüyen ve doğrusal olmayan tercih kurallarına dayalı bir simülasyon modeli önermiştir. Sonuç olarak, toplu taşıma ağlarının topolojik özelliklerinin çoğalarak iyi bir performans gösterdiğini belirtmiştir. Derrible ve Kennedy (2011)'e göre toplu taşıma sistemlerinin ağ yapıları iyi bilinmekte fakat topoloji açısından göreceli olarak sınırlı kalmaktadır. Bu yüzden iyi bir gelecek için planlamacılar daha iyi tasarlanmış toplu taşıma sistemleri

konusunda çok iş düşmektedir. Curtis ve Scheurer (2010), planlama politikalarında toplu taşıma modunun payının artırılması gerektiğini vurgulamıştır. Planlamada erişilebilirliğin önemine değinmiş ve medya aracılığıyla erişilebilirlik uygulamalarının yaygınlaştırılmasının, gelecekte kentsel hareketliliğe doğru katkıyı sağlayacağını belirtmiştir. Hadas (2013)'a göre toplu taşıma ağları çeşitli fiziksel özelliklerinden dolayı karmaşık bir yapıya sahiptir. Değerlendirilmesi zor olan bu ağları geliştirmek amacıyla yaptığı çalışmada GIS tekniklerine dayalı bir metodoloji ile analiz gerçekleştirmiştir. Bu analiz ile ulaşım ağını; kapsama seviyesi, ortalama hızlar, kesişim kapsama seviyesi, aktarma potansiyeli ve güzergah çakışmaları açısından incelemiştir. Toplu taşıma araçlarına ait güzergahları konu alan bu çalışmalardan hiçbiri güzergahlar rasyonelliklerini geometrik açıdan değerlendirmemiştir. Ulaşım planlaması, genellikle alternatif seçeneklerin karşılaştırıldığı bir rasyonellik sürecidir (Cascetta vd. 2015). Ulaşım rasyonellik kavramı bölgesel planlama ve ekonomi açısından ve sosyal, hukuki ve siyasi yönleriyle bazı araştırmacılar tarafından çalışılmıştır (Breheny ve Hooper 1985; Diesing 1962; Sager 1994). Rasyonelliğin bu araştırmalar göz önüne alınarak araçsal ve iletişimsel rasyonel olarak ikiye ayrıldığı başka bir çalışmada, ulaşımın planlama aşamaları ve insanlarla olan etkileşimi incelenmiştir (Langmyhr 2001). Başka bir çalışmada toplu taşıma ağ bağlantısı, Google Ulaşım kullanılarak yürüme mesafesi, konfor, ödeme metodu ve güvenlik yönleriyle ölçülerek, Paris şehrinin Londra ve Auckland şehirlerinden daha rasyonel olduğunu belirtilmiştir (Subeh Chowdhury vd. 2014). Buna ek olarak, bu çalışmaların hiçbirinde güzergah geometrisinin rasyonelliği çalışılmamıştır. Güzergahların yol ağları üzerinde oluşturduğu gereksiz ağ alanı kullanımlarına göre yol ağlarının rasyonelliklerinin ölçüldüğü farklı bir çalışmada ise toplu ulaşım boyutu incelenmemiştir (Matisziw ve Demir 2016). Yapılan çalışmalar toplu ulaşım odaklı politikalarda yetkililere güzergahların geometrisi açısından bir kalite değerlendirmesi yapma şansı sağlamamaktadır.

Güzergahlar, geometrilerine bağlı olarak yolculara daha kısa mesafe ya da yolculuk süresi gibi kazanımlar sağlayabilir. Bu kazanımlar daha önceki çalışmalarımızda otomobil ile yapılan yolculuklar için, rasyonellik skalaları adı altında farklı metotlarla ölçülmüştür (Akgöl ve Gunay 2018; Gunay vd. 2018). Bu ölçümler sayesinde yol ağlarında düzenleme yapılması gereken bölgeler kolaylıkla tespit edilebilmiştir. Benzer bir ölçümün toplu ulaşım güzergahları için yapılarak hizmet kalitesinin artırılması toplu taşımanın cazibesini artıracaktır. Toplu ulaşım sürdürülebilir bir başarı elde edilebilmesi için ölçülebilir başarı kriterleri oluşturulmalıdır (Canci ve Onden 2013). Bu doğrultuda toplu ulaşım araçlarına ait güzergahların rasyonelliklerini geometrileri açısından değerlendiren sayısal analiz yöntemleri geliştirmek, toplu taşımanın cazibesinin artırılmasına olanak sağlayacaktır.

Güzergahlara ait mesafelerin, öklid uzunluklarına oranı dolaştırma (circuitry) ya da rasyonellik (rationality) kavramları altında bir çok araştırmacı tarafından çalışılmıştır (Huang ve Levinson 2015). öklid mesafesi, “kuş uçuşu mesafe” olarak da tanımlanabilir. Bu oran, en kısa güzergahın öklid mesafesine eşit olması durumunda teorik olarak minimum 1 olabilmektedir. Ağ üzerinde yapılabilecek tüm yolculuklar için hesaplanan dolaştırma (rasyonellik) değerlerinin ortalaması ise yol ağının performansını göstermektedir (Vragović vd. 2005). Dolaştırma değeri ne kadar düşük olursa ağ o kadar verimlidir. Newell (1980) çalışmasında yol ağlarına ait ortalama dolaştırma

değerini yaklaşık 1,2 olarak hesaplamıştır. O'Sullivan ve Morrall (1996) ise çalışmasında toplu taşıma ağlarını incelemiş ve dolaştırma değerinin 1,21 ve 1,23 arasında çıktığını belirtmiştir. Fakat bu değerler her incelenen şehirde farklı sonuçlar verebilir. Ballou vd. (2002) ortalama dolaştırma değerine bakarak yolculuk mesafelerinin tahmin edilebileceğini belirtmiştir. Love ve Morris (1979) çalışmasında öncelikli tesis yeri problemleri için analitik modeller kullanarak iki nokta arasında uzunluğu bu şekilde tahmin etmiştir. Bu değerlendirme yöntemi Dill (2004) tarafından yaya ve bisiklet yolculukları içinde kullanılmış ve otomobil için gözlemlenen değerlerden daha yüksek dolaştırma değerleri elde edilmiştir. Wolf vd. (2004) çalışmasında yolcuların GPS izlerini inceleyerek, gerçek yolculukların beklenenden daha fazla dolaştırma değerine sahip olduğunu bulmuştur. Levinson ve El-Geneidy (2009) ise yaptığı çalışmada rastgele nokta çiftleri ve ev-iş yeri çiftleri arasındaki dolaştırma değerlerini hesaplamış ve işçilerin evlerinin konumuna karar verirken daha az dolaştırma değerine sahip yerleri tercih ettiği sonucuna ulaşmıştır.

Ulaşım planlamasında toplu ulaşım ağları genellikle politik nedenlerden dolayı geniş bir mekânsal kapsama alanı sağlamak için tasarlanmış ve bunun sonucunda dolambaçlı hatlar ortaya çıkmıştır (Black 1995; Murray vd. 1998; Taylor 1991). Bununla birlikte, yüksek dolaştırma değerleri insanları diğer ulaşım modlarına yönelmeye teşvik etmektedir (Walker 2012). Toplu ulaşım ağlarına erişimi en üst düzeye çıkarabilmek ve toplu taşımayı teşvik edebilmek için birçok şehirde ağ yatırım politikaları iyileştirilmiş ve genişletilmiştir (Cervero 2004). Toplu taşıma ağları genellikle daha geniş bir alanı kapsayacak şekilde tasarlandığından, toplu taşıma yolcuları için dolaştırma değerleri otomobil kullanıcılarına göre daha yüksektir (D. M. Levinson 1998). Otomobil kullanımı ile yol ağları ve toplu ulaşım ağlarının erişilebilirliği arasındaki korelasyonlar farklı çalışmalarda yerel ve metropoliten alanlarda incelenmiştir. Sonuçlar, toplu ulaşım erişilebilirliğindeki artışın, toplu taşıma yolculukları için harcanan zamanı ve otomobil kullanım oranını azaltacağı yönünde bir etkiye sahip olduğunu ortaya koymuştur (D. Levinson 2012; Owen ve Levinson 2015).

2.3. Optimizasyon Yöntemleri

Toplu taşıma güzergahlarının tasarımı konusunda başvurulacak en iyi çözüm optimizasyon yöntemleridir. Optimizasyon, belirli bir amaca yönelik ve istenilen koşullar altında tanımlanan problemde, mevcut çözümler arasındaki en uygun çözümün bulunması işlemidir.

2.3.1. Deterministik yöntemler

Sayısal yöntemler için teorik olarak elde edilen hata payıyla çözümün optimum olduğunu garanti eden ve yakınsama hızının oldukça yüksek olduğu yöntemlerdir. Bu yüzden süre olarak stokastik yöntemlere göre avantajlıdır. Deterministik yöntemler genellikle türevsel kökenli olup, amaç ve kısıtlar sürekli fonksiyonlar şeklinde tanımlanmalıdır. Bu da deterministik yöntemlerin kullanılma alanları sınırlar ve tasarım problemlerine uygulanmalarını oldukça zorlaştırır. En sık kullanılan deterministik yöntemlerden bazıları aşağıda belirtilmiştir.

Doğrusal Programlama: Amaç ve kısıtları doğrusal denklemlerden oluşan bir deterministik yöntemdir. Amaç fonksiyonu ve kısıtlar bir arada yazılarak doğrusal

denklem sistemine dönüştürülür. İteratif işlemler yapılarak optimum çözüm bulunur. Topping (1983) çalışmasında doğrusal programlama yöntemini elastik yapı tasarımı için kullanmıştır.

Lagrange Çarpan Yöntemi: Amaç ve kısıtlar lagrange çarpan fonksiyonları ile birleştirilerek tek bir fonksiyon elde edilir. Fonksiyondaki bilinmeyenlere göre türevleri alınarak denklem sistemi oluşturulur ve denklem çözülerek optimum sonuç elde edilir. Çözüm her ne kadar optimum sonucu verse de, sadece sürekli fonksiyonlara uygulanabilir. Ayrıca büyük ölçekli problemlerde çözümün çok zor olması sebebiyle kullanım alanı sınırlıdır (Lagrange 2013).

Optimallik Kriter Yöntemi: Yöntem, sürekli değişkenler için geliştirilmiştir (Prager 1968). Teorik olarak lagrange çarpan yönteminde oluşturulan birleşik fonksiyonu doğrudan çözmek yerine tekrarlı bir işlem uygulanarak optimum sonucu elde eder. Mühendislik problemlerinde sık kullanılan deterministik yöntemlerden biri olsa da stokastik yöntemlere göre daha az tercih edilir.

2.3.2. Stokastik / Meta-Sezgisel yöntemler

Stokastik yöntemler belirli bir olasılığa dayalı olarak optimum sonucu veren optimizasyon yöntemleridir. Yöntem rastgele ürettiği değerlerden yola çıkarak optimum sonuca ulaşır. Rastgele değerler üretilmesi çözümün bilgisayar ortamında oluşturulmasını gerektirir. Rastgele üretilen değerlerin seçilme olasılığı çözüme ait amaç fonksiyonu ile doğrudan ilişkilidir. Meta-Sezgisel yöntemler ise optimum sonucu ararken sezgiye dayalı çözümler üretir. Yeni üretilen çözümler önceki çözümlerden yararlanarak elde edilir. Öncelikle rastgele çözümler içeren bir havuz oluşturulur. Daha sonra eski çözümler güncellenerek yeni çözümleri oluşturur (Kaveh 2014). Yöntemde dikkat edilmesi gereken en önemli konu çeşitlendirme ve yoğunlaşma arasındaki dengenin sağlanmasıdır.

Stokastik/Meta-Sezgisel yöntemler türevsel işlemlere ihtiyaç duymamaktadır. Dolayısıyla doğrusal olmayan, parçalı fonksiyonlar içeren, ayrık değişkenli optimizasyon problemlerine rahatlıkla uygulanabilir. Bu yöntemlerle her zaman en iyi çözüm bulunamasa da belirli bir hata ile en iyi sonuca yakın sonuçlar üretilir. Hata miktarı minimuma indirgenerek çözümün optimum sonuca mümkün olduğunca yaklaştırılır. Genellikle bu yöntemler doğadaki hayvanların davranışından veya fiziksel olaylardan esinlenilerek geliştirilmiş yöntemlerdir.

Genetik Algoritma: Darwin'in evrim teorisinden esinlenerek, doğadaki güçlü olan bireylerin yaşayıp zayıf olan bireylerin elenmesi, bireylerin mutasyona uğraması ve yeni bireylerin oluşturulması gibi davranışların matematiksel modele dönüştürüldüğü optimizasyon yöntemidir (Goldberg ve Holland 1988).

Karınca Kolonisi Optimizasyonu: Yöntem, karıncaların yiyecek ile yuva arasında en kısa yolu bulma ilkesinden esinlenerek geliştirilmiştir. Karıncalar, "Pheromone" adı verilen bir kimyasal salgılayarak diğer karıncaları etkilemekte ve bir önceki karıncanın geçtiği yolun yakınından geçmesine yardımcı olmaktadır. Sabit bir zaman diliminde kısa yoldaki salgı miktarı uzun yoldakine daha fazla olacaktır. Dolayısıyla karıncalar bir süre sonra kısa yolu tercih etmeye başlayacaklardır (Dorigo

vd. 1996). Aydoğdu ve Saka (2012) düzensiz çelik çerçevelerin tasarımında bu yöntemi kullanmıştır.

Harmoni Arama: Enstrümanların mükemmel uyumu sağlamak için yaptıkları akort ayarından esinlenerek geliştirilmiştir. İlk olarak rastgele bir harmoni hafızası oluşturulmakta ve belirli oranda ince ayarlar yapılarak hafıza güncellenmektedir. Yeni oluşturulan harmoniler, hafızadaki harmonilerden daha iyi ise yeniler ile eskiler yer değiştirmektedir. Bu şekilde mükemmel harmoni uyumu yakalanmaya çalışılmaktadır (Geem vd. 2001). Hasançebi vd. (2009) bu algoritmayı yapısal optimizasyon problemlerinde açıklıkları, Akin ve Aydogdu (2015) ise çelik çerçeve boyutlarını optimize etmek için kullanmıştır.

Karaboga ve Basturk (2007) arı sürülerinin davranışlarını inceleyerek yapay arı kolonisi, Simon (2008) doğadaki göç, hayatta kalma ve yok olma kavramlarını baz alarak biyocoğrafya tabanlı optimizasyon, Shi (2011) bir grup insanın problemlerin çözümleri için ürettiği yolları baz alarak beyin fırtınası optimizasyonu ve Yang (2009) ateş böceği davranışlarını gözlemleyerek ateş böceği algoritmasını geliştirmiştir. Değertekin (2012) çerçevelerin optimum tasarımı için yapay arı kolonisi algoritmasından, Jalili vd. (2014) kafes sistemlerin boyutlandırmasında biyocoğrafya tabanlı optimizasyondan faydalanmıştır. Değertekin vd. (2006) çelik çerçevelerin tasarımı için tabu arama ve genetik algoritma yöntemlerini kullanmıştır.

2.3.3. Güzergah optimizasyonu

Günümüzde, güzergah tasarlamayla ilgili araç rotalama problemleri adı altında ya da bu problem türünün daha genel bir hali olan gezgin satıcı problemleri adı altında farklı algoritmalar kullanılarak çözüm arayışları sürmektedir. Bu çözüm yöntemleri en uygun sonuca ulaşıp ulaşmamasına göre kesin çözüm yöntemleri ve sezgisel yöntemler olarak ikiye ayrılmaktadır. Fakat kesin çözüm yöntemleriyle sonuca ulaşmak oldukça zordur. 1995 yılında en fazla 134 müşterilik bir problem çözülebilmıştır (Düzakın ve Demircioğlu 2009). Dolayısıyla sezgisel yöntemler çözüm konusunda ön plana çıkmıştır. Sezgisel yöntemler de klasik ve meta-sezgisel yöntemler olarak ikiye ayrılmaktadır. Meta-sezgisel yöntemlerin klasik sezgisel yöntemlere göre daha iyi sonuç verdiği literatürde bir çok çalışmada karşımıza çıkan bir olgudur (Osman ve Kelly 1996).

Shi vd. (2017) İngiltere’de kentsel alanlarda (Londra hariç) yol temelli toplu taşımacılığın planlanmasında yolcu ve planlamacıların ihtiyaçları arasında çelişkiler olduğunu belirtmiştir. Bu çelişki yol temelli toplu taşımacılıkta çok sık karşılaşılan bir durumdur. Yaptıkları çalışmada otobüs güzergahı tasarımı ve yolculuk frekansı arasında optimizasyon tekniklerinden tabu arama algoritmasını kullanarak Java için ArcObjects ortamında bir optimizasyon yöntemi geliştirmiştir. Krol (2017) toplu taşıma sistemlerinin verimlilik hesaplarının çok karmaşık optimizasyon problemleri olduğu dile getirmiş ve sezgisel yöntemlerin bu problemin çözümündeki önemini vurgulamıştır. Yine çalışmasında toplu taşıma şebekesinin optimum yapısını bulmak için genetik algoritmayı kullanmış, amaç fonksiyonunu ulaşım taleplerinin gerçekleşeceği toplam zaman üzerine oluşturmuş ve güney Polonya’da bir şehir için vaka çalışması yapmıştır. Nikolopoulou vd. (2017) tedarikçiler ve müşteriler arasındaki ulaşım taleplerini karşılamak için yerel arama optimizasyon yöntemini geliştirmiştir. Leong vd. (2016)

optimum güzergah problemlerinde gezgin satıcı problemlerinin yetersiz kaldığını belirtmiş ve arıların beslenme davranışlarını dikkate alarak yeni bir koloni algoritması geliştirmiştir.

Görüldüğü gibi farklı sezgisel optimizasyon algoritmaları birçok ulaşım probleminin çözümünde bugüne kadar defalarca kullanılmıştır. Fakat güzergah tasarımı konusunda genetik algoritma, karınca kolonisi, tabu arama vb. meta-sezgisel yöntemler her ne kadar başarılı sonuçlar üretse de çoklu güzergah tasarlama problemlerinde, durak sayısı arttıkça optimum sonuçlar üretmekte zorlanmakta ve gerçekçi olmayan sonuçlar üretebilmektedir. Dolayısıyla gerçek hayatta karşımıza çıkan büyük ölçekli otobüs güzergahlarının tasarlanması gibi problemler çözümsüz kalmaktadır. Bu çalışmada literatürdeki mevcut çözümlerden daha iyi sonuçlar üretebilmek adına meta-sezgisel yöntemlerden biri olan genetik algoritma modifiye edilmiş ve yeni matematiksel modellerle birleştirilerek hibrit bir model elde edilmiştir. Böylece meta-sezgisel yöntemlerin lokal optimum yakınsadığı noktalardan global optimum çözüme geçiş sağlanmıştır.

3. MATERYAL VE METOT

Bir güzergahın geometrik açıdan kalitesinden bahsedebilmenin en kolay yolu güzergah üzerinde katedilen mesafenin iki nokta arasındaki öklid mesafeye ne kadar yakın olduğuyla ilgilidir. Bu yakınlık arttıkça güzergahın bizi daha az dolaştırdığını dolayısıyla kalitesinin arttığını söylemek mümkündür. Aynı çerçevede daha az dolaştıran güzergahlara sahip toplu taşıma araçlarının yolculara daha cazip geleceği ve bu modun tercih edilmesinde etkili olacağını ifade etmek yanlış olmayacaktır. Bu yüzden bu çalışmada toplu taşıma güzergahlarının geometrik açıdan ne kadar kaliteli olduğu ölçebilmek için akış doğrultusu yöntemi oluşturulmuş ve bu kalitenin nasıl artırılabilirliği konusuna çözüm getirecek çoklu güzergah optimizasyonu yapan hibrit bir optimizasyon yöntemi geliştirilmiştir. Bu yöntemlerin test edilebilmesi için İstanbul şehrine ait mevcut otobüs hatlarının durumu incelenmiş ve bu hatlara ait güzergahlar optimize edilerek mevcut durumla karşılaştırılmıştır. Güzergahların optimize edilebilmesi için öncelikle mevcut hatlar kümelenmiş ve bu kümede kalan duraklar boş kalmayacak şekilde yeni rotalar oluşturulmuştur.

3.1. Veriler

Çalışmada örnek bir uygulama yapabilmek adına, ülkenin metropoliten şehirlerden biri olan İstanbul'a ait otobüs hatları incelenmiştir. Şehirde toplam 732 otobüs hattı, incelenen ağda 5602 adet otobüs durağı ve bu duraklar arasında 14664 kesit bulunmaktadır. Fakat bazı duraklardan birden fazla hat geçtiği için toplamda 29933 durakla hesaplamalar gerçekleştirilmiştir. Otobüs hatlarına ait veri tabanında tüm hatlara ait güzergahların sıralı olarak hangi duraklardan geçtiği, bu duraklara ait koordinat verileri ve duraklar arasındaki kesitlere ait mesafe verileri bulunmaktadır. Çalışma kapsamında hesaplarda her durak çifti arasında 1 yolculuk gerçekleşeceği varsayımı yapılmıştır. Fakat örnek bir yolculuk matrisi ile yolculuk sayılarının da dahil edildiği bir uygulama çalışma kapsamında çözülmüştür. Duraklar arasındaki mesafeler, öklid mesafeler olarak alınmıştır. Bu mesafeler koordinatlar yardımıyla ve Haversine metoduyla hesaplanmaktadır. (3.1)'de verilen bu formül dünya üzerindeki coğrafi koordinat noktaları arasında gerçek mesafelerin hesabı için kullanılmaktadır (Chopde ve Nichat 2013; Robusto 1957).

$$r = 2 \cdot r_w \cdot \sin^{-1} \sqrt{\sin^2 \left(\frac{x_j - x_i}{2} \right) + \cos(x_i) \cdot \cos(x_j) \cdot \sin^2 \left(\frac{y_j - y_i}{2} \right)} \quad (3.1)$$

Burada,

x_i : i'inci noktanın enlemi,

y_i : i'inci noktanın boylamı,

r : öklid mesafe,

r_w : dünyanın yarıçapı (6371 km).

Böylece gerçek mesafelere yakın değerler üretilerek çözüme gidilmiştir. Örnek olması açısından ise Google Maps'ten de veri çekilerek bir uygulama yapılmıştır.

3.2. Optimizasyon Problemi

Bir güzergahın geometrik açıdan kalitesinin ölçülebilmesi için güzergah mesafesinin öklid mesafesine olan oranını hesaplamak yeterlidir. Fakat toplu taşıma araçlarına ait güzergahların kalitesini ölçebilmek için o güzergah üzerinde yapılabilecek yolculukların da hesaba dahil edilmesi gerekmektedir. Dolayısıyla tüm yolculuklara ait bu oranların hesaplanarak yolculuk sayısına göre ağırlıklandırılması bir toplu taşıma aracının güzergah kalitesini ortaya koyacaktır. Bu durum güzergah üzerinde aynı doğrultuda yapılacak yolculuk sayılarının ve yolculuk mesafelerinin durumuna göre farklı sonuçlar doğuracaktır. Güzergah üzerinde aynı doğrultudaki yolculuk sayısı ve aynı doğrultuda katedilen mesafeler arttıkça güzergahın kalitesi artacaktır. Bu yüzden toplu taşıma güzergahlarının kalitesinin ölçülebilmesi için Akış Doğrultusu Yöntemi (ADY) geliştirilmiştir. Bu yöntem ile hesaplanan değerler de bu çalışmada Akış Doğrultusu Performansı (ADP) olarak adlandırılmıştır. Bu değer aynı zamanda optimizasyon probleminin amaç fonksiyonunu oluşturmaktadır. (3.2)'de verilen fonksiyon minimize edilerek istenilen optimum sonuca ulaşılmaktadır. Optimizasyonun kısıtları ise (3.3) ve (3.4)'te verilmiştir.

ADP değerinin minimizasyonu:

$$ADP = \sum_{j=2}^n \sum_{i=1}^{n-1} \frac{M_{ij}^{güzergah}}{M_{ij}} \cdot \varphi_{ij} / n_K \quad (3.2)$$

Kısıtlar:

$$\varphi_{ij} = 1 + \frac{\sum \Psi_{ij} - \Psi_{ij}}{\sum \Psi_{ij}} \quad (3.3)$$

$$n_K = \frac{n \cdot (n-1)}{2} \quad (3.4)$$

Burada,

n : güzergahtaki durak sayısı,

i, j : güzergahtaki durak numaraları indisleri,

M_{ij} : ağ üzerinde i 'inci duraktan j 'inci durağa olan yolculuk mesafesi ya da süresi,

$M_{ij}^{güzergah}$: güzergah üzerinde i 'inci duraktan j 'inci durağa olan yolculuk mesafesi ya da süresi,

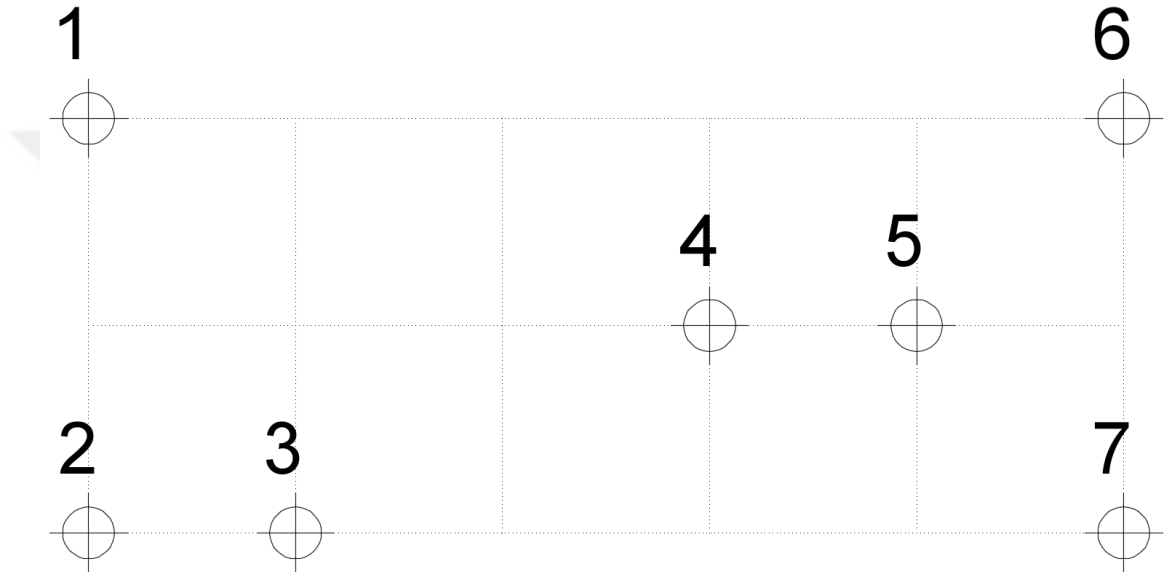
φ : yolculuk katsayısı,

Ψ_{ij} : i 'inci duraktan j 'inci durağa olan yolculuk sayısı,

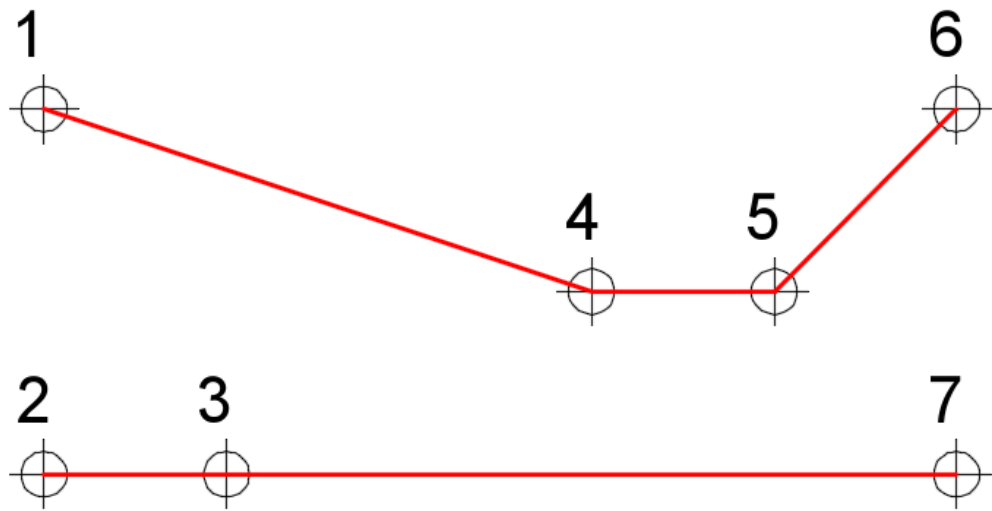
n_K : güzergahtaki toplam yolculuk kombinasyonu sayısıdır.

3.2.1. ADP hesabı için örnek bir uygulama

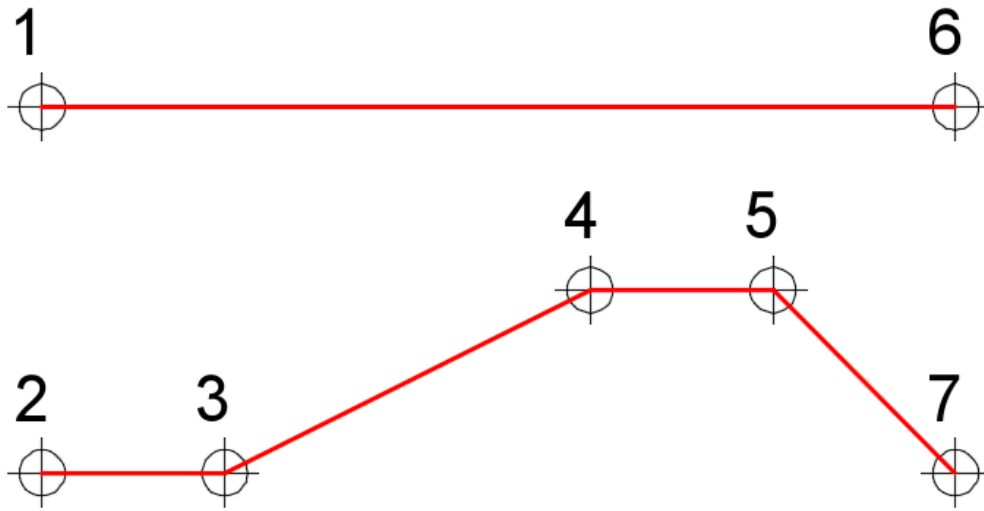
Yöntemi daha iyi anlayabilmek ve yolculuk sayılarının *ADP* üzerindeki etkisini inceleyebilmek adına Şekil 3.1’de görülen 7 duraklı bir ağ oluşturulmuştur. Bu ağ üzerinde 2 farklı otobüs güzergahı oluşturulacaktır. Bunlardan ilki 1 ile 6 nolu duraklar arasında ikincisi ise 2 ve 7 nolu duraklar arasındadır. 3 nolu durağın 2 ile 7 nolu durağa ait güzergahta bulunması istenmektedir. 4 ve 5 nolu durakların bu durak çiftlerinden hangisinde bulunması gerektiğine ise *ADP* değerlendirmesiyle karar verilecektir. Şekil 3.1’de duraklar arasındaki mesafelerin hesaplanabilmesi için şekil 1’er birimlik kare ızgara bir altlık üzerine oturtulmuştur. 4 ve 5 nolu duraklarla ilgili tüm alternatif planlamalar Şekil 3.2, Şekil 3.3, Şekil 3.4 ve Şekil 3.5’te gösterilmiştir.



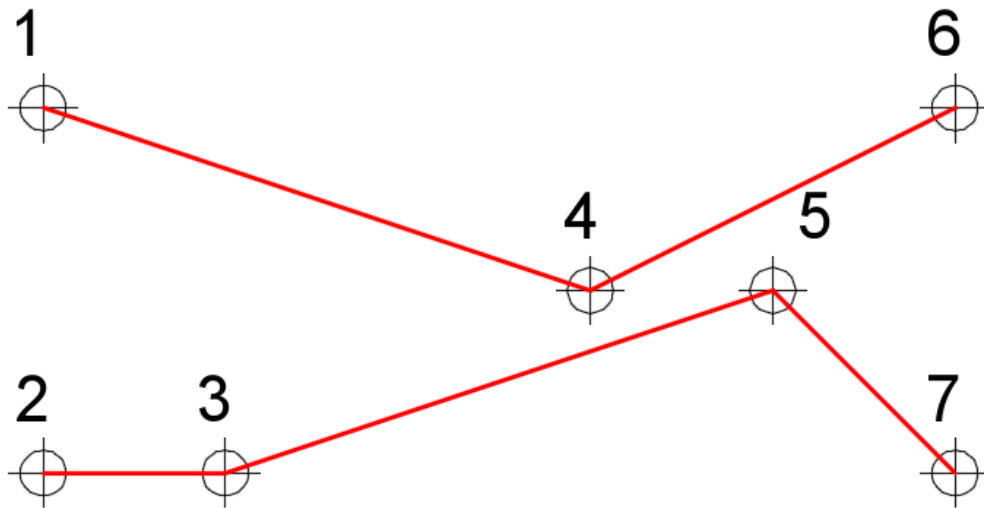
Şekil 3.1. ADP hesabı için örnek bir ağ ve duraklar



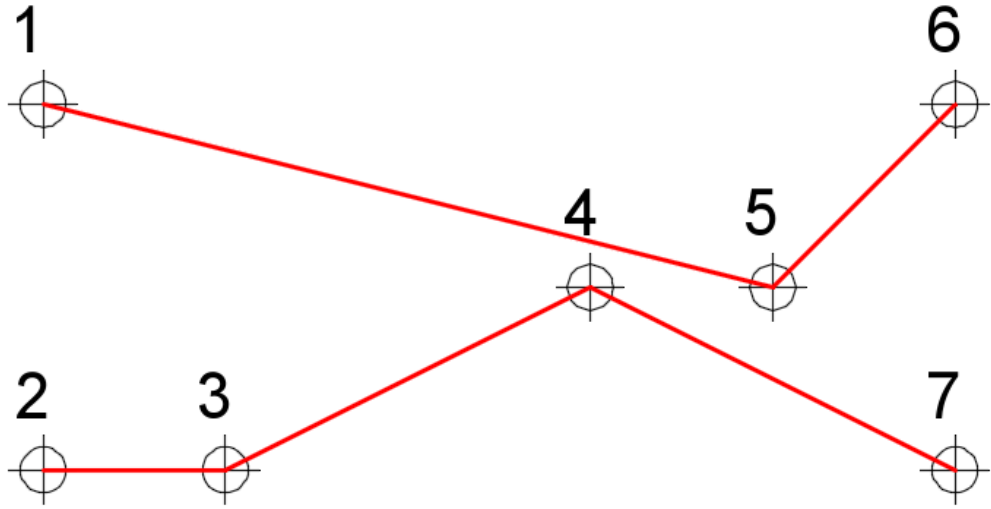
Şekil 3.2. 1 numaralı alternatif plan



Şekil 3.3. 2 numaralı alternatif plan



Şekil 3.4. 3 numaralı alternatif plan



Şekil 3.5. 4 numaralı alternatif plan

Yolculuk katsayıları hesaba katılmadığı takdirde *ADP* hesaplamaları aşağıdaki gibi olacaktır.

1. alternatif plan için güzergahlara ait *ADP* değerleri:

$$ADP_{(1-6)} = \frac{\frac{2,23}{2,23} + \frac{4,23}{4,12} + \frac{5,61}{5,00} + \frac{2,00}{2,00} + \frac{3,41}{3,16} + \frac{1,41}{1,41}}{\frac{4 \cdot (4 - 1)}{2}} = \frac{6,23}{6} = 1,038$$

$$ADP_{(2-7)} = \frac{\frac{1,00}{1,00} + \frac{5,00}{5,00} + \frac{4,00}{4,00}}{\frac{3 \cdot (3 - 1)}{2}} = \frac{3,00}{3} = 1,000$$

2. alternatif plan için güzergahlara ait *ADP* değerleri:

$$ADP_{(1-6)} = \frac{\frac{5,00}{5,00}}{\frac{2 \cdot (2 - 1)}{2}} = \frac{1,00}{1} = 1,000$$

$$ADP_{(2-7)} =$$

$$\frac{\frac{1,00}{1,00} + \frac{2,41}{2,23} + \frac{4,41}{4,12} + \frac{5,82}{5,00} + \frac{1,41}{1,41} + \frac{3,41}{3,16} + \frac{4,82}{4,00} + \frac{2,00}{2,00} + \frac{3,41}{3,16} + \frac{1,41}{1,41}}{\frac{5 \cdot (5 - 1)}{2}} = \frac{10,68}{10} = 1,068$$

3. alternatif plan için güzergahlara ait *ADP* değerleri:

$$ADP_{(1-6)} = \frac{\frac{2,23}{2,23} + \frac{5,39}{5,00} + \frac{3,16}{3,16}}{\frac{3 \cdot (3 - 1)}{2}} = \frac{3,08}{3} = 1,026$$

$$ADP_{(2-7)} = \frac{\frac{1,00}{1,00} + \frac{4,16}{3,16} + \frac{5,57}{5,00} + \frac{3,16}{3,16} + \frac{4,57}{4,00} + \frac{1,41}{1,41}}{\frac{4 \cdot (4 - 1)}{2}} = \frac{6,57}{6} = 1,095$$

4. alternatif plan için güzergahlara ait *ADP* değerleri:

$$ADP_{(1-6)} = \frac{\frac{4,12}{4,12} + \frac{5,53}{5,00} + \frac{1,41}{1,41}}{\frac{3 \cdot (3 - 1)}{2}} = \frac{3,11}{3} = 1,035$$

$$ADP_{(2-7)} = \frac{\frac{1,00}{1,00} + \frac{2,41}{2,23} + \frac{5,57}{5,00} + \frac{1,41}{1,41} + \frac{4,57}{4,00} + \frac{3,16}{3,16}}{\frac{4 \cdot (4 - 1)}{2}} = \frac{6,34}{6} = 1,056$$

Hesaplanan *ADP* sonuçları Çizelge 3.1’de bir arada görülmektedir.

Çizelge 3.1. Alternatif planlara ait *ADP* değerleri

Alternatif Plan	Güzergah	ADP	Toplam
1	1-4-5-6	1,038	2,038
	2-3-7	1,000	
2	1-6	1,000	2,068
	2-3-4-5-7	1,068	
3	1-4-6	1,026	2,121
	2-3-5-7	1,095	
4	1-5-6	1,035	2,091
	2-3-4-7	1,056	

Çizelge 3.1’e göre yolculuk sayıları dikkate alınmadan yapılan bu *ADP* değerlendirmesi sonucunda 1 numaralı alternatif plan en iyi, 3 numaralı alternatif ise en kötü seçenek olarak karşımıza çıkmaktadır. Aynı değerlendirme yolculuk sayıları dikkate alınarak tekrar yapılacaktır. Bunun için Çizelge 3.2’de verilen yolculuk sayıları matrisi kullanılacaktır.

Çizelge 3.2. Yolculuk sayıları matrisi

Duraklar	1	2	3	4	5	6	7
1	-	-	-	2	20	10	-
2		-	4	6	5	-	3
3			-	20	2	-	1
4				-	1	2	15
5					-	8	1
6						-	-
7							-

Çizelge 3.2’de bu duraklar arasında yolculuk yapmak isteyen 100 kişiye ait yolculuk sayıları belirtilmiştir. Burada (3.3) numaralı formüle göre yolculuk katsayıları hesaplanmıştır. Örneğin 1 ve 5 nolu duraklar için;

$$\varphi_{15} = 1 + \frac{\sum \Psi_{15} - \Psi_{15}}{\sum \Psi_{15}} = 1 + \frac{100 - 20}{100} = 1,8$$

şeklinde katsayı hesaplanmıştır. Bu katsayıların tamamı hesaplandıktan sonra R_{m2} değerleri bu katsayılar dahil edilerek tekrar hesaplanmıştır.

Yolculuk katsayıları hesaba katıldığı takdirde *ADP* hesaplamaları aşağıdaki gibi olacaktır.

1. alternatif plan için güzergahlara ait *ADP* değerleri:

$$ADP_{(1-6)} =$$

$$\frac{\frac{2,23}{2,23} \cdot (1,98) + \frac{4,23}{4,12} \cdot (1,8) + \frac{5,61}{5,00} \cdot (1,9) + \frac{2,00}{2,00} \cdot (1,99) + \frac{3,41}{3,16} \cdot (1,98) + \frac{1,41}{1,41} \cdot (1,92)}{\frac{4 \cdot (4 - 1)}{2}} = \frac{12,01}{6} = 2,001$$

$$ADP_{(2-7)} = \frac{\frac{1,00}{1,00} \cdot (1,96) + \frac{5,00}{5,00} \cdot (1,97) + \frac{4,00}{4,00} \cdot (1,99)}{\frac{3 \cdot (3 - 1)}{2}} = \frac{5,92}{3} = 1,973$$

2. alternatif plan için güzergahlara ait *ADP* değerleri:

$$ADP_{(1-6)} = \frac{\frac{5,00}{5,00} \cdot (1,9)}{\frac{2 \cdot (2 - 1)}{2}} = \frac{1,90}{1} = 1,900$$

$$ADP_{(2-7)} =$$

$$\begin{aligned} & \frac{\frac{1,00}{1,00} \cdot (1,96) + \frac{2,41}{2,23} \cdot (1,94) + \frac{4,41}{4,12} \cdot (1,96) + \frac{5,82}{5,00} \cdot (1,97) + \frac{1,41}{1,41} \cdot (1,8)}{\frac{5 \cdot (5 - 1)}{2}} \\ & + \frac{\frac{3,41}{3,16} \cdot (1,98) + \frac{4,82}{4,00} \cdot (1,99) + \frac{2,00}{2,00} \cdot (1,99) + \frac{3,41}{3,16} \cdot (1,85) + \frac{1,41}{1,41} \cdot (1,99)}{\frac{5 \cdot (5 - 1)}{2}} \\ & = \frac{20,76}{10} = 2,076 \end{aligned}$$

3. alternatif plan için güzergahlara ait ADP değerleri:

$$ADP_{(1-6)} = \frac{\frac{2,23}{2,23} \cdot (1,98) + \frac{5,39}{5,00} \cdot (1,9) + \frac{3,16}{3,16} \cdot (1,98)}{\frac{3 \cdot (3 - 1)}{2}} = \frac{6,01}{3} = 2,003$$

$$ADP_{(2-7)} =$$

$$\begin{aligned} & \frac{\frac{1,00}{1,00} \cdot (1,96) + \frac{4,16}{3,16} \cdot (1,95) + \frac{5,57}{5,00} \cdot (1,97) + \frac{3,16}{3,16} \cdot (1,98) + \frac{4,57}{4,00} \cdot (1,99) + \frac{1,41}{1,41} \cdot (1,99)}{\frac{4 \cdot (4 - 1)}{2}} \\ & = \frac{12,97}{6} = 2,161 \end{aligned}$$

4. alternatif plan için güzergahlara ait ADP değerleri:

$$ADP_{(1-6)} = \frac{\frac{4,12}{4,12} \cdot (1,8) + \frac{5,53}{5,00} \cdot (1,9) + \frac{1,41}{1,41} \cdot (1,92)}{\frac{3 \cdot (3 - 1)}{2}} = \frac{5,82}{3} = 1,940$$

$$ADP_{(2-7)} =$$

$$\begin{aligned} & \frac{\frac{1,00}{1,00} \cdot (1,96) + \frac{2,41}{2,23} \cdot (1,94) + \frac{5,57}{5,00} \cdot (1,97) + \frac{1,41}{1,41} \cdot (1,8) + \frac{4,57}{4,00} \cdot (1,99) + \frac{3,16}{3,16} \cdot (1,85)}{\frac{4 \cdot (4 - 1)}{2}} \\ & = \frac{12,17}{6} = 2,029 \end{aligned}$$

Yeniden hesaplanan ADP sonuçları Çizelge 3.3'teki gibidir.

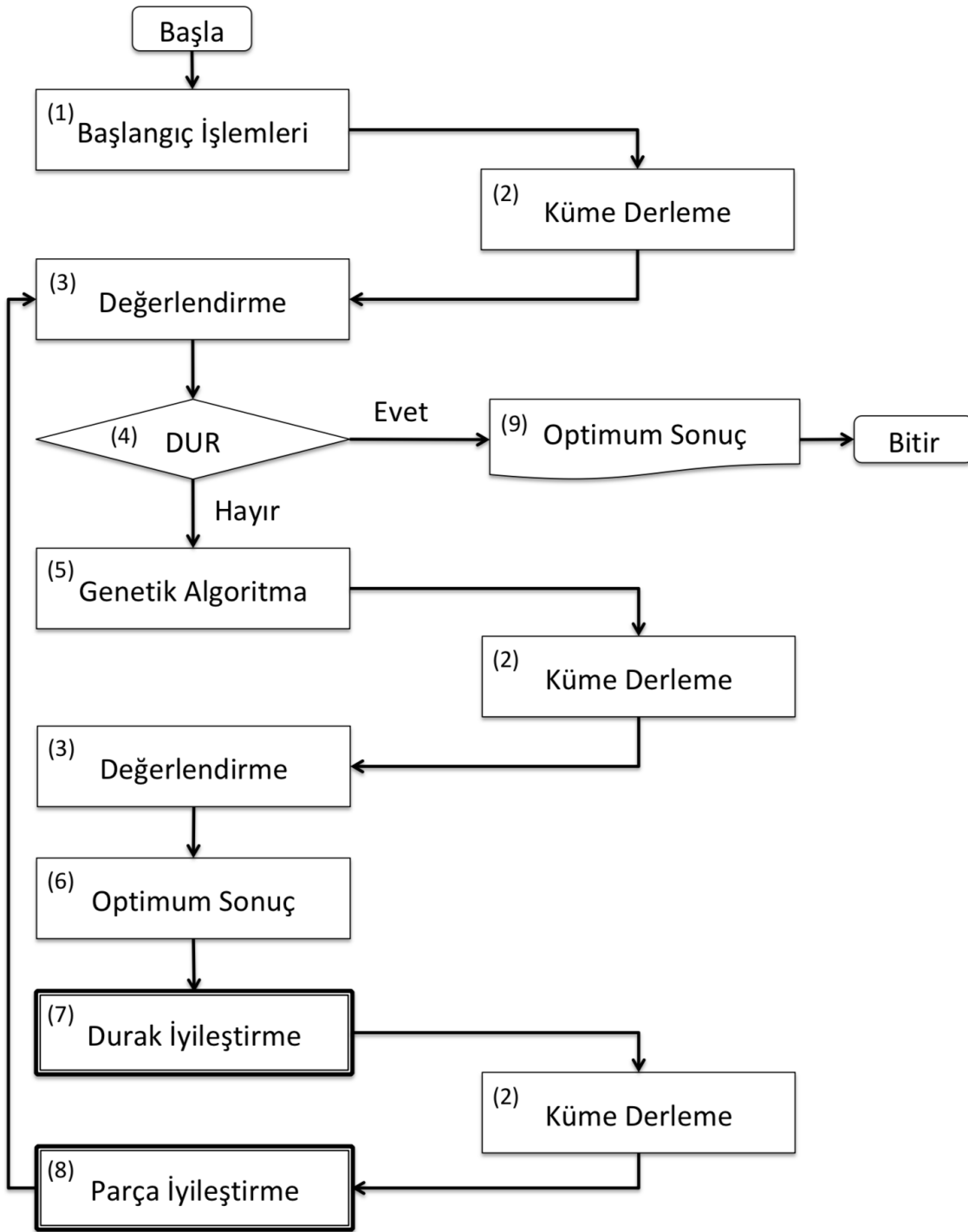
Çizelge 3.3. Yolcu sayılarına göre hesaplanan alternatif planlara ait ADP değerleri

Alternatif Plan	Güzergah	ADP	Toplam
1	1-4-5-6	2,001	3,974
	2-3-7	1,973	
2	1-6	1,900	3,976
	2-3-4-5-7	2,076	
3	1-4-6	2,003	4,164
	2-3-5-7	2,161	
4	1-5-6	1,940	3,969
	2-3-4-7	2,029	

Çizelge 3.3'te görüldüğü gibi en iyi alternatif plan 4 numaralı seçenektir. Yolcu sayıları hesaba katılmadığı durumda ise 1 numaralı seçenek en iyi alternatif çıkmıştır. Bu örnekte de görüldüğü eğer yolcuların tamamı için maksimum fayda sağlanmak isteniyorsa kesinlikle yolcu sayıları hesaba katılarak değerlendirme yapılmalıdır.

3.3. Akış Doğrultusu Yöntemi Kullanılarak Oluşturulan Hibrit Bir Meta-Sezgisel Yöntem

Güzergah hesaplamalarında meta-sezgisel yöntemlerden genetik algoritma literatürde karşımıza çıkan bir yöntemdir. Doğadaki evrimsel sürecin çalışma mekanizmasını prensip almıştır (Beasley vd. 1993; Holland 1992). Gezgin satıcı problemlerinin çözümünde literatürde verimli sonuçlar ürettiği tespit edildiği için bu çalışmada da kullanımı tercih edilmiştir. Genetik algoritma (GA) ile güzergah hesabı yaparken sabit boyutlu bir dizi oluşturulmakta ve dizi içerisinde düğüm noktaları rastgele sıralanmaktadır. Daha sonra bu dizilimin değişimiyle ilgili bazı işlemler uygulanarak en iyi dizilim aranmaktadır. Algoritmanın en temel özelliği sabit bir dizi uzunluğuna sahip olmasıdır. Fakat mevcut meta-sezgisel yöntemler örneklemin büyüklüğü arttıkça yeterli verimlilikte çözüm üretememektedir. Özellikle çoklu güzergah tasarımlarında bu sorun daha belirgin hale gelmektedir. Çoklu güzergah tasarımında sadece güzergah için belirlenen düğüm noktalarının sıralanması değil, aynı anda hangi düğüm noktasının hangi güzergah grubunda olacağı da optimize edilmektedir. Dolayısıyla bu tarz problemlerin çözümü için yeni arayışlar devam etmektedir. Bu çalışmada, toplu taşıma araçlarına yönelik optimum çoklu güzergah tasarımlarının oluşturulabilmesi için yeni bir yöntem önerilmektedir. Bu yöntem ile mevcut toplu taşıma hatlarına ait duraklar kümelerine ayrılarak, her bir küme içerisinde kalan hatlar için başlangıç ve bitiş durakları aynı olacak şekilde yeniden güzergah hesabı yapılmıştır. Amaç fonksiyonu *ADP* değerini minimize etmektedir. Problemdeki tüm durakların birer kere kullanılması kısıtı vardır. Durakların güzergah dizilimindeki yerleri optimizasyondaki tek değişkendir. Optimizasyona akış doğrultusu yöntemine yönelik iyileştirme adımları eklenerek genetik algoritmayla hibrit olarak çalışan bir model geliştirilmiştir (Şekil 3.6). Şekilde görüldüğü gibi model 9 ana bölümden oluşmaktadır. Bu bölümler genel akış şeması ile uyumlu olarak 3.3 numaralı başlığın alt başlıkları halinde detaylı olarak açıklanmıştır.



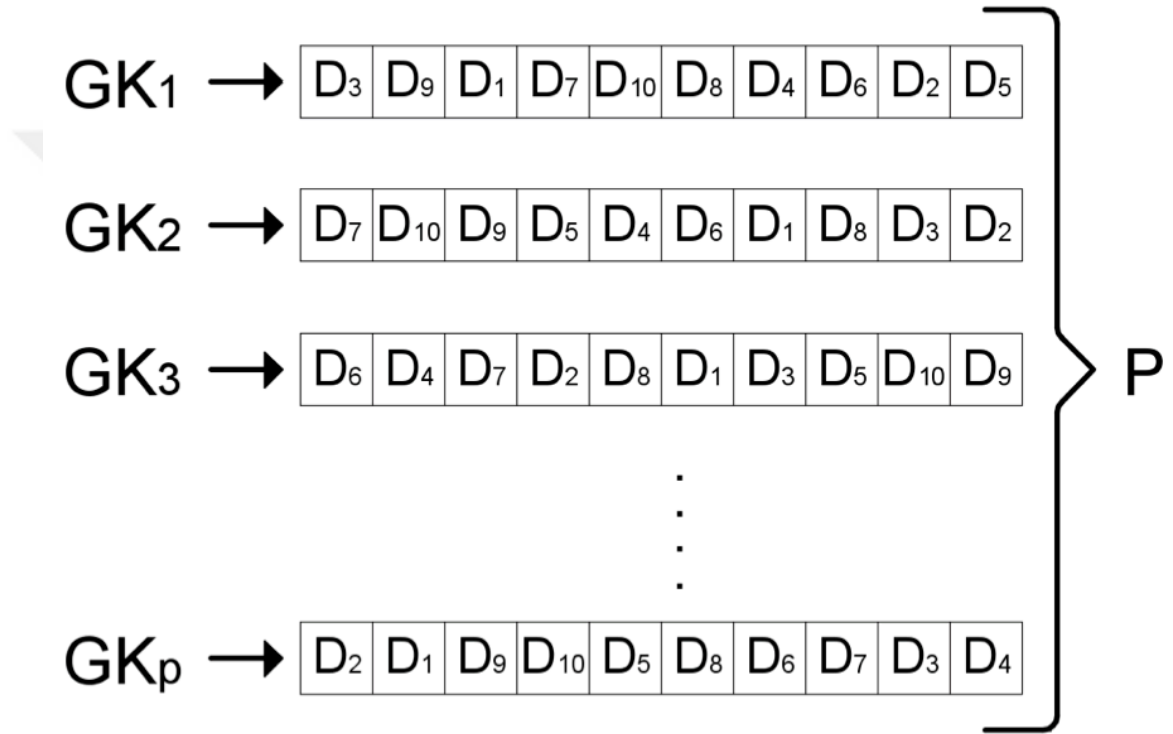
Şekil 3.6. Akış doğrultusu tabanlı optimum çoklu güzergah tasarımı yapan hibrit modelin genel akış şeması

3.3.1. Başlangıç işlemleri

Genetik algoritma kullanılarak hibrit bir model oluşturulacağı için bu modele uygun bir altyapı oluşturulması gerekmektedir. Bunun için ilk olarak rastgele sıralanmış ve sabit uzunlukta dizilerden oluşan bir topluluk oluşturulmuştur. Bu topluluk popülasyon, içerisindeki her bir dizi ise bir birey olarak düşünülmektedir. Bu bireyler

aslında optimum sonucu verecek durak dizilimlerini içeren güzergahlardır. Fakat ilk olarak rastgele durak sıralamayla elde edilmekte olan bu güzergahlar daha sonra üzerinde yapılacak değişikliklerle optimum sonucu verecek dizilimleri içerecektir.

Şekil 3.6’da genel akış diyagramı verilen algoritmada (1) numaralı bölümde burada bahsedilen belirli sayıda ve ağ üzerinde bulunan tüm durak numaralarının rastgele sıralanmasıyla elde edilen diziler oluşturulmaktadır. Birey olarak da adlandırılan bu diziler, içerisinde bir ya da daha fazla güzergah barındıran fakat güzergahların henüz belirgin olmadığı kümelerdir. Bu yüzden bu çalışmada bu diziler “Güzergah Kümeleri” ($GK_1, GK_2, GK_3, \dots, GK_p$), kümelerin tamamı ise “Popülasyon” (P) olarak adlandırılmıştır (Şekil 3.7).



Şekil 3.7. Rastgele durak dizilimlerinden oluşan bir popülasyon örneği

Şekillerde $D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9$ ve D_{10} olarak tanımlanan ifadeler durak numaralarını ifade etmektedir. Modelin çözüm aşamasından önce popülasyon içerisinde ne kadar güzergah kümesinin oluşturulacağı bilinmelidir. Popülasyon büyüklüğü optimizasyon problemin çözümü ve çözüm süresi üzerinde etkili bir değişkendir. Dolayısıyla popülasyonun ne çok büyük ne de çok küçük olması gerekmektedir. Bu yüzden popülasyonun büyüklüğü için bir alt limit getirilerek çözülecek problemdeki durak sayısına bağlı olarak belirlenmektedir. Optimizasyon yöntemiyle optimum sonuca ulaşabilmek için popülasyon üzerinde tekrarlı değişikliklerin uygulanması gerekmektedir. Popülasyon büyüklüğünde olduğu gibi tekrar sayısı da durak sayısına göre belirlenmektedir. Dolayısıyla modelin başlangıç işlemleri aşağıdaki gibi oluşturulmuştur.

$$N_p = \text{ceil}(100 + \frac{N_d}{4}) \quad (3.5)$$

$$N_i = 4000 + N_d \cdot 8 \quad (3.6)$$

$$RGK = \text{randperm}(N_d) \quad (3.7)$$

$$GK_p = RGK \quad \forall p \in \{1, 2, \dots, N_p\} \quad (3.8)$$

Burada,

N_d : problemdeki toplam durak sayısı,

N_p : popülasyondaki güzergah kümesi (birey) sayısı,

N_i : iterasyon sayısı,

RGK : rastgele üretilen güzergah kümesi dizisi,

GK_p : popülasyondaki p 'inci güzergah kümesidir.

Popülasyondaki birey sayısı minimum 100, popülasyon üzerinde uygulanacak işlemlerin tekrar sayısı ise minimum 4000 olacak şekilde (3.5) ve (3.6) formülleriyle belirlenmektedir. (3.5) numaralı formülde “*ceil*” fonksiyonla sayı yukarı yuvarlatılarak tam sayıya dönüştürülmektedir. (3.7) numaralı “*randperm*” fonksiyonu ile 1’den N_d ’ye yani toplam durak sayısına kadar olan, durak numaralarının rastgele sıralandığı bir dizi oluşturulmaktadır. Bu işlem N_p kadar tekrarlanarak (3.8) numaralı formülde görüldüğü gibi popülasyon oluşturulmaktadır.

3.3.2. Küme derleme

Şekil 3.6’da (2) numaralı bölümde kümelerin içerisindeki duraklar belirli kurallar çerçevesinde yeniden sıralanmaktadır. Bu sıralama işlemi küme derleme olarak adlandırılmıştır. Modelde aşağıdaki işlemler gerçekleştirilmektedir.

$$A = \{A_1, A_2, \dots, A_{N_{AB}}\}, B = \{B_1, B_2, \dots, B_{N_{AB}}\} \quad (3.9)$$

$$GK_p = D \setminus (A \cup \{B_{N_{AB}}\}) \quad (3.10)$$

$$GK_{p_1} = \{A_1\}, \quad GK_{p_{N_d}} = \{B_{N_{AB}}\} \quad (3.11)$$

$$A_j^* = B_{j-1}^* + 1 \quad \forall j \in \{2, \dots, N_{AB}\} \quad (3.12)$$

$$GK_{p_{A_j^*}} = \{A_j\} \quad (3.13)$$

Burada,

A : başlangıç durakları kümesi,

B : bitiş durakları kümesi,

D : tüm durakları içeren küme,

GK_{p_1} : popülasyondaki p 'inci güzergah kümesinin ilk elemanı,

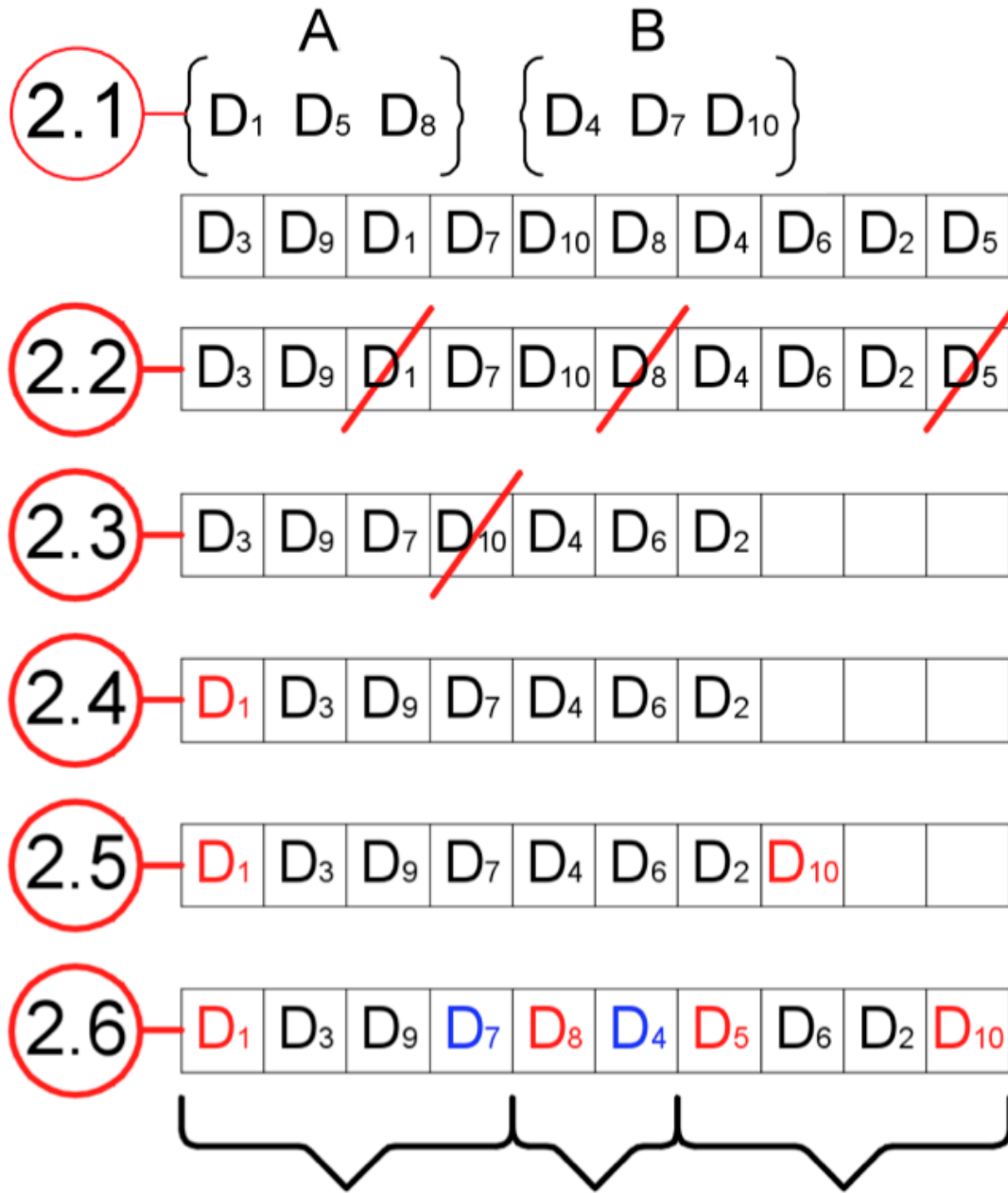
$GK_{p_{N_d}}$: popülasyondaki p 'inci güzergah kümesinin son elemanı,

$A_{N_{AB}}, B_{N_{AB}}$: A, B kümesinin N_{AB} 'inci elemanı,

A_j^*, B_j^* : A, B kümesinin j 'inci elemanının küme içindeki konumu,

N_{AB} : başlangıç bitiş durak çifti sayısıdır.

Küme derleme işlemi için ilk olarak güzergah kümesi içerisinde karışık olarak bulunan başlangıç ve bitiş durakları (3.9)'daki gibi tanımlanmaktadır. Başlangıç durakları ve sadece son bitiş durağı güzergah kümesinden (3.10)'daki çıkartıldıktan sonra ilk başlangıç durağı güzergah kümesinde ilk sıraya, son bitiş durağı da güzergah kümesinin en son sırasına (3.11)'deki formülle yerleştirilir. Diğer başlangıç duraklarının konumları ise (3.12)'de belirlenerek, (3.13)'teki formülle bu konuma yerleştirilir. Şekil 3.8'de küme derleme işlemi detaylı olarak gösterilmektedir.

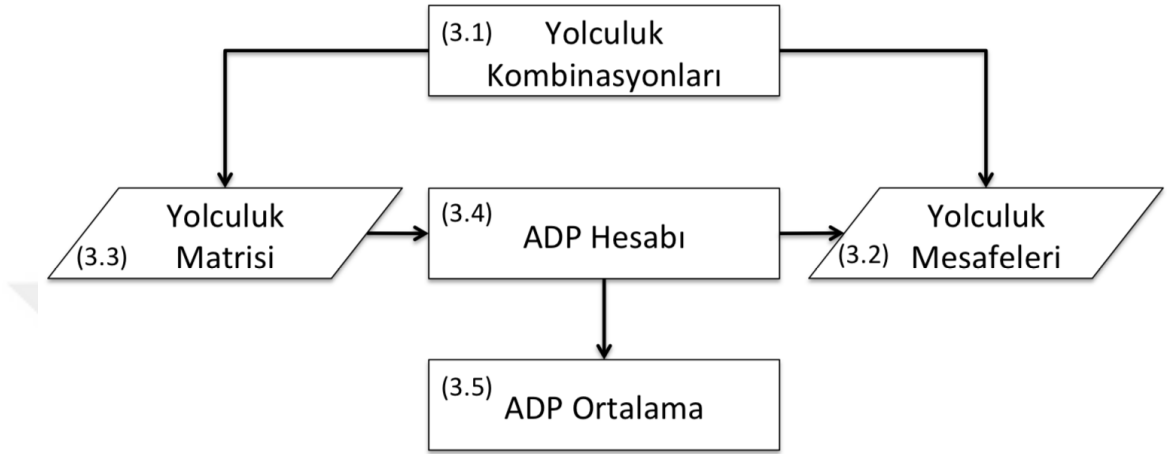


Şekil 3.8. Güzergah kümelerinin derleme işlemi örneği

Şekil 3.8’de (2.1) numaralı bölümde başlangıç ve bitiş durakları sırasıyla sisteme girilmektedir. Örnek olarak D_1 , D_5 ve D_8 başlangıç durakları, D_4 , D_7 ve D_{10} ise bitiş durakları olarak tanımlanmıştır. Sonra (2.2) numaralı bölümde güzergah kümesinden bütün başlangıç durakları ve (2.3) numaralı bölümde ise sisteme en son girilen bitiş durağı çıkarılmaktadır. Sisteme girilen ilk başlangıç durağı (2.4) numaralı bölümde birinci sıraya, sisteme son girilen bitiş durağı (2.5) numaralı bölümde son sıraya eklenmektedir. Sonra (2.6) numaralı bölümde, son bitiş durağı haricindeki bitiş duraklarının dizi içerisindeki sıralamada yerleri tespit edilmekte ve bu duraklardan sonra sırayla kalan başlangıç durakları diziyeye eklenmektedir. Böylece başlangıç bitiş durak çiftleri arasında kalan güzergah kümeleri belirlenmektedir.

3.3.3. Güzergah kümelerinin ADP hesabı

Belirlenen kümelerin performansları Şekil 3.6'da (3) numaralı bölümde, geliştirilen ADY ile ölçülmektedir. Bu ölçüm küme içerisinde bulunan güzergahların geometrisine ve yolcuların bu geometriden etkilenme durumlarına göre bir kalite değeri vermektedir. Şekil 3.9'da ADY ile yapılan hesapların detaylarını gösteren bir akış şeması verilmiştir.



Şekil 3.9. ADY ile güzergah kümelerinin performanslarının değerlendirilmesi akış şeması

Şemayı bir örnek üzerinden açıklamak gerekirse Şekil 3.8'de (2.6) numaralı bölümde oluşturulan güzergahlardan ilkinin ($D_1-D_3-D_9-D_7$) inceleyebiliriz. Öncelikle Şekil 3.9'da (3.1) numaralı bölümde bu durak dizisine yani güzergaha ait yolculuk kombinasyonları ve kombinasyon sayısı belirlenmektedir. Bu kombinasyonlar D_1-D_3 , D_1-D_9 , D_1-D_7 , D_3-D_9 , D_3-D_7 ve D_9-D_7 nolu duraklar arasında yapılabilecek yolculuklardır ve kombinasyon sayısı (n_k) 6'dır. Bu işlemler aynı kümede bulunan diğer güzergahlar içinde yapılmaktadır. Belirlenen yolculuk kombinasyonlarındaki duraklar arasındaki mesafe verileri (M_{ij}), duraklara ait koordinat verileri üzerinden (3.2) numaralı bölümde Google Maps uygulamasından bir yazılım ile çekilerek sisteme girilmektedir. Yine aynı yolculuk kombinasyonlarına ait duraklar arası yolculuk sayısı (Y_{ij}) verileri, şehirlerde yapılan ulaşım planlamasına yönelik anketlerle oluşturulan yolculuk matrislerinden (3.3) numaralı bölümde çekilmektedir. Sonra her bir güzergah için ADP (3.4) numaralı bölümde hesaplanmaktadır. Kümede bulunan her güzergah için ADP değerleri hesaplandıktan sonra ortalaması alınarak (3.5) numaralı bölümde güzergah kümesine ait ortalama ADP değeri belirlenmektedir. Bu değer küçük çıkması güzergah kümesinin daha iyi bir performansa sahip olduğu anlamına gelmektedir. Modelde hesaplamalar (3.2), (3.3) ve (3.4) numaralı formüllerle yapılmaktadır.

3.3.4. Durdurma kriteri

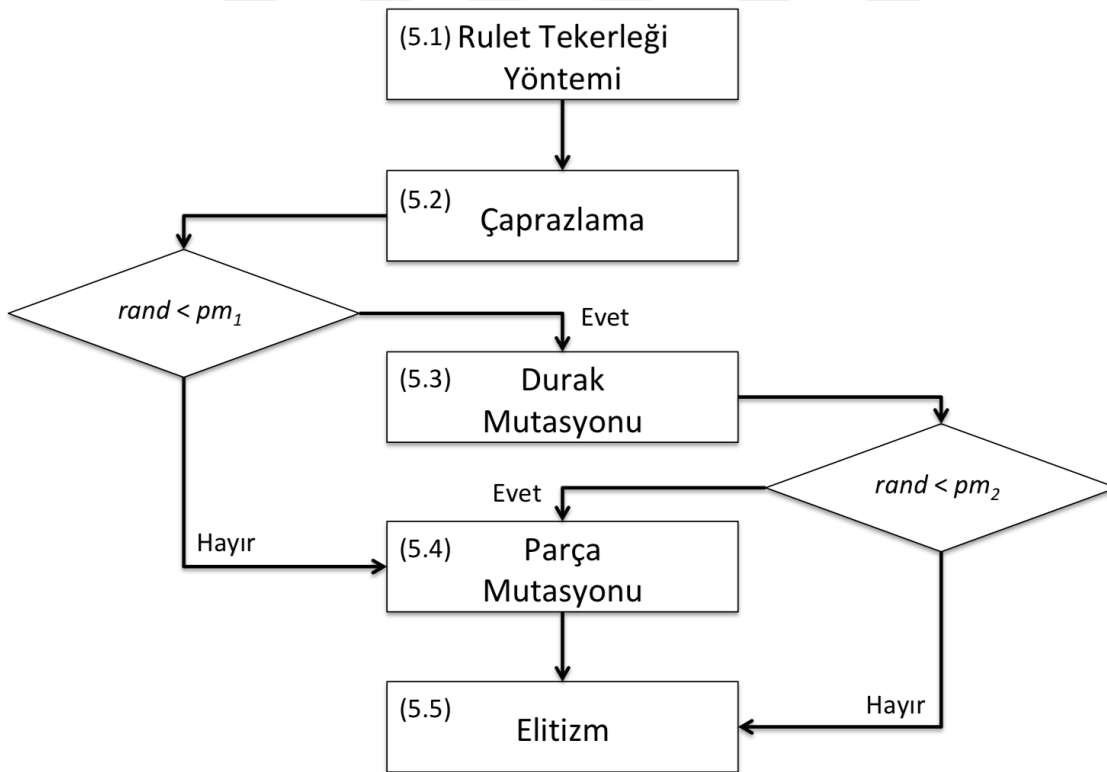
Bu bölümde Şekil 3.6'da (4) numaralı bölümde belirlenen iterasyon sayısına ulaşıp ulaşılamadığı kontrol edilir. Burada yeterli iterasyon sayısı (3.6) formülü ile hesaplanmaktadır. Yapılan denemeler sonucunda toplam durak sayısına bağlı olarak

hesaplanan bu değerin minimum 4000 olması uygun görülmüştür. Özellikle durak sayısının 200'ün üzerine çıktığı problemlerde 4000 iterasyonun altında yapılan çözümlerde verimin oldukça düştüğü görülmüştür. Burada belirlenen iterasyon sayısına ulaşıldığında güncelleştirme işlemleri son bularak optimum sonuç belirlenmektedir. Durdurma kriteri sağlanmadığı takdirde ise bir sonraki genetik algoritma adımına geçilmektedir.

3.3.5. Genetik algoritma

Şekil 3.6'da (5) numaralı bölümde, popülasyon içerisindeki her bir güzergah kümesine ait durakların dizilimleriyle ilgili güncellemeler yapılmaktadır. Bu işlem küme içerisinde bulunan güzergahların başlangıç ve bitiş duraklarının yerlerinde değişikliğe neden olabilmektedir. Bu yüzden burada yapılan işlemlerden sonra küme yapısının korunması için küme derleme işlemi (2) numaralı bölümde tekrar yapılarak kümelerin performansı (3) numaralı bölümde yeniden değerlendirilmektedir.

Durak dizilimleriyle ilgili yapılan güncellemeler genetik algoritma adımlarını içermektedir (Şekil 3.10).



Şekil 3.10. Genetik algoritma işlem adımları akış şeması

İlk olarak oluşturulan güzergah kümesi popülasyonu, Şekil 3.10'da (5.1) numaralı bölümde rulet tekerleği yöntemine göre yeniden oluşturulmaktadır. (5.2) numaralı bölümde bu popülasyon içerisindeki güzergah kümeleri iki gruba ayrılmaktadır. Bu gruptaki bireyler birbirleriyle eşleştirilerek, çiftler arasında çaprazlama olarak adlandırılan parça değişimleri yapılmaktadır. (5.3) numaralı bölümde popülasyonun içerisinde rastgele güzergah kümeleri seçilerek dizi içerisindeki bazı

durakların yerleri rastgele değiştirilmektedir. Bu işlem “*rand*” fonksiyonu ile 0-1 arasında üretilen rastgele sayının “ $pm_1 = 0,01$ ” mutasyon oranından küçük çıkması durumuna göre belirlenmektedir. Yine popülasyon içerisinde rastgele seçilen bireyler üzerinde (5.4) numaralı bölümde rastgele parça değişimleri uygulanmaktadır. Burada da rastgele seçim “ $pm_2 = 0,02$ ” mutasyon oranına göre yapılmaktadır. Optimizasyon yöntemi uygulanırken her iterasyonda popülasyondaki en iyi güzergah kümesi belirlenmektedir. Bu kümenin bir sonraki iterasyonda güncellenerek kaybolması ve daha kötü bir sonucun en optimum sonuç olma ihtimali vardır. Bunu engellemek adına her iterasyonda belirlenen optimum sonuç (5.5) numaralı bölümde elitizm adı altında bir sonraki popülasyona eklenmektedir.

3.3.5.1. Rulet tekerleği yöntemi

Bu yöntemle, optimizasyon sırasında popülasyonda kötü sonuç veren bireylerin elenerek gen aktarımına dahil edilmemesi istenmektedir. Böylece daha iyi sonuç veren bireyler üzerinden gen aktarımı yapılarak daha verimli sonuçlar üretilmesi planlanmaktadır. Bu problemde güzergahların daha verimli olması sayısal olarak *ADP* değeri ile ölçülmektedir. *ADP* değerinin küçük çıkması kümenin daha verimli olduğu anlamına gelmektedir. Dolayısıyla daha büyük *ADP* değerine sahip güzergah kümelerinin yerine daha küçük *ADP* değerine sahip güzergah kümeleri kopyalanarak popülasyon güncellenmektedir. Buradaki eleme ve seçim işlemi aşağıdaki gibi modellenmiştir.

$$P_i = \frac{1}{ADP_i} / \sum \frac{1}{ADP_i} \quad (3.14)$$

$$[sP, I] = sort(P) \quad (3.15)$$

$$SumP_i = \sum_{j=1}^i sP_j \quad \forall j \in (1, 2, \dots, i) \quad (3.16)$$

$$SumP_{i-1} > rand > SumP_i \rightarrow SGK = I_i \quad \forall i \in \{1, 2, \dots, N_p\} \quad (3.17)$$

Burada,

P : güzergah kümelerinin seçilme olasılığını gösteren dizi,

P_i : P dizisindeki i 'inci güzergah kümesinin seçilme olasılığı,

sP : P dizisinin küçükten büyüğe doğru sıralanması ile elde edilen dizi,

sP_i : sP dizisindeki i 'inci güzergah kümesinin seçilme olasılığı,

I : sP dizisinin P dizisine göre olan sıra numaraları,

$SumP_i$: sP dizisindeki i 'inci güzergah kümesinin kümülatif seçilme olasılığı,

SGK : seçilen güzergah kümesidir.

Popülasyondaki her güzergah kümesi ilk olarak (3.14) numaralı formülle ağırlıklandırılmakta ve seçilme olasılıkları hesaplanmaktadır. Sonra (3.15)'te “*sort*” fonksiyonu ile bu olasılıklar küçükten büyüğe doğru sıralanmaktadır. Sıralanan bu olasılık dizisi (3.16)'da kümülatif olarak toplanarak, 0 ile 1 arasında bir dağılım elde edilmektedir. Son olarak (3.17)'de popülasyondaki her bir birey için rastgele bir sayı üretilmektedir. Bu sayı kümülatif olarak sıralanan olasılık dizisinde hangi aralığa denk gelirse, o güzergah kümesi seçilmektedir.

3.3.5.2. Çaprazlama

Rulet tekerleği yöntemiyle güncellenen popülasyondaki güzergah kümelerinin tamamı ilk olarak çaprazlama işlemine tabi tutulmaktadır. Burada popülasyondaki her güzergah kümesi, popülasyondan başka bir güzergah kümesiyle çift olacak şekilde gruplandırılmaktadır. Her çift için rastgele bir kırılma noktası belirlenerek, iki küme de aynı noktadan iki parçaya ayrılmaktadır. Daha sonra kümelerdeki ilk parçalar kendi arasında yer değiştirerek tekrar birleşmektedir. Böylece iki güzergah kümesi arasında çaprazlama işlemi tamamlanmaktadır. Bu işlemler modelde aşağıdaki formüllerle gerçekleştirilmektedir.

$$GKCh^1 = GK_{1+2 \cdot (i-1)} \quad (3.18)$$

$$GKCh^2 = GK_{2+2 \cdot (i-1)} \quad (3.19)$$

$$c_s = \text{ceil}((N_d - 1) \cdot \text{rand}) \quad (3.20)$$

$$\text{ind}_k = \text{find}(GKCh^1 = GKCh_k^2) \quad \forall k \in \{1, 2, \dots, c_s\} \quad (3.21)$$

$$GKCh_{\text{ind}_k}^1 = GKCh_k^1 \quad (3.22)$$

$$GKCh_k^1 = GKCh_k^2 \quad (3.23)$$

$$GKCh_{2\text{ind}_k} = GKCh_k^2 \quad (3.24)$$

$$GKCh_k^2 = GKCh_k^1 \quad (3.25)$$

$$GK_{1+2 \cdot (i-1)} = GKCh^1 \quad (3.26)$$

$$GK_{2+2 \cdot (i-1)} = GKCh^2 \quad (3.27)$$

Burada,

$GKCh^1$: popülasyonda oluşturulan güzergah kümesi çiftlerinin birinci elemanı,

$GKCh^2$: popülasyonda oluşturulan güzergah kümesi çiftlerinin ikinci elemanı,

c_s : çaprazlamanın yapılacağı dizi içindeki kırılma noktası,

ind_k : ikinci güzergah kümesindeki k 'inci durağın birinci güzergah kümesindeki sırasıdır.

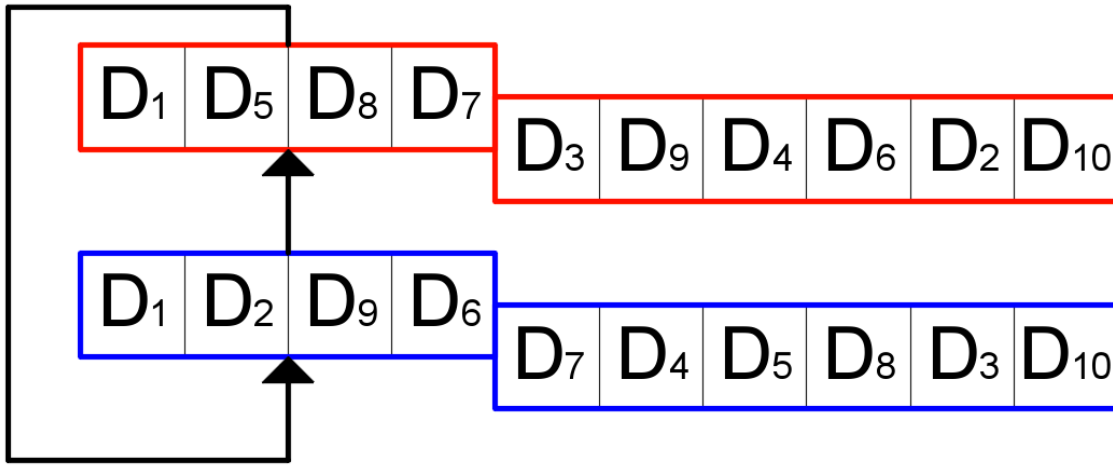
Çaprazlama işlemi için ilk olarak (3.18) ve (3.19)'daki formüllerle güzergah kümesinden bir çift seçilmektedir. Bu bireylere örnek olarak Şekil 3.11 oluşturulmuştur. Şekilde görüldüğü üzere her iki dizi de (3.20) hesaplanan kırılma noktası ile aynı noktadan iki parçaya ayrılmıştır.

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

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

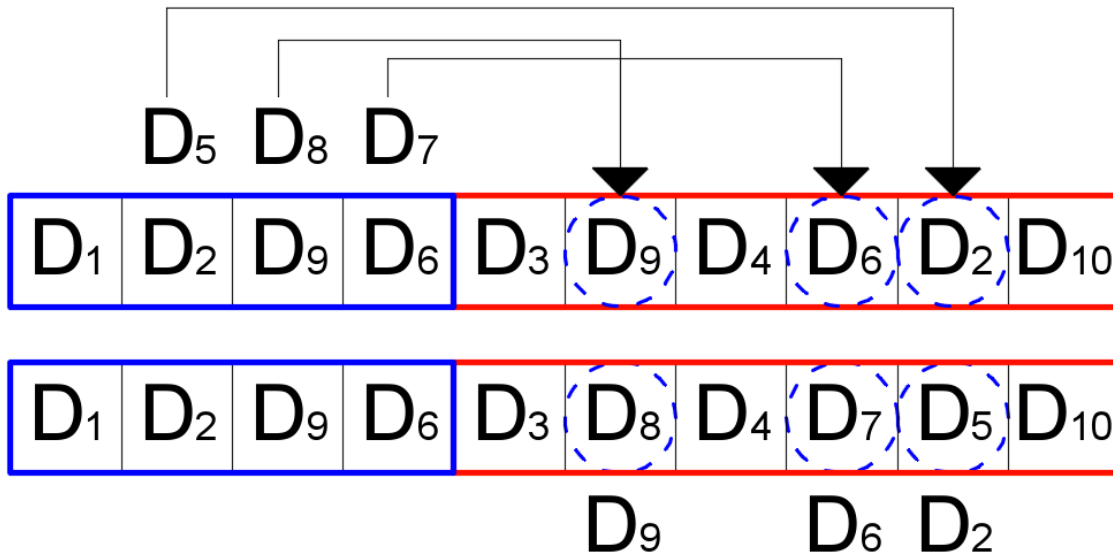
Şekil 3.11. Çaprazlama için örnek olarak seçilen $GKCh^1$ ve $GKCh^2$ güzergah kümeleri

Kırılma noktası belirlendikten sonra Şekil 3.12'de görüldüğü gibi iki dizinin ilk parçaları yer değiştirmektedir.



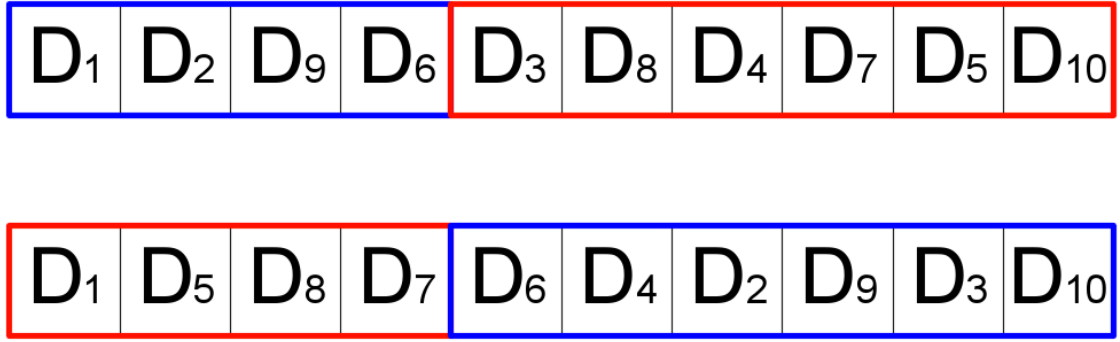
Şekil 3.12. İlk parçaların değişimi

Bu çaprazlama işlemi gerçekleştiğinde aynı dizi içerisinde bazı duraklar kaybolurken bazı duraklardan ikişer adet bulunabilmektedir. Şekil 3.13'te ilk dizinin parça değişimine uğramış hali görülmektedir. Değişim sonucunda dizi de D_5 , D_8 ve D_7 durakları kümenin dışında kalırken, D_2 , D_9 ve D_6 durakları kümede ikişer defa bulunmaktadır. Bu durumun önüne geçmek için dizinin ikinci parçasında bulunan aynı duraklar, küme dışında kalan duraklarla değiştirilmektedir. Bu düzenleme için (3.21), (3.22), (3.23), (3.24) ve (3.25) numaralı formüller kullanılmaktadır. (3.21) numaralı formülde “*find*” fonksiyonu ile $GKCh_k^2$ durağının $GKCh^1$ kümesindeki sıra numarası tespit edilmektedir.



Şekil 3.13. $GKCh^1$ güzergah kümesi için ikinci parçadaki düzenleme işlemi

İkinci parçalarda yapılan değişiklikler sonucunda Şekil 3.14'te görülen çaprazlanmış bireyler elde edilmektedir. Bunlar (3.26) ve (3.27) numaralı formüllerle yeniden popülasyondaki yerlerini almaktadır.



Şekil 3.14. $GKCh^1$ ve $GKCh^2$ güzergah kümelerinin çaprazlanmış halleri

3.3.5.3. Rastgele seçilen durakların yer değişimi mutasyonu

Genetik algoritma adımlarından biri olan gen mutasyonları ile yapılan işlemlerden biri dizi içerisinde rastgele seçilmiş düğüm noktalarının yer değiştirmesidir. Bu işlem güzergah kümesinin boyutu aşmayan bir rastgele sayı üretilir. Bu sayı kadar güzergah kümesi içerisinde durak rastgele seçilir. Bu durakların yerleri rastgele değiştirilir ve güzergah kümesi güncellenir. Şekil 3.10'daki (5.3) numaralı adımda gerçekleştirilen bu mutasyonlar popülasyondaki her bireye uygulanmamaktadır. Bu adımın hemen öncesinde yapılan bir sorgulamayla, popülasyonun yaklaşık %1'i bu işleme tabi tutulmaktadır. Bu değer mutasyon oranı olarak bilinmektedir. Bu oranın artırılması çözüm aralığının daralmasına yol açmaktadır. Bu yüzden yüksek oranlarda mutasyon uygulamak optimum çözüme ulaşmayı zorlaştırmaktadır. Bu mutasyon modelde aşağıdaki formüller kullanılarak gerçekleştirilmektedir.

$$rnc = \text{ceil}((N_d - 1) \cdot rand) \quad (3.28)$$

$$PGK = RGK(2:rnc) \quad (3.29)$$

$$GK_r(PGK) = PGK(\text{randperm}(rnc - 1)) \quad (3.30)$$

Burada,

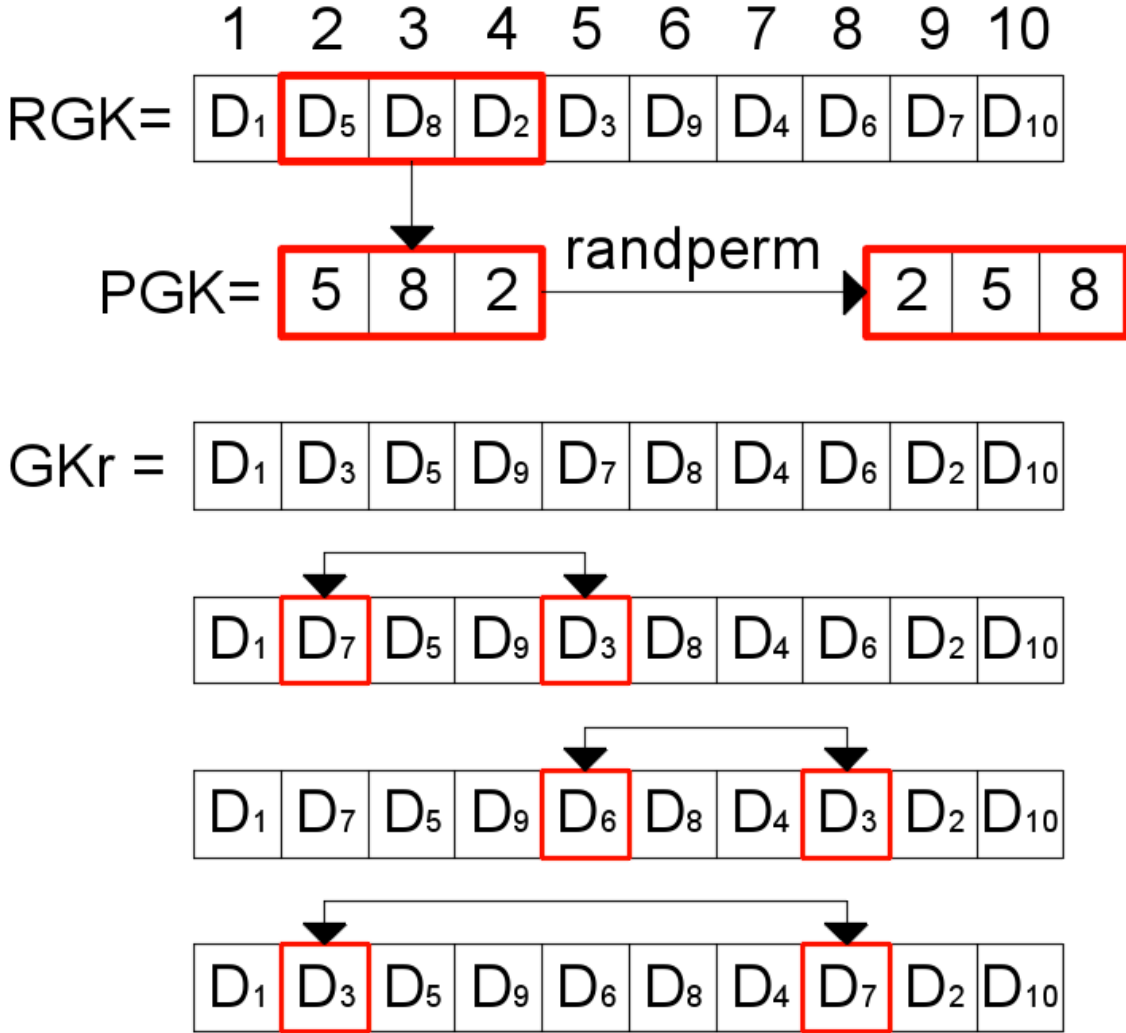
rnc : güzergah kümesinden rastgele seçilecek durak sayısı,

PGK : seçilen duraklardan oluşan rnc elemanlı güzergah kümesi parçası,

GK_r : rastgele seçilen bir güzergah kümesidir.

Şekil 3.15'te bu mutasyon için bir örnek hazırlanmıştır. Formül (3.28)'de $rnc = 4$ değeri için (3.29)'da önce rastgele bir güzergah kümesi oluşturulmuş, daha sonra bu küme içinde bulunan 2'den $rnc = 4$ 'e kadar olan duraklar seçilerek, PGK kümesine ait durak numaraları 5, 8 ve 2 olarak belirlenmiştir. Bu numaraların yerleri rastgele değiştirilerek 2, 5 ve 8 durak numaraları elde edilmiştir. Daha sonra rastgele

seçilen bir güzergah kümesi içerisinde belirlenen durak numaraları arasında sırayla yer değiştirme işlemi uygulanmıştır. Burada ilk olarak 5 ile 2 numaradaki duraklar, sonra 8 ile 5 numaradaki duraklar, son olarak ise 2 ile 8 nolu durakların yerleri değiştirilerek (3.30) numaralı formül ile mutasyon tamamlanmıştır. Mutasyona uğrayan bu güzergah kümesi popülasyondaki eski hali ile değiştirilerek popülasyon güncellenmiştir.



Şekil 3.15. Rastgele seçilen güzergah kümesinde durak yer değişimi örneği

3.3.5.4. Güzergahtaki iki parçanın değişimi mutasyonu

Genetik algoritma mutasyonlarından olan parça değişimi işlemi farklı parça sayılarında gerçekleştirilebilmektedir. Fakat bu çalışmada iki parçanın değişimi tercih edilmiştir. Dizinin ilk ve son durağı parça değişimi işleminin dışında tutulmaktadır. Önce dizi içerisinde bir kırılma noktası belirlenmektedir. Bu kırılan noktadan dizi iki parçaya ayrılmakta ve parçalar yer değiştirmektedir. Bu adım için mutasyon oranı %2'dir. Modelde işlemler aşağıdaki formüllerle gerçekleştirilmektedir.

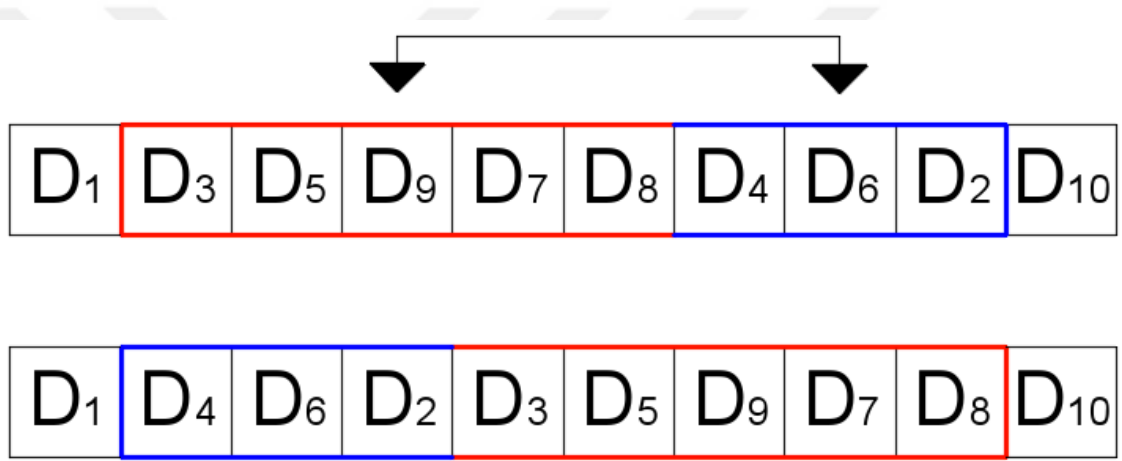
$$c_p = \text{ceil}((N_d - 2) \cdot \text{rand}) \quad (3.31)$$

$$GK_r = \left[GK_r((c_p + 1): N_d - 1) \quad GK_r(2: c_p) \right] \quad (3.32)$$

Burada,

c_p : parça değişiminin gerçekleşeceği dizi içerisindeki kırılma noktasıdır.

Şekil 3.16'da görüldüğü gibi kırılma noktası (3.31) numaralı formülle belirlendikten sonra (3.32) numaralı formülle yer değiştirme işlemi gerçekleştirilmektedir. Dizideki ilk ve son duraklar değişim işlemine tabi tutulmamaktadır.



Şekil 3.16. Rastgele seçilen güzergah kümesinde iki parçanın değişimi örneği

3.3.5.5. Elitizm

Genetik algoritmanın son adımı olarak elitiz işlemi uygulanmaktadır. Bu işlem optimum sonucun daha kötüye evrilmemesini sağlamak için gerçekleştirilmektedir. Popülasyon her iterasyonda güncellendikten sonra sonuçlar her zaman daha iyi çıkmayabilir. Bu durumda optimum sonuç bir önceki sonuca göre daha kötü bir çözüm olabilir. Bunun önüne geçmek için her iterasyondaki en optimum sonuç hiçbir işleme uğramadan bir sonraki iterasyon için popülasyona doğrudan aktarılır. Böylece optimum çözümünden daha kötü bir sonuca doğru evrilmenin önüne geçilmiş olur. Bu durum modelde aşağıdaki formüllerle gerçekleştirilmektedir.

$$[ADP_{min}^p, I_b^p] = \min(ADP^p) \quad (3.33)$$

$$GK_1 = GK_{I_b^p} \quad (3.34)$$

Burada,

ADP^p : her iterasyonda genetik algoritma adımları başlamadan önceki popülasyona ait güzergah kümelerinin ADP 'lerini içeren dizi,

ADP_{min}^p : ADP^p dizisindeki en küçük değer,

I_b^p : ADP^p dizisindeki en küçük değeri veren sıra numarasıdır.

(3.33) numaralı formül ile bir önceki iterasyona ait popülasyondaki en iyi ADP değerini veren güzergah kümesinin popülasyondaki sıra numarası ve ADP değeri belirlenmektedir. Bu güzergah kümesi (3.34) numaralı formülle bir sonraki popülasyonun ilk bireyi olarak sisteme eklenmektedir. Böylece genetik algoritma adımları tamamlanmaktadır.

3.3.5.6. Genetik algoritma karşılaştırma örnekleri

Genetik algoritma güzergah tasarımı için hibrit bir model ile birlikte kullanılmadan önce üç matematiksel değerlendirme fonksiyonu ile test edilmiştir. Bu fonksiyonlar Çizelge 3.4'te verilmiştir.

Çizelge 3.4. Karşılaştırma fonksiyonu özellikleri (Aydogdu vd. 2017)

Fonksiyon	Formülasyon	Aralık
Sphere	$F_1(\mathbf{x}) = \sum_{i=1}^n (x_i^2)$	$[-5,12; 5,12]$
Rastrigin	$F_2(\mathbf{x}) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$[-5,12; 5,12]$
Griewank	$F_3(\mathbf{x}) = \sum_{i=1}^n \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^n \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600; 600]$

Çizelge 3.4'teki fonksiyonları için $n=5$, $n=30$ ve $n=100$ olarak üç farklı karar değişkeni sayısı kullanılarak çözüm üretilmiştir. Bu çözümler Çizelge 3.5, Çizelge 3.6 ve Çizelge 3.7'de gösterilmiştir. Ayrıca genetik algoritma (GA) için üretilen sonuçlar Aydogdu vd. (2017)'nin çalışmasından alınan yapay arı kolonisi (YAK), biyocoğrafya tabanlı optimizasyon (BTO), beyin fırtınası optimizasyonu (BFO), harmoni arama optimizasyonu (HAO) ve ateş böceği algoritması (ABA) test verileriyle karşılaştırılmıştır.

Çizelge 3.5. Genetik algoritma karşılaştırma örnekleri istatistiksel veriler (n=5)

		YAK	BTO	BFO	HAO	ABA	GA
F1	Ortalama	5,28e-11	6,99e-5	2,29e-4	5,86e-5	3,73e-7	0,0004
	Std. Sp.	5,08e-11	6,54e-5	1,27e-4	1,03e-4	3,09e-5	0,0006
	Medyan	3,41e-11	4,35e-5	2,24e-4	1,77e-5	3,17e-5	0,0001
	Min.	4,2e-12	5,31e-6	3,42e-5	1e-7	6e-6	7,48e-8
	Maks.	1,78e-10	0,0003	4,84e-4	5,18e-4	1,54e-6	0,0028
F2	Ortalama	8,43e-6	0,0131	0,975	0,0215	1,791	0,0605
	Std. Sp.	8,43e-6	0,0136	0,637	0,0373	0,9143	0,0908
	Medyan	4,84e-6	0,0101	1,04	0,0051	1,9900	0,0178
	Min.	1,2e-7	0,0005	0,0231	0	0,0001	5,6e-5
	Maks.	3,16e-5	0,0651	2,17	0,1372	2,985	0,3876
F3	Ortalama	0,0136	0,0694	0,687	0,3492	0,0300	0,4590
	Std. Sp.	0,0065	0,0256	0,418	0,2301	0,0218	0,2315
	Medyan	0,0137	0,0626	0,643	0,2938	0,0229	0,4070
	Min.	0,0003	0,0272	0,163	0,0023	0,0101	0,1355
	Maks.	0,0292	0,1390	1,590	0,7713	0,0948	1,0110

Çizelge 3.6. Genetik algoritma karşılaştırma örnekleri istatistiksel veriler (n=30)

	F1	YAK	BTO	BFO	HAO	ABA	GA
F1	Ortalama	0,4114	0,000631	0,0419	1,0389	1,819e-5	0,00052
	Std. Sp.	0,281	0,000219	0,0051	0,5401	4,92e-6	0,00151
	Medyan	0,3935	0,000602	0,0416	0,8989	1,721e-5	9,89e-5
	Min.	0,0265	0,000306	0,0321	0,555	9,79e-6	2,62e-6
	Maks.	1,2233	0,001162	0,05	1,9489	3,303e-5	0,00823
F2	Ortalama	62,841	0,117184	42,7192	24,3984	29,7863	0,04996
	Std. Sp.	7,3062	0,044030	6,3649	21,9022	5,8596	0,10208
	Medyan	64,5771	0,107645	43,3136	18,7426	29,8518	0,00790
	Min.	52,4758	0,051570	29,717	9,3234	18,9075	6,89e-5
	Maks.	77,3247	0,221349	51,9668	62,5372	39,8013	0,50366
F3	Ortalama	1,9129	0,328980	15,9278	6,0126	0,0083	1,1129
	Std. Sp.	1,0371	0,093773	3,8231	0,2969	0,002	0,0424
	Medyan	1,5246	0,323460	15,7031	5,9205	0,0082	1,1100
	Min.	1,1524	0,151963	11,1064	5,6253	0,0054	1,0443
	Maks.	5,2249	0,516959	22,802	6,3507	0,0137	1,2249

Çizelge 3.7. Genetik algoritma karşılaştırma örnekleri istatistiksel veriler (n=100)

		YAK	BTO	BFO	HAO	ABA	GA
F1	Ortalama	230,82	0,0219	0,3696	76,08	2,211e-4	0,000554
	Std. Sp.	19,31	0,0030	0,0222	6,31	2,03e-5	0,000966
	Medyan	237,40	0,0219	0,372	78,08	2,216e-4	0,000140
	Min.	198,69	0,0170	0,3406	69,01	2,001e-4	2,287e-8
	Maks.	249,13	0,0302	0,3937	81,15	2,524e-4	0,004533
F2	Ortalama	753,30	1,5965	164,65	415,57	214,1549	3,890432
	Std. Sp.	12,26	0,2806	44,16	171,37	30,192	0,532961
	Medyan	756,67	1,6024	146,38	360,56	218,9371	3,892449
	Min.	735,14	0,9935	135,42	300,22	169,1787	3,146262
	Maks.	767,09	2,4601	230,41	713,72	246,7955	5,166629
F3	Ortalama	817,38	0,7864	83,53	271,33	0,0152	0,085426
	Std. Sp.	63,15	0,1172	18,45	28,44	0,0011	0,140084
	Medyan	844,57	0,7768	84,52	268,30	0,0151	0,047670
	Min.	722,34	0,6081	60,97	244,52	0,0136	0,000175
	Maks.	879,08	1,1775	104,08	301,17	0,0165	0,661835

Çizelge 3.5, Çizelge 3.6 ve Çizelge 3.7'deki test sonuçlarına göre genetik algoritma matematiksel değerlendirme fonksiyonlarında tatmin edici sonuçlar ortaya koymuş hatta bazı fonksiyonlarda tüm algoritmalar arasında en iyi çözümü üretmiştir. Dolayısıyla güzergah optimizasyonu için oluşturulacak modelde kullanılması uygundur.

3.3.6. Genetik algoritma sonundaki optimum güzergah kümesinin belirlenmesi

Genetik algoritma adımları tamamlandıktan sonra en iyi performansı veren güzergah kümesi Şekil 3.6'da (6) numaralı bölümde tespit edilmektedir. Bu işlem daha sonra uygulanacak iyileştirme adımlarının uygulanacağı güzergah kümesini belirlenmesini sağlamaktadır. Modelde aşağıdaki formüllerle belirlenmektedir.

$$[ADP_{min}^g, I_b^g] = \min(ADP^g) \quad (3.35)$$

$$GK_{opt}^g = GK_{I_b^g} \quad (3.36)$$

Burada,

ADP^g : genetik algoritma adımları sonrasındaki ADP 'lerini içeren dizi,

ADP_{min}^g : ADP^g dizisindeki en küçük değer,

I_b^g : ADP^g dizisindeki en küçük değeri veren sıra numarası,

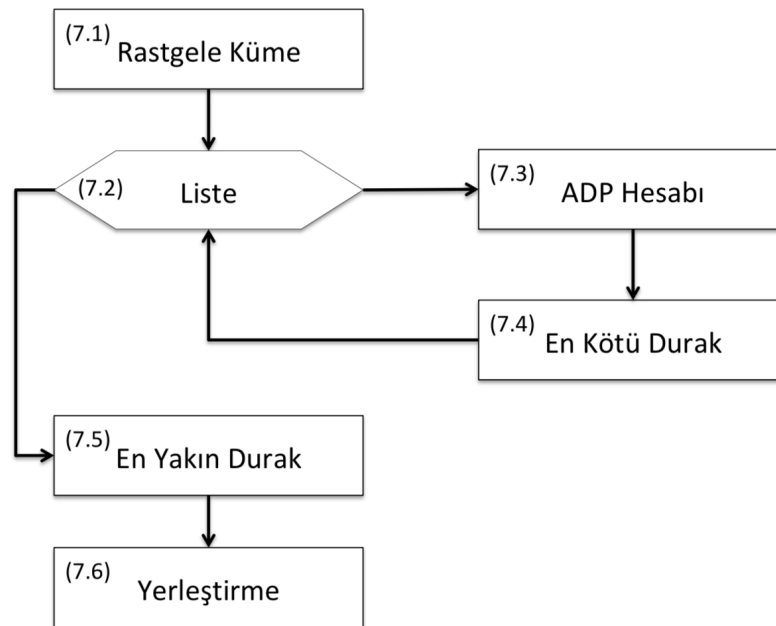
GK_{opt}^g : genetik algoritma adımları sonrasındaki optimum güzergah kümesidir.

Bu adımda (3.35) numaralı formülle genetik algoritma işlemleri sonundaki popülasyonda en optimum güzergah kümesini veren bireyin sıra numarası tespit edilir. Bu birey (3.36) numaralı formül ile optimum güzergah kümesi olarak atanır.

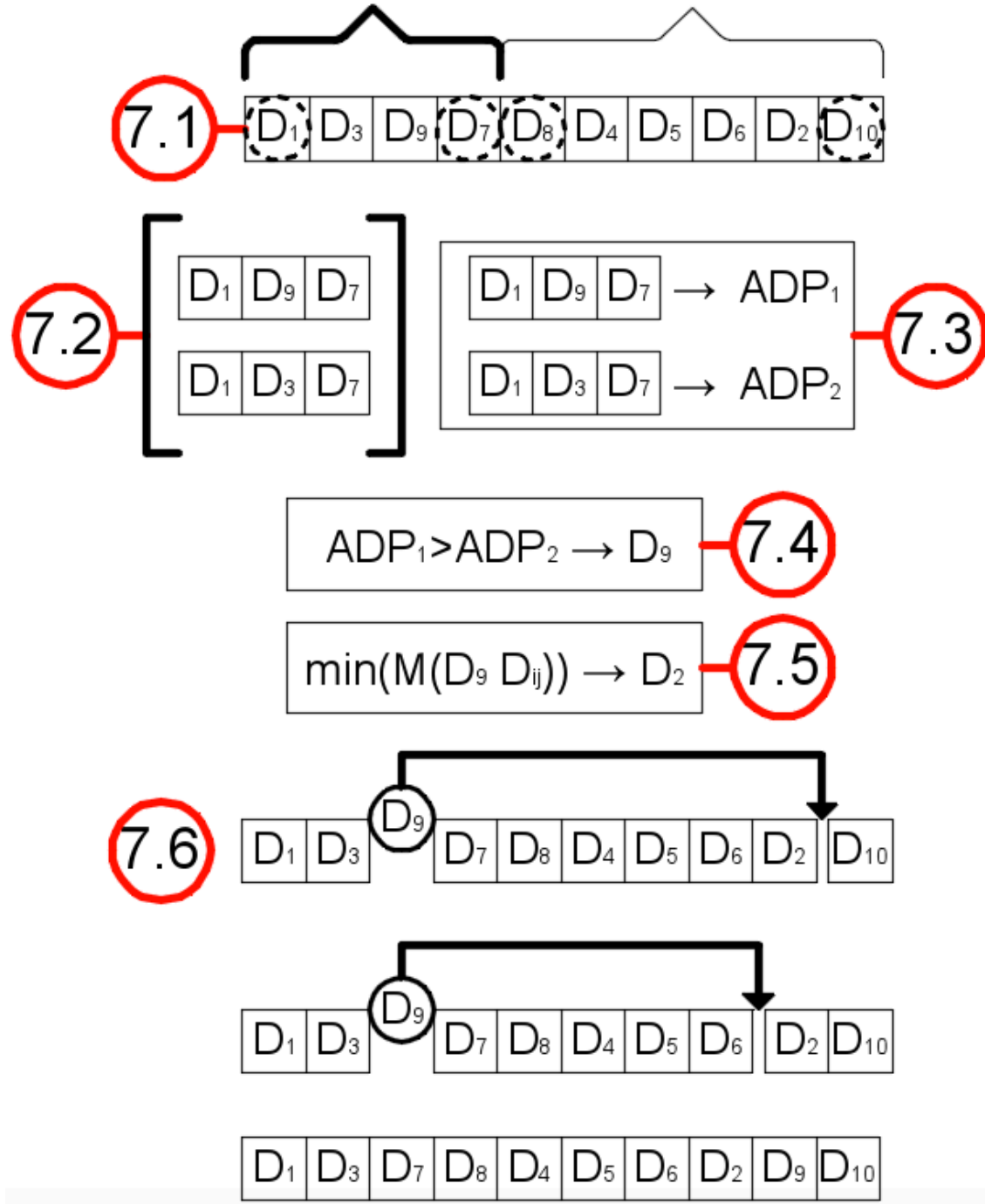
3.3.7. Durağın en uygun konuma yerleştirilmesi iyileştirilmesi

Şekil 3.6'da (7) numaralı bölümde optimum güzergah kümesi üzerinde seçilen bir durağın iyileştirme adımları kapsamında yeri değiştirilmektedir. Şekil 3.17'de detayları verilen bu iyileştirme adımı Şekil 3.18'de örneklendirilmiştir. Burada popülasyondaki her birey için işlemler tekrarlanmaktadır. İşleminin başlaması için öncelikle 0 ile 1 arasında rastgele bir sayı üretilerek, bu sayının 0,15'ten küçük olması durumu kontrol edilmektedir. Bu kontrol işlemi mutasyon oranlarıyla aynı mantıkla çalışmaktadır. Kontrol edilen sayı yapılan denemeler sonucunda en verimli oran olarak belirlenmiştir. Rastgele üretilen sayı 0,15'ten küçük ise bu işleme tabi tutulan optimum kümeyle popülasyondaki birey değiştirilmektedir. Eğer rastgele üretilen sayı 0,15'ten büyükse o birey üzerinde hiçbir işlem yapılmadan bir sonraki birey için yeniden rastgele sayı üretilmektedir. Bu işlemler popülasyondaki tüm bireyler için gerçekleştikten sonra yeni popülasyon ADY ile değerlendirme adımı (3) tekrar gönderilmektedir.

Şekil 3.17'de öncelikle (7.1) numaralı adımla güzergah kümesi kendini oluşturan güzergahlara ayrılmakta ve rastgele biri seçilmektedir. Seçilen güzergahtan başlangıç ve bitiş durakları hariç kalan durakların birer tanesinin olmadığı kombinasyonlar (7.2) numaralı adımda listelenmektedir. Bu listedeki güzergahlar ADY'ye göre değerlendirilmekte (7.3) ve güzergahtaki en kötü konumu veren durak (7.4) tespit edilmektedir. Daha sonra (7.5) numaralı adımda bu durağa en yakın durak güzergah kümesi içinde belirlenmektedir. Son olarak kötü konumlu durak kendine en yakın durağın öncesine ve sonrasına eklenerek iki seçenekten ADP değeri daha iyi olan seçenek (7.6) numaralı adımda optimum güzergah olarak atanmaktadır.



Şekil 3.17. ADY'ye göre en iyi performansa sahip güzergah kümesi üzerinde yapılan iyileştirme işlemi akış şeması



Şekil 3.18. ADY'ye göre en iyi performansa sahip güzergah kümesi üzerinde yapılan iyileştirme işlemi için bir örnek

Şekil 3.18'de (7.1) numaralı adımda güzergah kümesi D_1, D_3, D_9, D_7 ve $D_8, D_4, D_5, D_6, D_2, D_{10}$ olarak iki güzergaha ayrılmıştır. Bu güzergah rastgele seçim sonucu ilki üzerinde işlemler gerçekleştirilmiştir. (7.2) numaralı adımda ilk güzergahtan ara duraklarından birer tanesinin olmadığı kombinasyonlar listelenmiştir. (7.3) numaralı adımda bu listedeki güzergahlar için ADP değerleri hesaplanmıştır. (7.4) numaralı adımda ise en büyük ADP değerini veren güzergahtaki çıkarılan durak olan D_9 durağı tespit edilmiştir. (7.5) numaralı adımda mesafe olarak D_9 durağına en yakın durak D_2

olarak hesaplanmıştır. (7.6) numaralı son adımda ise D_9 durağı güzergah kümesindeki D_2 durağının öncesine ve sonrasına yerleştirilmiştir. Bu iki alternatiften ADP değeri küçük olan güzergah kümesi seçilerek popülasyonda güncelleme yapılmıştır.

3.3.7.1. Güzergah kümesini güzergahlara ayırma

İyileştirme adımlarında, güzergah kümesi içerisindeki işlemlerin yapılabilmesi güzergahların başlangıç ve bitiş duraklarına göre güzergah parçalarına ayrılması gerekmektedir. Bu ayırma işlemi modelde aşağıdaki formüllerle yapılmaktadır.

$$Q_i = \text{sort} \left(\text{find}(GK_i = (A \cup B)) \right) \quad \forall i \in \{1, 2, \dots, N_p\} \quad (3.37)$$

$$G_i^c = GK_i(Q_i(j): Q_i(j)) \quad \forall j \in \{1, 2, \dots, N_{A \cup B}\} \quad (3.38)$$

$$\forall c \in \{1, 2, \dots, N_{AB}\}$$

$$rc = \text{ceil}(\text{rand} \cdot N_{AB}) \quad (3.39)$$

Burada,

Q_i : i 'inci güzergah kümesindeki başlangıç ve bitiş durak konumlarının sıralaması,

G_i^c : i 'inci güzergah kümesindeki c 'inci güzergah dizisi,

$N_{A \cup B}$: başlangıç ve bitiş duraklarının toplam sayısı,

N_{AB} : başlangıç ve bitiş durak çifti sayısı,

rc : güzergah kümesi içerisinde rastgele seçilen güzergah numarasıdır.

İlk olarak (3.37) numaralı formülle güzergah kümesi içerisindeki başlangıç ve bitiş duraklarının sıra numaraları küçükten büyüğe doğru sıralanmaktadır. Sonra bu sıra numaraları arasında kalan durak dizilimleri (3.38) numaralı formülde güzergah parçası olarak atanmaktadır. Son olarak (3.39) numaralı formülle güzergahlardan biri rastgele seçilmektedir.

3.3.7.2. Güzergahlardan durak çıkarma

Bu adımda rastgele seçilen güzergah parçası içerisindeki duraklardan hangisinin bu güzergahta olmamasının diğerlerine göre ADP değeri açısından daha çok avantaj sağlayacağı belirlenmektedir. Bunun içinde birer durak eksiltiyle oluşturulmuş alternatif güzergahlar listelenmektedir. Bu liste modelde aşağıdaki formülle hesaplanmaktadır.

$$G_i^{cef} = G_i^c - G_i^{cf} \quad \forall f \in \{2, 3, \dots, N_{G_i^c} - 1\} \quad (3.40)$$

Burada,

G_i^{cf} : i 'inci güzergah kümesindeki c 'inci güzergahın f 'inci durağı,

G_i^{cef} : i 'inci güzergah kümesindeki c 'inci güzergahtan f 'inci durağın çıkarılmış hallerini içeren küme,

$N_{G_i^c}$: G_i^c güzergahındaki durak sayısıdır.

3.3.7.3. ADY ile güzergahları değerlendirme

Bu adımda listelenen alternatif güzergahlar için ADP değerleri belirlenmektedir. Bu değerler (3.2)'de verilen formülle her bir alternatif için ayrı ayrı hesaplanmaktadır.

3.3.7.4. Güzergahtaki en kötü konuma sahip durağı belirleme

Hesaplanan ADP değerlerinden en yüksek değere sahip alternatif güzergahtaki çıkarılan durak, rastgele seçilen güzergah içerisindeki en kötü konuma sahip duraktır. Bu durağın sıra numarası modelde aşağıdaki belirlenir.

$$[ADP_{max}^{ce}, I_w^{ce}] = \max(ADP^{ce}) \quad \forall ce \in \{2, 3, \dots, N_{AB} - 1\} \quad (3.41)$$

$$I_w^{GK_i} = \text{find}(GK_i = G_i^{rc}(I_w^{ce})) \quad (3.42)$$

Burada,

G_i^{rc} : i 'inci güzergah kümesi içerisinde rastgele seçilen güzergah,

ADP^{ce} : G_i^{ce} kümesine ait ADP değerleri,

ADP_{max}^{ce} : ADP^{ce} kümesindeki en büyük ADP değeri,

I_w^{ce} : G_i^c güzergahına ait ADP değerini en çok yükselten durağın sıra numarası,

$I_w^{GK_i}$: GK_i kümesinde $G_i^c(I_w^{ce})$ durağının sıra numarasıdır.

Modelde (3.41) numaralı formül rastgele seçilen güzergahtaki en kötü konuma sahip durağın sıra numarası belirlerken, (3.42) numaralı formülle bu sıra numarasının güzergah kümesinde hangi sıra numarasını temsil ettiği hesaplanmaktadır.

3.3.7.5. En kötü konuma sahip durağa en yakın durağı belirleme

Bir durağın güzergah kümesi içerisindeki konumunun, kendine yakın mesafedeki bir durağın öncesinde ya da sonrasında olmasının sağlanması, güzergah kümesinin performansını artırma potansiyeline sahiptir. Bu yüzden en kötü konuma sahip durağa en yakın durak belirlenmektedir. Fakat burada iki ihtimal söz konusudur. Birincisi en yakın durağın aynı güzergah parçasında bulunması ihtimali, diğeri ise farklı güzergah parçalarında bulunma ihtimalidir. Bu iki durum göz önünde bulundurularak hesaplarda en yakın güzergahın aranacağı küme için iki farklı alternatif oluşturulmuştur. Modelde hesaplamalar aşağıdaki formüllerle yapılmaktadır.

$$rand < 0,7 \rightarrow K_i = GK_i - G_i^{rc} \quad (3.43)$$

$$rand \geq 0,7 \rightarrow K_i = GK_i \quad (3.44)$$

$$[CD, I_c^{ce}] = \min(M(I_w^{ce}, K_i)) \quad (3.45)$$

$$I_c^{GK_i} = \text{find}(GK_i = K_i(I_c^{ce})) \quad (3.46)$$

Burada,

K_i : i 'inci güzergah kümesi içerisinde rastgele seçilen güzergahın çıkarılması sonucu elde edilen kümesi,

$M(I_w^{ce}, K_i)$: I_w^{ce} durağı ile K_i kümesi içerisindeki duraklar arası mesafeler,

CD : K_i kümesinde, I_w^{ce} durağına en yakın durak mesafesi,

I_c^{ce} : K_i kümesinde, I_w^{ce} durağına en yakın durağın sıra numarası,

$I_c^{GK_i}$: GK_i kümesinde, I_w^{ce} durağına en yakın durağın sıra numarasıdır.

Modelde (3.43) numaralı formül ile en yakın durağın %70 ihtimalle farklı güzergah parçalarında aranması, (3.44) numaralı formülle ise %30 ihtimalle güzergah kümesinin tamamında aranması sağlanmıştır. Aranan durağın konumu arandığı küme içerisinde (3.45) numaralı formülle tespit edilmektedir. Bu durağın güzergah kümesindeki konumu ise (3.46) numaralı formülle hesaplanmaktadır.

3.3.7.6. Durağın doğru konuma yerleştirilmesi

Bir durağın güzergahtaki sıralamasında, kendine yakın bir durağın yanında bulunması iki farklı şekilde olabilir. İlki yakın olarak belirlenen durağın öncesine yerleştirilmesi, ikincisi ise sonrasına yerleştirilmesi durumudur. Dolayısıyla bu iki seçenekte denenerek hangi durumun daha avantajlı olacağına karar verilmelidir. Bu durum modelde aşağıdaki gibi belirlenmektedir.

$$Alt1 = [GK_i(1:I_c^{GK_i}), GK_i(I_w^{GK_i}), GK_i(I_c^{GK_i} + 1:N_d)] \quad (3.47)$$

$$Alt2 = [GK_i(1:I_c^{GK_i} - 1), GK_i(I_w^{GK_i}), GK_i(I_c^{GK_i}:N_d)] \quad (3.48)$$

$$ADP^{Alt1} < ADP^{Alt2} \rightarrow GK_i = Alt1 \quad (3.49)$$

$$ADP^{Alt1} \geq ADP^{Alt2} \rightarrow GK_i = Alt2 \quad (3.50)$$

Burada,

$Alt1$: i 'inci güzergah kümesine $I_w^{GK_i}$ durağının $I_c^{GK_i}$ durağından sonrasına eklenmiş alternatif küme,

$Alt2$: i 'inci güzergah kümesine $I_w^{GK_i}$ durağının $I_c^{GK_i}$ durağından öncesine eklenmiş alternatif kümedir.

(3.47) ve (3.48) numaralı formüllerle yakın olarak belirlenen durağın öncesine ve sonrasına yerleştirilen iki alternatif oluşturulmaktadır. Daha sonra (3.49) ve (3.50) numaralı formüllerle bu alternatifler için hesaplanan ADP değerleri kıyaslanarak güzergah kümesi popülasyonda güncellenmektedir.

3.3.8. Güzergahlar arası parça değiştirme iyileştirmesi

Şekil 3.6'da (8) numaralı bölümde optimum güzergah kümesi üzerinde farklı güzergahlar içerisinde seçilen iki parçanın iyileştirme adımları kapsamında yeri değiştirilmektedir. Burada parçalar rastgele seçilmemektedir. Bir güzergah kümesi içinde farklı iki güzergah rastgele seçilerek, bu güzergahlarda bulunan durakların birbirlerine olan yakınlıkları hesaplanmaktadır. İki güzergahta da birbirine en yakın ikişer durak seçilmektedir. Daha sonra bu duraklar arasında kalan parçaların yerleri değiştirilmektedir. Bu işlemler modelde aşağıdaki gibi hesaplanmaktadır. Örnek olarak Şekil 3.19'daki parçaların değişimi verilmiştir.

$$\begin{bmatrix} rc_1 \\ rc_2 \end{bmatrix} = rand \left(\begin{matrix} N_{AB} \\ 2 \end{matrix} \right) \quad (3.51)$$

$$\left[M_{min}^{rc_1 rc_2}, \left(I_{rw}^k \right) \right] = \min \left(M(G_i^{rc_1}, G_i^{rc_2}) \right) \quad \forall k \in \{1,2\} \quad (3.52)$$

$$G_i^{rc_1} = [G_i^{rc_1}(1:I_{rw}^1 - 1), G_i^{rc_2}(I_{cl}^1:I_{cl}^2), G_i^{rc_1}(I_{rw}^2 + 1:N_e^1)] \quad (3.53)$$

$$G_i^{rc_2} = [G_i^{rc_2}(1:I_{cl}^1 - 1), G_i^{rc_1}(I_{rw}^1:I_{rw}^2), G_i^{rc_2}(I_{cl}^2 + 1:N_e^2)] \quad (3.54)$$

Burada,

rc_1 : güzergah kümesinden rastgele seçilen ilk güzergahın numarası,

rc_2 : güzergah kümesinden rastgele seçilen ikinci güzergahın numarası,

$M(G_i^{rc_1}, G_i^{rc_2})$: $G_i^{rc_1}$ güzergahındaki duraklar ile $G_i^{rc_2}$ güzergahındaki duraklar arası yolculuk mesafesi matrisi,

$M_{min}^{c_1c_2}$: $G_i^{rc_1}$ ve $G_i^{rc_2}$ güzergahlarındaki iki durak arası en kısa mesafe,

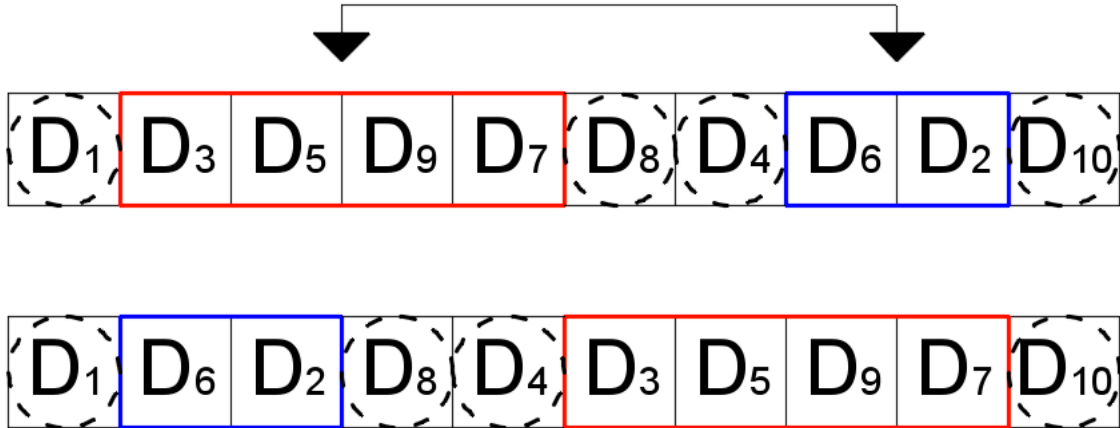
I_{rw}^k : $G_i^{rc_1}$ ve $G_i^{rc_2}$ güzergahlarındaki birbirine en yakın duraklardan $G_i^{rc_1}$ güzergahında bulunan k 'inci durağın sıra numarası,

I_{cl} : $G_i^{rc_1}$ ve $G_i^{rc_2}$ güzergahlarındaki birbirine en yakın duraklardan $G_i^{rc_2}$ güzergahında bulunan k 'inci durağın sıra numarası,

N_e^1 : $G_i^{rc_1}$ güzergahındaki durak sayısı,

N_e^2 : $G_i^{rc_2}$ güzergahındaki durak sayısıdır.

İlk olarak (3.51) numaralı formülle güzergah kümesi içerisinde hangi iki güzergahın seçileceği belirlenmektedir. Sonra bu güzergahlardaki duraklar arasındaki mesafeler ve birbirine en yakın ikişer durağın sıra numarası (3.52) numaralı formülle hesaplanmaktadır. Son olarak (3.53) ve (3.54) numaralı formüllerle belirlenen ikişer durak arasında kalan parçalar yer değiştirmektedir.



Şekil 3.19. Rastgele seçilen güzergah kümesinde içinde yer alan güzergahlar arasında parça değişimi örneği

Şekil 3.19'daki örneğe bakarsak güzergah kümesinden iki güzergah $D_1, D_3, D_5, D_9, D_7, D_8$ ve D_4, D_6, D_2, D_{10} olarak seçilmiştir. Örnekte zaten iki güzergah bulunduğu için ikisi de seçilmiştir. Bu güzergahlarda bulunan duraklardan $D_3 - D_6$ ve $D_7 - D_2$ durakları birbirine en yakın duraklar olarak belirlenmiştir. Daha sonra bu

duraklar arasında kalan parçalar yer değiştirilmiştir. Böylece iyileştirme adımı tamamlanarak güzergah kümesi popülasyonda güncellenmiştir.

3.3.9. Optimum sonuç

Şekil 3.6'da (4) numaralı bölümde yeterli iterasyon sayısına ulaşıldığında popülasyondaki en iyi performansa sahip güzergah kümesi (9) numaralı bölümde optimum sonuç olarak atanmaktadır. Bu atama modelde aşağıdaki gibi yapılmaktadır.

$$[ADP_{min}^p, I_b^p] = \min(ADP^p) \quad (3.55)$$

$$GK_{opt}^g = GK_{I_b^g} \quad (3.56)$$

Burada,

ADP^p : iterasyon tamamlandıktan sonraki popülasyona ait güzergah kümelerinin ADP 'lerini içeren dizi,

ADP_{min}^p : ADP^p dizisindeki en küçük değer,

I_b^p : ADP^p dizisindeki en küçük değeri veren sıra numarası,

GK_{opt}^p : iterasyon tamamlandıktan sonra popülasyondaki optimum güzergah kümesidir.

Bu adımda (3.55) numaralı formülle tüm işlem adımları tamamlandıktan sonra popülasyondaki optimum güzergah kümesini veren bireyin sıra numarası tespit edilir. Bu güzergah kümesi (3.56) numaralı formül ile modelin çözümü olarak atanır. Böylece algoritma sonlandırılır.

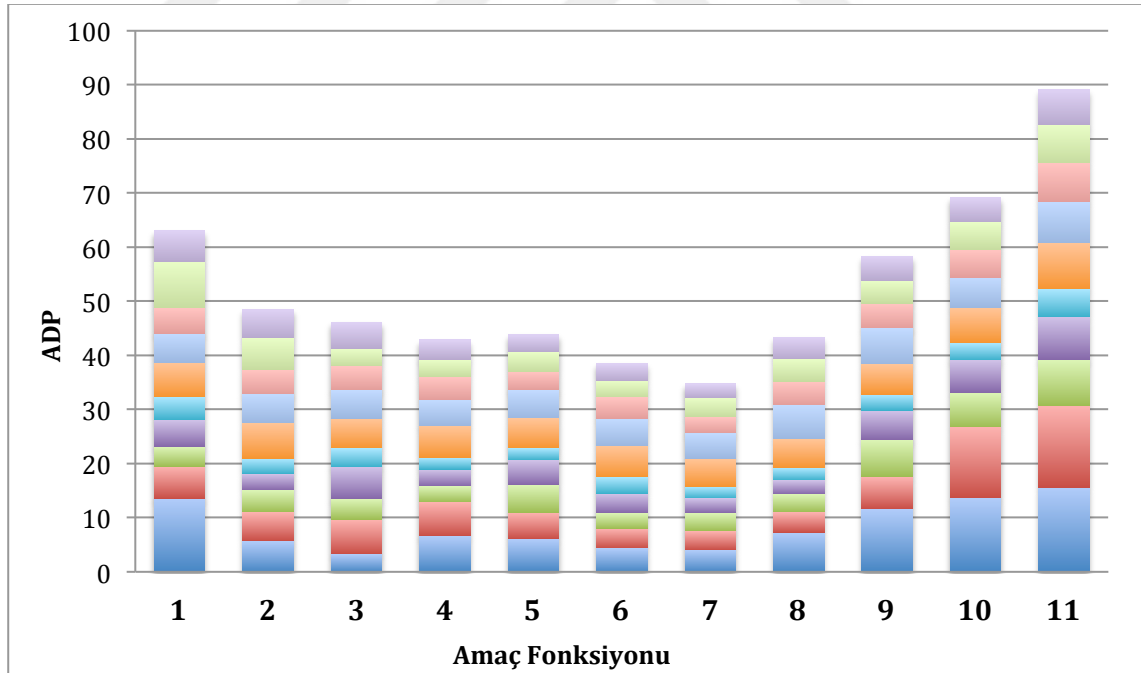
3.3.10. Sonuçların değerlendirilmesi

Geliştirilen model için dinamik bir amaç fonksiyonu oluşturulmuştur. Optimizasyon ilk olarak oluşturulan güzergahların toplam yolculuk mesafelerini, sonra ise akış doğrultusu yöntemine göre hesaplanan akış doğrultusu performanslarını (ADP) minimize etmektedir. Burada optimizasyondaki iterasyonun ne kadarında en kısa mesafeye ne kadarında ise ADP 'ye göre değerlendirme yapılacağına karar vermek için Çizelge 3.8'de verilen amaç fonksiyonları oluşturulmuştur.

Çizelge 3.8. Değerlendirme yöntemi için belirlenen en kısa mesafe ve ADP yüzdeleri

Amaç Fonksiyonu	En Kısa Mesafe (%)	ADP (%)
1	0	100
2	10	90
3	20	80
4	30	70
5	40	60
6	50	50
7	60	40
8	70	30
9	80	20
10	90	10
11	100	0

Çizelge 3.8’de verilen 11 farklı amaç fonksiyonu ile uygulamalara geçmeden önce bir ön çalışma olarak, seçilen 10 adet rastgele güzergah kümesi üzerinde deneme çalışması yapılmıştır. Her güzergah kümesi 11 farklı amaç fonksiyonu için ayrı ayrı çözümlenerek ve optimum *ADP* değerleri elde edilmiştir. Elde edilen *ADP* değerleri farklı amaç fonksiyonları için gruplandırılarak kümülatif olarak toplanmıştır (Şekil 3.20).

**Şekil 3.20.** Farklı değerlendirme yüzdelerine göre ADP değerleri

Şekil 3.20’de de görüldüğü üzere 7 numaralı amaç fonksiyonu diğer seçeneklere göre daha verimli sonuçlar üretmiştir. Yani optimizasyonda %60’a kadar en kısa mesafeye göre sonrasında ise *ADP* ‘ye göre minimizasyon yapılması faydalı olacaktır.

3.4. Kümeleme

Bir şehre ait tüm otobüs hatlarını aynı anda optimize etmek yerine bunları bulundukları bölgelere göre kümeleyerek aynı kümede yer alanları optimize etmek hem zaman hem de sonuçların verimliliği açısından daha faydalı olacaktır. Bu yüzden 732 otobüs hattı birbirlerine olan yakınlıklarına göre kümeleme işlemine tabi tutulmuştur. Optimizasyon yönteminin performansını değerlendirebilmek açısından bu kümelerde farklı sayılarda hat ve durak bulunmasına özen gösterilmiştir. Önce araçların geçtikleri durakların konumlarına göre hatlar ağırlıklandırılarak ağırlık merkezleri hesaplanmıştır. Sonra bu merkez noktalar birbirine olan yakınlıklarına göre k-means yöntemi kullanılarak kümelendi. K-means algoritmasının amacı, n boyutlu bir nokta kümesini k adet kümeye ayırmaktır. Bunu yaparken kümenin içinde kalan noktaların kümenin merkezine olan toplam uzaklık karelerini minimize etmektedir (John A Hartigan 1975; J. A. Hartigan ve Wong 1979; Telgarsky ve Vattani 2010). Burada uzaklıkları koordinatlar arasında kalan öklid mesafeleri hesaplayarak işleme dahil etmektedir. İterasyona bağlı bir çözüm ürettiği için yakınsamayı hızlı bir şekilde sağlamaktadır. Kısa sürede çözüm üretebildiği için bu çalışma tercih edilmiştir. Yapılan denemeler sonucunda yeterli çeşitliliği sağladığı düşünülen 250 küme oluşturulmuştur (Çizelge 3.9).

Çizelge 3.9. K-means kümeleme sonucunda oluşturulan kümeler ve içerdikleri hat sayıları

Kümede Bulunan Hat Sayısı	Küme Sayısı	Toplam Hat Sayısı
1	54	54
2	40	80
3	87	261
4	30	120
5	27	135
6	5	30
7	6	42
10	1	10
$\Sigma = 250$		$\Sigma = 732$

Geliştirilen model Çizelge 3.9’da oluşturulan 250 adet küme için ayrı ayrı hesap yapılmıştır. Yönteme göre, bu hesaplama sırasında o küme içerisinde bulunan tüm hatlara ait duraklar bir havuzda toplanmıştır. Bu kümede bulunan, mevcut hatlara ait başlangıç - bitiş durakları tespit edilmiş ve bu durak çiftleri arasında kalan aynı sayıda yeni güzergahlar hesaplanmıştır.

4. BULGULAR VE TARTIŞMA

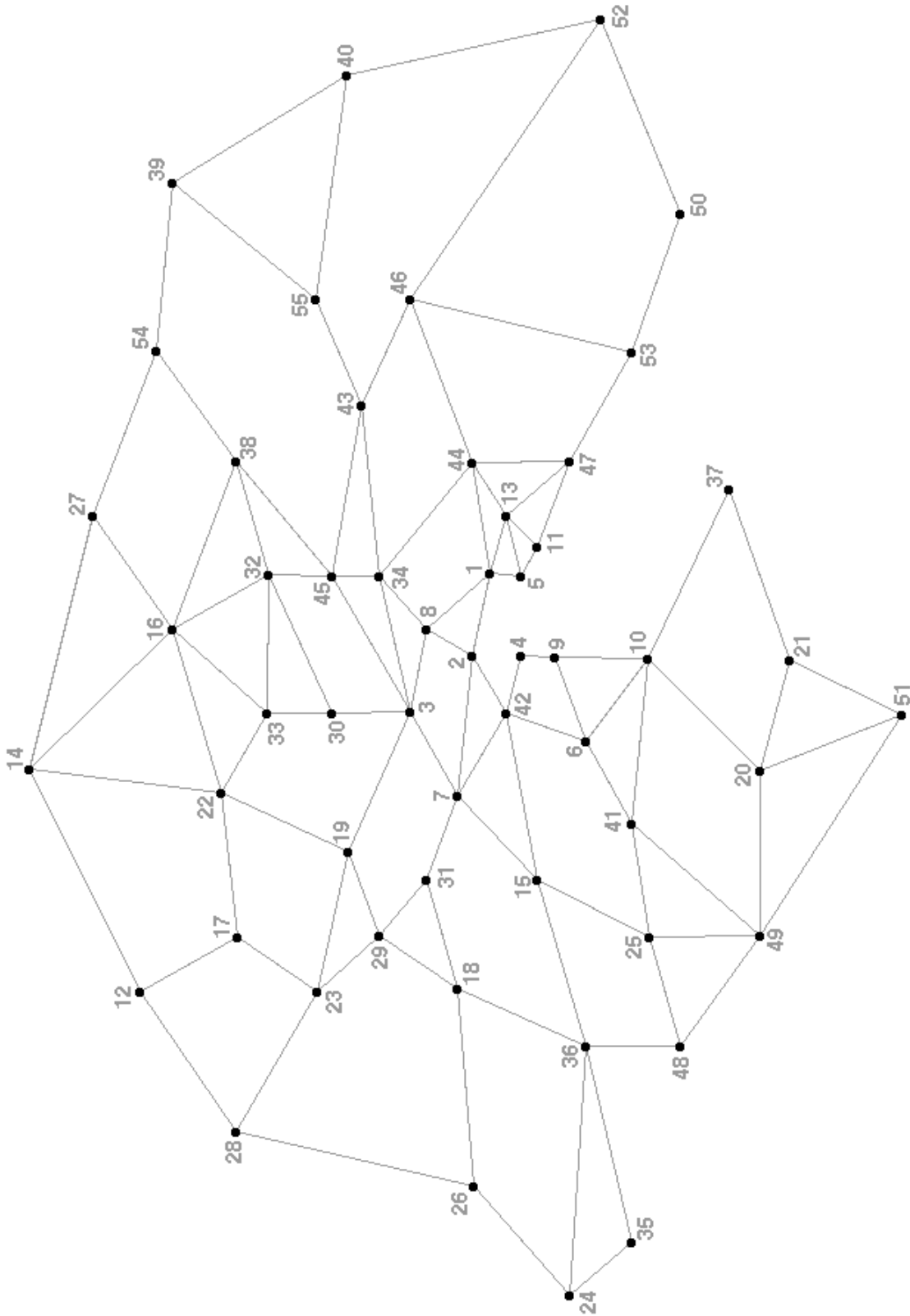
Geliştirilen model ile İstanbul kentine ait otobüs hatları için uygulama yapmadan önce küçük bir örnek uygulama çözülmüştür. Bu örnek uygulamada modelin verdiği ara sonuçları inceleyebilmek adına bazı iterasyonlar detaylı olarak açıklanmıştır. Daha sonra İstanbul şehri için toplu taşıma güzergahları yeniden hesaplanarak mevcut durumla karşılaştırılmıştır.

4.1. 18 Duraklı Swain Yol Ağı Örneği

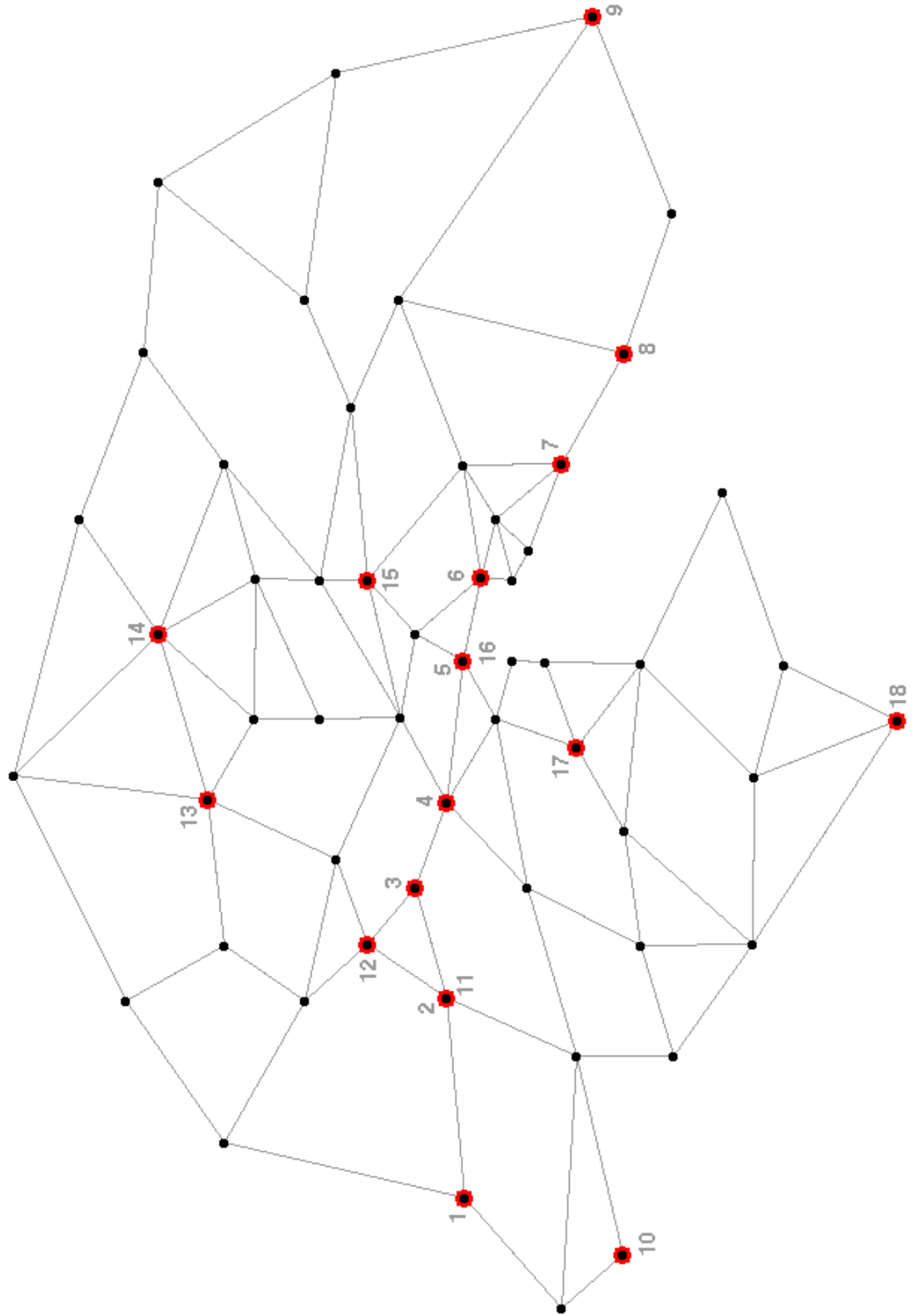
Örnek uygulama için 55 düğüm noktasına sahip Niblett vd. (2016) çalışmasında kullandığı Swain yol ağı olarak bilinen bir yol ağı kullanılmıştır (Şekil 4.1). Bu ağ üzerindeki 55 düğüm noktasının 16'sı seçilerek, bu duraklardan 2 adet hat geçecek şekilde bir örnek oluşturulmuştur (Şekil 4.2). Ayrıca her iki hattında seçilen aynı 2 duraktan geçmesi istenmektedir. Böylece problemdeki durak sayısı 18'e çıkmıştır. Bu problem için başlangıç durakları 1 ve 10, bitiş durakları ise 9 ve 18 olarak belirlenmiştir. Niblett vd. (2016) çalışmasından alınan yol ağına ait görsel Autocad dosyasına aktararak sayısallaştırılmıştır. Sonra buradan düğüm noktalarına ait koordinat noktaları belirlenmiştir. Bu 55 düğüm noktasına ait koordinatlar üzerinden yol ağındaki tüm kesitlere ait mesafeler hesaplanmıştır. Mesafe verileri ve yol ağı üzerindeki bağlantılar kullanılarak, tüm düğüm noktaları arasındaki en kısa güzergahı verecek yolculuk matrisi Dijkstra algoritması yardımıyla hesaplanmıştır. Bu matristen 18 durağa ait olan veriler çekilerek Çizelge 4.1 oluşturulmuştur. Mesafe verilerinin çizelgeye sığması için değerler yuvarlatılarak yazılmıştır. Örnek olarak oluşturulan bu veriler birimsizdir.

Çizelge 4.1. Duraklar arasındaki yolculuk mesafesi matrisi

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-	7	11	15	20	23	30	36	50	12	7	12	24	31	25	20	24	29
2	7	-	4	8	13	16	23	29	43	16	-	5	17	24	17	13	17	22
3	11	4	-	4	9	12	19	24	38	20	4	4	15	22	13	9	13	18
4	15	8	4	-	5	8	15	21	35	20	8	7	17	20	9	5	9	14
5	20	13	9	5	-	3	10	15	30	23	13	12	19	20	7	-	8	13
6	23	16	12	8	3	-	7	12	27	27	16	15	21	21	8	3	11	16
7	30	23	19	15	10	7	-	6	20	33	23	22	28	27	13	10	18	23
8	36	29	24	21	15	12	6	-	14	39	29	28	33	32	19	15	24	29
9	50	43	38	35	30	27	20	14	-	53	43	42	40	37	27	30	38	43
10	12	16	20	20	23	27	33	39	53	-	16	21	33	40	30	23	26	28
11	7	-	4	8	13	16	23	29	43	16	-	5	17	24	17	13	17	22
12	12	5	4	7	12	15	22	28	42	21	5	-	12	18	15	12	17	22
13	24	17	15	17	19	21	28	33	40	33	17	12	-	7	16	19	27	32
14	31	24	22	20	20	21	27	32	37	40	24	18	7	-	13	20	28	33
15	25	17	13	9	7	8	13	19	27	30	17	15	16	13	-	7	15	20
16	20	13	9	5	-	3	10	15	30	23	13	12	19	20	7	-	8	13
17	24	17	13	9	8	11	18	24	38	26	17	17	27	28	15	8	-	5
18	29	22	18	14	13	16	23	29	43	28	22	22	32	33	20	13	5	-



Şekil 4.1. Örnek uygulama için seçilen 55 düğüm noktalı yol ağı



Şekil 4.2. Yol ağı üzerinde seçilen 18 durak

Şekil 4.2’de görülen örneğin geliştirilen optimizasyon modeliyle çözümü için ilk olarak popülasyondaki birey sayısının ve iterasyon sayısının hesaplanması gerekmektedir. Örneklemede durak sayısı ($N_d = 18$) az olduğu için (3.5) ve (3.6) numaralı formüller kullanılmadan $N_p = 40$ ve $N_i = 500$ şeklinde belirlenmiştir. (3.7) numaralı “*randperm*” fonksiyonu ile 1’den 18’e kadar olan durak numaralarının rastgele sıralandığı bir dizi oluşturulmuştur. Bu işlem 40 defa tekrarlanarak (3.8) numaralı formülle 1. iterasyon için Çizelge 4.2’deki popülasyon oluşturulmuştur. Popülasyonu temsil eden bu ve diğer çizelgelerde satırlar bireyleri yani güzergah kümelerini, sütunlar güzergah kümesi içerisindeki durak numaralarını vermektedir. (3.9) numaralı formülle başlangıç ve bitiş durak numaraları $A = \{1, 10\}$, $B = \{9, 18\}$ şeklinde tanımlanmıştır. Daha sonra (3.10), (3.11), (3.12) ve (3.13) numaralı formüller yardımıyla Çizelge 4.3 elde edilmiştir. Çizelgelerde güzergah kümesi içerisinde kalan güzergahların daha rahat fark edilebilmesi için başlangıç durakları yeşil, bitiş durakları ise kırmızı renkle gösterilmiştir.

Geliştirilen modele göre Çizelge 4.3 oluşturulduktan sonra bu popülasyondaki her bir güzergah kümesi için *ADP* değerleri hesaplanmalıdır. Fakat yapılan denemeler sonucunda değerlendirme yönteminde toplam iterasyon sayısının yaklaşık %60’ına kadar en kısa yolculuk mesafesinin, kalan %40’ında ise *ADP* değerlerinin kullanılması daha verimli çözümler üretildiği göstermiştir. Bu yüzden modelde değerlendirme yöntemi iki aşamalı olarak düzenlenmiştir. Dolayısıyla $N_i = 500$ olarak belirlenen iterasyon sayısının 300’e kadar olan iterasyonlarında en kısa yolculuk mesafesi, 300 ve sonrasındaki iterasyonlarda ise *ADP* değerleri üzerinden değerlendirme yapılmıştır. Çizelge 4.3, henüz 1. iterasyona ait olduğu için en kısa yolculuk mesafeleri hesaplanmıştır. Örneğin Çizelge 4.3’teki 1 numaralı güzergah kümesini incelersek, duraklar arasındaki kesit mesafeleri Çizelge 4.1’den Şekil 4.3’teki gibi alınarak tüm mesafeler toplanmış ve yolculuk mesafesi 306,01 olarak belirlenmiştir.

1	8	3	15	7	12	9	10	2	13	4	16	17	14	6	11	5	18
35,63	24,14	13,08	13,39	22,21	41,98	15,98	17,22	17,36	5,06	8,08	28,13	21,45	16,21	13,05	13,04		

Şekil 4.3. 1 numaralı güzergah kümesi için kesit mesafeleri

Benzer bir şekilde tüm güzergah kümeleri için aynı hesaplar yapılarak Çizelge 4.4’teki yolculuk mesafesi verileri hesaplanmıştır. Sonra (3.14) numaralı formülde *ADP* değerleri yerine hesaplanan mesafe verileri kullanılarak her bir güzergah kümesinin seçilme olasılığı belirlenmiştir. Son olarak (3.15), (3.16) ve (3.17) numaralı formüllerle her birey için popülasyondan bir güzergah kümesi seçimi yapılmıştır. Bu işlemler sonucu aynı güzergah kümesi birden fazla kez seçilebilir.

Çizelge 4.2. 1. iterasyon için başlangıç popülasyonu

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

Çizelge 4.3. 1. iterasyon için küme derleme işlemi

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

Çizelge 4.4. 1. iterasyon için rulet tekerleği yöntemine göre seçilen bireyler

GK_i	Yolculuk Mesafesi	P_i	Seçilen Güzergah Kümeleri
1	306,01	0,03040	3
2	10244,65	0,00091	35
3	300,19	0,03099	33
4	10236,44	0,00091	33
5	297,98	0,03122	35
6	308,41	0,03017	12
7	310,31	0,02998	28
8	10302,26	0,00090	26
9	10277,87	0,00091	20
10	10286,00	0,00090	40
11	290,21	0,03206	22
12	291,65	0,03190	25
13	274,39	0,03391	21
14	279,50	0,03329	33
15	235,09	0,03958	40
16	237,77	0,03913	14
17	10271,65	0,00091	12
18	286,81	0,03244	27
19	264,17	0,03522	40
20	244,68	0,03802	35
21	289,49	0,03214	19
22	274,16	0,03394	18
23	10278,05	0,00091	15
24	281,63	0,03304	37
25	253,60	0,03669	20
26	261,63	0,03556	1
27	304,01	0,03060	11
28	262,56	0,03544	16
29	308,70	0,03014	13
30	262,55	0,03544	1
31	299,01	0,03112	36
32	10284,20	0,00090	20
33	316,66	0,02938	21
34	10255,13	0,00091	14
35	267,34	0,03480	15
36	289,26	0,03217	20
37	289,10	0,03218	25
38	10305,40	0,00090	7
39	316,33	0,02941	29
40	304,26	0,03058	24

Rulet yöntemine göre seçilerek belirlenen güzergah kümeleri ile genetik algoritma adımının ilk popülasyon güncellemesi gerçekleşmiştir. Örneğin Çizelge 4.4'teki 15 numaralı birey için, 40 numaralı güzergah kümesi seçilmiştir. Bu seçim işlemi Çizelge 4.3'teki 40 numaralı güzergahın, Çizelge 4.5'e 15 numaralı birey olarak aktarılması anlamına gelmektedir. Bu aktarma işlemi popülasyondaki tüm bireyler için gerçekleştikten sonra Çizelge 4.5'te bireyler çaprazlama için çift olarak gruplandırılmaktadır. Her bir çift için (3.20) numaralı formülle bir kırılma noktası belirlenmiştir. Aynı kırılma noktasından parçalanan çiftler arasında (3.21), (3.22), (3.23), (3.24), (3.25), (3.26) ve (3.27) numaralı formüller yardımıyla parça değişimleri gerçekleştirilmiştir. Örneğin 9 ve 10 numaralı güzergah kümeleri için kırılma noktası $c_s = 9$ olarak belirlenmiş ve ilk 9 durak yer değiştirmiştir. Bu işlemin popülasyondaki tüm bireylere uygulanması sonucu Çizelge 4.6 oluşturulmuştur.

Çaprazlama işleminden sonra popülasyon üzerinde mutasyon işlemleri uygulanmaya başlamaktadır. Fakat mutasyon işlemleri belirlenen mutasyon oranları çerçevesinde uygulanmaktadır. Yer değişimleri için belirlenen mutasyon oranı %1'dir. Örnek olarak seçilen popülasyon küçük olduğu için %1 oranıyla popülasyonda her zaman güncelleme yapmak mümkün olmayabilir. Bu sebepten ötürü popülasyon, Çizelge 4.7'ye herhangi bir değişikliğe uğramadan aktarılmıştır. Parça değişimi mutasyonu ise %2 oranında uygulanacaktır. Örnekte sadece 25 numaralı güzergah kümesi mutasyona uğramıştır. Çizelge 4.8'deki popülasyonda mutasyona uğrayan güzergah kümesi görülmektedir. Burada (3.31) numaralı formülle $c_p = 11$ olarak belirlenmiş ve (3.32) numaralı formülle mutasyon uygulanmıştır. Elitizm ile popülasyon genetik algoritmanın son adımını tamamlamış ve popülasyon güncellenmiştir (Çizelge 4.9). Bu adım için (3.35) numaralı formül yardımıyla Çizelge 4.4'teki en kısa yolculuk mesafesine sahip güzergah kümesi tespit edilmiştir. 15 numaralı birey olan bu güzergah kümesi Çizelge 4.3'ten alınarak Çizelge 4.9'a ilk birey olarak atanmıştır.

Genetik algoritma adımları tamamlandıktan sonra iyileştirme adımları uygulanmaktadır. İlk olarak durağın en uygun konuma yerleştirilmesi adımının uygulanması gerekmektedir. Fakat bu iyileştirme adımı için Çizelge 4.4'te yolculuk mesafesi verileri yerine *ADP* değerlerinin hesaplanması gerekmektedir. Bu yüzden bu adım optimizasyonun %40'lık dilimiyle yani 300. iterasyonla başlayacaktır. Güzergahlar arası parçaların yer değiştirmesi iyileştirmesi ise ilk iterasyondan itibaren uygulanmaktadır. Çizelge 4.9'daki popülasyona uygulanan iyileştirme adımı popülasyonun yaklaşık %15'ine uygulanmaktadır. Bu iterasyonda 3, 6, 10, 25, 27, 33, 34 ve 39 numaralı güzergah kümelerine (3.51), (3.52), (3.53) ve (3.54) numaralı formüller uygulanarak popülasyon güncellenmiştir (Çizelge 4.10). Böylece iterasyon tamamlanmıştır.

Çizelge 4.5. 1. iterasyon için seçilen kümelerin çift haline getirilmesi

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

Çizelge 4.6. 1. iterasyon için çaprazlanmış güzergah kümeleri

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

Çizelge 4.7. 1. iterasyon için yer değişimi mutasyonu

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

Çizelge 4.8. 1. iterasyon için parça değişimi mutasyonu

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

Çizelge 4.9. 1. iterasyon için elitizm

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

Çizelge 4.10. 1. iterasyon için parça değiştirme iyileştirmesi

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

Optimizasyon 300. iterasyona kadar aynı işlemler tekrarlanarak devam etmektedir. Bu iterasyondan sonra artık *ADP* değerleri hesaplanmakta ve çözüm genetik algoritmanın lokal optimuma yakalandığı noktadan kurtularak global optimum seviyelere taşınmıştır. İlk olarak bir önceki iterasyona ait güncellemelerin yapıldığı son popülasyon alınarak yeni popülasyona dahil edilir (Çizelge 4.11). Bu popülasyon için *ADP* değerleri ve buna bağlı olarak güzergah kümelerinin seçilme olasılıkları hesaplanır. Bu olasılık değerlerine göre rulet tekerleği yöntemine göre güzergah kümelerinin seçilme işlemi gerçekleştirilir (Çizelge 4.12).

Çizelge 4.12’de belirlenen güzergah kümeleri ile yeni popülasyon Çizelge 4.13’te oluşturulmuştur. Bu popülasyon üzerinde çaprazlama işlemine hazırlık olarak güzergah kümeleri çift haline getirilmiştir. Çizelge 4.14’te ise çaprazlama yapılmış olan güzergah kümeleri görülmektedir. Bu adımdan sonra durakların yer değişimi mutasyonu uygulanmaktadır. Çizelge 4.15’te 35 numaralı güzergah kümesine (3.28), (3.29) ve (3.30) numaralı formüller uygulanarak güzergah kümesinde yer değiştirme işlemleri gerçekleştirilmiştir. (3.28) numaralı formülde $rnc = 15$ olarak hesaplandığı için 15 durak kendi arasında yer değiştirmiştir. Yer değişimi için seçilen duraklar; ilk durak, 8 nolu ve son durak haricinde tüm duraklardır. Güncellenen popülasyon üzerinde parça değişimi mutasyonları uygulanarak Çizelge 4.16, elitiz işlemi uygulanarak Çizelge 4.17’deki popülasyonlar elde edilmiştir. Böylece genetik algoritma adımları bu iterasyon için tamamlanmıştır.

Çizelge 4.11. 300. iterasyon için önceki iterasyondan gelen popülasyon

300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
2	1	15	16	17	5	7	8	9	10	2	3	11	12	13	4	6	14	18
3	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
4	1	11	12	17	13	16	15	14	6	5	7	8	9	10	2	4	3	18
5	1	13	7	15	14	2	5	17	4	6	3	11	12	8	9	10	16	18
6	1	8	6	3	11	4	15	9	10	2	12	16	17	13	5	7	14	18
7	1	12	4	6	5	7	8	9	10	2	3	11	16	14	13	17	15	18
8	1	4	6	15	16	17	8	9	10	2	3	11	13	14	12	5	7	18
9	1	15	16	17	5	7	8	9	10	2	3	11	12	13	4	6	14	18
10	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18
11	1	4	6	5	7	17	8	9	10	2	3	11	12	13	16	15	14	18
12	1	4	6	15	16	7	8	9	10	2	3	11	12	13	14	5	17	18
13	1	6	5	7	8	9	10	2	4	3	11	13	14	15	16	12	17	18
14	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
15	1	15	16	17	5	7	8	9	10	3	12	13	14	4	6	11	2	18
16	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	14	13	18
17	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	15	16	18
18	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	15	16	18
19	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
20	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
21	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
22	1	4	16	17	5	7	8	9	10	2	3	11	12	15	6	14	13	18
23	1	4	6	15	16	17	8	9	10	2	5	7	13	14	3	11	12	18
24	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	3	11	12	13	16	15	7	8	9	10	2	17	5	14	18
27	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
28	1	4	6	3	11	12	13	14	15	16	8	7	9	10	17	5	2	18
29	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	2	17	18
30	1	4	6	5	7	8	9	10	2	11	12	13	16	15	17	3	14	18
31	1	13	14	15	16	8	7	9	10	2	5	17	4	6	3	11	12	18
32	1	12	14	13	15	6	3	11	7	8	9	10	2	17	5	4	16	18
33	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
34	1	6	5	7	8	9	10	2	4	3	17	11	15	16	12	14	13	18
35	1	12	14	13	15	6	3	11	2	8	9	10	4	7	16	5	17	18
36	1	4	16	6	5	7	8	9	10	13	3	11	2	17	15	12	14	18
37	1	4	6	3	11	12	8	9	10	17	5	7	13	14	15	16	2	18
38	1	15	16	17	5	7	8	9	10	2	4	6	3	13	11	12	14	18
39	1	4	6	15	16	17	8	9	10	2	3	11	12	13	5	7	14	18
40	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18

Çizelge 4.12. 300. iterasyon için rulet tekerleği yöntemine göre seçilen bireyler

GK_i	ADP	P_i	Seçilen Güzergah Kümeleri
1	1,466	0,04120	26
2	2,102	0,02873	26
3	2,591	0,02331	32
4	2,373	0,02545	40
5	2,807	0,02151	18
6	2,954	0,02045	2
7	2,103	0,02872	11
8	2,301	0,02625	6
9	2,102	0,02873	1
10	1,551	0,03893	26
11	2,275	0,02655	37
12	1,790	0,03374	17
13	2,155	0,02803	20
14	1,614	0,03742	32
15	129,087	0,00047	20
16	2,119	0,02850	29
17	1,927	0,03134	39
18	1,927	0,03134	39
19	2,213	0,02729	36
20	2,213	0,02729	36
21	2,231	0,02707	39
22	1,793	0,03368	29
23	2,563	0,02357	13
24	1,789	0,03375	40
25	2,213	0,02729	36
26	2,745	0,02200	16
27	2,298	0,02628	37
28	2,602	0,02321	38
29	2,568	0,02352	28
30	103,907	0,00058	16
31	2,297	0,02630	14
32	2,291	0,02636	13
33	2,281	0,02648	32
34	2,631	0,02295	4
35	335,382	0,00018	9
36	204,033	0,00030	14
37	2,610	0,02314	3
38	2,448	0,02467	2
39	2,224	0,02716	40
40	2,298	0,02628	37

Çizelge 4.13. 300. iterasyon için seçilen kümelerin çift haline getirilmesi

300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	6	5	7	8	9	10	2	4	3	11	13	14	15	16	12	17	18
2	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
3	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
4	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
5	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
6	1	15	16	17	5	7	8	9	10	2	3	11	12	13	4	6	14	18
7	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
8	1	11	12	17	13	16	15	14	6	5	7	8	9	10	2	4	3	18
9	1	4	6	5	7	17	8	9	10	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
11	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
12	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
13	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
14	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
15	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	2	17	18
16	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	15	16	18
17	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
18	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
19	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
20	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	15	16	18
21	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	14	13	18
22	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
23	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
24	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
25	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
26	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
30	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
31	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	14	13	18
32	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
33	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
34	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
35	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	14	13	18
36	1	13	14	15	16	8	7	9	10	2	5	17	4	6	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
39	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18
40	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18

Çizelge 4.14. 300. iterasyon için çaprazlanmış güzergah kümeleri

300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	6	15	16	17	8	2	5	3	11	13	14	7	12	9	10	18
2	1	6	5	7	8	9	10	17	16	2	3	11	13	14	15	4	12	18
3	1	4	16	6	5	7	8	9	10	2	3	11	12	14	15	17	13	18
4	1	4	6	5	7	8	9	10	2	17	11	12	13	3	14	15	16	18
5	1	15	16	17	5	7	8	9	10	2	3	11	12	4	14	13	6	18
6	1	13	14	15	16	8	7	9	10	2	3	11	12	17	4	6	5	18
7	1	11	12	17	13	16	15	9	10	2	3	4	7	5	14	8	6	18
8	1	4	16	6	5	7	8	14	17	13	12	15	9	10	2	11	3	18
9	1	4	6	5	7	8	9	10	17	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	17	8	2	3	12	11	13	14	15	16	9	10	18
11	1	4	6	5	11	12	13	14	7	16	8	9	10	2	15	3	17	18
12	1	4	6	5	7	8	9	10	2	3	12	11	13	14	16	15	17	18
13	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
14	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
15	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	16	15	18
16	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	17	2	18
17	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
18	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
19	1	4	6	5	7	8	17	9	10	2	3	11	13	14	16	15	12	18
20	1	4	6	15	16	17	8	9	10	2	3	11	12	13	14	5	7	18
21	1	4	16	6	5	7	8	9	10	2	3	11	12	15	17	14	13	18
22	1	4	6	5	7	8	9	10	2	17	3	11	12	13	14	15	16	18
23	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
24	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	15	6	5	7	8	17	2	3	12	11	13	14	4	16	9	10	18
30	1	4	6	5	7	8	9	10	17	2	3	11	12	13	14	15	16	18
31	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	6	17	18
32	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	13	14	18
33	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
34	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
35	1	13	14	15	16	8	7	9	10	2	3	11	12	5	17	6	4	18
36	1	4	6	5	7	8	9	10	2	17	15	16	13	14	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
39	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
40	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18

Çizelge 4.15. 300. iterasyon için yer değişimi mutasyonu

300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	6	15	16	17	8	2	5	3	11	13	14	7	12	9	10	18
2	1	6	5	7	8	9	10	17	16	2	3	11	13	14	15	4	12	18
3	1	4	16	6	5	7	8	9	10	2	3	11	12	14	15	17	13	18
4	1	4	6	5	7	8	9	10	2	17	11	12	13	3	14	15	16	18
5	1	15	16	17	5	7	8	9	10	2	3	11	12	4	14	13	6	18
6	1	13	14	15	16	8	7	9	10	2	3	11	12	17	4	6	5	18
7	1	11	12	17	13	16	15	9	10	2	3	4	7	5	14	8	6	18
8	1	4	16	6	5	7	8	14	17	13	12	15	9	10	2	11	3	18
9	1	4	6	5	7	8	9	10	17	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	17	8	2	3	12	11	13	14	15	16	9	10	18
11	1	4	6	5	11	12	13	14	7	16	8	9	10	2	15	3	17	18
12	1	4	6	5	7	8	9	10	2	3	12	11	13	14	16	15	17	18
13	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
14	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
15	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	16	15	18
16	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	17	2	18
17	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
18	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
19	1	4	6	5	7	8	17	9	10	2	3	11	13	14	16	15	12	18
20	1	4	6	15	16	17	8	9	10	2	3	11	12	13	14	5	7	18
21	1	4	16	6	5	7	8	9	10	2	3	11	12	15	17	14	13	18
22	1	4	6	5	7	8	9	10	2	17	3	11	12	13	14	15	16	18
23	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
24	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	15	6	5	7	8	17	2	3	12	11	13	14	4	16	9	10	18
30	1	4	6	5	7	8	9	10	17	2	3	11	12	13	14	15	16	18
31	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	6	17	18
32	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	13	14	18
33	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
34	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
35	1	14	13	3	5	8	11	15	4	7	6	2	16	9	10	12	17	18
36	1	4	6	5	7	8	9	10	2	17	15	16	13	14	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
39	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
40	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18

Çizelge 4.16. 300. iterasyon için parça değişimi mutasyonu

300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	6	15	16	17	8	2	5	3	11	13	14	7	12	9	10	18
2	1	6	5	7	8	9	10	17	16	2	3	11	13	14	15	4	12	18
3	1	4	16	6	5	7	8	9	10	2	3	11	12	14	15	17	13	18
4	1	4	6	5	7	8	9	10	2	17	11	12	13	3	14	15	16	18
5	1	15	16	17	5	7	8	9	10	2	3	11	12	4	14	13	6	18
6	1	12	17	4	6	5	13	14	15	16	8	7	9	10	2	3	11	18
7	1	11	12	17	13	16	15	9	10	2	3	4	7	5	14	8	6	18
8	1	4	16	6	5	7	8	14	17	13	12	15	9	10	2	11	3	18
9	1	4	6	5	7	8	9	10	17	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	17	8	2	3	12	11	13	14	15	16	9	10	18
11	1	4	6	5	11	12	13	14	7	16	8	9	10	2	15	3	17	18
12	1	4	6	5	7	8	9	10	2	3	12	11	13	14	16	15	17	18
13	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
14	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
15	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	16	15	18
16	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	17	2	18
17	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
18	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
19	1	4	6	5	7	8	17	9	10	2	3	11	13	14	16	15	12	18
20	1	4	6	15	16	17	8	9	10	2	3	11	12	13	14	5	7	18
21	1	4	16	6	5	7	8	9	10	2	3	11	12	15	17	14	13	18
22	1	4	6	5	7	8	9	10	2	17	3	11	12	13	14	15	16	18
23	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
24	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	15	6	5	7	8	17	2	3	12	11	13	14	4	16	9	10	18
30	1	4	6	5	7	8	9	10	17	2	3	11	12	13	14	15	16	18
31	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	6	17	18
32	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	13	14	18
33	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
34	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
35	1	14	13	3	5	8	11	15	4	7	6	2	16	9	10	12	17	18
36	1	4	6	5	7	8	9	10	2	17	15	16	13	14	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
39	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
40	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18

Çizelge 4.17. 300. iterasyon için elitizm

300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
2	1	6	5	7	8	9	10	17	16	2	3	11	13	14	15	4	12	18
3	1	4	16	6	5	7	8	9	10	2	3	11	12	14	15	17	13	18
4	1	4	6	5	7	8	9	10	2	17	11	12	13	3	14	15	16	18
5	1	15	16	17	5	7	8	9	10	2	3	11	12	4	14	13	6	18
6	1	12	17	4	6	5	13	14	15	16	8	7	9	10	2	3	11	18
7	1	11	12	17	13	16	15	9	10	2	3	4	7	5	14	8	6	18
8	1	4	16	6	5	7	8	14	17	13	12	15	9	10	2	11	3	18
9	1	4	6	5	7	8	9	10	17	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	17	8	2	3	12	11	13	14	15	16	9	10	18
11	1	4	6	5	11	12	13	14	7	16	8	9	10	2	15	3	17	18
12	1	4	6	5	7	8	9	10	2	3	12	11	13	14	16	15	17	18
13	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
14	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
15	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	16	15	18
16	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	17	2	18
17	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
18	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
19	1	4	6	5	7	8	17	9	10	2	3	11	13	14	16	15	12	18
20	1	4	6	15	16	17	8	9	10	2	3	11	12	13	14	5	7	18
21	1	4	16	6	5	7	8	9	10	2	3	11	12	15	17	14	13	18
22	1	4	6	5	7	8	9	10	2	17	3	11	12	13	14	15	16	18
23	1	4	6	15	16	17	8	9	10	2	3	11	13	14	7	5	12	18
24	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	15	6	5	7	8	17	2	3	12	11	13	14	4	16	9	10	18
30	1	4	6	5	7	8	9	10	17	2	3	11	12	13	14	15	16	18
31	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	6	17	18
32	1	4	6	5	7	8	9	10	2	17	3	11	12	15	16	13	14	18
33	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
34	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
35	1	14	13	3	5	8	11	15	4	7	6	2	16	9	10	12	17	18
36	1	4	6	5	7	8	9	10	2	17	15	16	13	14	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
39	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
40	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18

Genetik algoritma tamamlandıktan sonra (3.35) ve (3.36) numaralı formüllerle Çizelge 4.17'deki popülasyonda optimum güzergah kümesi, 13 numaralı küme olarak belirlenmiş ve iyileştirme adımlarına geçilmiştir. Seçilen optimum birey (3.37), (3.38) ve (3.39) numaralı formüller güzergahlara ayrılmış ve bu güzergahlardan biri rastgele seçilmiştir. (3.40) numaralı formülle rastgele seçilen güzergahtan başlangıç bitiş durakları hariç kalan duraklardan birer tanesinin olmadığı güzergahlar listelenmiştir. Bu listedeki güzergahların *ADP* değerleri hesaplanarak, %50 oranında (3.41) ve (3.42) numaralı formüllerle, %50 oranında ise rulet yöntemiyle en kötü konuma sahip duraklar belirlenmiştir. Burada rulet yönteminin kullanılması her seferinde aynı durağın seçilmesini önlemektedir. En kötü konuma sahip durak belirlendikten sonra (3.43), (3.44), (3.45) ve (3.46) numaralı formüllerle bu durağa en yakın durak belirlenmiştir. Son olarak (3.47), (3.48), (3.49) ve (3.50) numaralı formüllerle durağın yerleştirme işlemi tamamlanmıştır. Durağın en uygun konuma yerleştirilmesi iyileştirmesi popülasyonun yaklaşık %15'ine uygulanmaktadır. Bu bireyler rastgele 23, 29 ve 32 numaralı güzergah kümeleri şeklinde seçilmiş ve her birey için aynı işlemler tekrar uygulanmıştır. Bu iyileştirme adımı için 13 numaralı güzergah kümesinde gerçekleşen değişiklikler sonucu popülasyon Çizelge 4.18'deki halini almıştır. Ayrıca bu değişiklikler Şekil 4.4'te gösterilmiştir.

13	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
23	1	12	4	16	6	5	7	8	9	10	2	3	11	13	14	15	17	18
29	1	4	16	6	5	17	7	8	9	10	2	3	11	12	13	14	15	18
32	1	4	16	8	6	5	7	9	10	2	3	11	12	13	14	15	17	18

Şekil 4.4. 300. iterasyon için 13 numaralı optimum güzergah kümesine uygulanan durağın en uygun konuma yerleştirilmesi iyileştirmesi

Son olarak güzergahlar arası parça değiştirme iyileştirmesi ile 12, 13, 18, 19, 21, 30, 34 ve 38 numaralı güzergahlar güncellenmiş ve 300. iterasyon Çizelge 4.19'un elde edilmesiyle tamamlanmıştır.

Çizelge 4.18. 300. iterasyon için durak konumu değiştirme iyileştirmesi

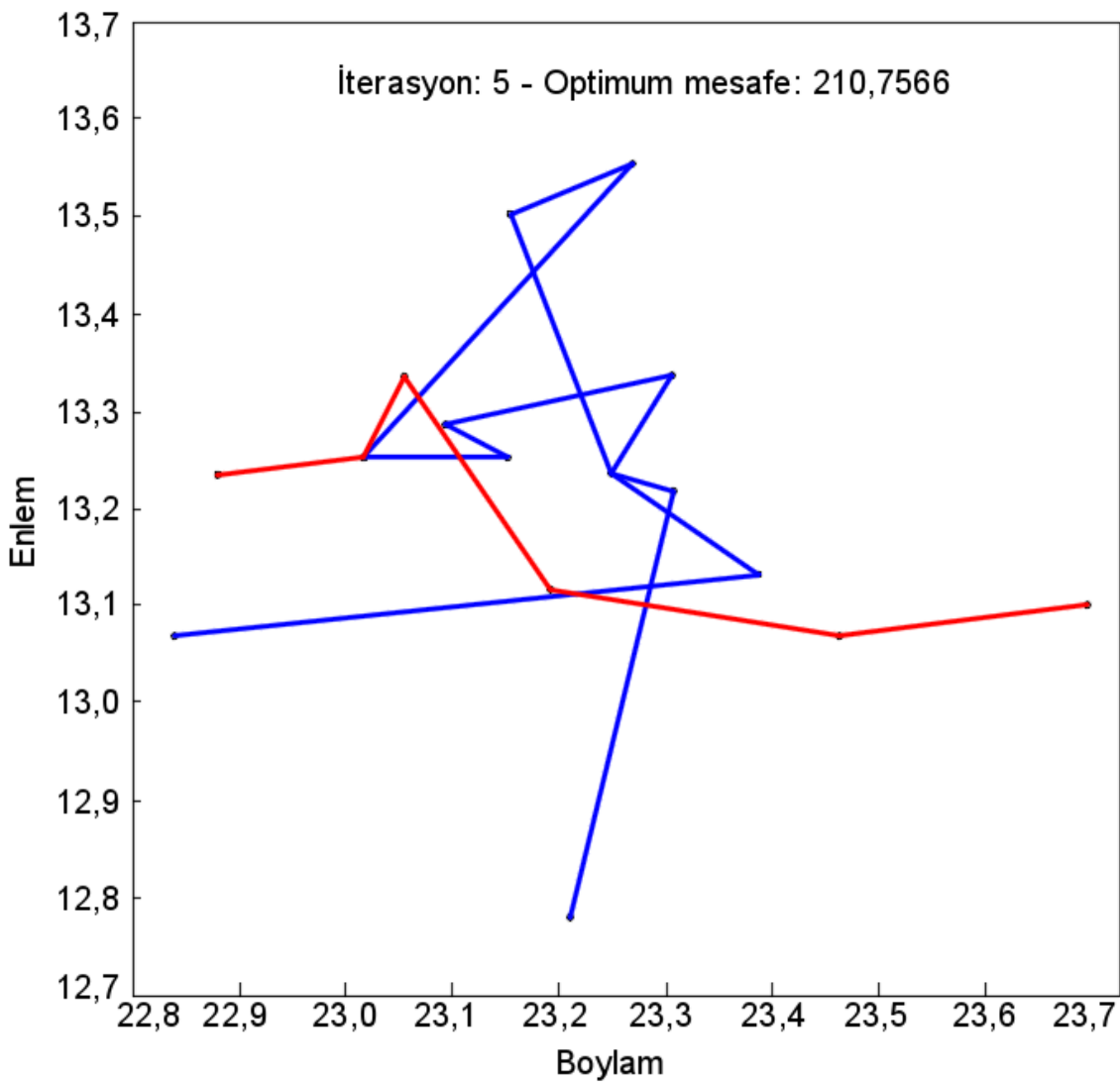
300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
2	1	6	5	7	8	9	10	17	16	2	3	11	13	14	15	4	12	18
3	1	4	16	6	5	7	8	9	10	2	3	11	12	14	15	17	13	18
4	1	4	6	5	7	8	9	10	2	17	11	12	13	3	14	15	16	18
5	1	15	16	17	5	7	8	9	10	2	3	11	12	4	14	13	6	18
6	1	12	17	4	6	5	13	14	15	16	8	7	9	10	2	3	11	18
7	1	11	12	17	13	16	15	9	10	2	3	4	7	5	14	8	6	18
8	1	4	16	6	5	7	8	14	17	13	12	15	9	10	2	11	3	18
9	1	4	6	5	7	8	9	10	17	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	17	8	2	3	12	11	13	14	15	16	9	10	18
11	1	4	6	5	11	12	13	14	7	16	8	9	10	2	15	3	17	18
12	1	4	6	5	7	8	9	10	2	3	12	11	13	14	16	15	17	18
13	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
14	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
15	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	16	15	18
16	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	17	2	18
17	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
18	1	15	6	5	7	8	17	9	10	2	3	11	12	13	14	4	16	18
19	1	4	6	5	7	8	17	9	10	2	3	11	13	14	16	15	12	18
20	1	4	6	15	16	17	8	9	10	2	3	11	12	13	14	5	7	18
21	1	4	16	6	5	7	8	9	10	2	3	11	12	15	17	14	13	18
22	1	4	6	5	7	8	9	10	2	17	3	11	12	13	14	15	16	18
23	1	12	4	16	6	5	7	8	9	10	2	3	11	13	14	15	17	18
24	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	4	16	6	5	17	7	8	9	10	2	3	11	12	13	14	15	18
30	1	4	6	5	7	8	9	10	17	2	3	11	12	13	14	15	16	18
31	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	6	17	18
32	1	4	16	8	6	5	7	9	10	2	3	11	12	13	14	15	17	18
33	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
34	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
35	1	14	13	3	5	8	11	15	4	7	6	2	16	9	10	12	17	18
36	1	4	6	5	7	8	9	10	2	17	15	16	13	14	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
39	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
40	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18

Çizelge 4.19. 300. iterasyon için parça değiştirme iyileştirmesi

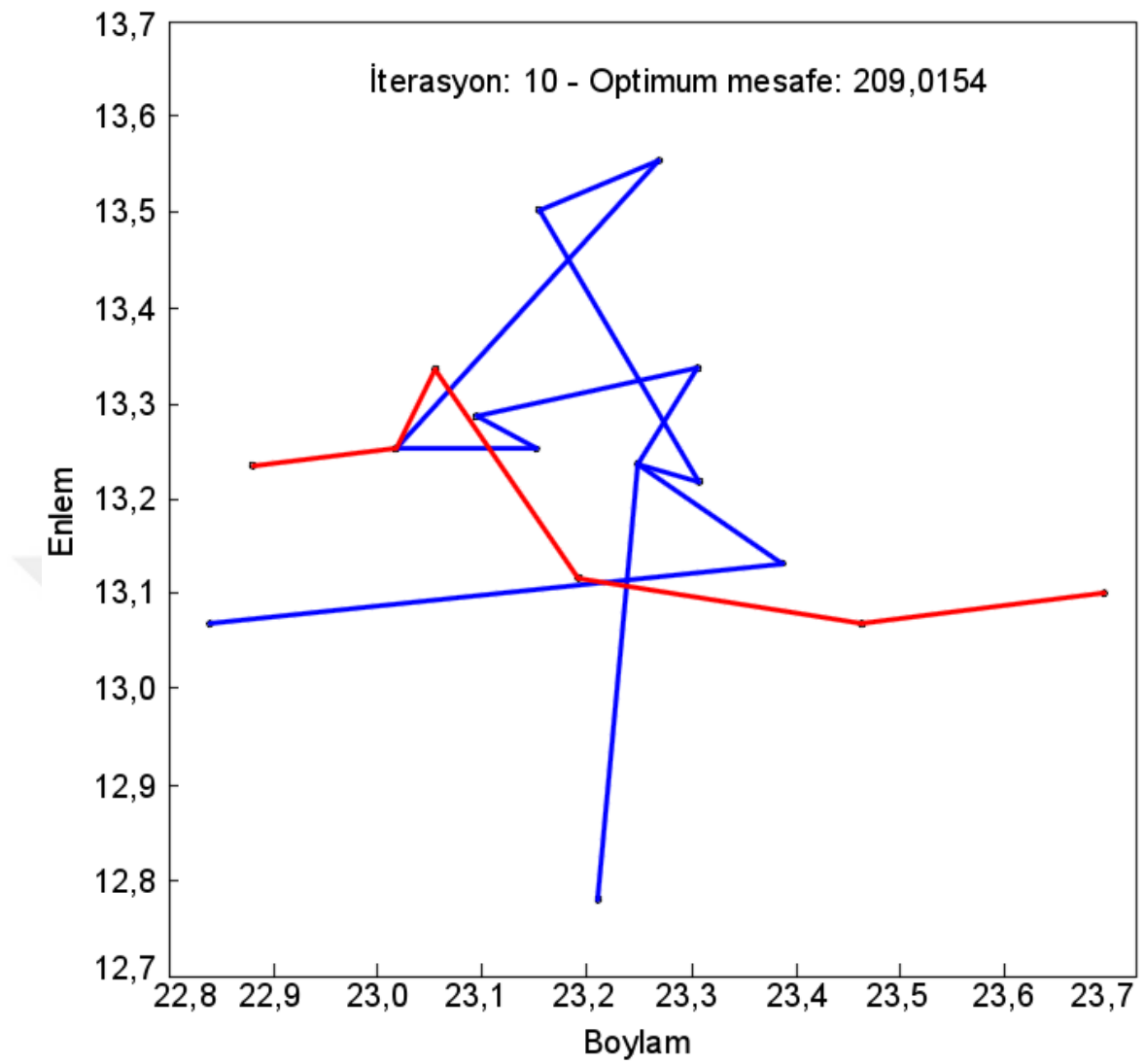
300	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	4	16	6	5	7	8	9	10	2	3	11	12	13	14	15	17	18
2	1	6	5	7	8	9	10	17	16	2	3	11	13	14	15	4	12	18
3	1	4	16	6	5	7	8	9	10	2	3	11	12	14	15	17	13	18
4	1	4	6	5	7	8	9	10	2	17	11	12	13	3	14	15	16	18
5	1	15	16	17	5	7	8	9	10	2	3	11	12	4	14	13	6	18
6	1	12	17	4	6	5	13	14	15	16	8	7	9	10	2	3	11	18
7	1	11	12	17	13	16	15	9	10	2	3	4	7	5	14	8	6	18
8	1	4	16	6	5	7	8	14	17	13	12	15	9	10	2	11	3	18
9	1	4	6	5	7	8	9	10	17	2	3	11	12	13	16	15	14	18
10	1	4	6	5	7	17	8	2	3	12	11	13	14	15	16	9	10	18
11	1	4	6	5	11	12	13	14	7	16	8	9	10	2	15	3	17	18
12	1	4	16	6	5	3	11	12	13	14	15	9	10	2	7	8	17	18
13	1	4	16	6	5	3	11	12	13	14	15	9	10	2	7	8	17	18
14	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
15	1	4	6	5	7	8	17	9	10	2	3	11	12	13	14	16	15	18
16	1	4	6	3	11	12	13	16	15	5	7	8	9	10	14	17	2	18
17	1	4	6	5	7	8	9	10	2	17	11	12	13	14	15	16	3	18
18	1	4	16	6	5	3	11	12	13	14	15	9	10	2	7	8	17	18
19	1	4	16	6	5	12	13	14	15	17	9	10	2	3	11	7	8	18
20	1	4	6	15	16	17	8	9	10	2	3	11	12	13	14	5	7	18
21	1	4	16	12	13	14	15	17	7	8	9	10	2	3	11	6	5	18
22	1	4	6	5	7	8	9	10	2	17	3	11	12	13	14	15	16	18
23	1	12	4	16	6	5	7	8	9	10	2	3	11	13	14	15	17	18
24	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
25	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
26	1	4	6	5	7	8	9	10	2	3	12	11	13	14	15	16	17	18
27	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
28	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
29	1	4	16	6	5	17	7	8	9	10	2	3	11	12	13	14	15	18
30	1	12	13	14	15	17	6	5	7	8	9	10	2	3	11	4	16	18
31	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	6	17	18
32	1	4	16	8	6	5	7	9	10	2	3	11	12	13	14	15	17	18
33	1	4	6	15	16	17	8	9	10	2	3	11	12	13	7	5	14	18
34	1	4	16	3	11	12	13	14	15	7	8	9	10	2	6	5	17	18
35	1	14	13	3	5	8	11	15	4	7	6	2	16	9	10	12	17	18
36	1	4	6	5	7	8	9	10	2	17	15	16	13	14	3	11	12	18
37	1	4	6	5	11	12	13	14	7	16	8	9	10	2	3	15	17	18
38	1	3	11	12	13	14	15	6	5	7	8	9	10	2	4	16	17	18
39	1	13	14	15	16	8	7	9	10	2	3	11	12	4	5	17	6	18
40	1	4	6	5	7	8	9	10	2	3	11	12	13	14	15	16	17	18

Geliştirilen optimizasyon yöntemiyle, iterasyonlar sonucu ortaya çıkan optimum çözüm kümelerinin takibi açısından, güzergah kümeleri güzergahlara ayrılarak grafikleri çizdirilmiştir. Böylece çözümün nasıl değiştiği, hangi parametrelerin çözümü ne şekilde etkilediği gibi sonuçlar kolayca gözlemlenmiştir. Fakat problemin çözüm süresinin uzamasından dolayı her 5 iterasyonda bir grafik çizdirilmiştir. Örnek olarak 5, 10, 15, 20, 65, 295, 300, 450 ve 460 numaralı iterasyon sonuçları verilmiştir.

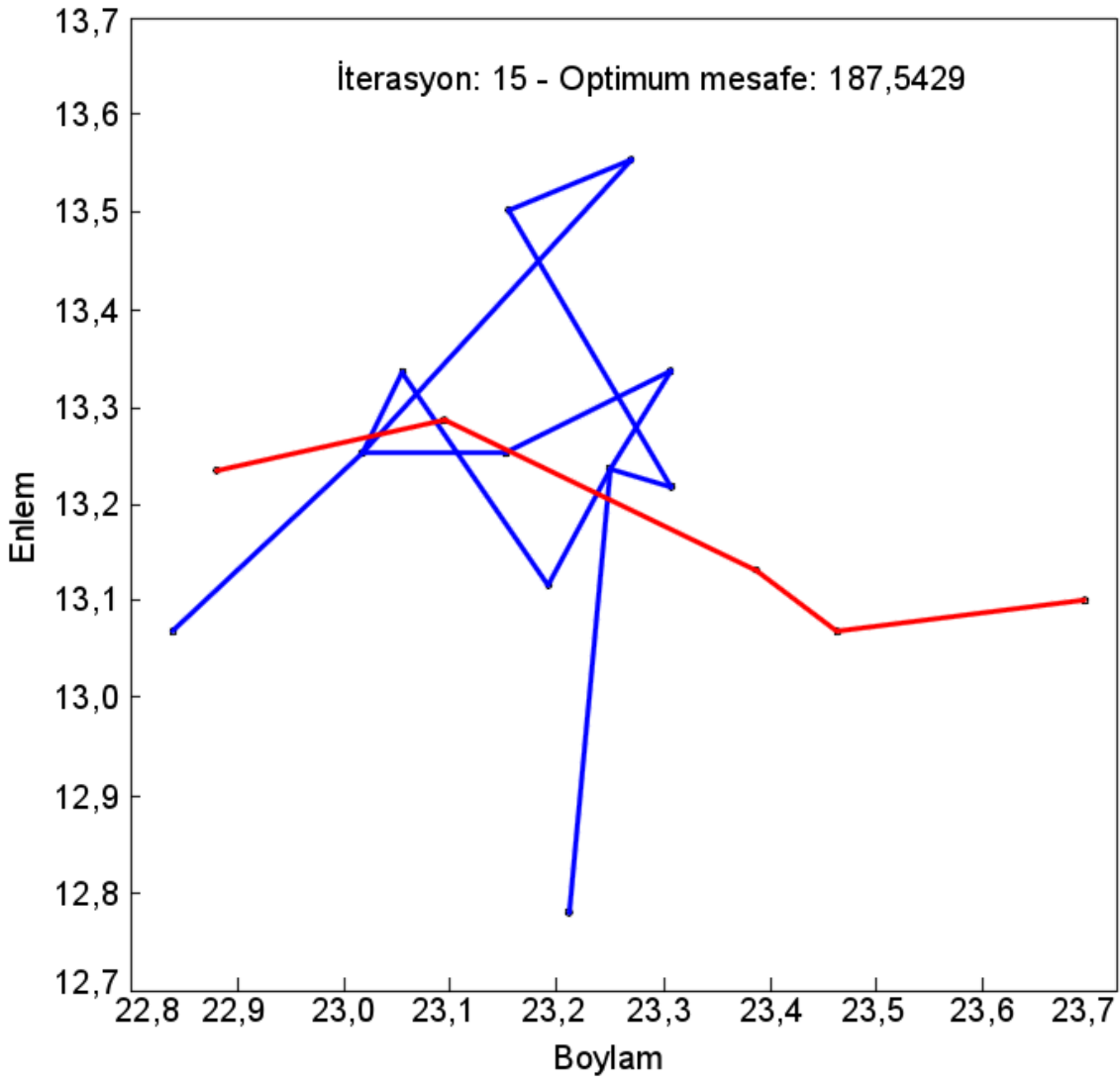
Şekil 4.5'te 5. iterasyon sonucunda optimum güzergah kümesi için yolculuk mesafesi 210,7566; Şekil 4.6'da 10. iterasyon sonucunda 209,0154; Şekil 4.7'de 15. iterasyon sonucunda 187,5429; Şekil 4.8'de 20. iterasyon sonucunda 181,423; Şekil 4.9'da 65. iterasyon sonucunda 164,1852; Şekil 4.10'da 295. iterasyon sonucunda 137,9383'tür.



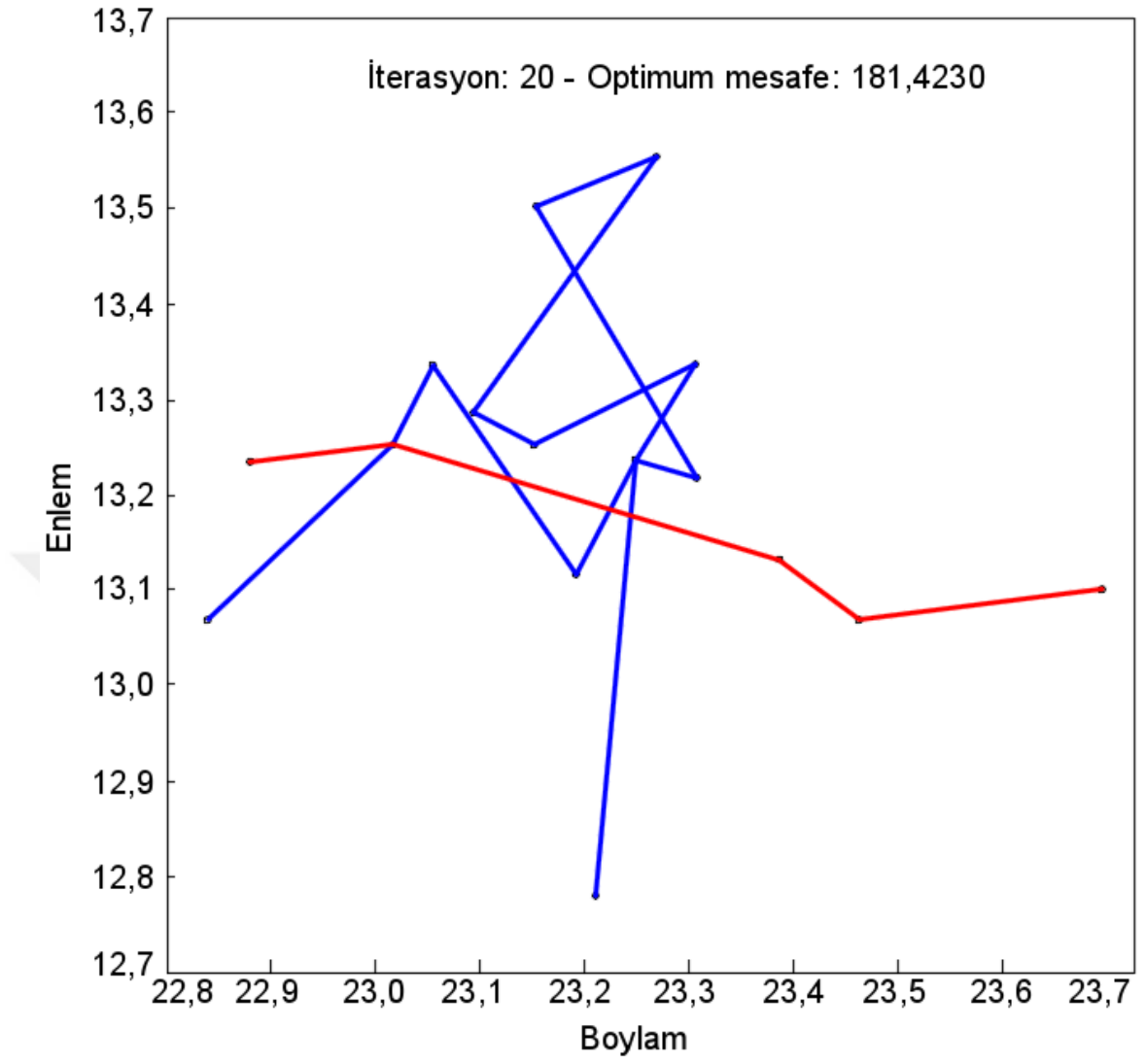
Şekil 4.5. 5. iterasyondaki optimum çözüm



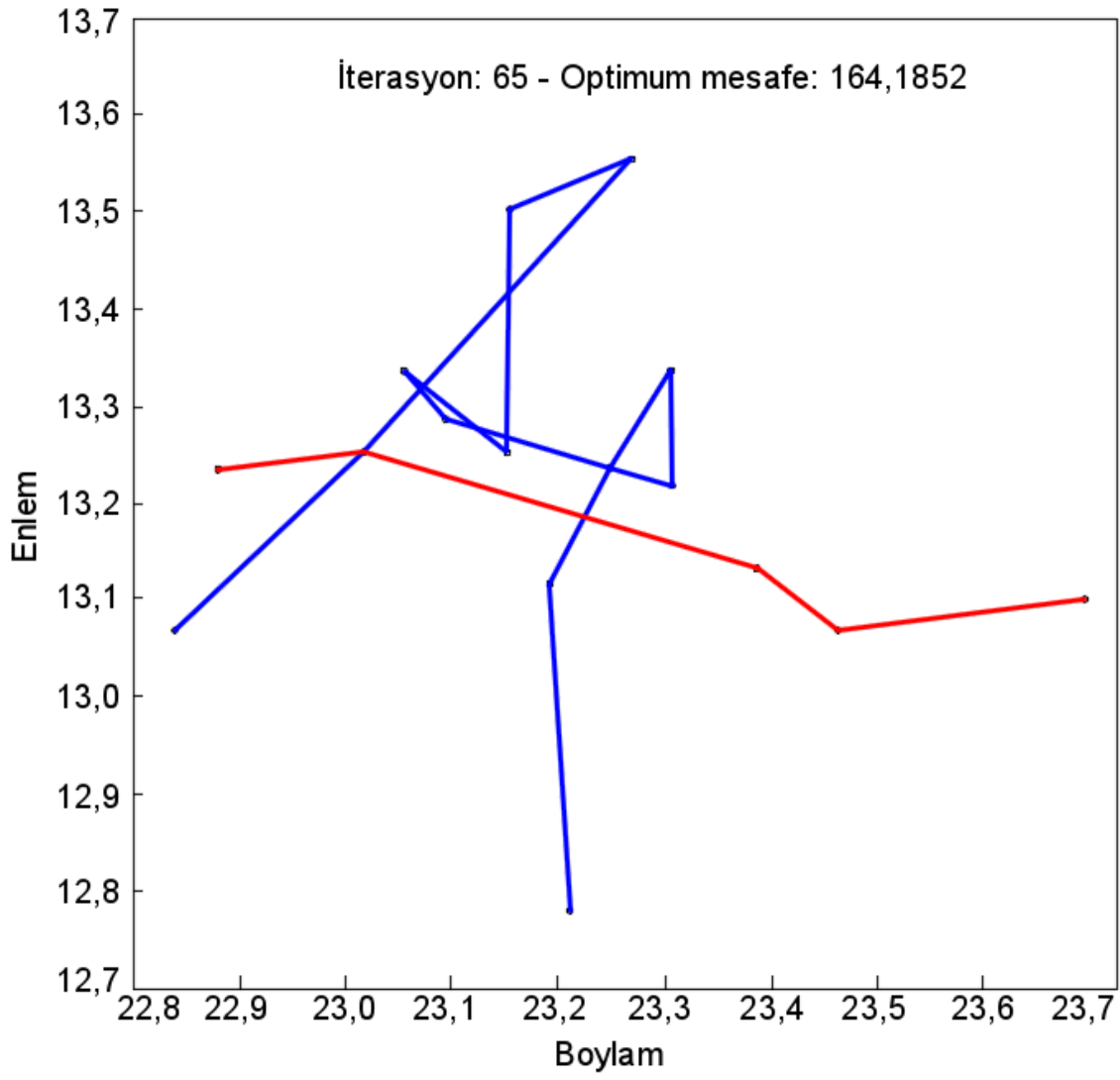
Şekil 4.6. 10. iterasyondaki optimum çözüm



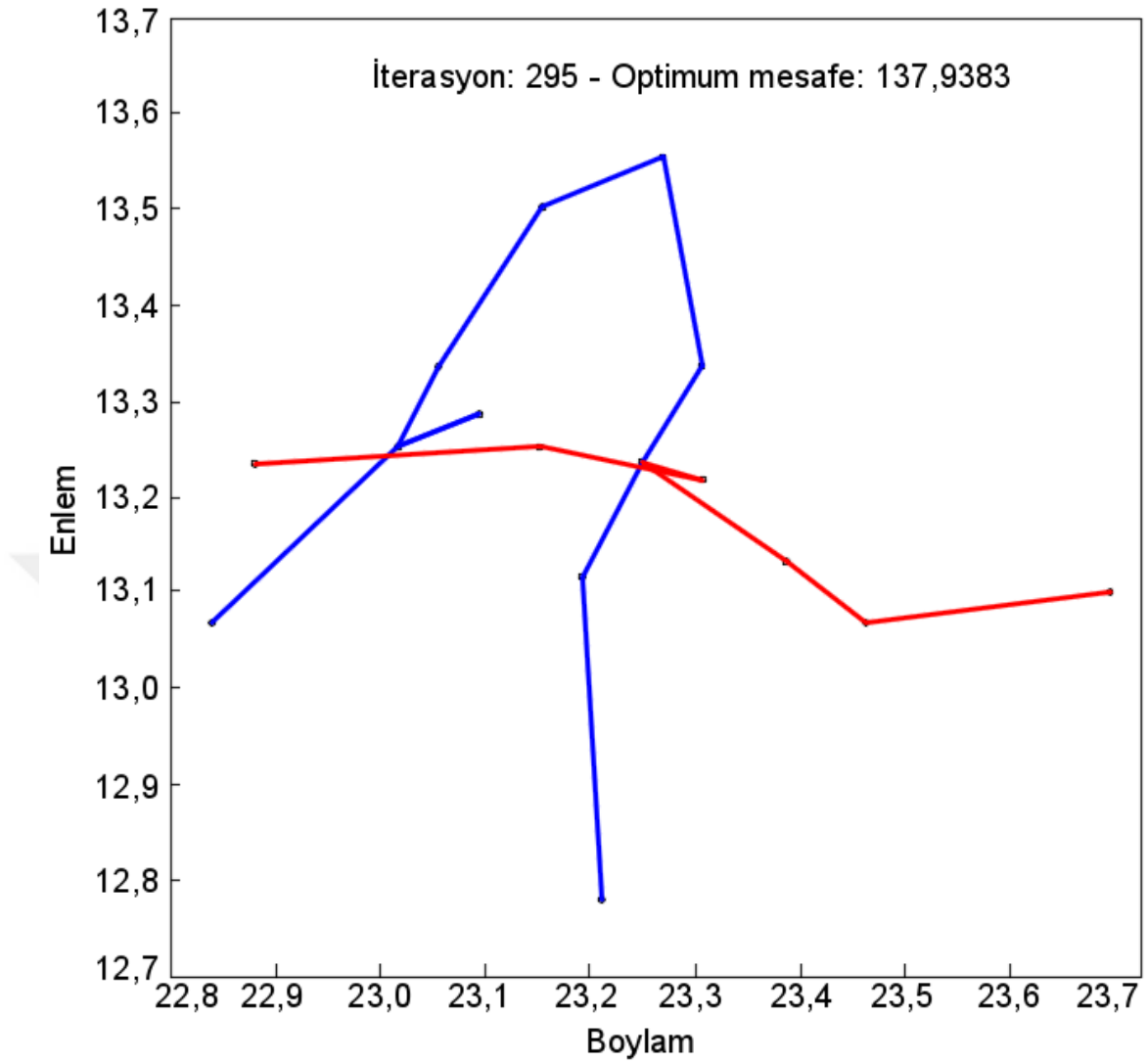
Şekil 4.7. 15. iterasyondaki optimum çözüm



Şekil 4.8. 20. iterasyondaki optimum çözüm



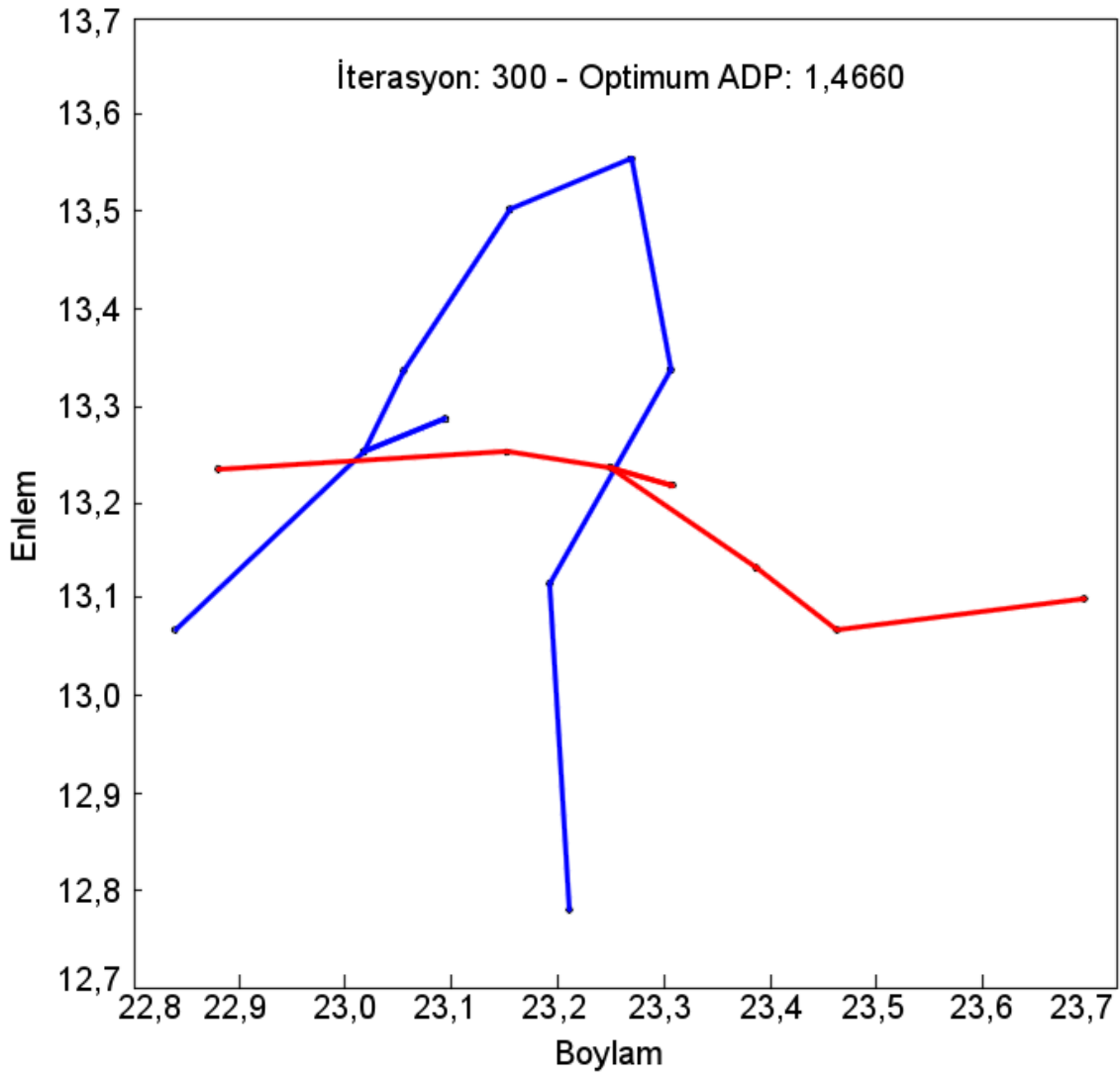
Şekil 4.9. 65. iterasyondaki optimum çözüm



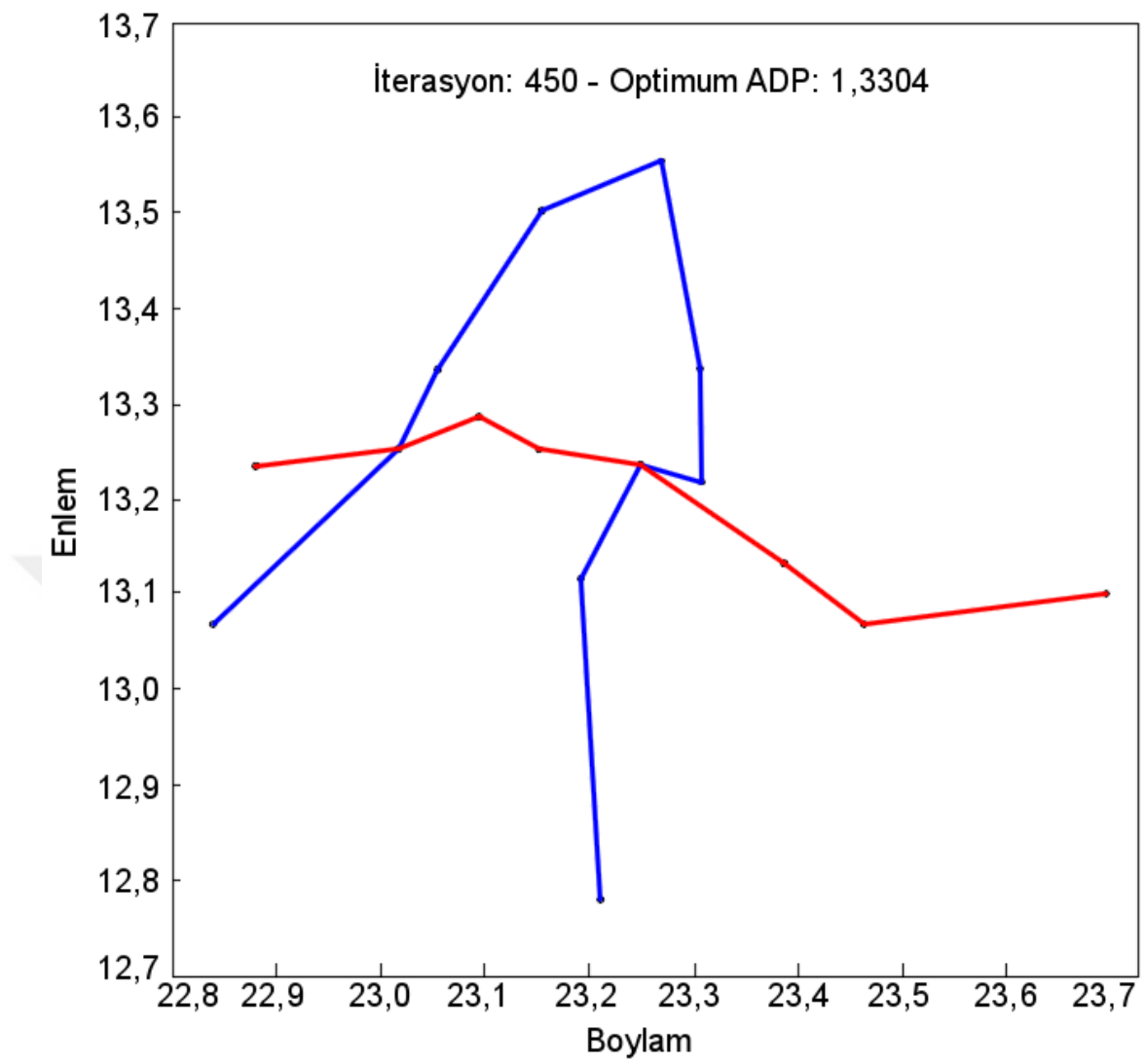
Şekil 4.10. 295. iterasyondaki optimum çözüm

Şekil 4.11'deki 300. iterasyona kadar güzergah kümeleri yolculuk mesafesine göre değerlendirilirken, 300. iterasyondan sonra değerlendirmeler *ADP* değerine yapılmıştır. 300. iterasyona kadar olan değişimlere bakıldığında çözümün genellikle güzergahlar arasında durak alış verişi ile değil güzergahın kendi içindeki durak dizilimlerinin değiştirilmesiyle geliştiği görülmektedir. Bu durum çoklu güzergah tasarımlarında, genetik algoritmanın problemin çözümünde tıkanmasına yol açan bir etkidir.

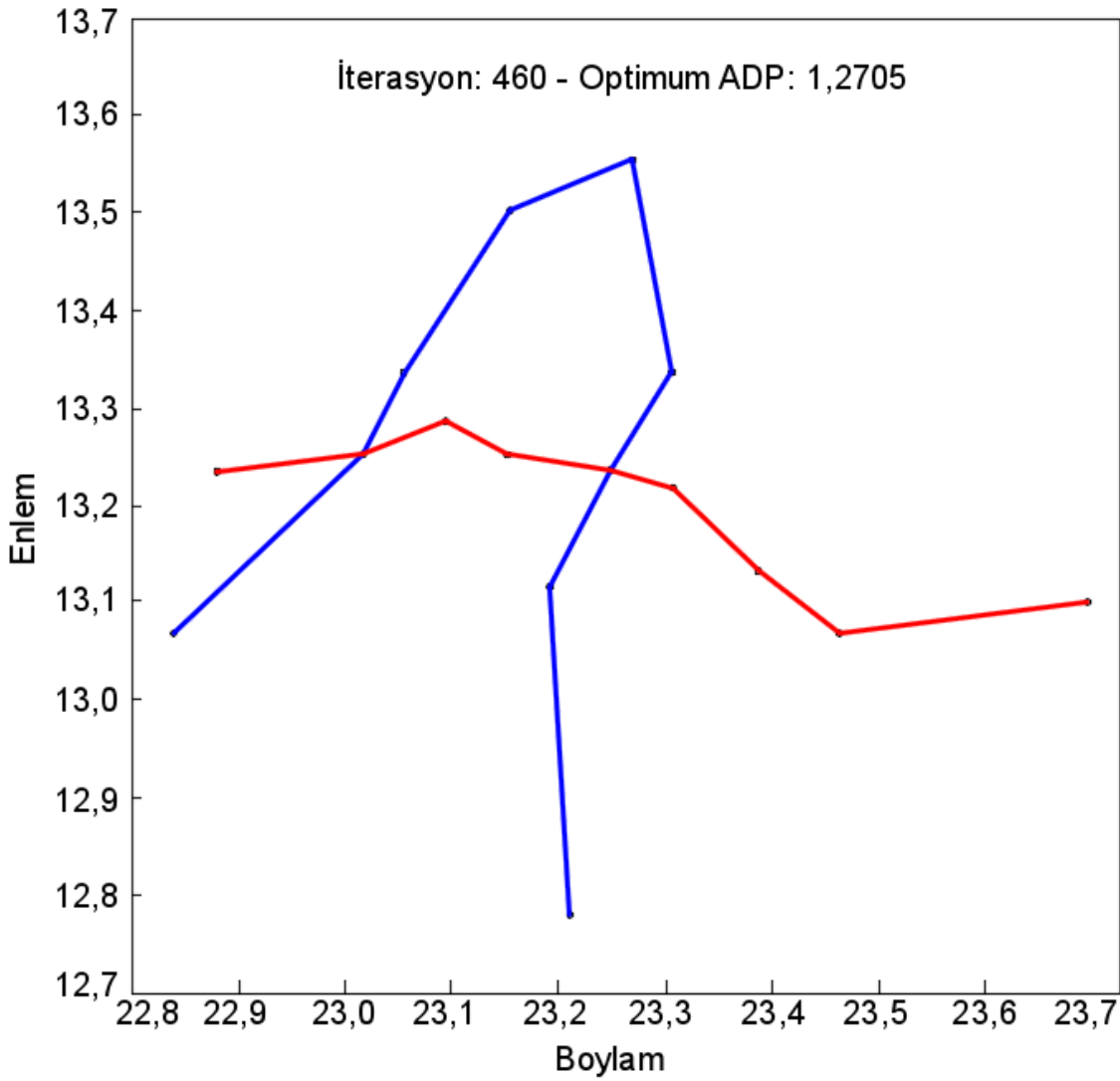
Şekil 4.11'de 300. iterasyon sonucunda $ADP = 1,466$; Şekil 4.12'de 450. iterasyon sonucunda $ADP = 1,3304$; Şekil 4.13'te ise 460. iterasyon sonucunda $ADP = 1,2705$ çıkmıştır. Bundan sonrasında ise sonuç değişmemiştir.



Şekil 4.11. 300. iterasyondaki optimum çözüm



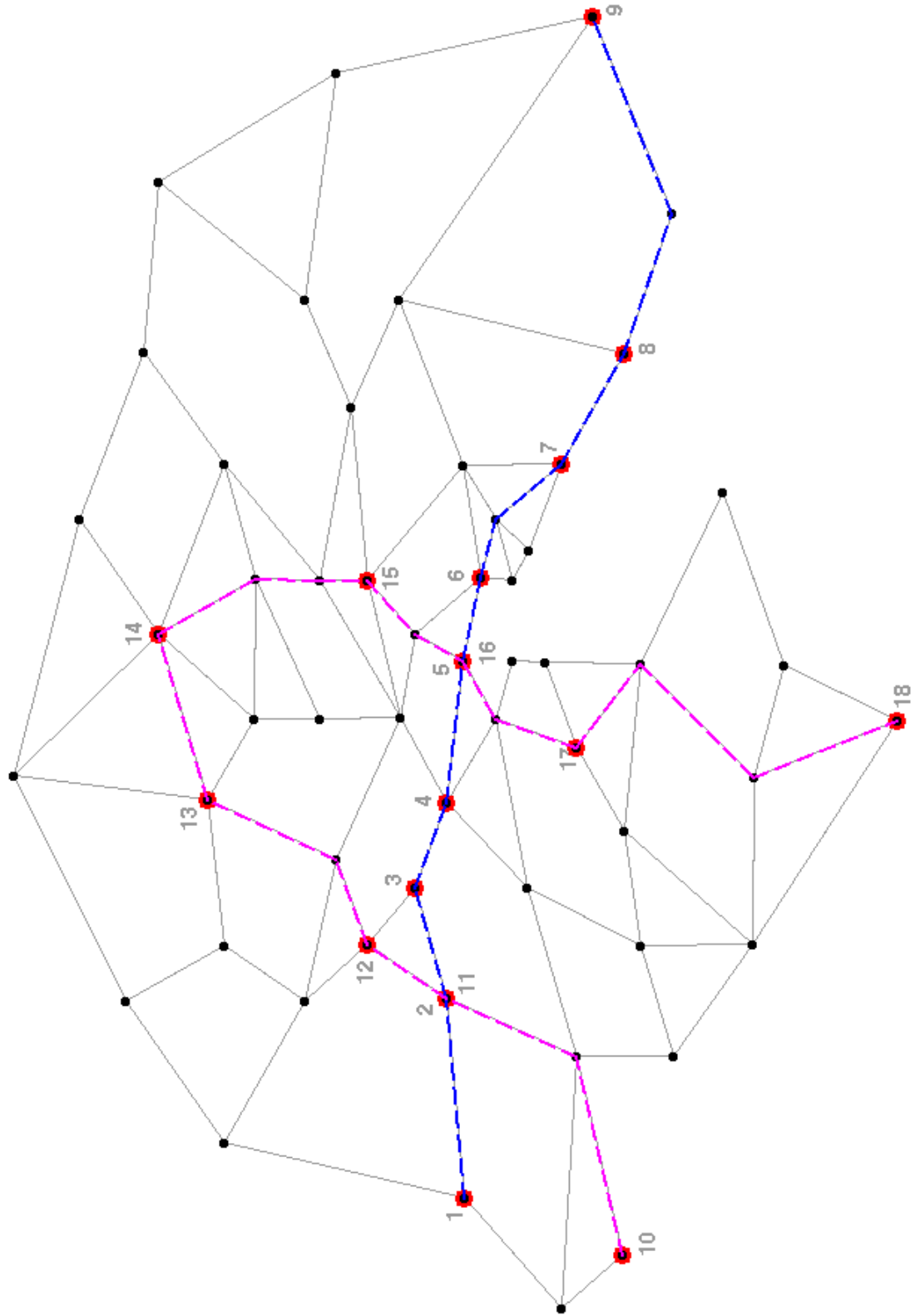
Şekil 4.12. 450. iterasyondaki optimum çözüm



Şekil 4.13. 460. iterasyondaki optimum çözüm

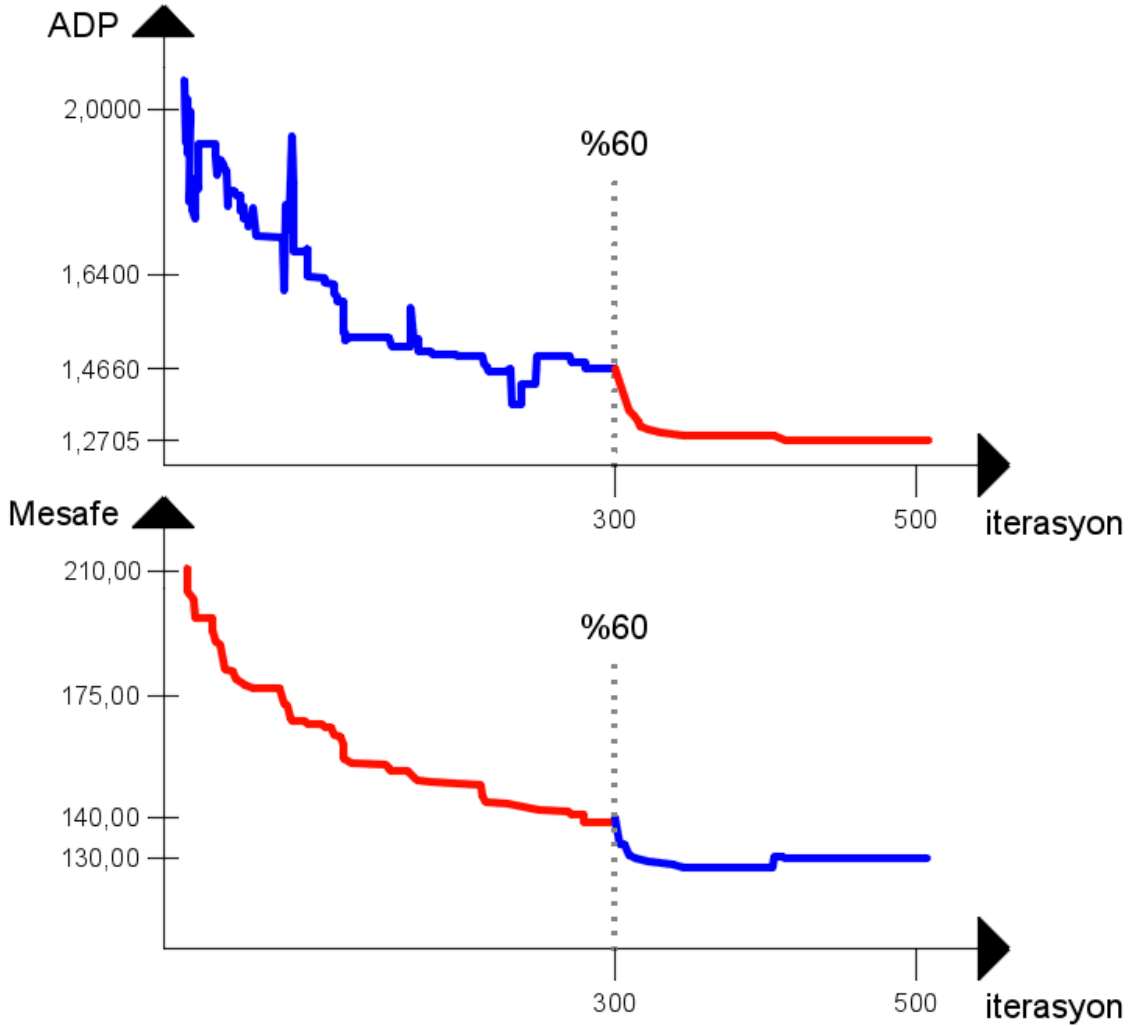
300. iterasyondan sonraki çözümler incelendiğinde, güzergahlar arasında durak geçişleri sağlayarak farklı çözüm arayışlarının önü açarak ve daha verimli sonuçlar üreten bir yaklaşım görülmektedir. Bu da genetik algoritmanın lokal optimum çözümlerden global optimum çözümlere geçişini sağlamaktadır.

Son olarak üretilen optimum güzergah kümesinin ağ üzerindeki izlediği yol Şekil 4.14'te çizdirilerek elde edilen güzergahlar gösterilmiştir. Şekilde, oluşturulan bu örnek için ideal bir çözüm üretildiği görülmektedir.



řekil 4.14. Ağ üzerinde hesaplanan optimum güzergah kümesi

Geliştirilen yöntem için iterasyon sayısına göre güzergah mesafesi ve *ADP* değişimlerini gösteren Swain örneğine ait grafik Şekil 4.15'te verilmiştir.



Şekil 4.15. İterasyon sayısına göre mesafe ve *ADP* değişimleri

Şekil 4.15'te görüldüğü üzere *ADP* değerleri ilk %60'a kadar olan iterasyonlarda bazen azalmakta bazen ise artmaktadır. Aynı durum mesafe değerleri için ise %60'tan sonraki iterasyonlarda görülmektedir. Bunun sebebi optimizasyonda %60'a kadar mesafe, sonrasında ise *ADP* değerine göre minimizasyon yapılmış olmasıdır. Optimizasyonda bu şekilde dinamik bir amaç fonksiyonu kullanılması, sonuçların tek bir amaç fonksiyonuna göre yapıldığı durumlardan daha verimli sonuçlar üretmiştir. Yapılan denemeler sonucunda, değişimin %60 civarında gerçekleşmesinin daha verimli sonuçlar ürettiği görülmüştür.

Geliştirilen optimizasyon yöntemi ile İstanbul şehrindeki otobüs hatlarına ait güzergahların yeniden tasarlanarak yöntemin daha büyük örneklerde test edilmesi aşamasına geçilmiştir.

4.2. İstanbul Şehrine Ait Otobüs Hatları İçin Uygulama

İstanbul şehrinde otobüs hatları için bu uygulamanın bir kere de yapılması verimli sonuçlar vermeyeceği gibi işlem uzunluğu da çok uzun sürecektir. Bu yüzden Çizelge 3.9’da otobüs hatları birbirlerine olan yakınlıklarına göre k-means yöntemiyle gruplandırılmıştır. Böylece 250 farklı problem oluşturulmuştur.

Yolculuk sayısı verisi bulunmadığı için her hat için tüm yolculuk kombinasyonları arasında 1 yolculuk yapıldığı varsayımı yapılmıştır. Mesafe verileri Google Maps’ten çekilebilmektedir. Fakat tüm hatlar için yapılacak veri çekme işlemin süresi yaklaşık 10 yıldan fazla sürecektir. Bu yüzden mesafe verileri olarak koordinatlar yardımıyla hesaplanan öklid mesafe verileri kullanılmıştır. Sonrasında sadece bir örnek için Google Maps’ten çekilen verilerle de problem çözülmüştür.

4.2.1. Öklid mesafe ile sonuçlar

Oluşturulan 250 küme için öncelikle mevcut *ADP* değerleri hesaplanmıştır. Daha sonra hem sadece genetik algoritma (GA) ile hem de geliştirilen hibrit modelle optimizasyon işlemleri gerçekleştirilmiştir. 250 kümeye ait sonuçlar EK-1’de verilmiştir. Küme, içinde bulundurduğu hat sayılarına göre 8 gruba ayrılmıştır. Bu gruplar için elde edilen ortalama *ADP* değerleri Çizelge 4.20’de verilmiştir.

Çizelge 4.20. 8 grup için mevcut (M *ADP*) ortalamaları ile hibrit model (H *ADP*) ve genetik algoritma (GA *ADP*) sonucu elde edilen *ADP* ortalamaları

Kümeye Bulunan Hat Sayısı	M <i>ADP</i> ortalamaları	H <i>ADP</i> ortalamaları	GA <i>ADP</i> ortalamaları
1	1,630	1,487	1,565
2	1,888	1,561	2,409
3	1,624	1,451	5,475
4	1,758	1,499	5,657
5	1,933	1,552	10,140
6	1,373	1,376	11,067
7	3,033	1,846	12,702
10	3,704	1,954	14,918
Σ	1,754	1,503	4,989

Çizelge 4.20’de görüldüğü üzere hibrit model ile çözüm üretilen 250 kümeden sadece 6 hat bulunan gruba ait *ADP* değeri ortalaması mevcut duruma göre daha yüksek çıkmıştır. Genetik algoritma ile üretilen çözümlerde ise sadece 1 hat bulunan kümeye ait *ADP* değeri ortalaması mevcut duruma göre daha iyi çıkmıştır. Bu sonuç genetik algoritmanın çoklu güzergah tasarımlarında başarısız olduğunu göstermektedir. Hibrit model ile yapılan çözümlerin, hem tek güzergah tasarımında hem de çoklu güzergah tasarımlarında genetik algoritmaya göre üstün bir başarı gösterdiğini söylemek mümkündür. 250 kümenin tamamına bakacak olursak *ADP* değeri genetik algoritma ile hesaplandığında sonuçlar %184,3 oranında mevcut değere göre daha kötü çıkmıştır.

Hibrit model ile yapılan sonuçlar ise %16,7 oranında daha iyidir. Bu verimlilik yüzdeleri 250 küme içerisindeki 8 grup için ayrı ayrı Çizelge 4.21’de verilmiştir.

Çizelge 4.21. Hibrit modelin mevcut duruma göre verimlilik yüzdeleri

Kümeye Bulunan Hat Sayısı	Hibrit Modelin Verimlilik Yüzdesi
1	9,62
2	20,95
3	11,92
4	17,28
5	24,55
6	-0,22
7	64,30
10	89,59
Σ	16,70

Optimizasyon hesapları sonucunda 250 kümenin yüzde 74,8’inde mevcut duruma göre daha iyi *ADP* değerlerine sahip kümeler elde edilmiştir (Çizelge 4.22).

Çizelge 4.22. Optimizasyon sonucunda daha iyi performansa sahip kümeler

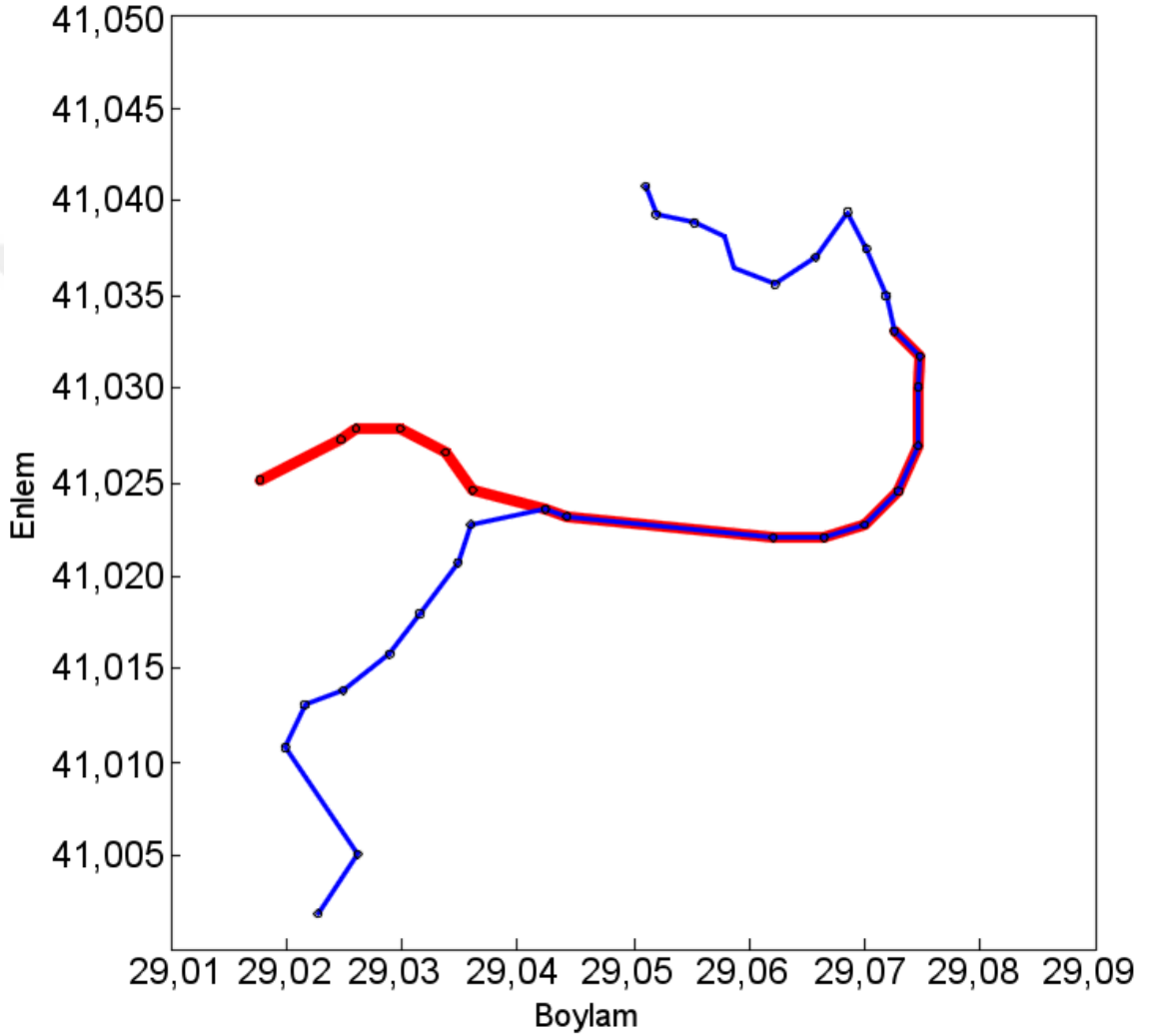
Kümeye Bulunan Hat Sayısı	Küme Sayısı	Hibrit modele göre <i>ADP</i> ’si daha iyi çıkan kümelerin sayısı	Hibrit modele göre <i>ADP</i> ’si daha iyi çıkan kümelerin yüzdesi
1	54	46	85,19
2	40	33	82,50
3	87	59	67,82
4	30	24	80,00
5	27	16	59,26
6	5	3	60,00
7	6	5	83,33
10	1	1	100,00
Σ	Σ =250	Σ =187	74,80

Hibrit model ile genetik algoritmanın *ADP* değerleri kıyaslandığında ise hibrit model %69,8 daha verimli sonuçlar üretmiştir. 250 küme içindeki sadece mevcut güzergahlara göre daha iyi çıkan kümeler incelendiğinde ise *ADP* değerlerinin ortalaması mevcut durum için 1,893; hibrit model için ise 1,540 çıkmıştır. Dolayısıyla mevcut hatlara göre verim oranı %16,7 ‘ten %22,9’a çıkmaktadır.

ADP değerleri 250 küme yerine, bu kümeler içerisinde yer alan toplam 732 hat için incelendiğinde mevcut durum için *ADP* ortalaması 1,831; hibrit model için ise 1,534 çıkmıştır. Burada %19,7’lik bir fark vardır. Sadece daha iyi sonuç veren kümeler incelendiğinde ise hibrit model mevcut duruma göre %28,5’lik bir verim sağlamıştır.

4.2.2. Google Maps ve öklid mesafe karşılaştırması

250 küme üzerinde yapılan çalışmada, öklid mesafeler kullanılmasına rağmen *ADP* değerleri mevcut durumla kıyaslandığında daha iyi performansa sahip sonuçların üretildiği görülmüştür. Burada öklid yerine Google Maps'ten elde edilecek gerçek verilerin kullanılması bu performansı daha da arttıracaktır. Bunun için öklid mesafe kullanılarak gerçekleştirilen değerlendirme sonucu, performansı daha kötü çıkan bir örnek ele alınmıştır (Şekil 4.16).



Şekil 4.16. 184 numaralı güzergah kümesine ait mevcut güzergah

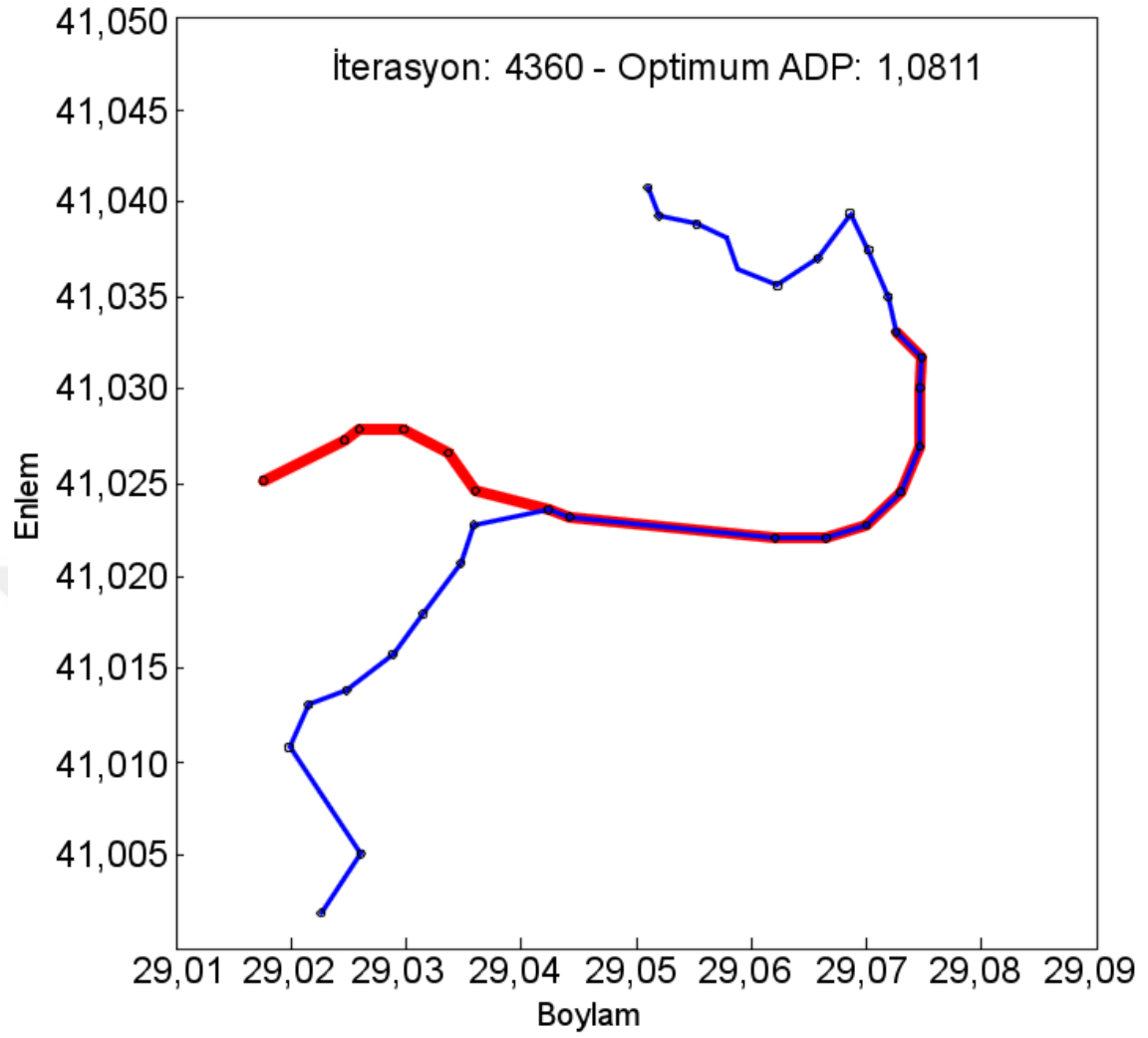
Şekil 4.16'da 184 numaralı kümede bulunan mevcut güzergahlar gösterilmiştir. Modelde (3.2) numaralı formülde görüldüğü gibi payda ve payda da kullanılmak üzere iki farklı mesafe verisi kullanılmaktadır. Bu fonksiyondaki mesafe verileri Çizelge 4.23'te görüldüğü gibi 3 farklı şekilde düzenlenerek aynı optimizasyon modeliyle işleme sokulmuştur.

Çizelge 4.23. Alternatif ADP hesapları

No	ADP Fonksiyonu
1	$\frac{Google\ Maps_{ij}^{güzergah}}{Google\ Maps_{ij}^{en\ kısa}}$
2	$\frac{Google\ Maps_{ij}^{güzergah}}{Öklid_{ij}^{en\ kısa}}$
3	$\frac{Öklid_{ij}^{güzergah}}{Öklid_{ij}^{en\ kısa}}$

Çizelge 4.23'te *ADP* fonksiyonlarında Google Maps ve öklid olarak *i* ve *j* nolu duraklar arasında kalan yolculuk mesafeleri yer almaktadır. Güzergahta bulunan *i* ve *j* nolu duraklar arasındaki yolculuk mesafesi için durak dizilimlerinin güzergahta belirtilen dizilimi takip etmesi gerekmektedir. En kısa mesafedeki *i* ve *j* nolu duraklar arasındaki mesafe için ise güzergahta bulunan durak dizilimlerine ihtiyaç yoktur. Google Maps'te bulunan en kısa mesafe yol ağında bulunan en kısa mesafe, öklid olarak tanımlanan en kısa mesafe ise iki nokta arasındaki kuş uçuşu mesafedir.

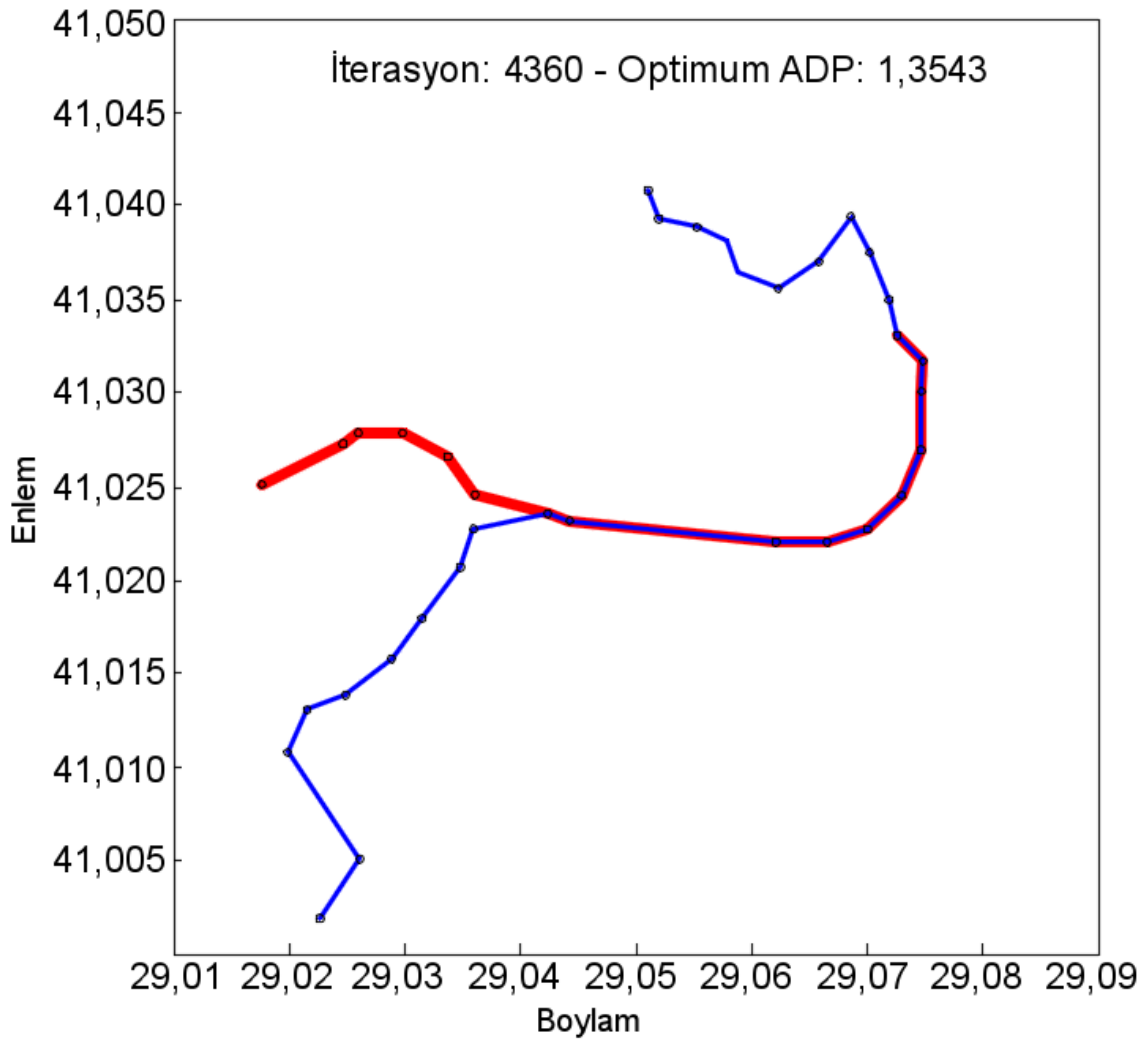
Çizelge 4.23'te 3 farklı *ADP* hesabı yöntemi bulunmaktadır. Bu alternatif fonksiyonların güzergah tasarımına olan etkisi Şekil 4.16'daki 184 numaralı güzergah kümesi üzerinde test edilmiştir.



Şekil 4.17. 1 numaralı ADP fonksiyonu

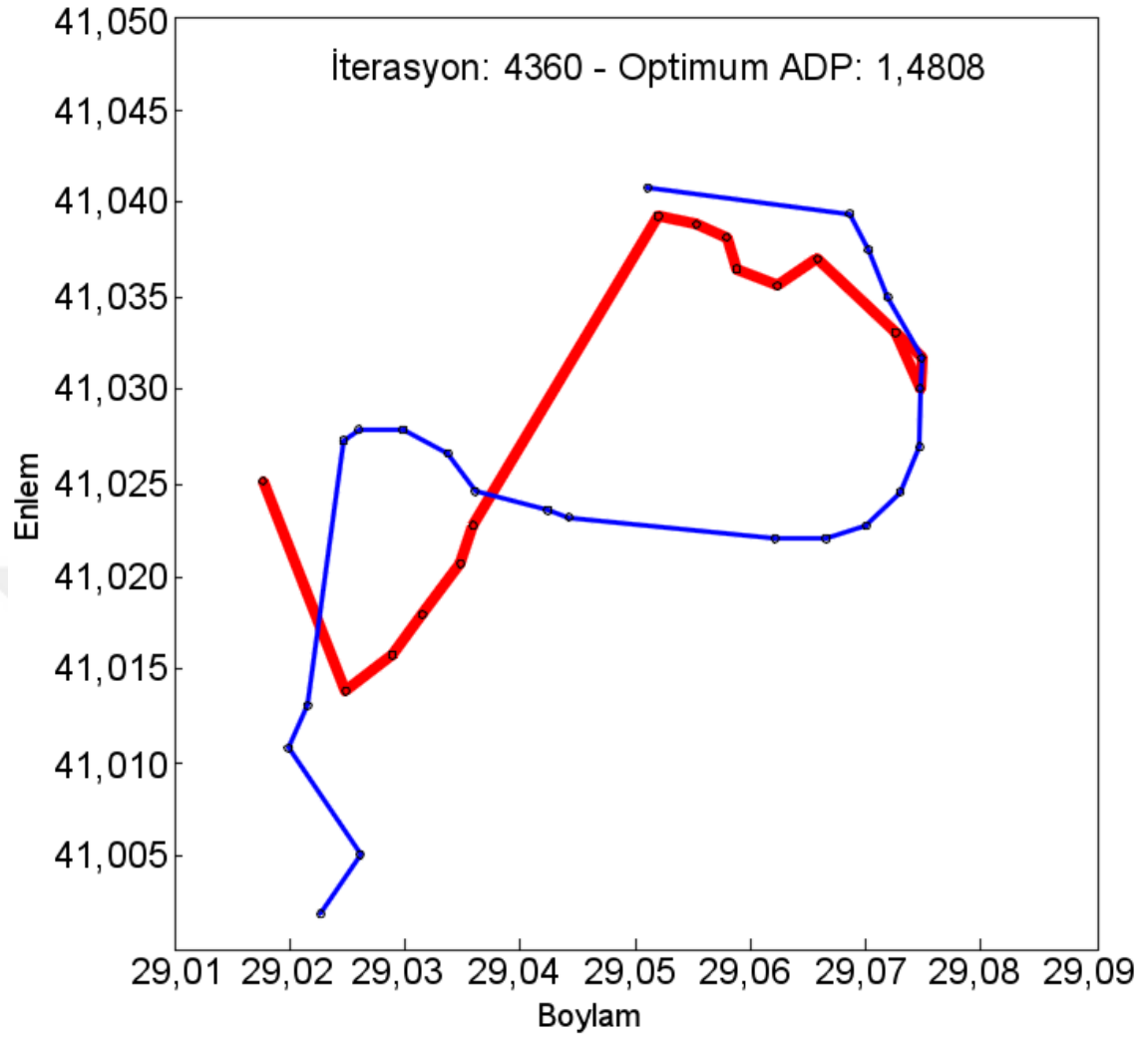
Şekil 4.17’de, yolculuk mesafesi matrisi için verilerin Google Maps’ten çekildiği 1 numaralı *ADP* fonksiyonuna göre yapılan çözümde 4360 iterasyon sonucunda $ADP = 1,0811$ çıkmıştır. Bu Şekil 4.16’daki mevcut güzergahlarla aynı sonuçtur. 250 küme için yapılan hesaplarda ise mevcut güzergah kümesi için $ADP = 1,268$ çıkmıştır. Aslında farklı gibi gözüken bu değerler birbirine denktir. Çünkü Google Maps’ten alınan mesafeler için yol ağı işin içine girmektedir. Yol ağındaki mesafeler, öklid mesafelerle aynı oranda değişmemektedir. Dolayısıyla aynı değerın çıkması beklenemez.

Şekil 4.18’de pay için yolculuk mesafesi matrisi Google Maps’ten çekilen verilerle oluşturulurken, payda için öklid mesafeler yani kuş uçuşu mesafeler kullanılmaktadır.



Şekil 4.18. 2 numaralı ADP fonksiyonu

Şekil 4.18’de 2 numaralı *ADP* fonksiyonuna göre yapılan çözümde 4360 iterasyon sonucunda $ADP = 1,3543$ çıkmıştır. Bu yine Şekil 4.16’daki mevcut güzergahlarla aynı sonuçtur. Fakat payda da öklid mesafeler kullanıldığı için *ADP* değeri farklı çıkmıştır. Ama bu değer bir önceki çözümdeki *ADP* değerine denktir.



Şekil 4.19. 3 numaralı ADP fonksiyonu

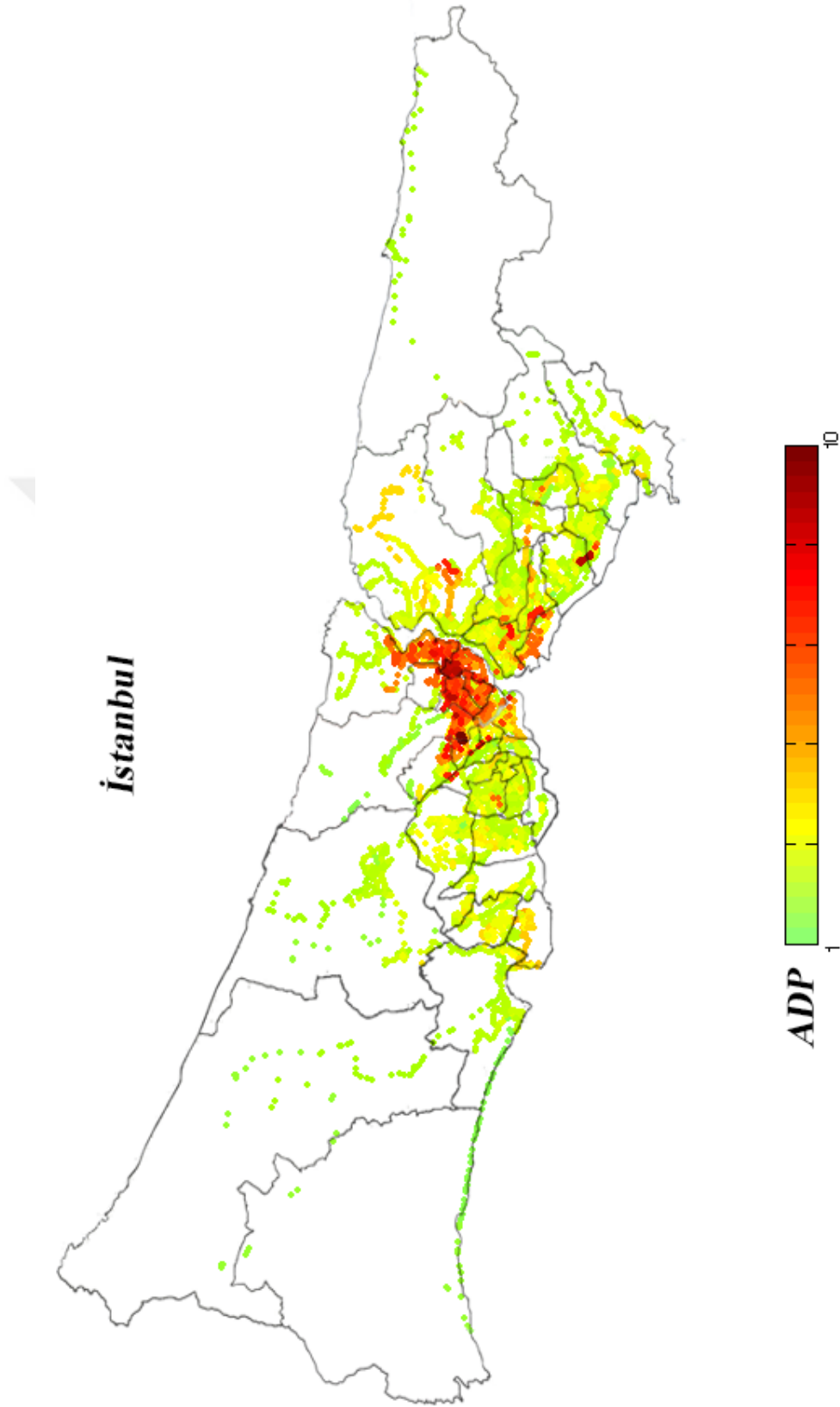
Şekil 4.19'da 3 numaralı *ADP* fonksiyonuna göre yapılan çözümde 4360 iterasyon sonucunda $ADP = 1,4808$ çıkmıştır. Bu değer, 250 küme için yapılan çözümle aynı sonucu vermiştir. Bu örnekte görüldüğü gibi payda kullanılan öklid mesafeler gerçek mesafeleri her zaman doğru yansıtamamaktadır. Bu da güzergah tasarımlarında çözümün yanlış yönlere sapmasına sebep olabilmektedir.

1 ve 2 numaralı *ADP* fonksiyonlarında payda Google Maps'ten alınan mesafelerin kullanılması çözümü doğru sonuçlara götürmeye yetmiştir. Payda da kullanılan verinin Google Maps'ten çekilmesinin ya da öklid mesafe olarak kullanılmasının ise bir önemi yoktur. 3 numaralı fonksiyonda payda kullanılan öklid mesafe doğru çözümü bulmada yeterli olmamıştır. Fakat 250 küme için 3 numaralı fonksiyonla yapılan çözümler bile mevcut güzergahların %74,8'inde daha verimli sonuç üretmeye yetmiştir. Dolayısıyla geliştirilen hibrit model ile yetkililer gerçek verileri temin ederek daha fazla kümede daha verimli sonuçlar üretebilir.

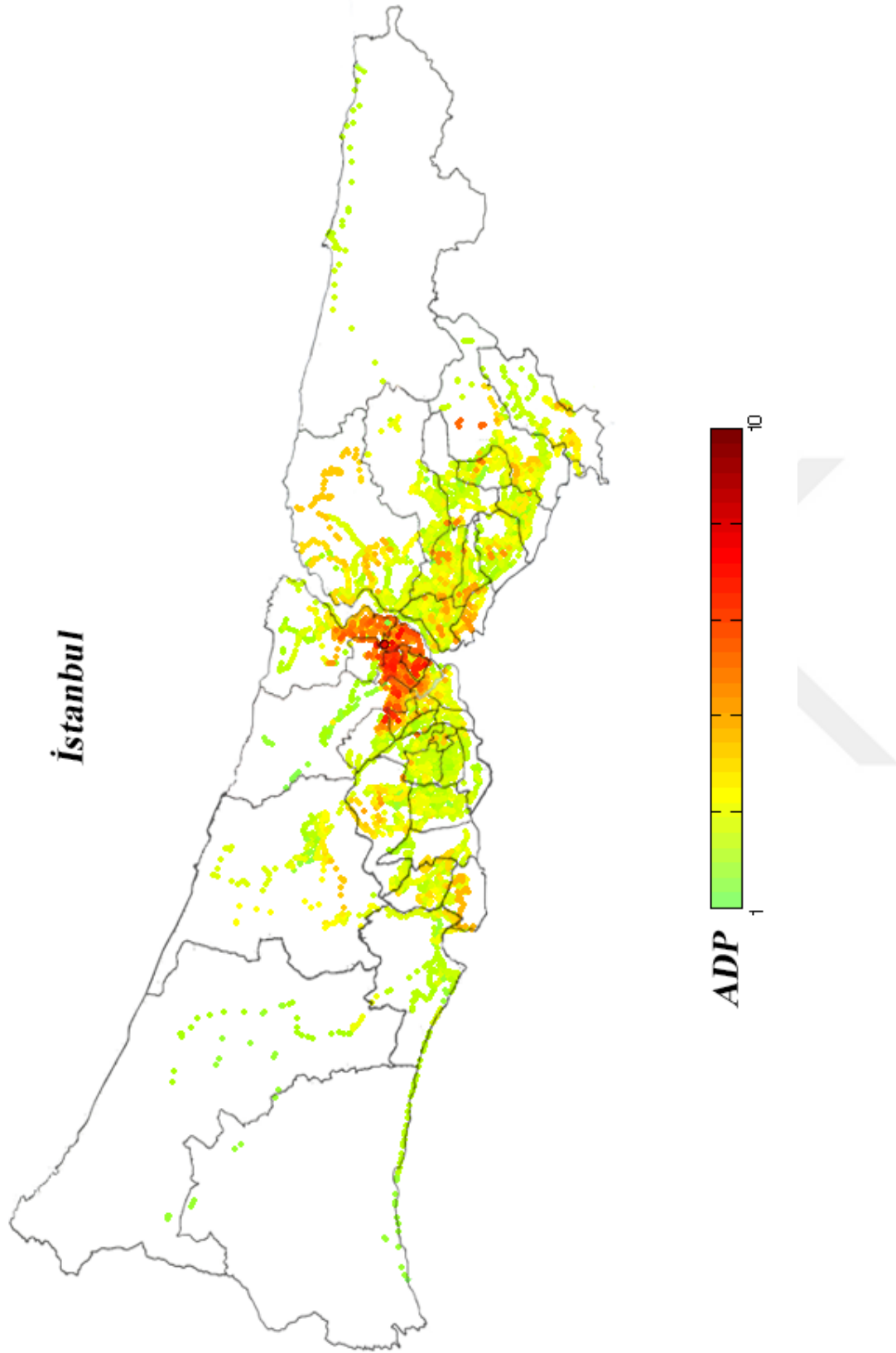
4.2.3. ADP haritaları

Geliştirilen yöntemin sonuçlarını bölgesel olarak inceleyebilmek için akış doğrultusu performansı haritaları oluşturulmuştur. Bu haritaların oluşturulabilmesi için otobüs hatlarına ait durakların koordinatları kullanılmıştır. Ağ üzerinde toplam 5602 durak bulunmaktadır. Çoğu duraktan birden fazla hat geçtiği için hesaplarda toplam 29933 durak kullanılmıştır. Çalışmada 732 hat için aynı zamanda 29933 durağa ait *ADP* değeri hesaplanmıştır. Bu durumda 5602 durak bulunan ağda bazı duraklar için birden fazla kez *ADP* değeri hesaplanmıştır. Dolayısıyla her durak için hesaplanan *ADP* değerlerinin ortalaması alınarak, durağa özgü *ADP* değerleri belirlenmiştir. 1 ile 10 arasında değişen bu değerler için yeşilden kırmızıya dönüşen bir skala oluşturulmuştur. İlk olarak otobüs güzergahlarının optimize edilmemiş mevcut halleri için duraklara ait hesaplanan *ADP* değerlerine göre bir harita oluşturulmuştur (Şekil 4.20). Şekilde de görüldüğü üzere mevcut durum için çevre bölgelerde sonuçlar oldukça iyidir. Merkezi bölgelere gelindiğinde ise sonuçlar aniden değişerek durum tersine dönmektedir.

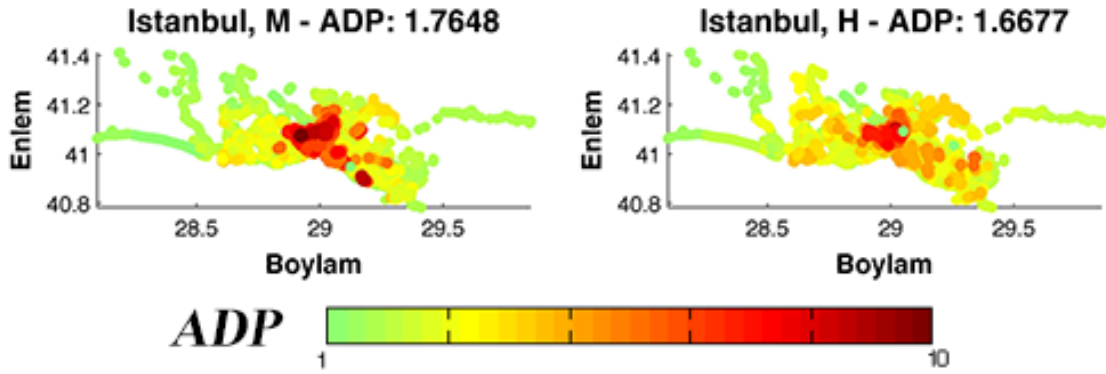
Şekil 4.21’de geliştirilen hibrit model sonucu hesaplanan *ADP* değerlerine göre oluşturulan harita görülmektedir. Bu haritada Şekil 4.20’ye göre çevre bölgelerdeki performansta biraz düşüş gözlemlenmektedir. Fakat merkez bölgedeki kötü sonuçların etkisi ise azalmıştır. Böylece farklı bölgeler arasında daha dengeli bir geçiş sağlanmıştır. Yani merkezi bölgenin yaşadığı aşırı olumsuz durum çevre bölgelere dağıtılarak azaltılmıştır. İstanbul şehri için hesaplanan bu 5602 durağa ait *ADP* değerlerinin ortalaması alınarak, sonuçlar mevcut durum ve hibrit model için kıyaslanmıştır (Şekil 4.22). Mevcut durum için $ADP = 1,7648$; hibrit model için ise $ADP = 1,6677$ çıkmıştır. Sonuç olarak durak bazında da %6’lık bir iyileşme gerçekleşmiştir.



Şekil 4.20. İstanbul şehrinin mevcut otobüs hatlarına göre ADP haritası



Şekil 4.21. İstanbul şehrinin hibrit optimizasyon sonucu otobüs hatlarına göre ADP haritası



Şekil 4.22. İstanbul için mevcut durum ile hibrit modelin karşılaştırılması

Şekil 4.22’de yapılan karşılaştırmayı daha detaylı inceleyebilmek adına tüm ilçeler bazında mevcut durum ile optimize edilen model sonuçlarının karşılaştırması grafik olarak EK-2’de verilmiştir. Ayrıca ilçeler için hesaplanan ortalama ADP değerleri için Çizelge 4.24 oluşturulmuştur.

Çizelge 4.24. İstanbul şehri ilçelerine ait ortalama mevcut ve hibrit model ADP değerleri

İlçe	M ADP	H ADP	İlçe	M ADP	H ADP
Üsküdar	1,6845	1,5827	Zeytinburnu	1,4133	1,4581
Kadıköy	2,4087	1,8352	Eyüp	2,2972	1,9629
Ataşehir	1,4912	1,4730	Bayrampaşa	1,3669	1,3767
Maltepe	1,4018	1,4341	Bahçelievler	1,3816	1,3648
Ümraniye	1,4871	1,5340	Bağcılar	1,4494	1,4157
Çekmeköy	1,4148	1,4931	Bakırköy	1,3523	1,4023
Sancaktepe	1,3573	1,4498	Güngören	1,4403	1,4438
Beykoz	1,6886	1,6492	Başakşehir	1,5038	1,6211
Pendik	1,3970	1,5765	Esenler	1,4356	1,3470
Sultanbeyli	1,4001	1,4502	Gaziosmanpaşa	2,0040	1,7914
Kartal	1,7151	1,4375	Sultangazi	2,0996	1,6549
Tuzla	1,4234	1,5702	Küçükçekmece	1,4585	1,4474
Sarıyer	2,5946	2,0846	Beylikdüzü	1,7849	1,7467
Beşiktaş	3,7028	2,7700	Avcılar	1,4291	1,5543
Beyoğlu	2,7391	2,3108	Büyükçekmece	1,2908	1,3470
Arnavutköy	1,3084	1,5190	Esenyurt	1,5470	1,5755
Şişli	2,8997	2,8295	Şile	1,2070	1,3187
Fatih	2,0012	1,6005	Silivri	1,0820	1,2057
Kağıthane	3,7920	3,1209	Çatalca	1,1559	1,2238

Optimizasyon yöntemi toplam verimi baz alarak hesap yaptığı için bazı ilçelerdeki ADP ortalamaları düşerken bazılarının ise artması normaldir. Çizelge 4.24’te İstanbul şehrine ait 38 ilçeden 18’inde hibrit model ile hesaplanmış ADP değerleri

mevcut duruma göre daha iyi çıkmıştır. Bu modelin daha kötü bir sonuç verdiği anlamına gelmemektedir. Çünkü ilçeler sahip olduğu durak ve bu duraklardan geçen hat sayılarına göre toplam ADP ortalamasına farklı oranlarda katkısı sağlamaktadır. İstanbul'un tamamına bakıldığında Şekil 4.22'de görüldüğü üzere hibrit model ile hesaplanan ortalama ADP değerinin mevcut ADP değerinden küçük çıkmış olması modelin daha verimli sonuçlar verdiğini göstermiştir.



5. SONUÇLAR

Trafik sorunlarına karşı toplu taşıma kullanımının teşvik edilmesi bu konudaki en etkili çözüm önerilerindendir. Fakat toplu taşıma araçlarına ait güzergahlar bazen başlı başına bu ulaşım modunun tercih edilmeme sebebi olabilmektedir. Dolayısıyla toplu taşıma araçlarına ait güzergahlarda, yolcular için maksimum faydayı sağlayacak şekilde düzenlemeler yapılması, özel araç kullanımına olan ilgiyi azaltabileceği gibi bu ulaşım türünden yolcu çekme potansiyeline de sahiptir. Bu çalışmada, yolculara sağlanacak faydayı ölçebilecek bir yöntem baz alınarak çoklu güzergah tasarımı yapan hibrit bir model geliştirilmiştir. Model problemde kullanılmayan durak kalmayacak şekilde belirlenen birden fazla başlangıç ve bitiş noktası arasında optimum güzergahı hesaplamaktadır. Dolayısıyla hem durak dizilimleri hem de durakların hangi güzergahta bulunacağı aynı anda optimize edilmektedir. Genetik algoritmayla beraber çalışan hibrit model ile genetik algoritmanın lokal optimuma yakalandığı noktalardan global optimum çözümlere geçiş sağlanmıştır. Yolcular açısından maksimum faydanın sağlanması, toplu taşıma araçları için minimum mesafe ya da süreyi veren güzergahın hesaplanmasıyla elde edilemez. Burada o hattı kullanan ve farklı durak çiftleri arasında yolculuk yapan yolcular bulunmaktadır. Dolayısıyla her yolcu için mümkün olduğunca gitmek istediği yönde istikamet eden güzergahlar tasarlanmalıdır. Bunun için akış doğrultusu yöntemi geliştirilmiştir. Bu yöntem güzergah tasarımı yapan modelin ana amacını oluşturmuştur. Bu amaç doğrultusunda genetik algoritma modifiye edilmiş ve iyileştirme adımları eklenerek hibrit model tamamlanmıştır.

Geliştirilen model ile önce küçük örnekler üzerinde denemeler yapılmıştır. Bu denemeler modele eklenen iyileştirme adımlarının örneklem üzerinde ne kadar değişiklik yapılacağı aşamasında faydalı olmuştur. Çünkü yapılan değişikliklerin seviyesi az olursa yeterince iyi çözümler üretilmemektedir. Bu seviye arttığında ise çözüm tıkanmaktadır. Daha sonra ise İstanbul kenti için otobüs hatlarına ait veriler kullanılarak otobüs hatları için hibrit model ile yeni güzergahlar tasarlanmıştır. Optimizasyon sonucu elde edilen güzergahlar ve güzergahların mevcut halleri akış doğrultusu yöntemine göre değerlendirilerek optimizasyon yönteminin sağladığı fayda tespit edilmiştir.

İstanbul şehrine ait 732 otobüs hattı çalışma kapsamında değerlendirilerek yeniden tasarlanmıştır. Bu optimizasyon modeli için çok büyük bir örneklem olduğu için hatlar birbirlerine olan yakınlıklarına göre kümelenerek 250 adet güzergah kümesi oluşturulmuştur. Daha sonra her küme için tek tek geliştirilen hibrit yöntemle çözüm üretilmiştir. Bu kümelerde yer alan her güzergah için hesaplanan *ADP* değerinin ortalaması alınarak 250 küme için ortalama *ADP* değeri hesaplandığında kümelerin %74,8'inde daha verimli sonuçların üretildiği görülmüştür. *ADP* değerlerine bakıldığında ise 250 küme için %19,7'lik iyileşme olduğu görülmüştür. Sadece daha iyi sonuç veren kümeler incelendiğinde ise bu oran %28,5'e çıkmaktadır. 732 hat için belirlenen mevcut ve hibrit model sonucu tüm güzergahların *ADP* ortalamasına bakıldığında ise %16,7 oranında daha verimli güzergahlar tasarlanmıştır. Bu orana sadece %74,8'lik kümede kalan güzergahlar için bakıldığında ise verim %22,9'a çıkmaktadır. Son olarak *ADP* değerleri durak bazında incelenmiştir. Bir duraktan geçen bütün hatlara ait *ADP* değerlerinin ortalaması alınarak duraklara ait ortalama *ADP* değerleri hesaplanmış ve burada da %6 daha iyi sonuçlar çıktığı görülmüştür. Ayrıca

durak bazında yapılan bu değerlendirme ile İstanbul şehri ve ilçeleri için bölgesel olarak otobüs hattı kullanıldığında maruz kalınacak verimin haritası oluşturulmuştur. Bu haritalar bir şehirde toplu taşıma konusunda çalışma yapılması planlanan bölgeler ile ilgili bir öncelik sıralaması yapılması konusunda fikir vermektedir. İlçelere ait mevcut *ADP* değerlerine bakıldığında Kadıköy, Sarıyer, Beşiktaş, Beyoğlu, Kağıthane, Eyüp, Sultangazi, Fatih, Şişli ve Gaziosmanpaşa ilçelerinde *ADP* ortalamasının 2'nin üstünde olduğu görülmektedir. Dolayısıyla öncelikli olarak bu bölgelerde yapılacak düzenlemeler faydalı olacaktır. Hibrit model ile tasarlanan güzergahlarda Kadıköy, Eyüp, Sultangazi, Fatih, Gaziosmanpaşa ilçelerinde bu değerler 2'nin altına düşürülmüştür. Bu çalışmada bir hat içerisinde yapılabilecek her yolculuk alternatifinde 1 yolculuk bulunduğu kabulü yapılmış ve kesit verileri yaklaşık olarak hesaplanmıştır. Dolayısıyla güzergahlarda yapılacak düzenlemelerin, yetkililerin kendi yolculuk matrisleri ve kesit verileriyle yeniden optimize edilerek planlanması gerekmektedir.

Bu çalışmada şehiriçi karayolu ağını kullanan toplu taşıma hatları, bu hatlarda yapılabilecek tüm yolculuk kombinasyonları açısından değerlendirilerek yeni güzergahlar tasarlanmıştır. Mevcut güzergahlara göre daha verimli sonuçlar üreten güzergahlar oluşturulmuştur. Böylece toplu ulaşım ile ilgili karar alacak yetkililerin daha verimli sonuçlara ulaşmasını sağlayacak bir yöntem oluşturulmuştur. Güzergahlarda yapılacak iyileştirmeler toplu taşımanın daha cazip hale getirilmesine katkıda bulunacaktır. Toplu ulaşımın cazip hale getirilerek bir mod kayması oluşturulması zincirleme olarak başka faydalar da sağlayacaktır. Yapılacak altyapı yatırımlarından önce ekstra maliyet gerektirmeyen güzergah düzenlemeleri gibi çözümler, ülke ekonomisine fayda sağlayacak tasarruflar oluşturacaktır. Geliştirilen optimizasyon yönteminin değerlendirme aşamasına, yolculuk anketleriyle elde edilecek kişisel davranış parametreleri eklenerek gelecekte farklı amaçlara hizmet eden güzergahların üretilmesi de planlanmaktadır.

6. KAYNAKLAR

- Akgol, K. ve Gunay, B. 2018. The Use of the Monte Carlo Method for the Rationality Calculations of Urban Road Networks. *Teknik Dergi*, 29(1): 8153-8165.
- Akin, A. ve Aydogdu, I. 2015. Optimum design of steel space frames by hybrid teaching-learning based optimization and harmony search algorithms. *World Academy of Science, Engineering and Technology Civil and Environmental Engineering*, 2(7): 739-745.
- Aydogdu, I., Carbas, S. ve Akin, A. 2017. Effect of Levy Flight on the discrete optimum design of steel skeletal structures using metaheuristics. *Steel and Composite Structures*, 24(1): 93-112.
- Aydoğdu, I. ve Saka, M. P. 2012. Ant colony optimization of irregular steel frames including elemental warping effect. *Advances in Engineering Software*, 44(1): 150-169.
- Ballou, R. H., Rahardja, H. ve Sakai, N. 2002. Selected country circuitry factors for road travel distance estimation. *Transportation Research Part A: Policy and Practice*, 36(9): 843-848.
- Beasley, D., Bull, D. R. ve Martin, R. R. 1993. An overview of genetic algorithms: Part 1, fundamentals. *University computing*, 15(2): 56-69.
- Beirão, G. ve Sarsfield Cabral, J. A. 2007. Understanding attitudes towards public transport and private car: A qualitative study. *Transport Policy*, 14(6): 478-489.
- Black, A. 1995. Urban mass transportation planning.
- Breheny, M. G. ve Hooper, A. J. 1985. The role of rationality in urban and regional planning. *Rationality in Planning*, Pion, London, pp. 1-14
- Buchari, E. ve Ndiaye, A. B. 2008. Reformed Regulation: A Solution of Problems of Implementing Public Transport Routes, Hong Kong Soc Transportation Studies Ltd, Hong Kong.
- Canci, M. ve Onden, I. 2013. Determining the success criteria for sustainable bus system creating. TRANSIST 6. Transportation Symposium and Exhibition, 25-26 December 2013, Istanbul, Turkey.
- Cascetta, E., Carteni, A., Pagliara, F. ve Montanino, M. 2015. A new look at planning and designing transportation systems: A decision-making model based on cognitive rationality, stakeholder engagement and quantitative methods. *Transport Policy*, 38: 27-39.
- Cervero, R. 2004. Transit-oriented development in the United States: Experiences, challenges, and prospects, Transportation Research Board, 102.
- Chopde, N. R. ve Nichat, M. 2013. Landmark Based Shortest Path Detection by Using A* and Haversine Formula. *International Journal of Innovative Research in Computer and Communication Engineering*, 1: 298-302.
- Chowdhury, S., Ceder, A. ve Sachdeva, R. 2014. The effects of planned and unplanned transfers on public transport users' perception of transfer routes. *Transportation Planning and Technology*, 37(2): 154-168.

- Chowdhury, S., Ceder, A. ve Velly, B. 2014. Measuring Public-Transport Network Connectivity Using Google Transit with Comparison across Cities. *Journal of Public Transportation*, 17(4): 76-92.
- Curtis, C. ve Scheurer, J. 2010. Planning for sustainable accessibility: Developing tools to aid discussion and decision-making. *Progress in Planning*, 74(2): 53-106.
- Davison, L. J. ve Knowles, R. D. 2006. Bus quality partnerships, modal shift and traffic decongestion. *Journal of Transport Geography*, 14(3): 177-194.
- Degertekin, S. 2012. Optimum design of geometrically non-linear steel frames using artificial bee colony algorithm. *Steel and Composite Structures*, 12(6): 505-522.
- Değertekin, S. Ö., Ülker, M. ve Hayalioğlu, M. S. 2006. Uzay Çelik Çerçevelerin Tabu Arama Ve Genetik Algoritma Yöntemleriyle Optimum Tasarımı. *Teknik Dergi*, 17(83).
- Derrible, S. ve Kennedy, C. 2011. Applications of Graph Theory and Network Science to Transit Network Design. *Transport Reviews*, 31(4): 495-519.
- Diesing, P. 1962. Reason in Society: Five Types of Decisions and Their Social Conditions, University of Illinois Press, Urbana.
- Dill, J. 2004. Measuring network connectivity for bicycling and walking. 83rd Annual Meeting of the Transportation Research Board, Washington, DC.
- Dorigo, M., Maniezzo, V. ve Coloni, A. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1): 29-41.
- Düzakın, E. ve Demircioğlu, M. 2009. Araç Rotalama Problemleri ve Çözüm Yöntemleri. *Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 13(1): 68-87.
- Garcia-Sierra, M., Van Den Bergh, J. C. J. M. ve Miralles-Guasch, C. 2015. Behavioural economics, travel behaviour and environmental-transport policy. *Transportation Research Part D-Transport and Environment*, 41: 288-305.
- Geem, Z. W., Kim, J. H. ve Loganathan, G. V. 2001. A new heuristic optimization algorithm: harmony search. *simulation*, 76(2): 60-68.
- Goldberg, D. E. ve Holland, J. H. 1988. Genetic algorithms and machine learning. *Machine learning*, 3(2): 95-99.
- Gunay, B., Akgol, K., Andreasson, I. ve Terzi, S. 2016. Estimation of modal shift potential for a new form of dial-a-ride service. *Journal of Public Transportation*, 19(2): 75-88.
- Gunay, B., Akgol, K. ve Raveau, S. 2018. Rationality of route layouts as a quality indicator of urban road networks. *Urban Design International*, 23(2): 132-144.
- Hadas, Y. 2013. Assessing public transport systems connectivity based on Google Transit data. *Journal of Transport Geography*, 33: 105-116.
- Hadas, Y. ve Ranjitkar, P. 2012. Modeling public-transit connectivity with spatial quality-of-transfer measurements. *Journal of Transport Geography*, 22: 137-147.

- Hartigan, J. A. 1975. Clustering algorithms.
- Hartigan, J. A. ve Wong, M. A. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): 100-108.
- Hasançebi, O., Erdal, F. ve Saka, M. P. 2009. Adaptive harmony search method for structural optimization. *Journal of Structural Engineering*, 136(4): 419-431.
- Holland, J. H. 1992. Genetic algorithms. *Scientific american*, 267(1): 66-73.
- Huang, J. ve Levinson, D. M. 2015. Circuitry in urban transit networks. *Journal of Transport Geography*, 48: 145-153.
- Jalili, S., Hosseinzadeh, Y. ve Kaveh, A. 2014. Chaotic biogeography algorithm for size and shape optimization of truss structures with frequency constraints. *Periodica Polytechnica Civil Engineering*, 58(4): 397-422.
- Karaboga, D. ve Basturk, B. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3): 459-471.
- Kaveh, A. 2014. Advances in metaheuristic algorithms for optimal design of structures, Springer.
- Krol, A. 2017. Application of the Genetic Algorithm for Optimization of the Public Transportation Lines. In G. Sierpinski (Ed.), *Intelligent Transport Systems and Travel Behaviour*, Springer Int Publishing Ag, Cham, 505, pp. 135-146
- Lagrange, J. L. 2013. Analytical mechanics, Springer Science & Business Media, 191.
- Langmyhr, T. 2001. The rationality of transport investment packages. *Transportation*, 28(2): 157-178.
- Leong, K. H., Abdul-Rahman, H., Wang, C., Onn, C. C. ve Loo, S. C. 2016. Bee Inspired Novel Optimization Algorithm and Mathematical Model for Effective and Efficient Route Planning in Railway System. *Plos One*, 11(12): 24.
- Levinson, D. 2012. Network structure and city size. *Plos One*, 7(1): e29721.
- Levinson, D. ve El-Geneidy, A. 2009. The minimum circuitry frontier and the journey to work. *Regional science and urban economics*, 39(6): 732-738.
- Levinson, D. M. 1998. Accessibility and the journey to work. *Journal of Transport Geography*, 6(1): 11-21.
- Lin, J. ve Ban, Y. 2013. Complex Network Topology of Transportation Systems. *Transport Reviews*, 33(6): 658-685.
- Love, R. F. ve Morris, J. G. 1979. Mathematical models of road travel distances. *Management Science*, 25(2): 130-139.
- Matisziw, T. C. ve Demir, E. 2016. Measuring Spatial Correspondence among Network Paths. *Geographical Analysis*, 48(1): 3-17.
- Murray, A. T., Davis, R., Stimson, R. J. ve Ferreira, L. 1998. Public transportation access. *Transportation Research Part D: Transport and Environment*, 3(5): 319-328.

- Newell, G. F. 1980. Traffic flow on transportation networks.
- Niblett, T. J. ve Church, R. L. 2016. The Shortest Covering Path Problem: A New Perspective and Model. *International Regional Science Review*, 39(1): 131-151.
- Nikolopoulou, A. I., Repoussis, P. P., Tarantilis, C. D. ve Zachariadis, E. E. 2017. Moving products between location pairs: Cross-docking versus direct-shipping. *European Journal of Operational Research*, 256(3): 803-819.
- O'sullivan, S. ve Morrall, J. 1996. Walking distances to and from light-rail transit stations. *Transportation research record: journal of the transportation research board*(1538): 19-26.
- Osman, I. H. ve Kelly, J. P. 1996. Meta-heuristics: an overview. Meta-heuristics, Springer, pp. 1-21
- Owen, A. ve Levinson, D. M. 2015. Modeling the commute mode share of transit using continuous accessibility to jobs. *Transportation Research Part A: Policy and Practice*, 74: 110-122.
- Prager, W. 1968. Optimality criteria in structural design. *Proceedings of the National Academy of Sciences*, 61(3): 794-796.
- Raveau, S., Guo, Z., Muñoz, J. C. ve Wilson, N. H. M. 2014. A behavioural comparison of route choice on metro networks: Time, transfers, crowding, topology and socio-demographics. *Transportation Research Part A: Policy and Practice*, 66: 185-195.
- Robusto, C. 1957. The cosine-haversine formula. *American Mathematical Monthly*: 38-40.
- Rojo, M., Dell'olio, L., Gonzalo-Orden, H. ve Ibeas, Á. 2015. Inclusion of quality criteria in public bus service contracts in metropolitan areas. *Transport Policy*, 42: 52-63.
- Sager, T. 1994. Communicative Planning Theory, Avebury, Aldershot.
- Shi, Y. 2011. Brain Storm Optimization Algorithm, Berlin, Heidelberg.
- Shi, Y. J., Blainey, S. ve Hounsell, N. 2017. Using GIS to assess the potential for centralised planning of bus networks. *Transportation Planning and Technology*, 40(1): 119-142.
- Simon, D. 2008. Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6): 702-713.
- Sun, D., Chen, S. K., Zhang, C. ve Shen, S. W. 2016. A bus route evaluation model based on GIS and super-efficient data envelopment analysis. *Transportation Planning and Technology*, 39(4): 407-423.
- Taylor, B. D. 1991. Unjust Equity: An Examination of California's Transportation Development Act.
- Telgarsky, M. ve Vattani, A. 2010. Hartigan's Method: k-means Clustering without Voronoi. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.

- Topping, B. 1983. Shape optimization of skeletal structures: a review. *Journal of Structural Engineering*, 109(8): 1933-1951.
- Vitale, A., Festa, D. C., Guido, G. ve Rogano, D. 2014. A Decision Support System based on smartphone probes as a tool to promote public transport. In J. F. Desousa, J. P. Desousa, A. Costa, T. Farias ve S. Melo (Ed.), *Transportation: Can We Do More with Less Resources? - 16th Meeting of the Euro Working Group on Transportation - Porto 2013*, Elsevier Science Bv, Amsterdam, 111, pp. 224-231
- Vragović, I., Louis, E. ve Díaz-Guilera, A. 2005. Efficiency of informational transfer in regular and complex networks. *Physical Review E*, 71(3): 036122.
- Vreeswijk, J. D., Landman, R. L., Van Berkum, E. C., Hegyi, A., Hoogendoorn, S. P. ve Van Arem, B. 2015. Improving the road network performance with dynamic route guidance by considering the indifference band of road users. *Iet Intelligent Transport Systems*, 9(10): 897-906.
- Walker, J. 2012. *Human transit: How clearer thinking about public transit can enrich our communities and our lives*, Island Press.
- Wolf, J., Schönfelder, S., Samaga, U., Oliveira, M. ve Axhausen, K. 2004. Eighty weeks of global positioning system traces: approaches to enriching trip information. *Transportation research record: journal of the transportation research board*(1870): 46-54.
- Yan, X.-Y. ve Wang, M.-S. 2009. Topological Properties of Public Transport Networks: A Temporal Perspective. ICCTP 2009, pp. 1-6
- Yang, X.-S. 2009. *Firefly Algorithms for Multimodal Optimization*, Berlin, Heidelberg.
- Yildirim, R. O. 2013. Measurement model of service quality. TRANSİST 6. Transportation Symposium and Exhibition, 25-26 December 2013, Istanbul, Turkey.

7. EKLER

EK-1. Güzergah Kümelerine Ait Hesap Sonuçları

Çizelge 7.1. 1-36 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
1	3	168	5340	1,290	1,295	8,918	3,871	3,957	26,754
2	2	149	5190	1,281	1,280	5,196	2,562	2,599	10,392
3	1	23	4180	1,524	1,524	1,524	1,524	1,524	1,524
4	4	126	5005	1,411	1,291	3,220	5,643	5,164	12,882
5	1	79	4630	1,201	1,194	1,194	1,201	1,221	1,194
6	1	88	4700	1,965	2,157	2,626	1,965	2,353	2,626
7	5	269	6150	1,195	1,189	12,516	5,975	5,946	62,580
8	3	173	5380	1,369	1,354	6,818	4,107	4,061	20,454
9	1	29	4230	1,128	1,128	1,128	1,128	1,128	1,128
10	6	256	6045	1,345	1,325	10,587	8,071	8,396	63,523
11	3	80	4640	1,374	1,287	3,097	4,123	3,860	9,291
12	3	119	4950	1,497	1,572	5,163	4,492	4,716	15,490
13	1	32	4255	2,081	2,076	2,021	2,081	2,076	2,021
14	3	95	4760	1,246	1,248	1,716	3,739	3,743	5,148
15	6	251	6005	1,484	1,449	10,862	8,906	8,693	65,173
16	4	153	5220	1,181	1,183	3,794	4,722	4,827	15,175
17	3	143	5140	1,378	1,364	5,590	4,134	4,653	16,769
18	5	301	6405	1,322	1,463	15,550	6,608	7,400	77,752
19	2	55	4440	3,303	2,254	2,736	6,606	4,508	5,472
20	3	69	4550	2,035	1,522	2,035	6,105	4,567	6,104
21	3	190	5520	3,605	2,421	9,814	10,814	7,264	29,442
22	1	52	4415	1,211	1,211	1,211	1,211	1,211	1,211
23	7	254	6030	1,489	1,461	8,715	10,425	10,226	61,008
24	2	70	4560	1,488	1,487	1,895	2,977	3,203	3,791
25	1	72	4575	1,310	1,355	1,355	1,310	1,376	1,355
26	1	15	4120	1,174	1,174	1,174	1,174	1,174	1,174
27	2	104	4830	1,194	1,196	1,962	2,388	2,443	3,924
28	1	43	4340	1,283	1,257	1,257	1,283	1,257	1,257
29	1	65	4520	1,118	1,169	1,169	1,118	1,169	1,169
30	1	36	4285	1,278	1,256	1,256	1,278	1,256	1,256
31	4	142	5135	1,288	1,341	5,891	5,153	5,366	23,565
32	1	42	4335	1,714	1,468	1,514	1,714	1,468	1,514
33	1	17	4135	2,264	2,018	2,018	2,264	2,018	2,018
34	3	165	5320	1,191	1,353	8,056	3,574	4,060	24,167
35	3	116	4925	1,469	1,340	3,698	4,408	4,019	11,093
36	4	156	5245	3,626	2,632	5,661	14,506	10,528	22,644

Çizelge 7.2. 37-76 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
37	2	37	4295	2,269	1,856	2,030	4,537	3,712	4,061
38	4	153	5220	1,347	1,415	3,481	5,389	6,167	13,923
39	3	173	5380	1,245	1,171	8,854	3,735	3,514	26,563
40	3	122	4975	1,667	1,596	4,485	5,001	4,788	13,456
41	3	79	4630	2,037	1,490	1,819	6,112	4,470	5,456
42	4	168	5340	3,412	2,296	7,883	13,649	9,185	31,531
43	3	130	5040	1,391	1,364	3,339	4,172	4,553	10,017
44	4	167	5335	1,221	1,241	6,889	4,885	4,964	27,557
45	2	95	4760	4,899	3,103	4,727	9,798	6,207	9,454
46	3	115	4920	1,468	1,348	2,455	4,404	4,044	7,364
47	2	57	4455	1,249	1,252	1,364	2,498	2,540	2,727
48	3	92	4735	1,284	1,282	2,127	3,852	3,845	6,381
49	3	137	5095	1,788	1,760	7,067	5,363	5,279	21,201
50	5	179	5430	1,380	1,466	5,296	6,902	7,544	26,481
51	4	112	4895	1,955	1,573	2,443	7,819	6,292	9,773
52	4	122	4975	1,353	1,332	3,927	5,413	5,327	15,708
53	2	142	5135	3,643	2,159	5,580	7,287	4,317	11,160
54	4	123	4980	2,097	1,702	3,657	8,389	6,808	14,626
55	4	246	5965	1,309	1,258	15,902	5,235	5,587	63,608
56	5	268	6140	1,304	1,379	18,456	6,522	6,894	92,279
57	4	157	5255	2,059	1,370	6,002	8,235	5,480	24,006
58	2	72	4575	1,238	1,236	1,988	2,477	2,473	3,975
59	2	70	4560	1,266	1,187	1,412	2,532	2,374	2,825
60	3	88	4700	3,062	2,254	3,047	9,187	6,762	9,141
61	1	58	4460	1,333	1,333	1,333	1,333	1,333	1,333
62	3	82	4655	1,195	1,258	1,935	3,586	4,367	5,805
63	3	155	5240	1,295	1,366	5,248	3,886	4,335	15,743
64	3	75	4600	1,121	1,112	1,353	3,362	3,336	4,058
65	6	297	6375	1,371	1,430	17,086	8,226	8,656	102,517
66	5	212	5695	1,431	1,350	8,868	7,155	6,752	44,342
67	5	300	6400	1,116	1,142	15,676	5,578	5,728	78,378
68	2	74	4590	3,476	2,429	2,912	6,953	4,857	5,824
69	3	132	5055	1,320	1,293	4,353	3,960	3,880	13,059
70	2	57	4455	1,465	1,263	1,574	2,931	2,526	3,149
71	1	52	4415	1,425	1,399	1,406	1,425	1,399	1,406
72	4	140	5120	1,158	1,154	5,590	4,633	4,616	22,361
73	1	23	4180	1,692	1,663	1,663	1,692	1,663	1,663
74	2	25	4200	1,044	1,044	1,065	2,088	2,088	2,129
75	5	241	5925	1,450	1,416	14,994	7,252	7,081	74,970
76	5	329	6630	1,569	1,668	22,374	7,845	8,340	111,870

Çizelge 7.3. 77-116 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
77	2	92	4735	1,775	1,689	2,760	3,549	3,379	5,521
78	7	260	6080	4,048	1,998	8,009	28,338	13,989	56,060
79	3	128	5020	1,529	1,450	5,729	4,587	4,351	17,187
80	3	82	4655	1,407	1,416	1,956	4,222	4,646	5,869
81	3	114	4910	1,392	1,381	2,361	4,175	4,142	7,083
82	1	48	4380	1,533	1,533	1,533	1,533	1,533	1,533
83	3	185	5480	1,391	1,327	18,424	4,174	3,981	55,271
84	5	150	5200	1,690	1,651	6,051	8,448	8,254	30,254
85	4	86	4685	1,301	1,298	1,982	5,204	5,190	7,928
86	3	165	5320	1,296	1,301	8,159	3,889	4,107	24,476
87	5	153	5220	2,176	1,651	4,061	10,878	8,255	20,304
88	5	191	5525	1,312	1,382	7,047	6,558	6,912	35,235
89	3	181	5445	2,624	1,802	5,276	7,872	5,405	15,829
90	5	235	5880	1,463	1,577	16,336	7,317	7,884	81,679
91	5	182	5455	2,151	1,646	7,121	10,754	8,230	35,607
92	3	92	4735	3,019	2,334	3,186	9,057	7,001	9,557
93	4	86	4685	1,224	1,195	2,080	4,898	4,779	8,319
94	3	136	5085	1,564	1,485	8,954	4,693	4,879	26,863
95	3	159	5270	1,481	1,511	9,032	4,443	4,534	27,095
96	3	221	5765	2,534	1,796	16,696	7,602	5,389	50,088
97	1	25	4200	1,115	1,115	1,115	1,115	1,115	1,115
98	1	84	4670	1,364	1,437	1,380	1,364	1,437	1,380
99	1	32	4255	1,321	1,321	1,321	1,321	1,321	1,321
100	4	78	4620	1,213	1,188	1,620	4,852	4,752	6,480
101	3	102	4815	1,445	1,262	2,342	4,336	3,787	7,027
102	5	169	5350	1,260	1,313	5,303	6,299	7,301	26,514
103	1	65	4520	1,261	1,261	1,261	1,261	1,261	1,261
104	4	205	5640	1,670	1,485	14,492	6,681	5,940	57,970
105	1	77	4615	1,277	1,376	1,402	1,277	1,376	1,402
106	1	27	4215	1,491	1,491	1,491	1,491	1,491	1,491
107	3	148	5180	1,220	1,221	5,967	3,660	3,662	17,902
108	5	178	5420	1,352	1,360	7,178	6,759	7,248	35,892
109	1	24	4190	1,274	1,269	1,269	1,274	1,269	1,269
110	4	264	6110	4,288	2,635	13,090	17,150	10,540	52,358
111	3	94	4750	1,569	1,522	2,208	4,707	4,565	6,624
112	3	148	5180	1,476	1,482	6,879	4,428	4,447	20,638
113	1	18	4140	1,854	1,854	1,854	1,854	1,854	1,854
114	3	140	5120	1,466	1,510	7,197	4,399	4,531	21,592
115	3	205	5640	1,270	1,444	8,161	3,811	4,332	24,482
116	2	66	4525	1,407	1,370	1,773	2,813	2,740	3,545

Çizelge 7.4. 117-156 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
117	5	150	5200	4,512	2,150	3,862	22,558	10,749	19,310
118	3	187	5495	1,372	1,373	14,692	4,116	5,188	44,077
119	2	81	4645	1,185	1,170	2,119	2,370	2,400	4,238
120	5	151	5205	1,636	1,402	3,277	8,180	7,010	16,387
121	5	223	5780	3,872	2,203	10,257	19,360	11,013	51,284
122	7	426	7405	1,158	1,225	20,994	8,106	8,574	146,959
123	2	95	4760	2,365	2,079	2,835	4,730	4,158	5,670
124	2	55	4440	1,315	1,309	1,657	2,631	2,617	3,315
125	5	215	5720	1,491	1,369	13,160	7,455	6,847	65,799
126	3	93	4740	1,340	1,288	2,120	4,021	4,149	6,360
127	1	23	4180	2,463	2,137	2,462	2,463	2,137	2,462
128	3	162	5295	1,174	1,177	7,891	3,523	3,530	23,674
129	7	313	6500	1,368	1,344	13,796	9,576	9,408	96,570
130	1	16	4125	1,615	1,603	1,603	1,615	1,603	1,603
131	3	111	4885	1,666	1,633	2,882	4,998	4,899	8,645
132	2	67	4535	3,367	2,315	3,085	6,735	4,630	6,169
133	1	59	4470	5,049	2,637	3,613	5,049	2,637	3,613
134	1	73	4580	1,297	1,297	1,297	1,297	1,297	1,297
135	2	72	4575	1,262	1,248	1,600	2,524	2,495	3,200
136	4	139	5110	3,368	2,239	3,094	13,470	8,956	12,376
137	1	38	4300	3,090	2,672	2,761	3,090	2,672	2,761
138	3	65	4520	1,838	1,843	2,334	5,514	5,642	7,002
139	3	156	5245	1,517	1,424	6,720	4,550	4,271	20,161
140	2	64	4510	1,703	1,584	1,781	3,406	3,168	3,562
141	2	126	5005	1,223	1,216	4,724	2,446	2,523	9,447
142	3	52	4415	1,891	1,442	1,853	5,672	4,325	5,558
143	1	19	4150	1,119	1,119	1,119	1,119	1,119	1,119
144	3	115	4920	1,165	1,159	3,113	3,496	3,476	9,339
145	1	44	4350	1,476	1,472	1,472	1,476	1,472	1,472
146	4	158	5260	1,476	1,368	7,173	5,904	5,473	28,690
147	3	98	4780	2,116	1,808	2,963	6,348	5,423	8,890
148	3	69	4550	2,610	1,745	2,510	7,830	5,235	7,530
149	3	83	4660	1,397	1,445	2,106	4,191	4,336	6,318
150	2	68	4540	1,359	1,306	2,089	2,718	2,611	4,178
151	10	475	7800	3,704	1,954	14,918	37,039	19,536	149,181
152	4	97	4775	1,443	1,202	1,648	5,773	4,809	6,594
153	3	104	4830	1,636	1,631	2,484	4,907	4,893	7,453
154	2	52	4415	2,352	1,620	1,929	4,705	3,240	3,859
155	3	210	5680	1,358	1,425	23,498	4,073	4,499	70,494
156	2	82	4655	1,327	1,312	2,361	2,653	2,623	4,723

Çizelge 7.5. 157-196 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
157	3	157	5255	1,407	1,543	7,944	4,220	4,628	23,833
158	4	195	5560	1,398	1,370	9,335	5,593	5,479	37,339
159	3	108	4860	1,271	1,252	5,171	3,814	3,757	15,513
160	6	208	5660	1,482	1,369	10,383	8,892	8,212	62,300
161	7	369	6950	5,696	2,587	14,014	39,875	18,106	98,095
162	3	152	5215	1,455	1,487	11,180	4,365	4,679	33,539
163	2	108	4860	1,162	1,170	1,724	2,325	2,339	3,448
164	1	6	4045	1,159	1,159	1,159	1,159	1,159	1,159
165	3	105	4840	1,253	1,239	5,492	3,758	3,769	16,476
166	1	3	4020	1,116	1,116	1,116	1,116	1,116	1,116
167	1	20	4160	1,456	1,327	1,331	1,456	1,327	1,331
168	2	104	4830	2,308	1,767	5,277	4,617	3,534	10,554
169	4	166	5325	2,681	2,100	7,651	10,725	8,400	30,602
170	2	111	4885	4,855	2,433	3,325	9,711	4,867	6,650
171	3	123	4980	1,422	1,286	6,731	4,266	3,858	20,194
172	1	70	4560	1,487	1,506	1,506	1,487	1,506	1,506
173	5	253	6020	1,293	1,529	20,632	6,466	8,015	103,160
174	3	69	4550	1,324	1,332	1,750	3,973	3,997	5,251
175	2	74	4590	1,588	1,586	2,101	3,177	3,172	4,203
176	5	244	5950	1,292	1,263	11,299	6,460	6,316	56,497
177	1	58	4460	1,300	1,329	1,300	1,300	1,329	1,300
178	4	85	4680	1,281	1,275	1,789	5,123	5,167	7,157
179	1	50	4400	1,870	1,718	1,861	1,870	1,718	1,861
180	5	216	5725	1,208	1,196	7,421	6,038	5,981	37,106
181	2	48	4380	1,370	1,351	2,133	2,739	2,702	4,266
182	5	226	5805	1,467	1,447	11,788	7,336	7,236	58,940
183	2	51	4405	3,382	2,422	3,029	6,763	4,844	6,058
184	2	47	4375	1,268	1,481	1,763	2,536	2,962	3,525
185	1	54	4430	1,355	1,355	1,355	1,355	1,355	1,355
186	3	85	4680	1,436	1,392	2,062	4,308	4,175	6,187
187	3	165	5320	4,545	2,175	7,099	13,635	6,526	21,297
188	3	219	5750	1,228	1,253	17,165	3,685	4,512	51,496
189	1	33	4260	1,379	1,362	1,362	1,379	1,362	1,362
190	1	42	4335	1,094	1,094	1,094	1,094	1,094	1,094
191	3	90	4720	1,280	1,243	3,268	3,839	3,729	9,804
192	1	29	4230	1,355	1,355	1,355	1,355	1,355	1,355
193	1	23	4180	1,387	1,288	1,288	1,387	1,288	1,288
194	3	133	5060	1,556	1,516	6,408	4,668	4,547	19,223
195	3	190	5520	1,274	1,263	7,291	3,823	3,789	21,872
196	3	199	5590	4,740	2,733	8,196	14,221	8,198	24,587

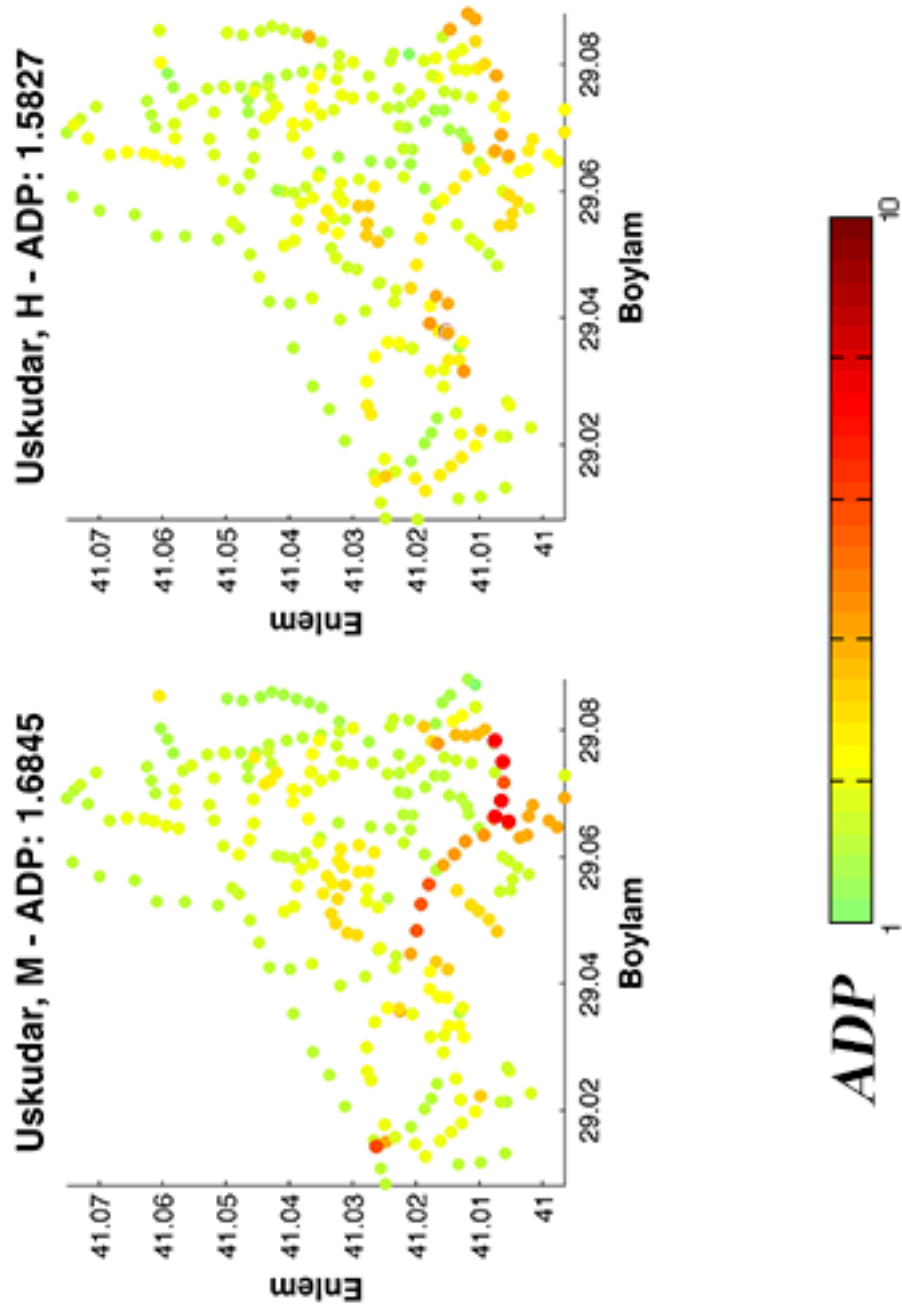
Çizelge 7.6. 197-236 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
197	3	105	4840	1,148	1,126	2,153	3,445	3,377	6,460
198	3	220	5760	1,309	1,364	17,569	3,926	4,092	52,707
199	3	140	5120	1,404	1,316	4,228	4,213	3,947	12,685
200	4	170	5360	1,386	1,237	6,558	5,544	4,947	26,232
201	3	91	4725	1,287	1,292	2,013	3,860	3,987	6,039
202	2	41	4325	1,167	1,165	1,587	2,334	2,346	3,174
203	4	194	5550	1,788	1,804	8,484	7,151	7,217	33,935
204	1	13	4100	1,275	1,275	1,275	1,275	1,275	1,275
205	1	32	4255	1,505	1,505	1,508	1,505	1,508	1,508
206	1	22	4175	1,103	1,103	1,103	1,103	1,103	1,103
207	1	39	4310	1,446	1,446	1,446	1,446	1,446	1,446
208	4	139	5110	1,208	1,215	7,333	4,834	4,880	29,331
209	3	117	4935	1,336	1,284	6,443	4,008	3,852	19,328
210	3	72	4575	1,821	1,628	1,917	5,464	4,883	5,751
211	1	51	4405	1,096	1,096	1,099	1,096	1,096	1,099
212	3	79	4630	1,209	1,197	1,548	3,627	3,592	4,643
213	3	131	5045	2,734	1,536	5,689	8,201	4,609	17,066
214	3	66	4525	1,190	1,172	1,554	3,571	3,515	4,661
215	3	177	5415	1,267	1,261	7,216	3,800	3,784	21,648
216	5	173	5380	4,213	2,412	4,559	21,064	12,058	22,795
217	6	234	5870	1,181	1,308	6,417	7,088	7,847	38,501
218	3	105	4840	1,471	1,437	3,193	4,413	4,310	9,579
219	1	80	4640	1,366	1,364	1,378	1,366	1,469	1,378
220	2	48	4380	2,480	1,666	2,440	4,959	3,331	4,879
221	2	85	4680	1,184	1,202	1,675	2,367	2,404	3,350
222	1	34	4270	1,297	1,297	1,297	1,297	1,297	1,297
223	3	120	4960	1,330	1,297	2,324	3,991	3,890	6,971
224	2	51	4405	1,249	1,231	1,547	2,499	2,461	3,094
225	4	133	5060	1,232	1,222	5,409	4,927	4,889	21,637
226	5	164	5310	1,281	1,312	6,233	6,406	6,558	31,163
227	1	18	4140	4,303	2,088	3,541	4,303	2,088	3,541
228	2	83	4660	1,494	1,525	1,714	2,989	3,093	3,427
229	3	110	4880	1,158	1,266	2,436	3,475	3,797	7,308
230	1	36	4285	4,828	3,016	3,804	4,828	3,016	3,804
231	2	43	4340	1,076	1,077	1,237	2,153	2,153	2,474
232	5	227	5815	5,479	2,363	8,041	27,393	11,816	40,204
233	3	168	5340	1,318	1,380	8,949	3,953	4,340	26,848
234	3	89	4710	1,385	1,267	2,070	4,155	3,800	6,211
235	5	174	5390	2,278	1,618	6,429	11,391	8,089	32,146
236	3	72	4575	1,210	1,137	1,328	3,630	3,410	3,984

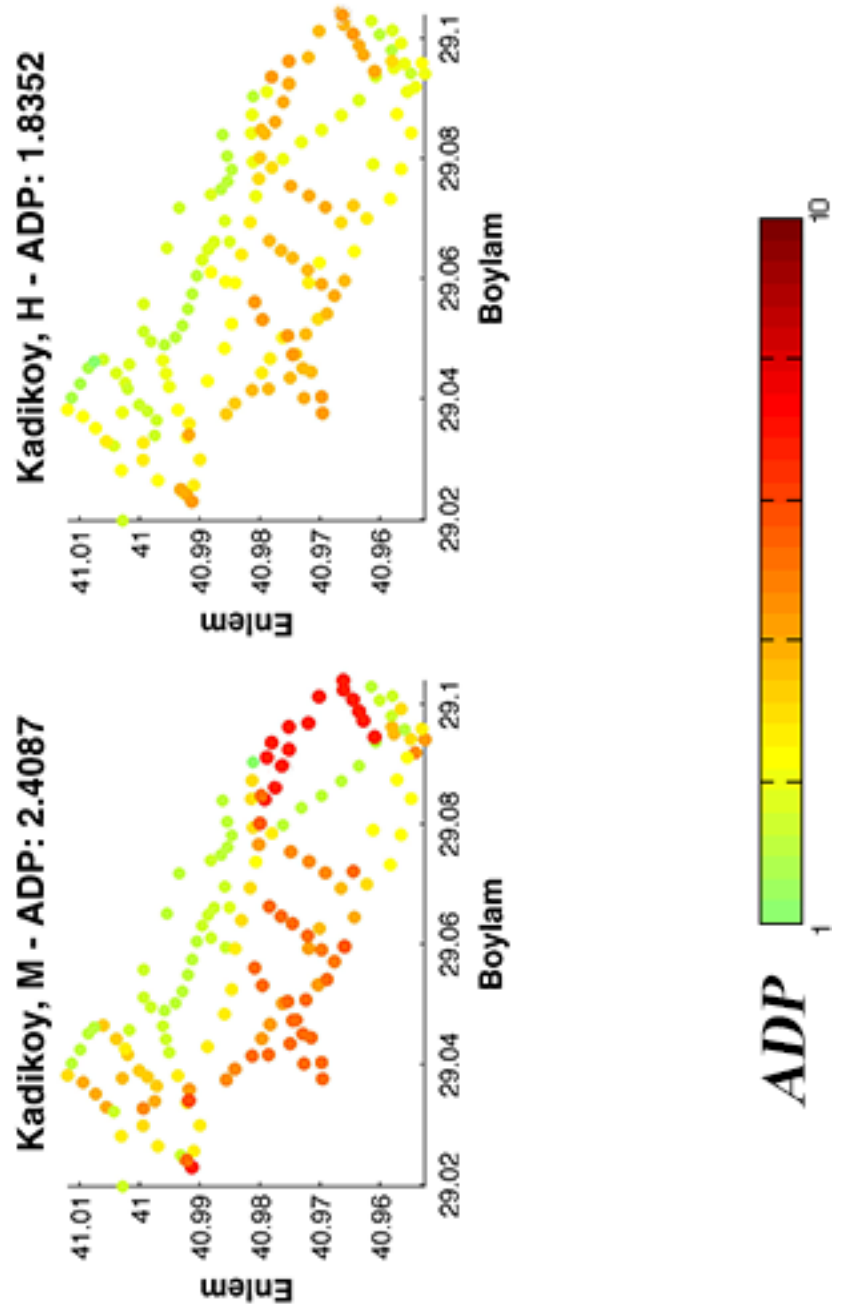
Çizelge 7.7. 237-250 numaralı güzergah kümelerine ait hesap sonuçları

No	GS	DS	İS	M ADP	H ADP	GA ADP	KM ADP	KH ADP	KGA ADP
237	2	73	4580	1,277	1,230	2,367	2,554	2,460	4,733
238	7	294	6350	4,436	2,461	10,684	31,054	17,227	74,786
239	3	99	4790	1,389	1,336	1,959	4,167	4,009	5,878
240	2	53	4420	1,187	1,180	1,291	2,373	2,360	2,582
241	3	139	5110	1,376	1,406	7,467	4,129	4,218	22,401
242	3	105	4840	1,329	1,272	2,173	3,987	3,815	6,519
243	4	63	4500	1,219	1,215	1,442	4,877	4,915	5,768
244	3	139	5110	1,252	1,718	4,126	3,756	5,208	12,379
245	4	116	4925	1,161	1,131	2,192	4,645	4,524	8,768
246	3	151	5205	1,484	1,417	6,212	4,452	4,917	18,636
247	1	53	4420	1,217	1,217	1,217	1,217	1,217	1,217
248	3	115	4920	1,447	1,286	2,496	4,340	3,857	7,487
249	1	55	4440	1,343	1,326	1,326	1,343	1,326	1,326
250	3	99	4790	3,158	1,441	2,802	9,473	4,323	8,406

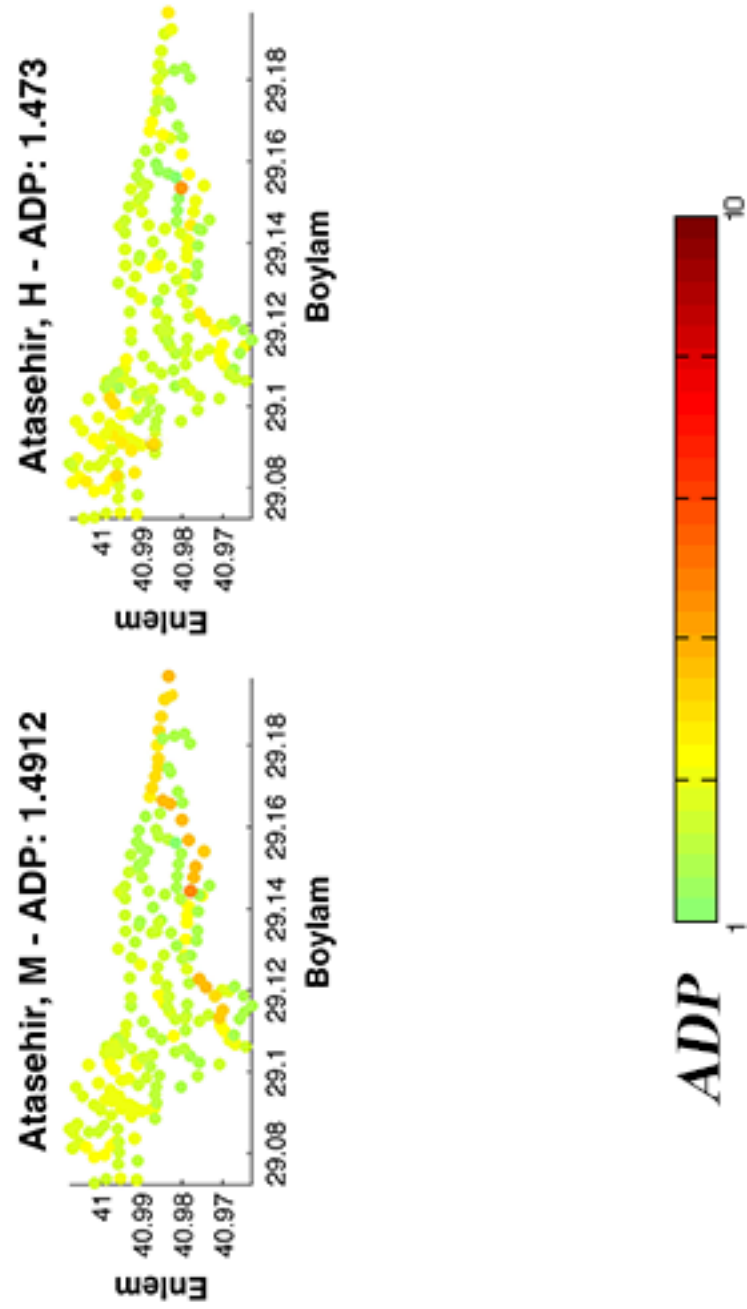
EK-2. İstanbul İlçelerine Ait ADP Sonuçları



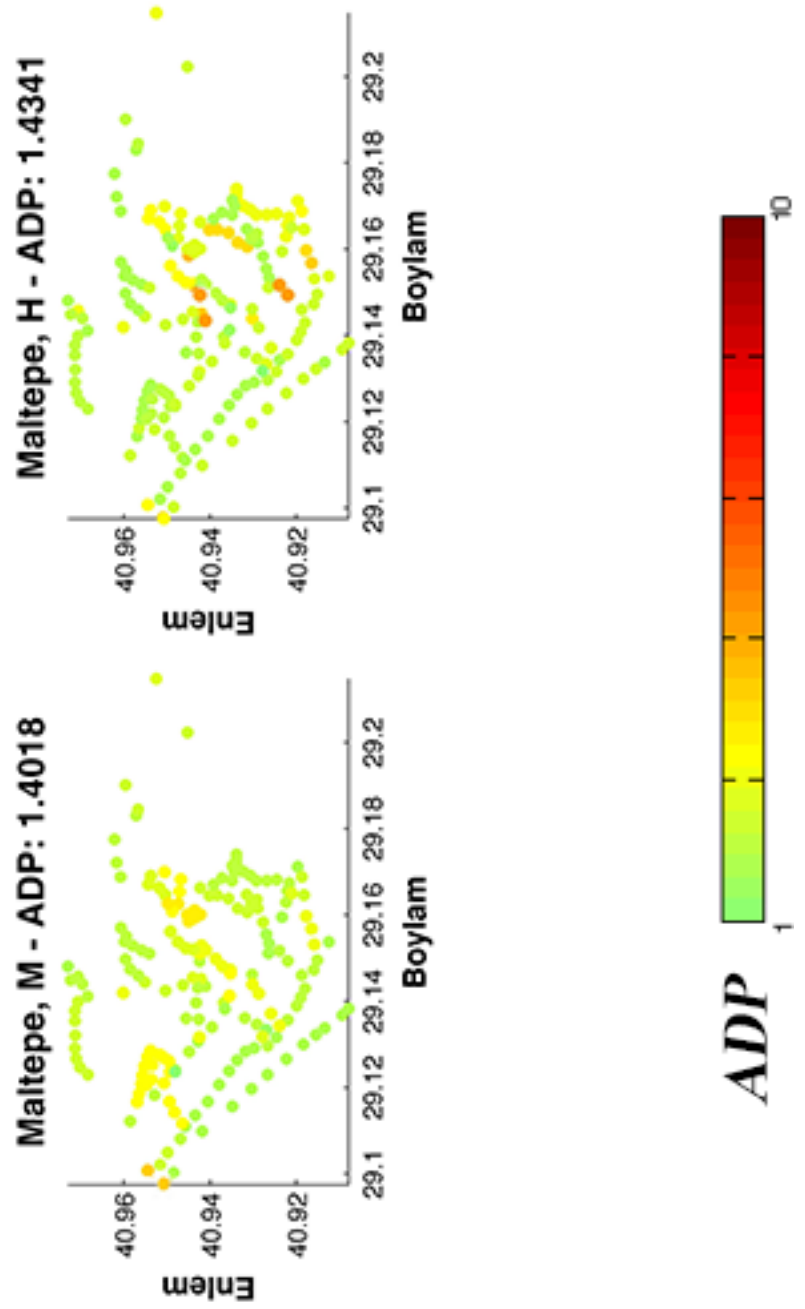
Şekil 7.1. Üsküdar ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



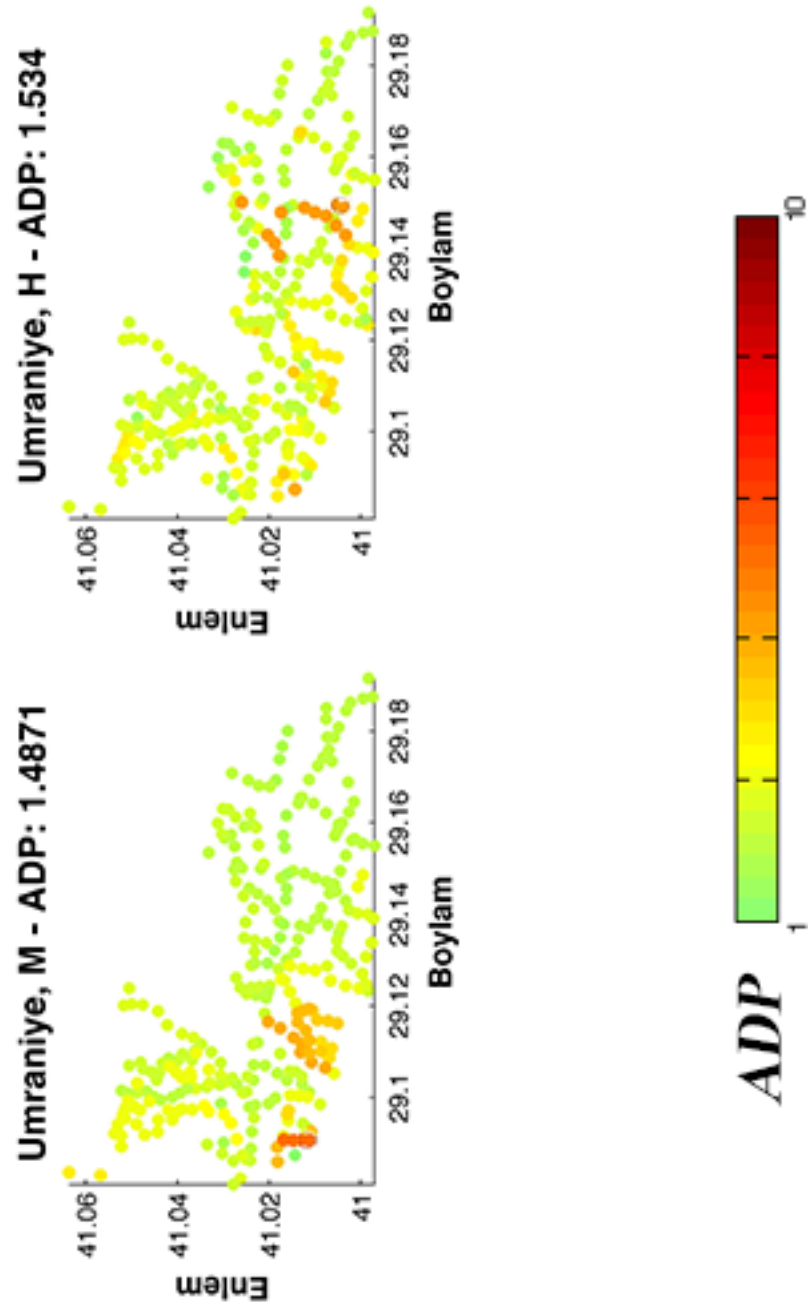
Şekil 7.2. Kadıköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



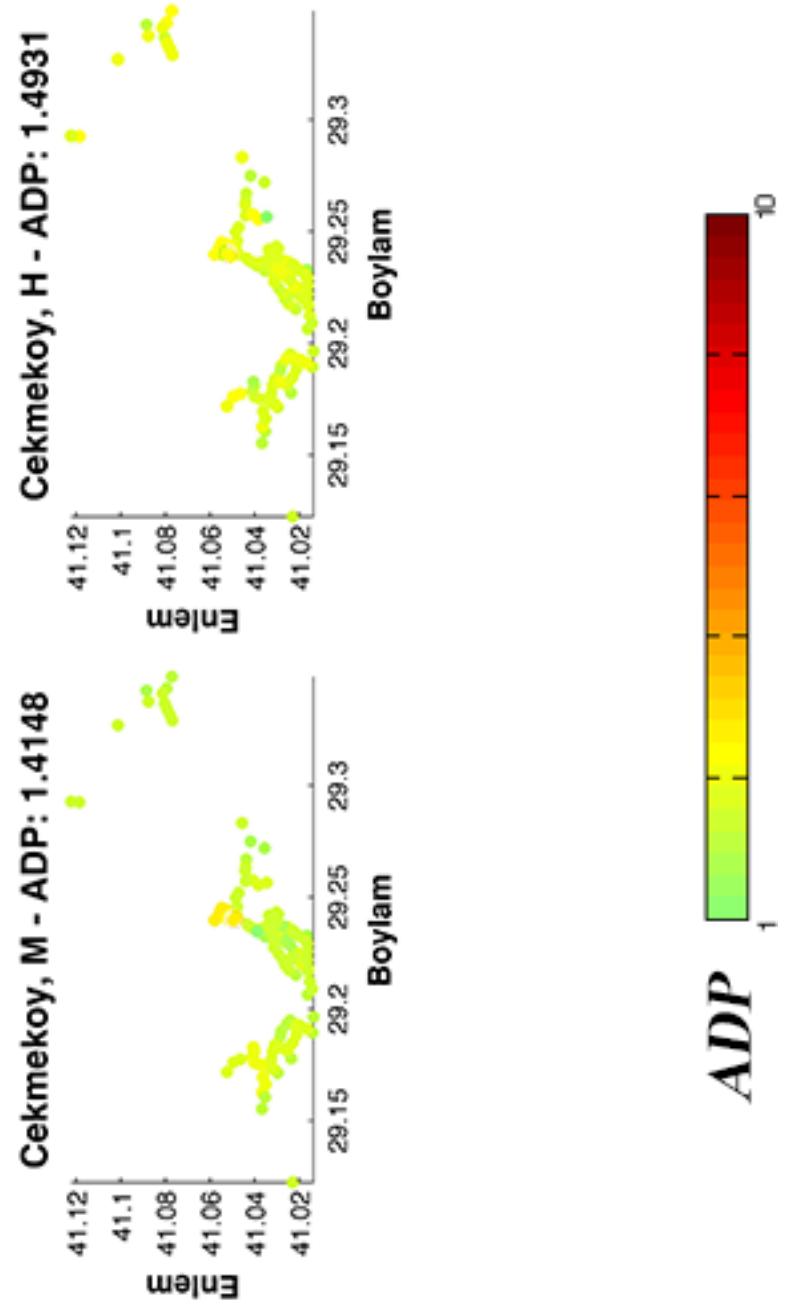
Şekil 7.3. Ataşehir ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



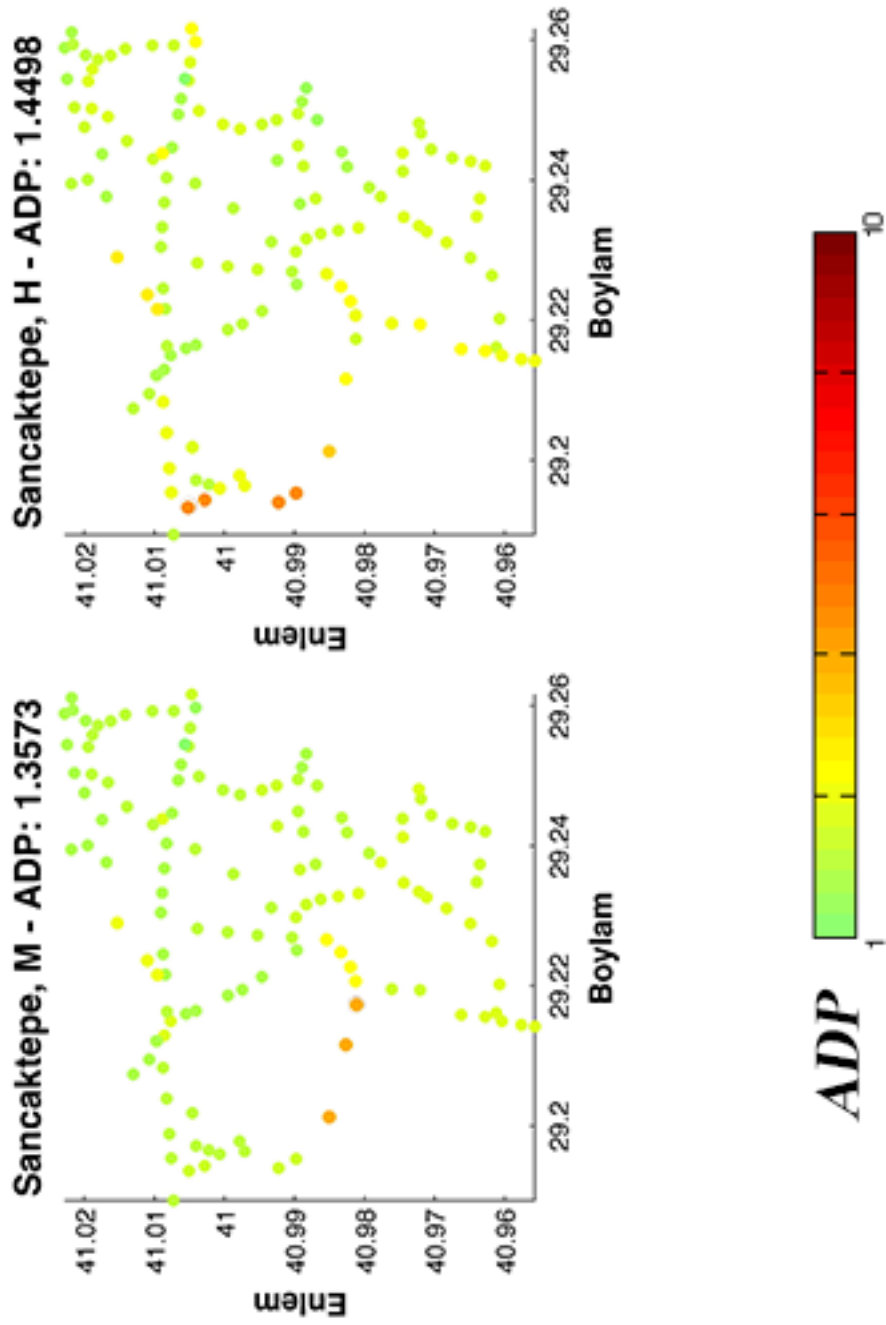
Şekil 7.4. Maltepe ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



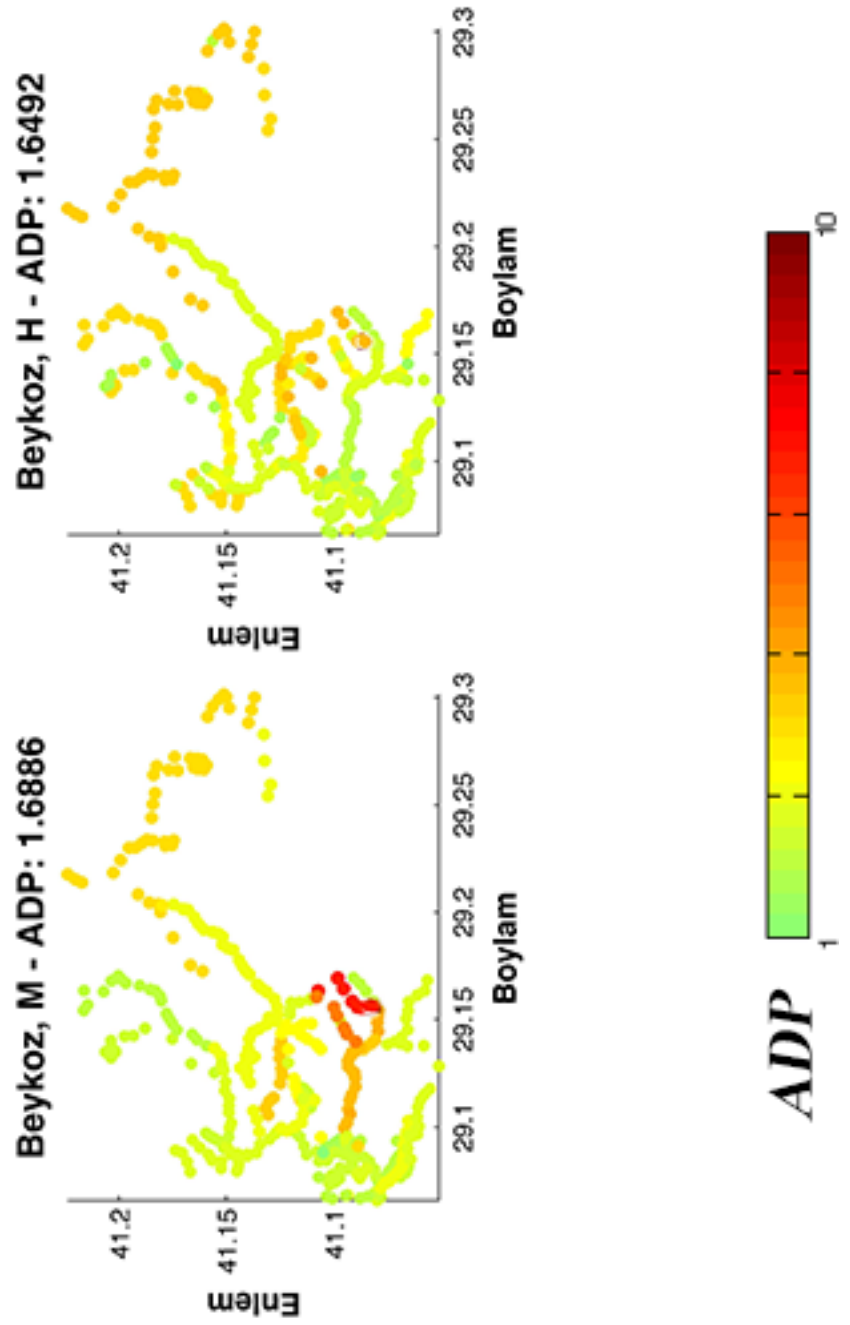
Şekil 7.5. Ümraniye ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



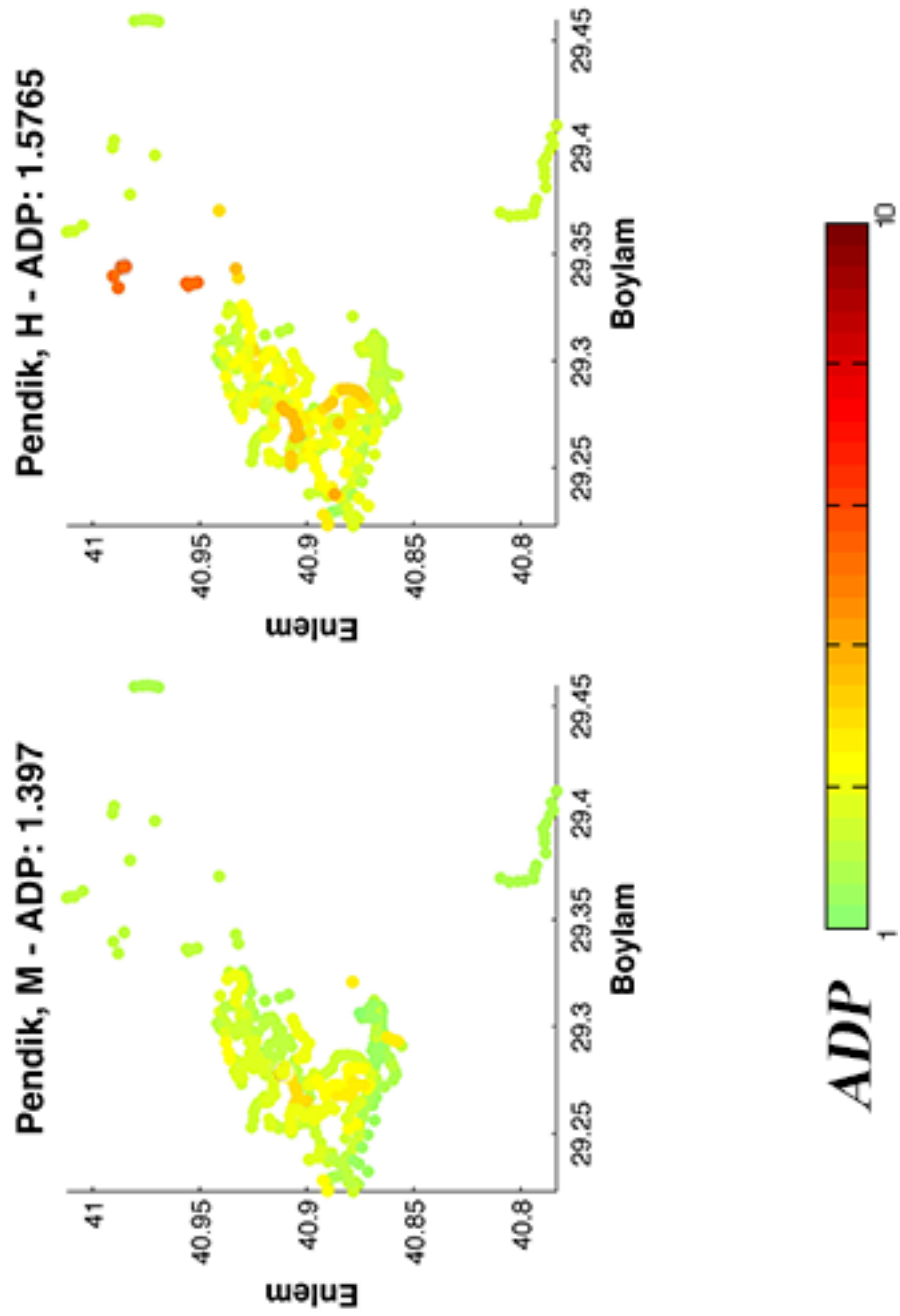
Şekil 7.6. Çekmeköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



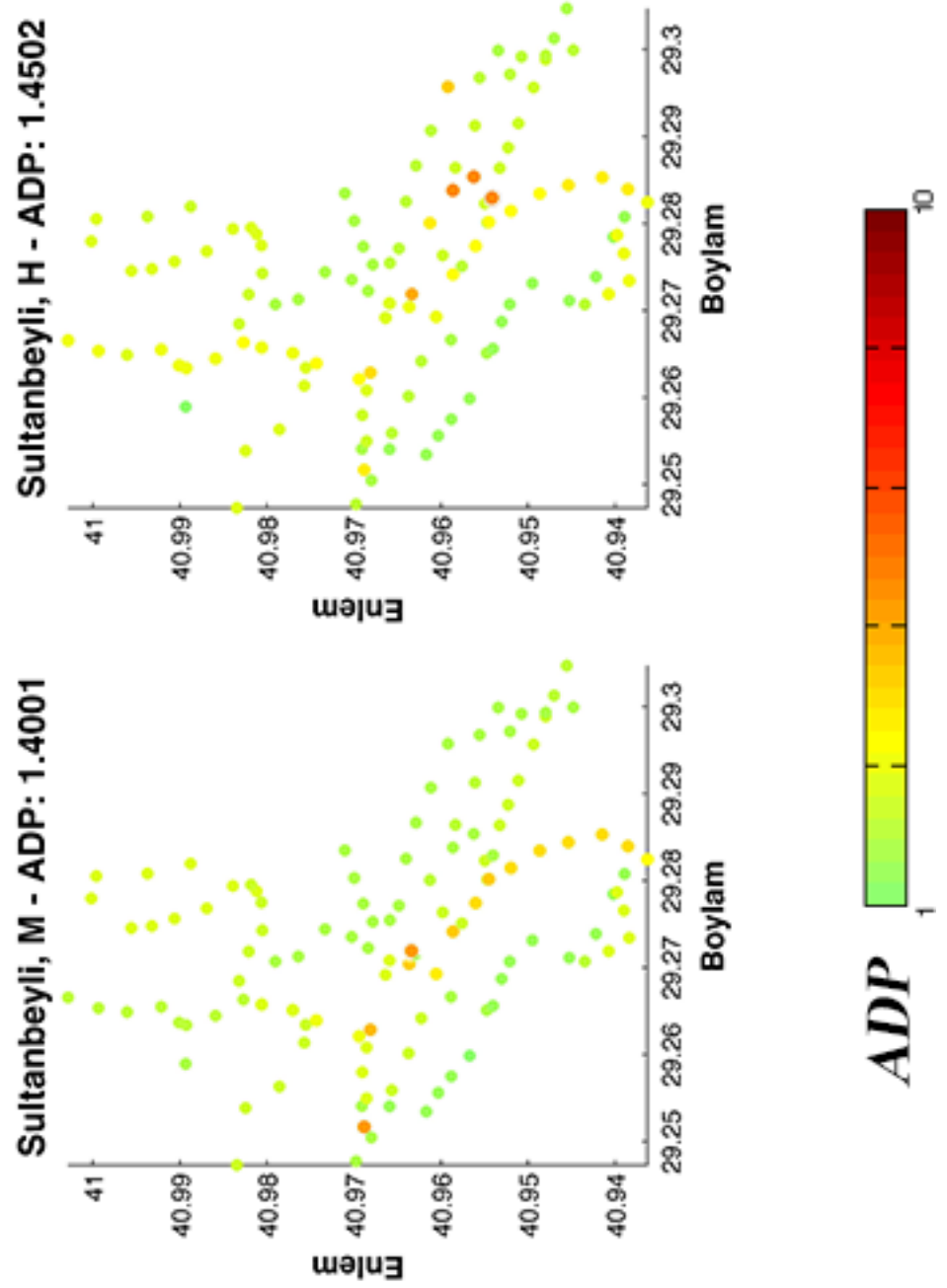
Şekil 7.7. Sancaktepe ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



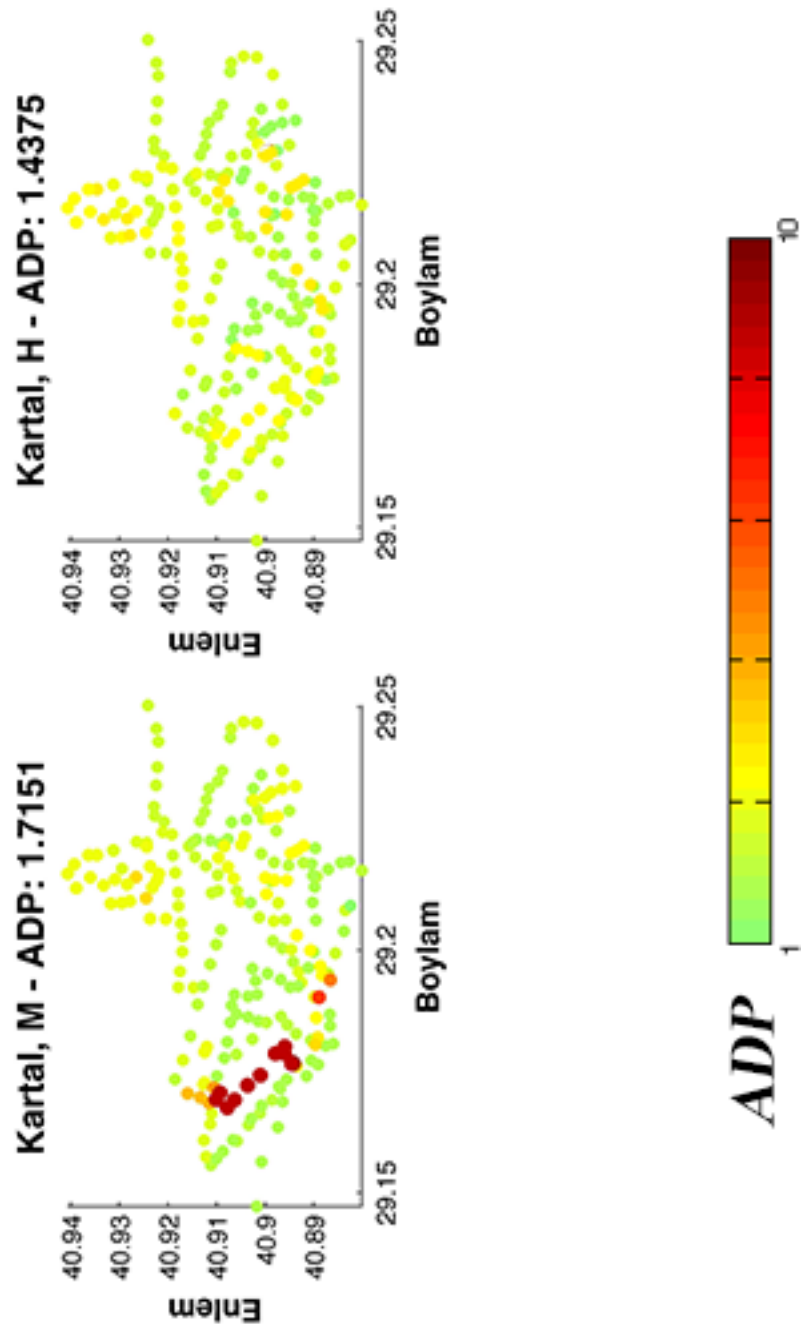
Şekil 7.8. Beykoz ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



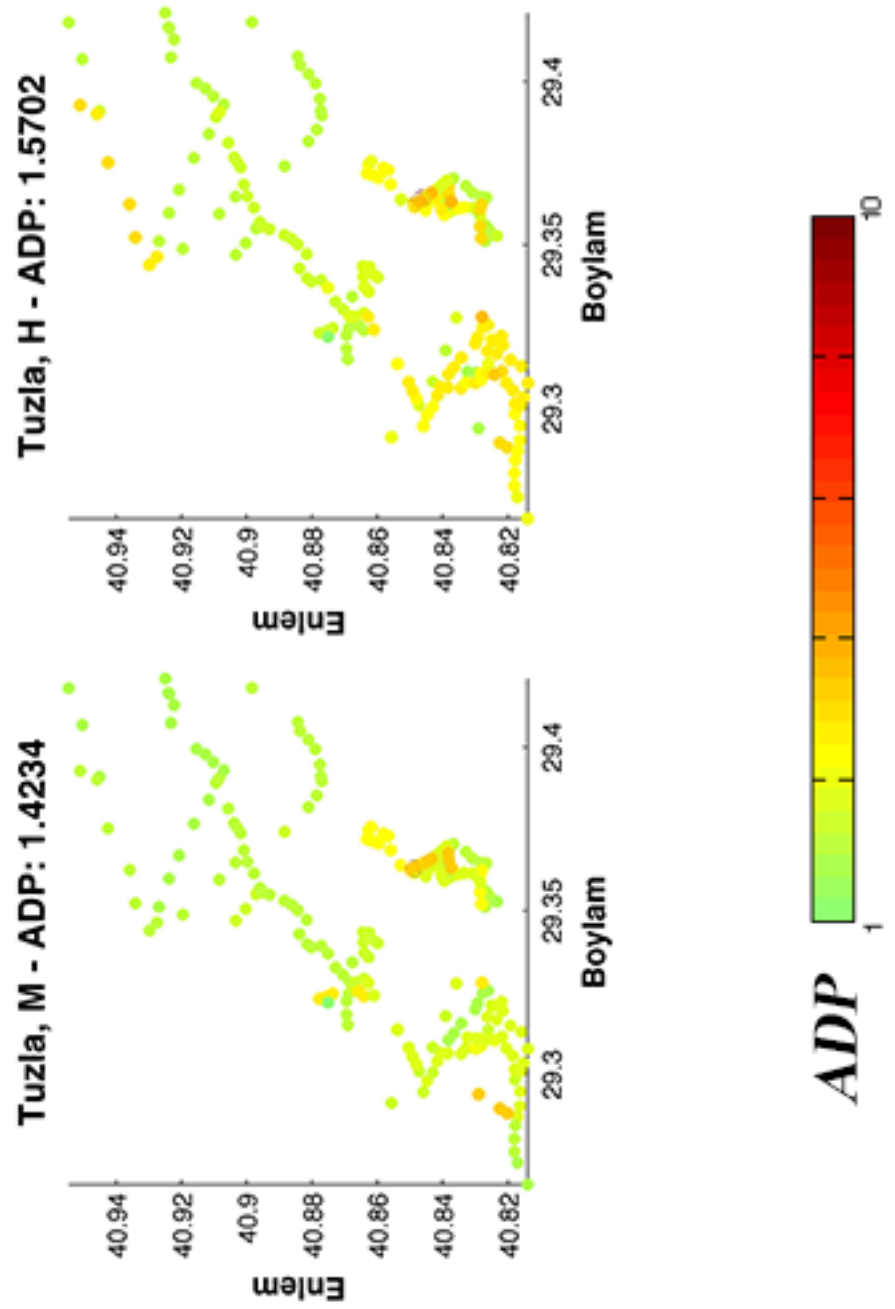
Şekil 7.9. Pendik ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



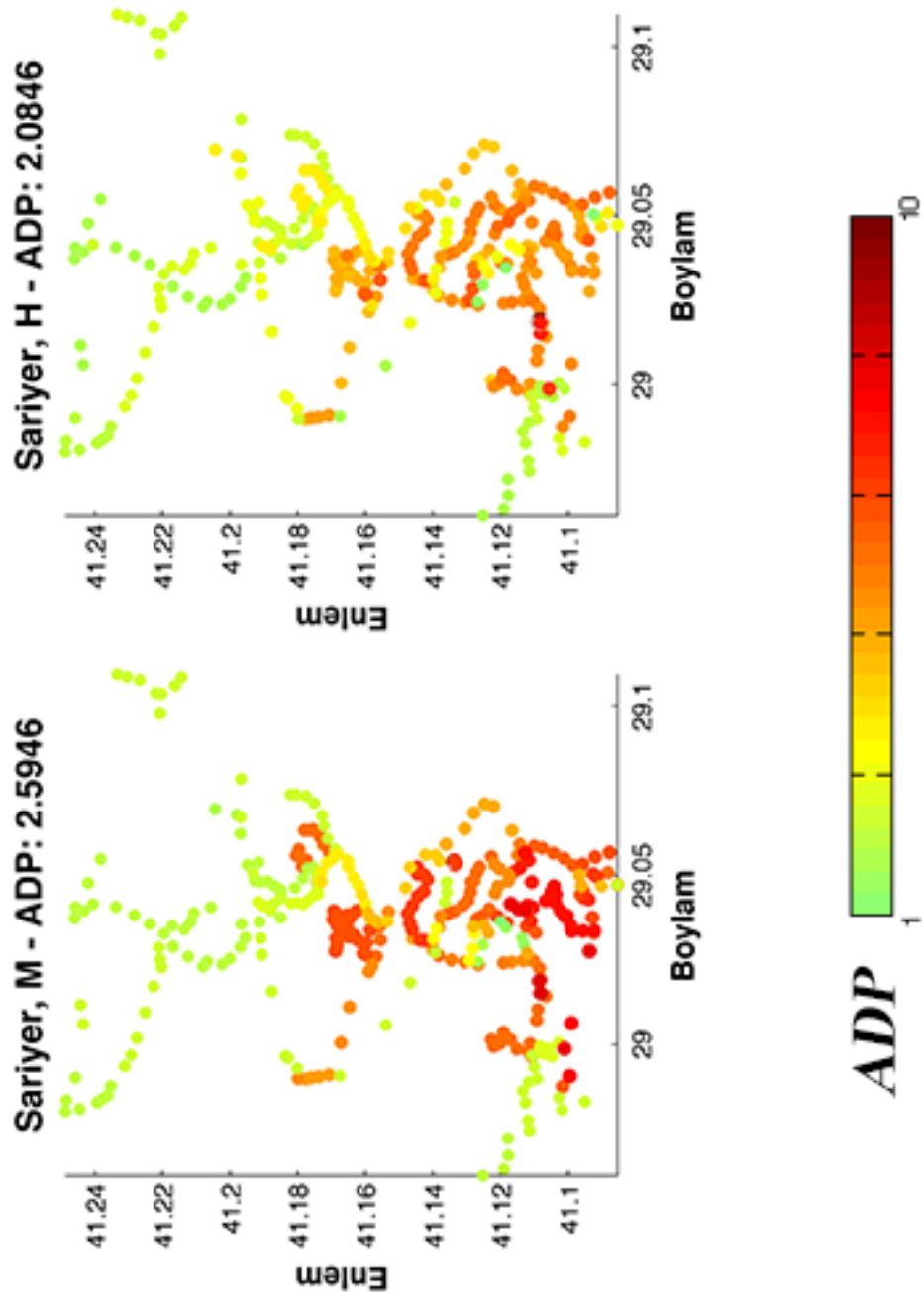
Şekil 7.10. Sultanbeyli ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



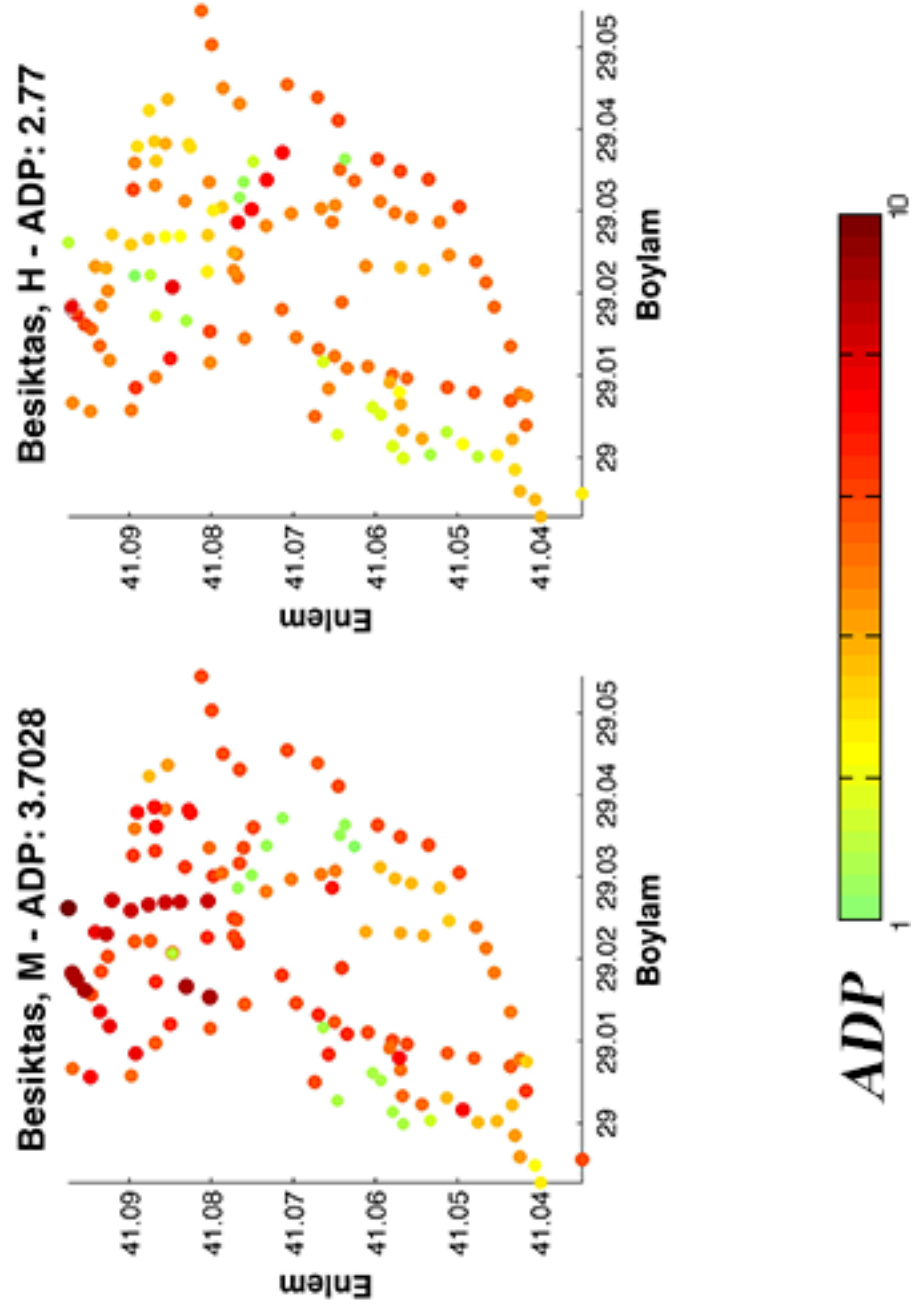
Şekil 7.11. Kartal ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



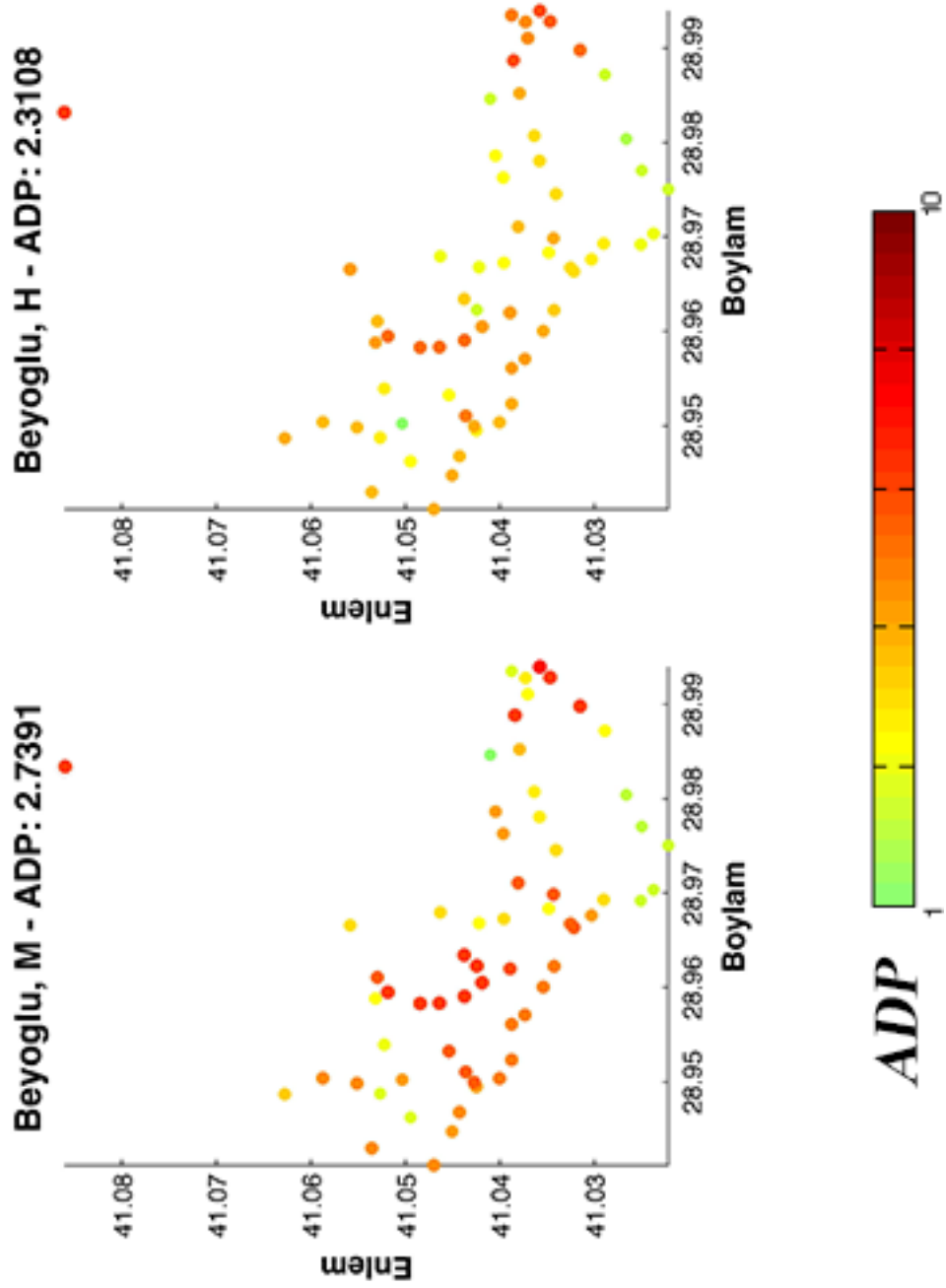
Şekil 7.12. Tuzla ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



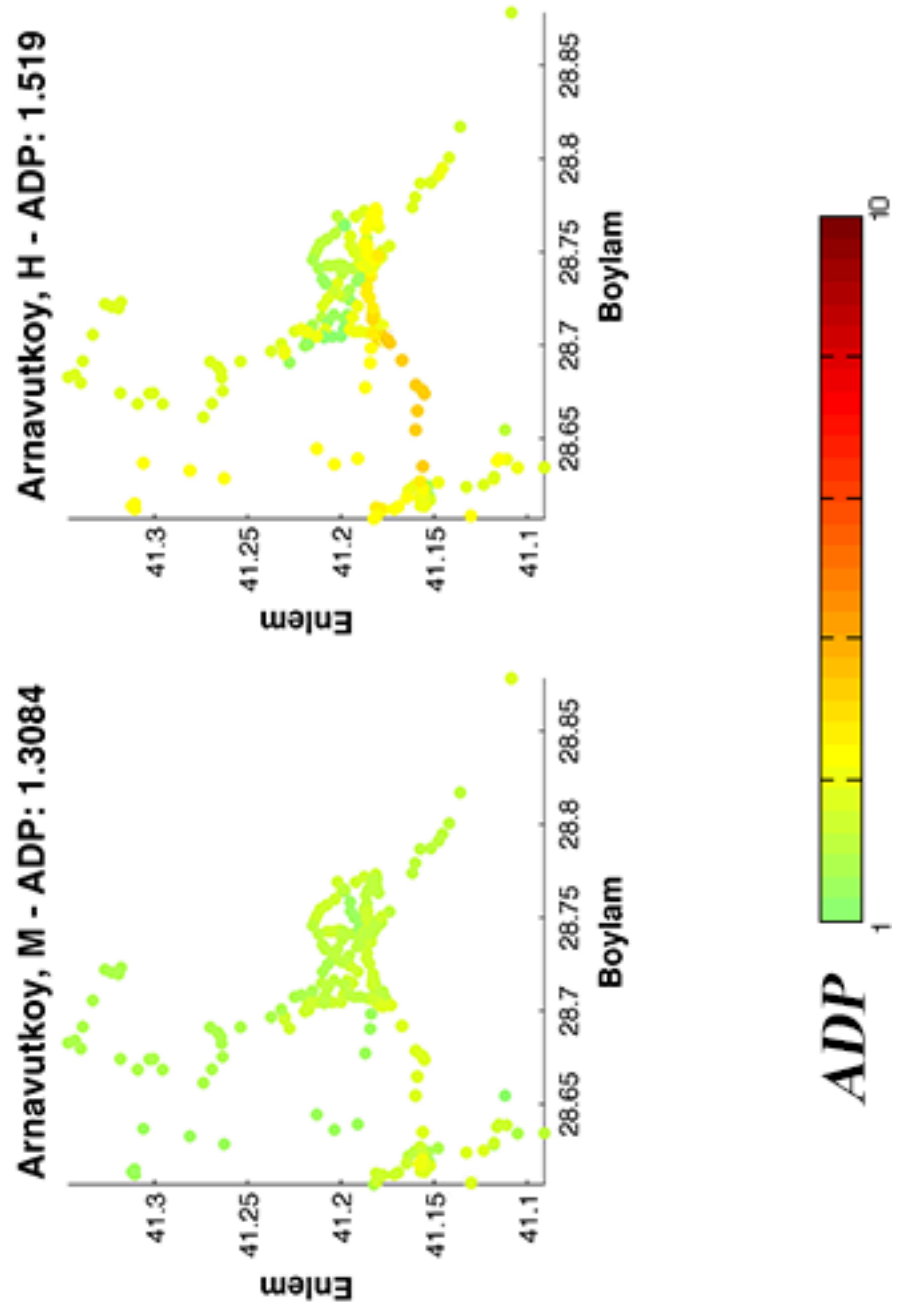
Şekil 7.13. Sariyer ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



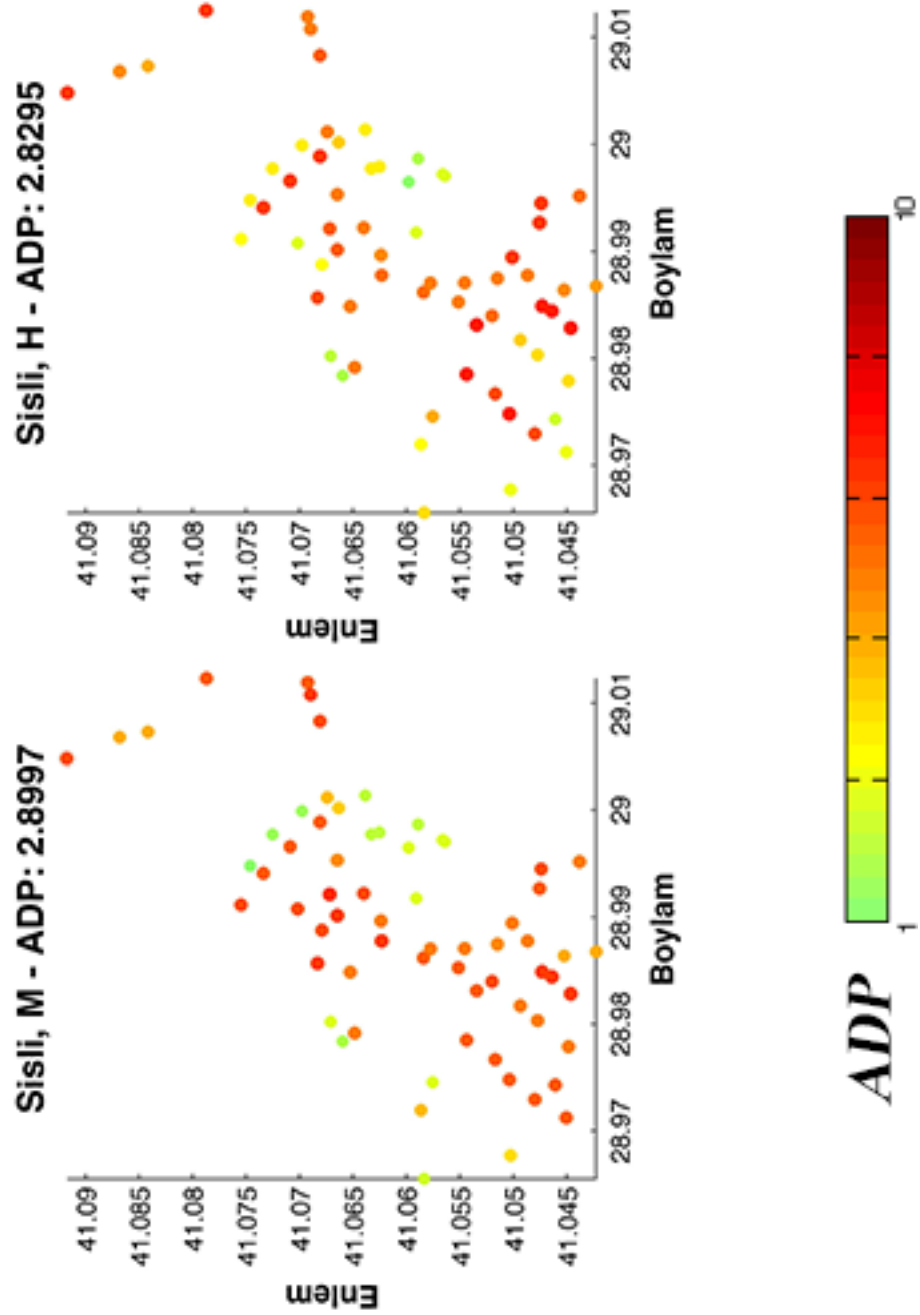
Şekil 7.14. Beşiktaş ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



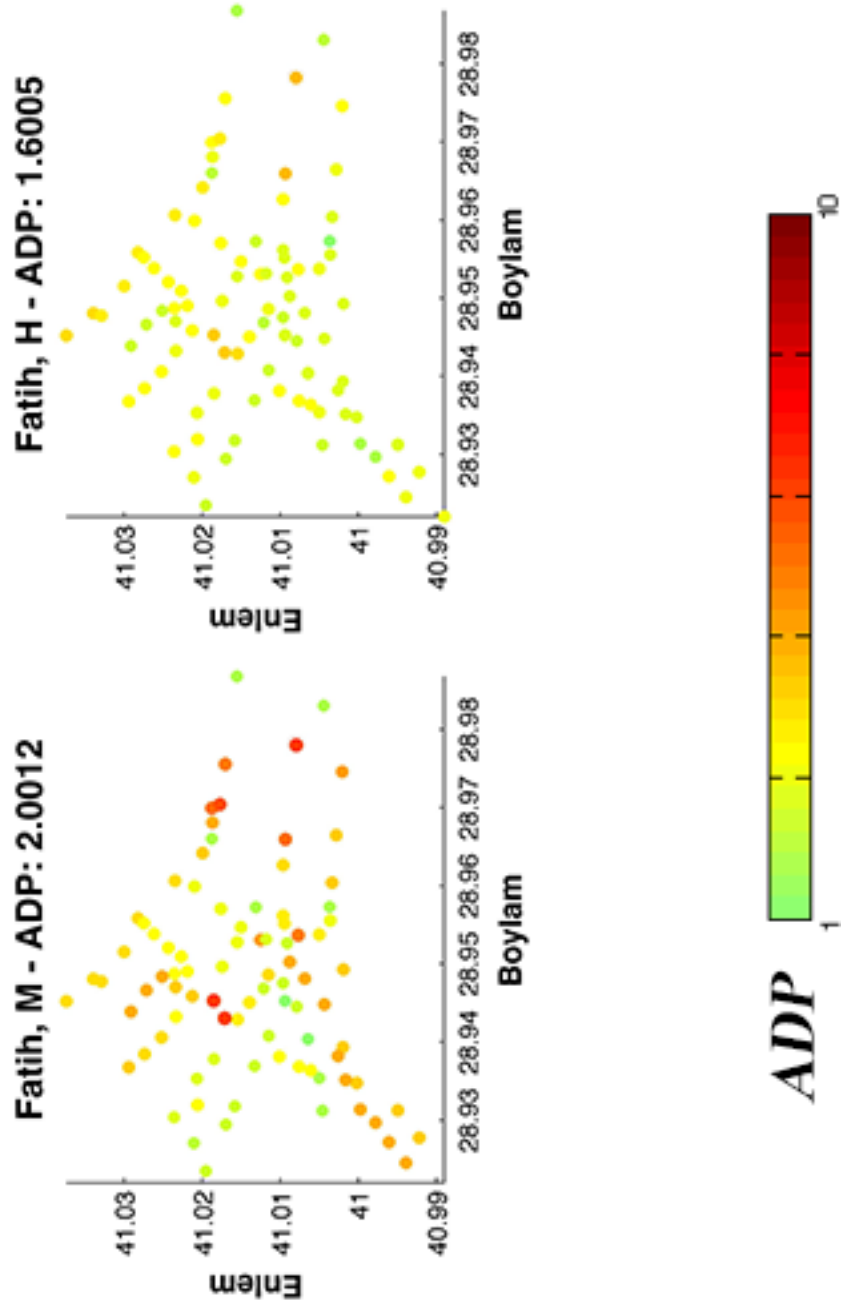
Şekil 7.15. Beyoğlu ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



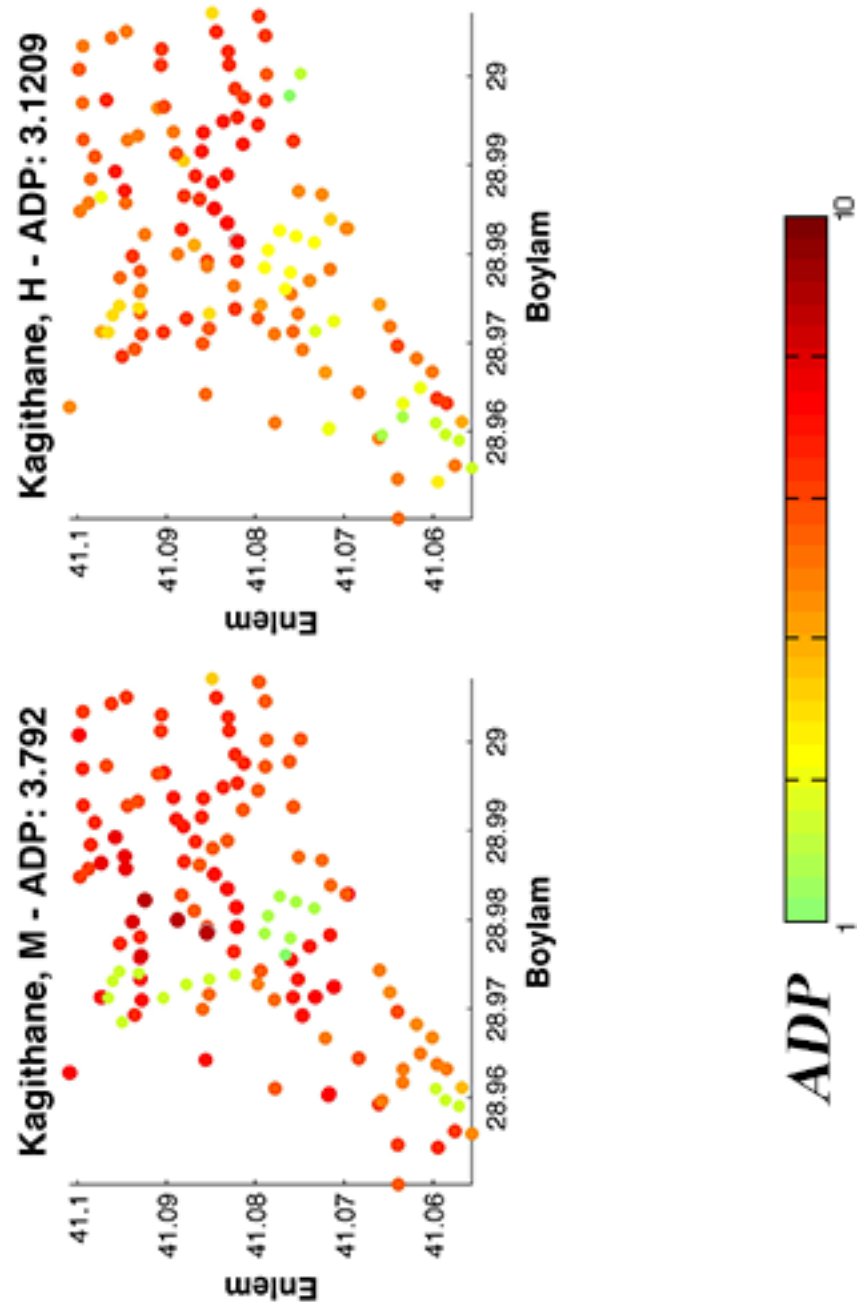
Şekil 7.16. Arnavutköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



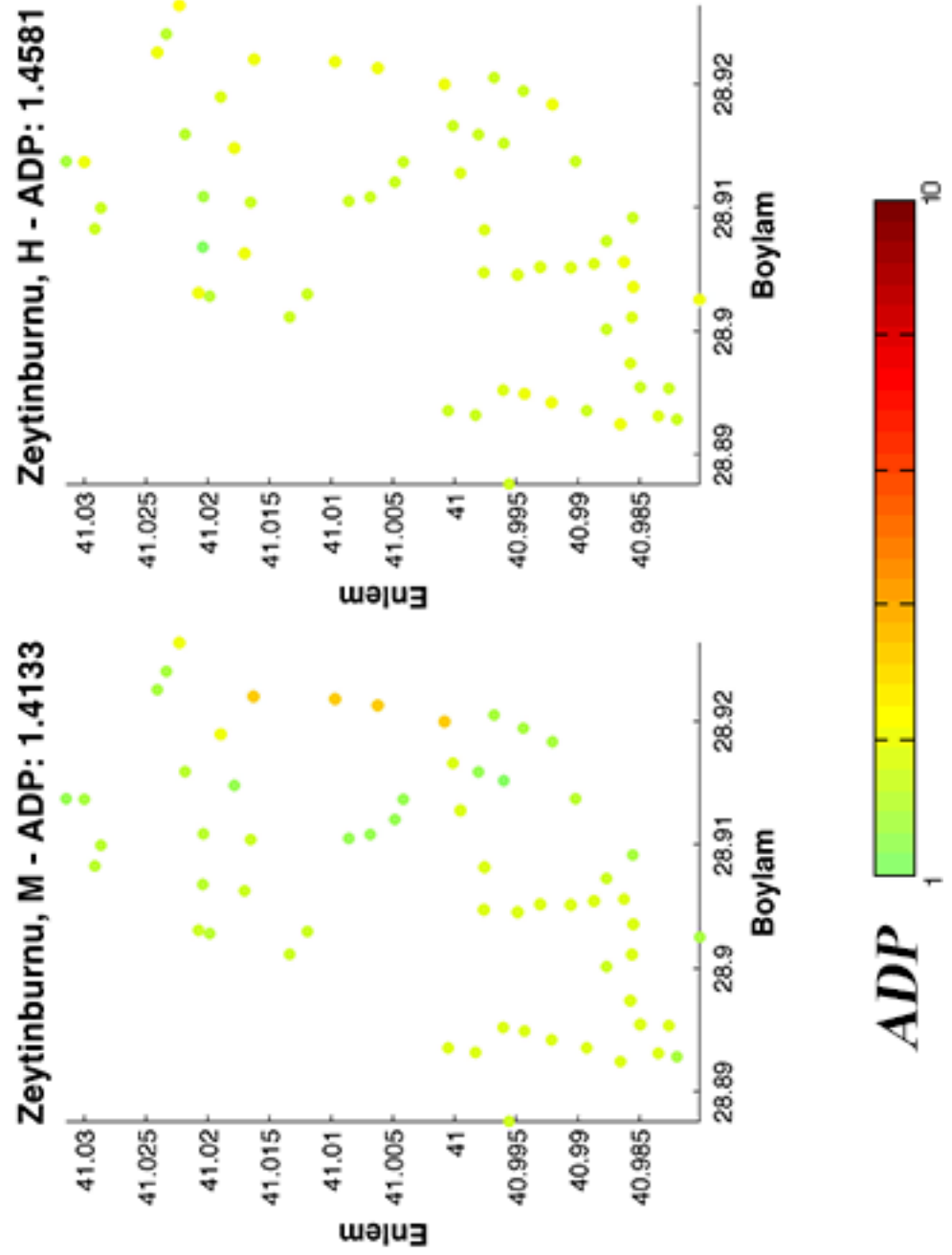
Şekil 7.17. Şişli ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



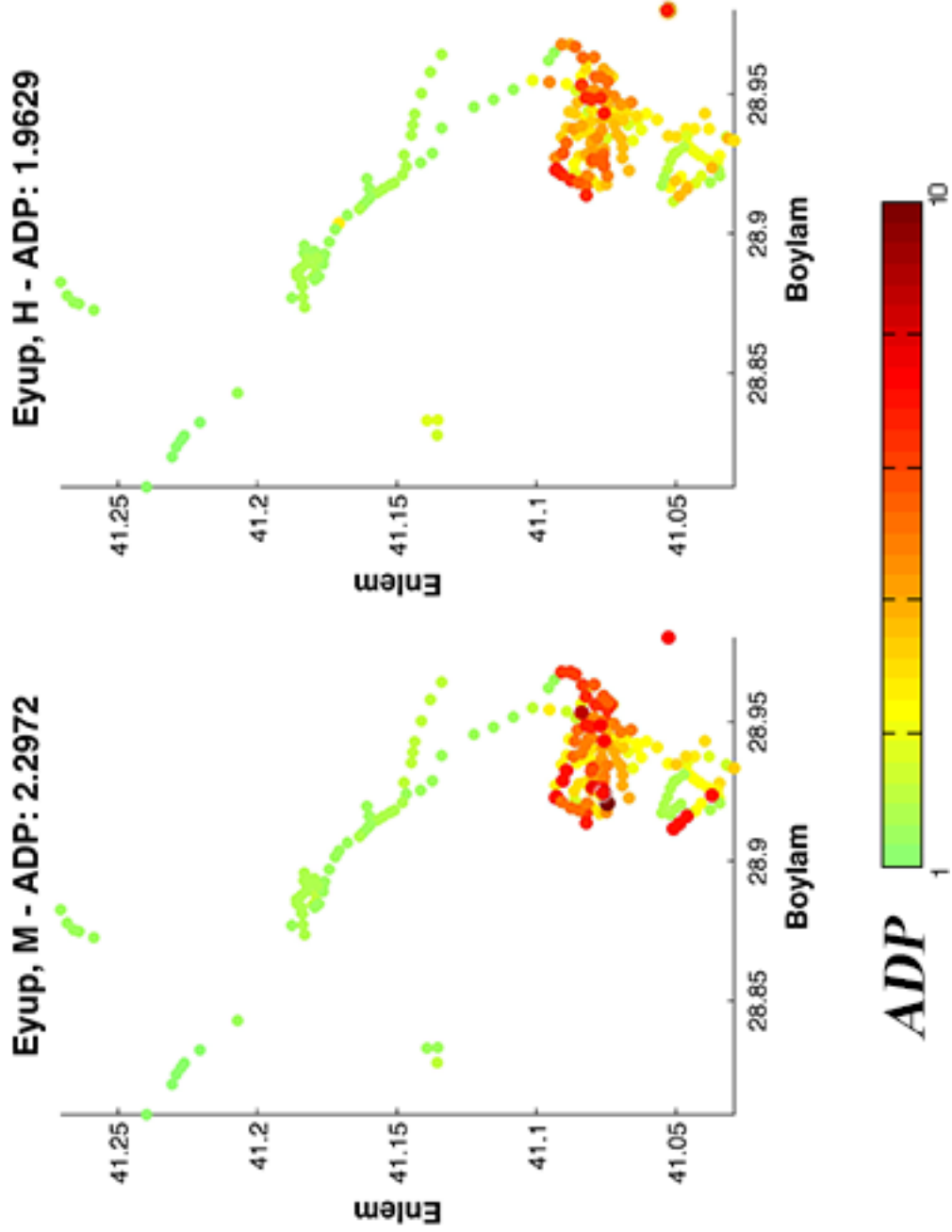
Şekil 7.18. Fatih ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



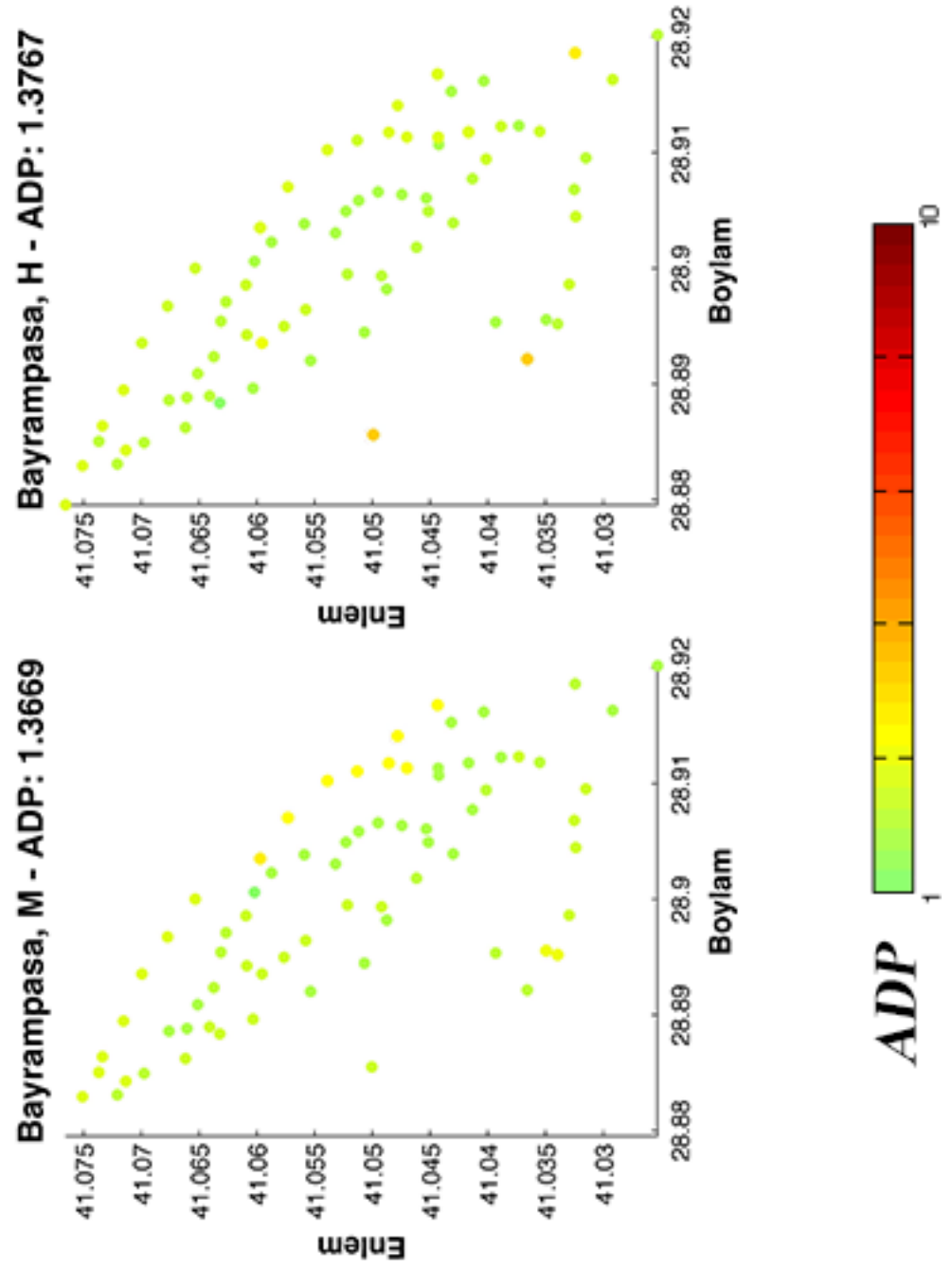
Şekil 7.19. Kağıthane ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



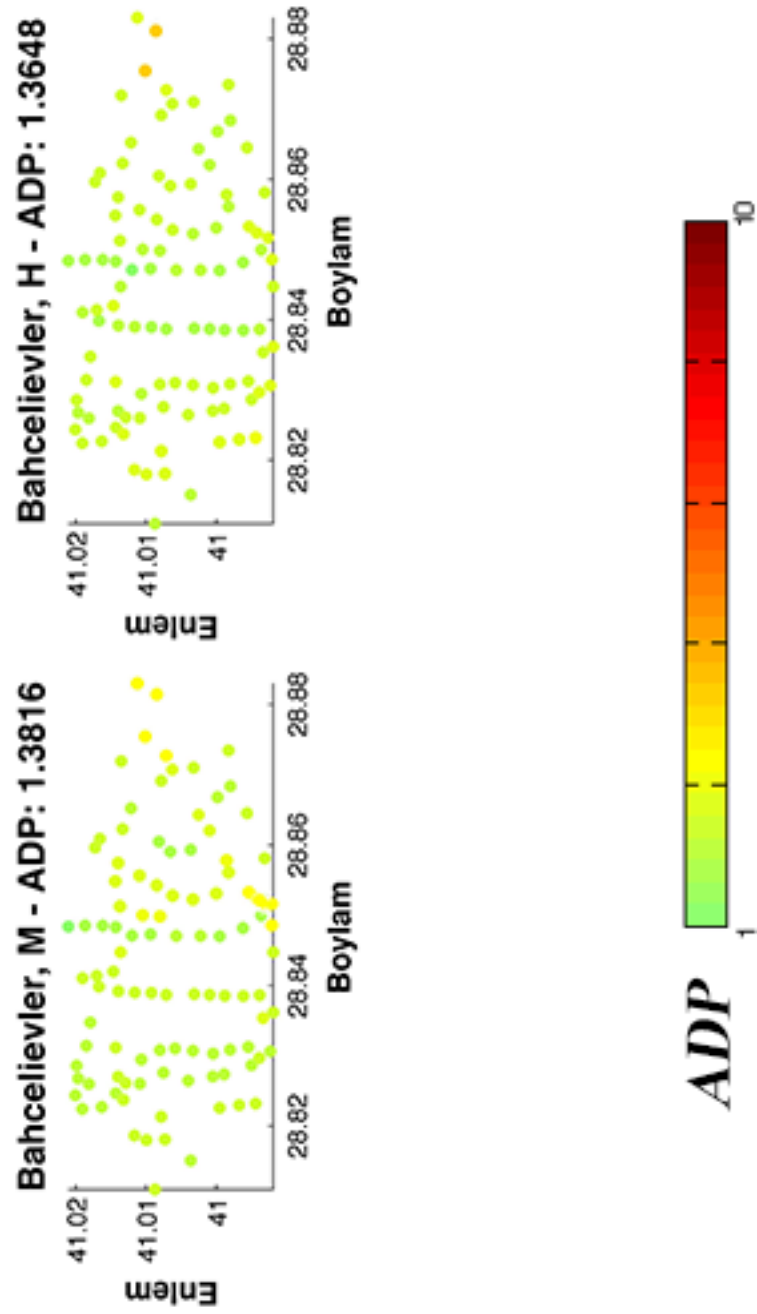
Şekil 7.20. Zeytinburnu ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



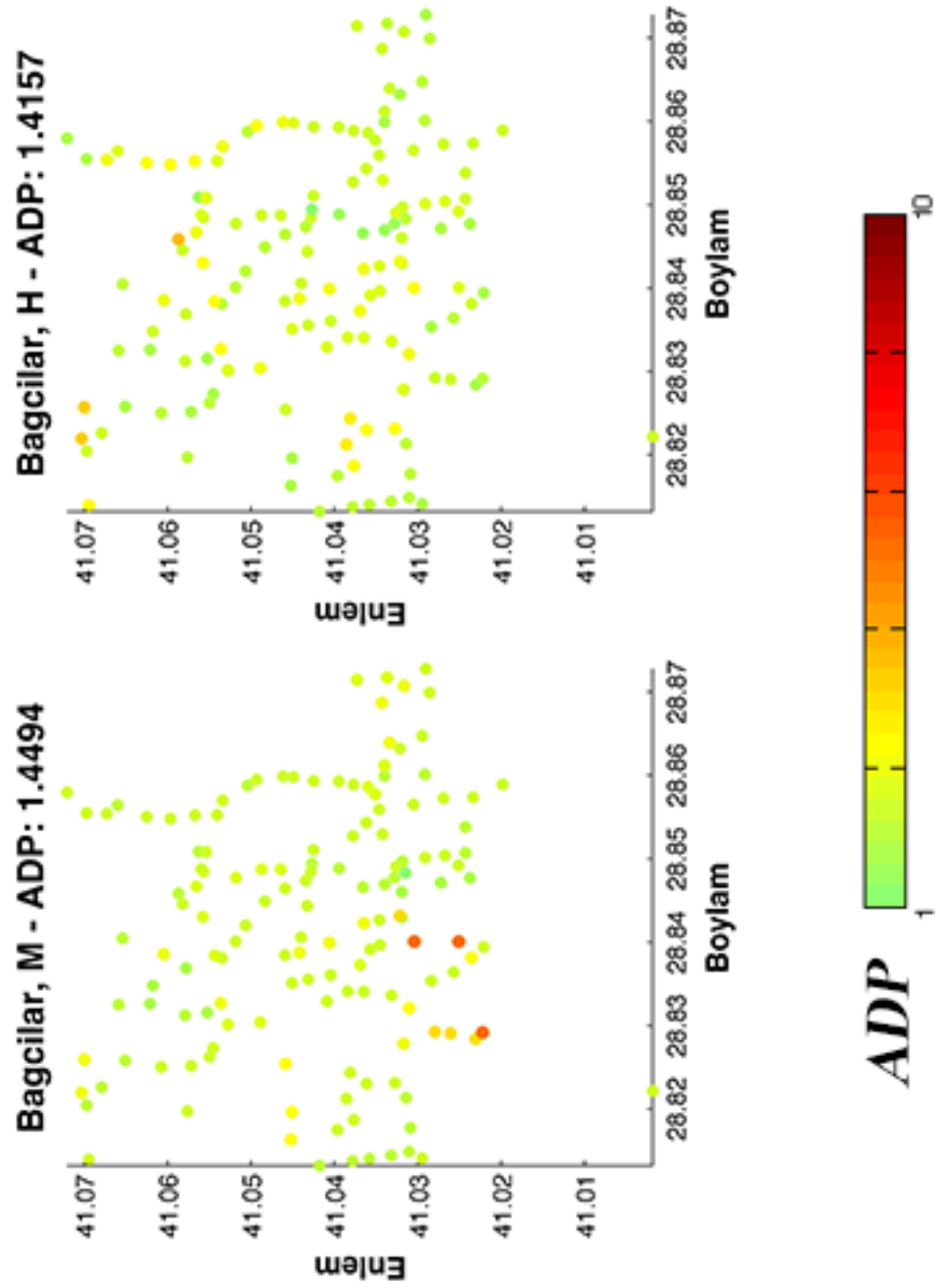
Şekil 7.21. Eyüp ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



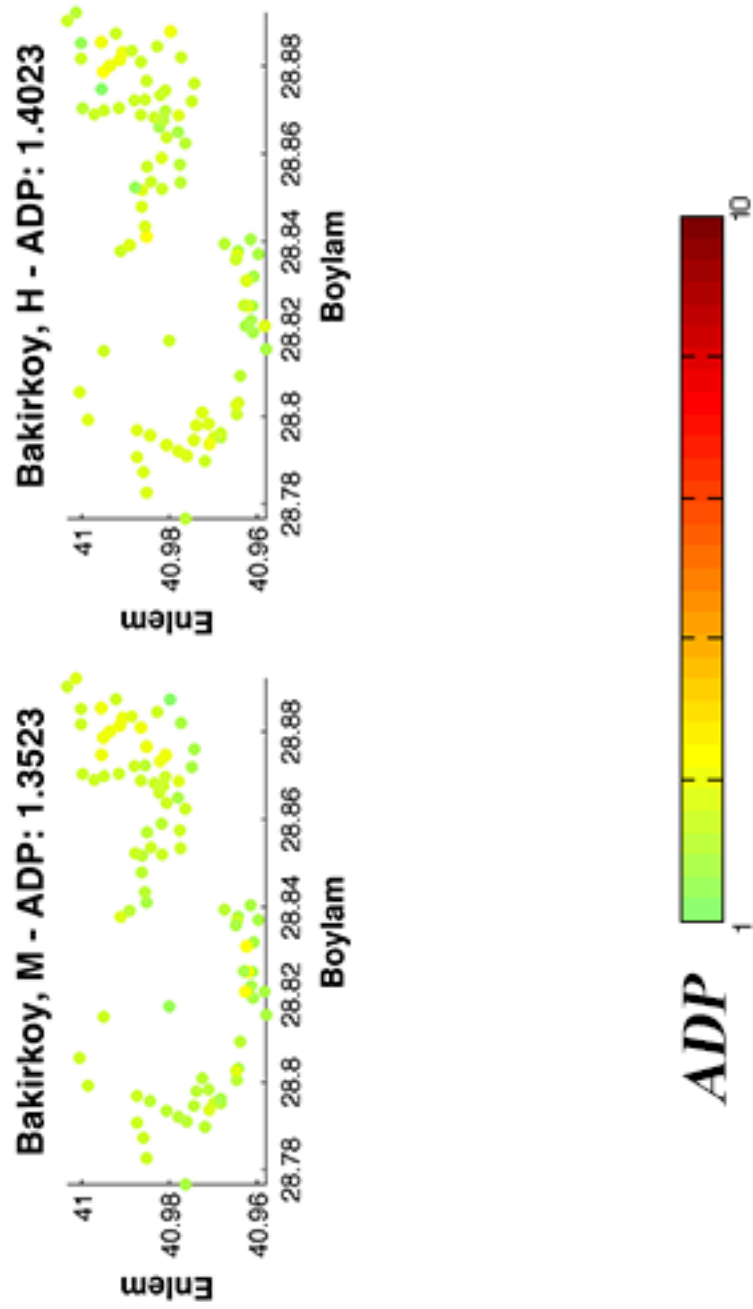
Şekil 7.22. Bayrampaşa ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



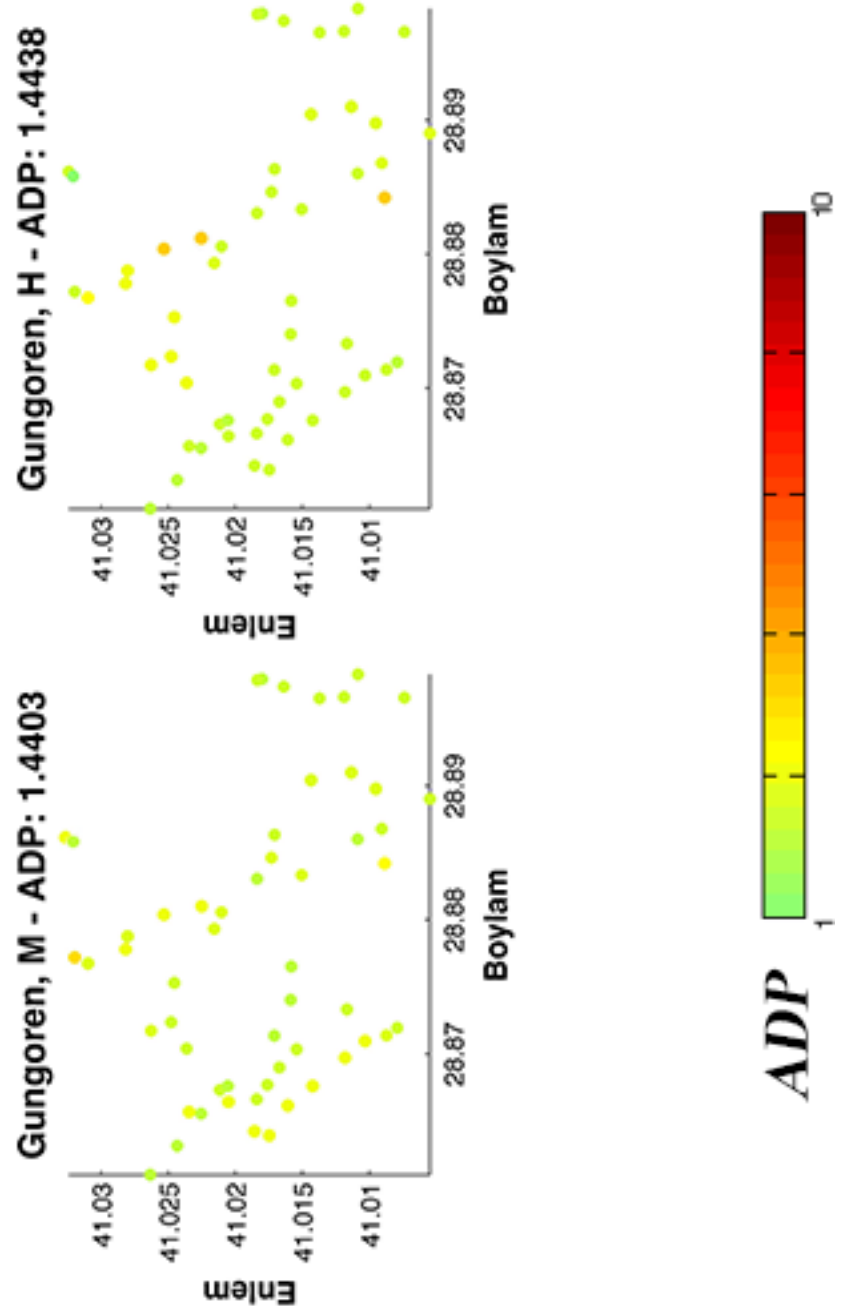
Şekil 7.23. Bahçelievler ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



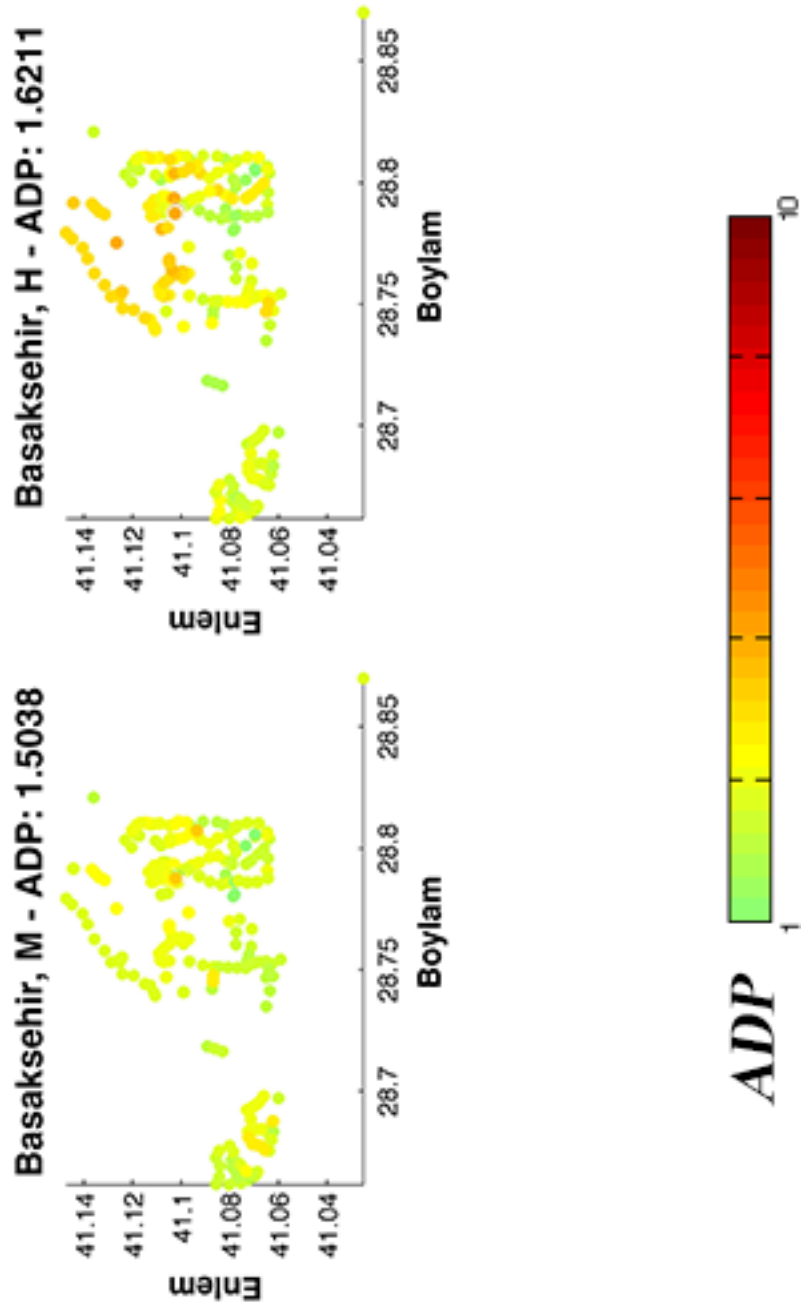
Şekil 7.24. Bağcılar ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



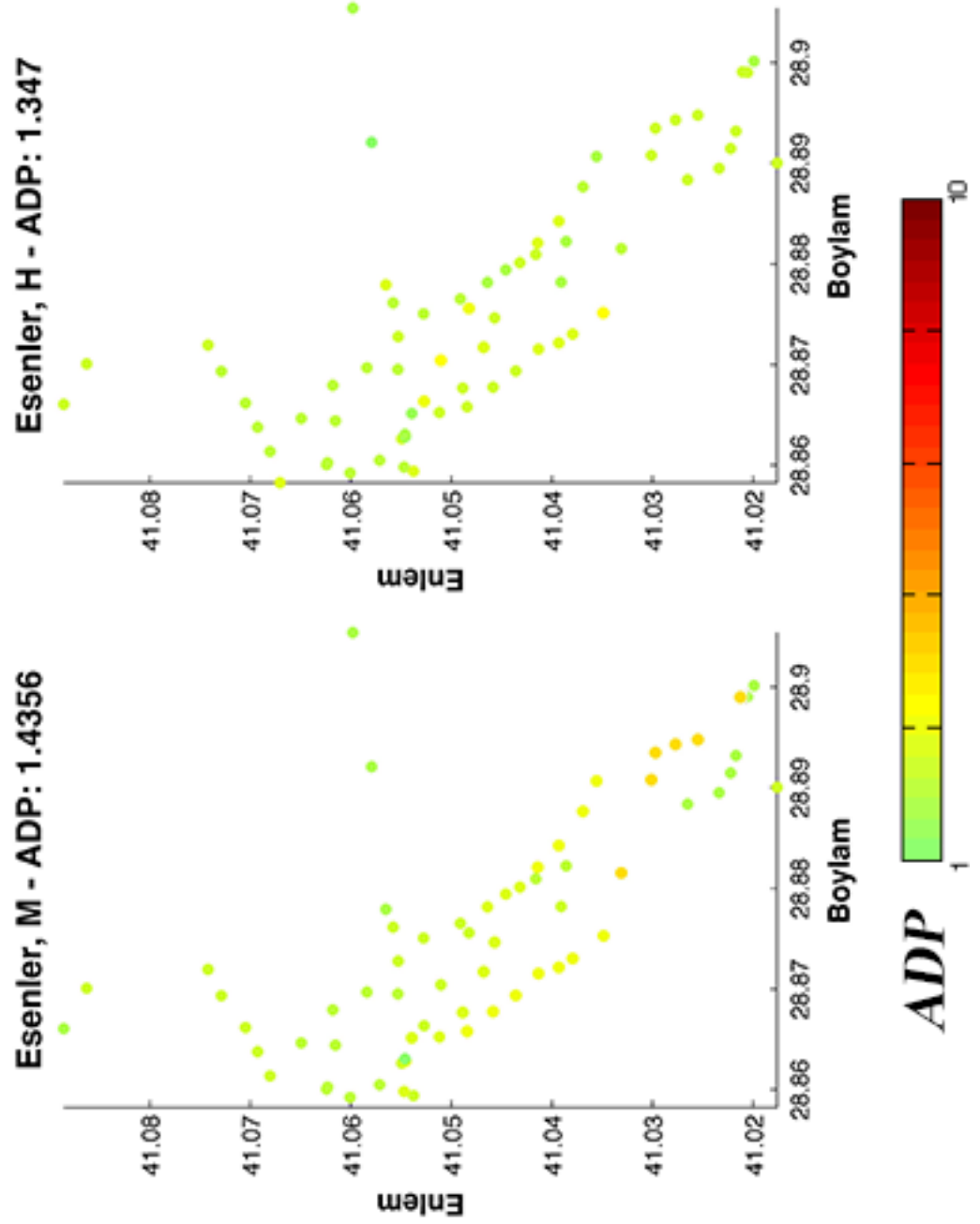
Şekil 7.25. Bakırköy ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



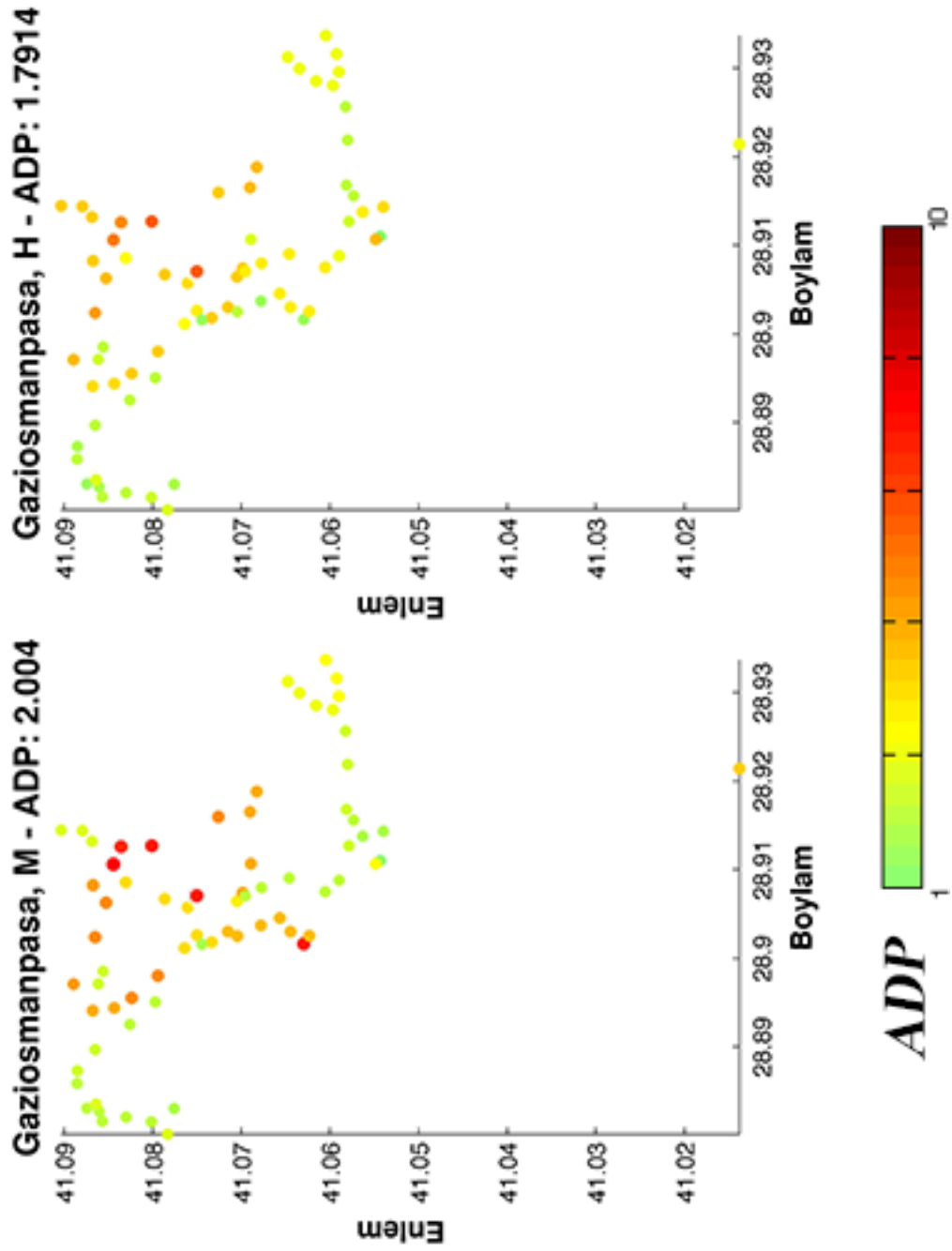
Şekil 7.26. Gungören ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



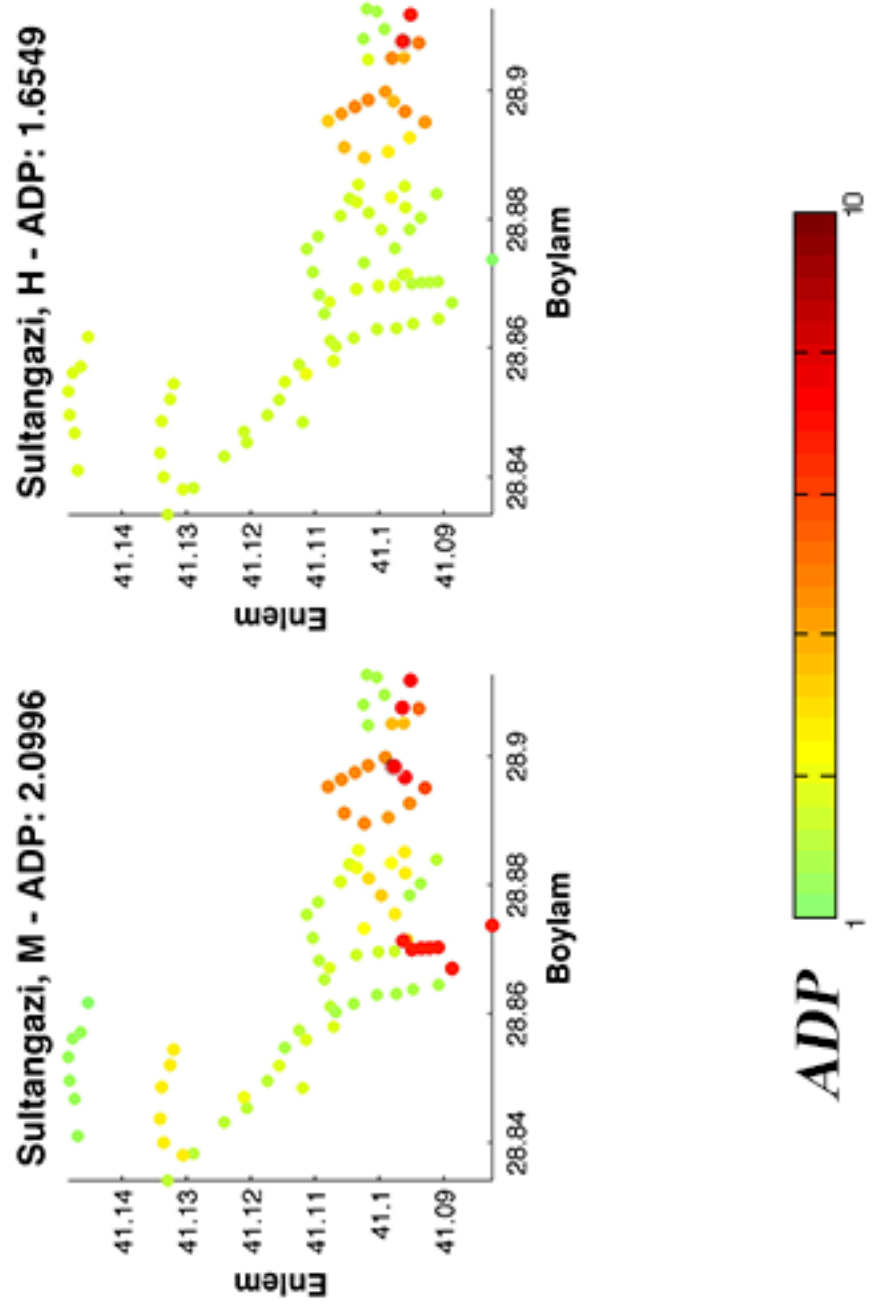
Şekil 7.27. Başakşehir ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



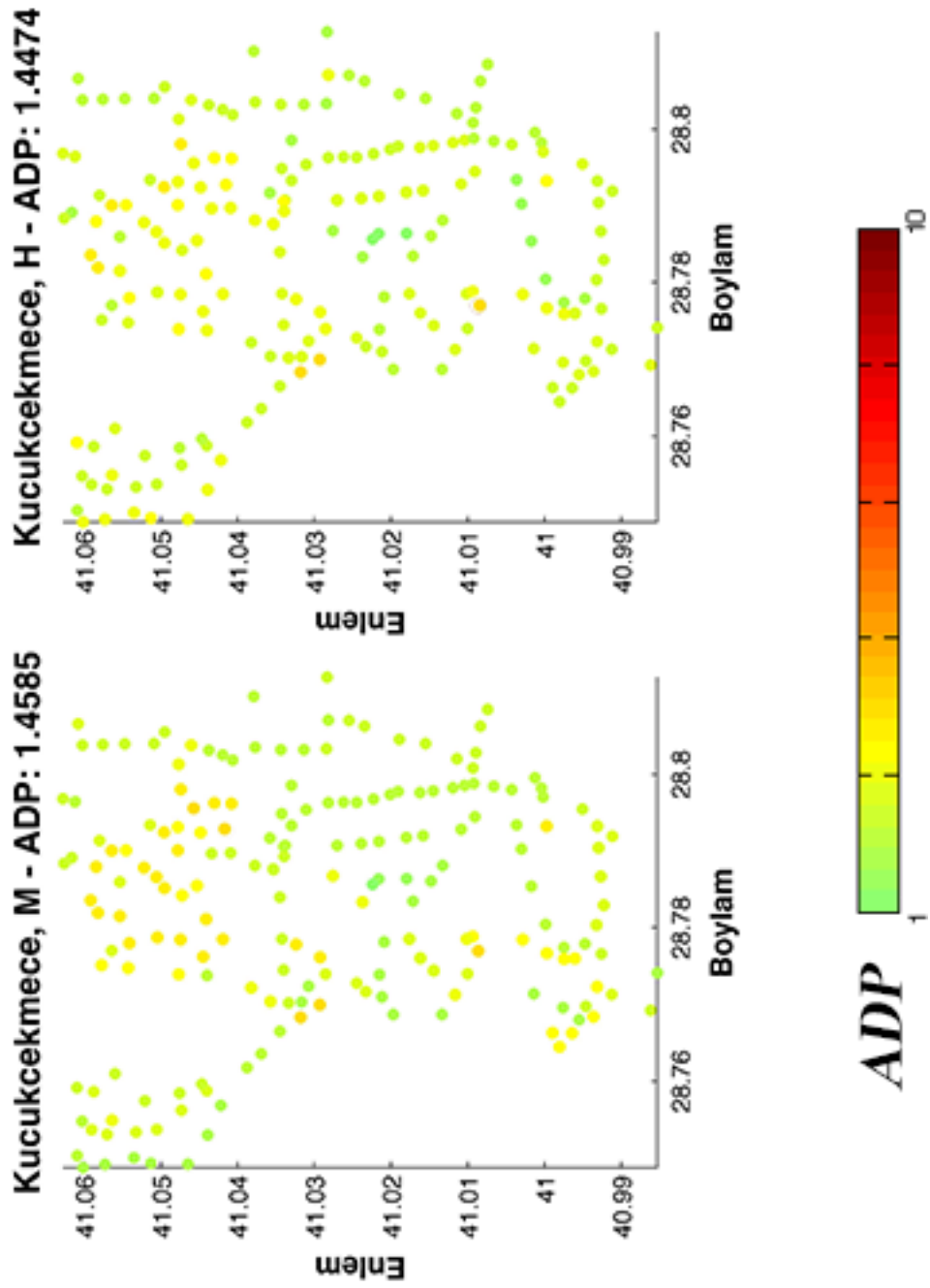
Şekil 7.28. Esenler ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



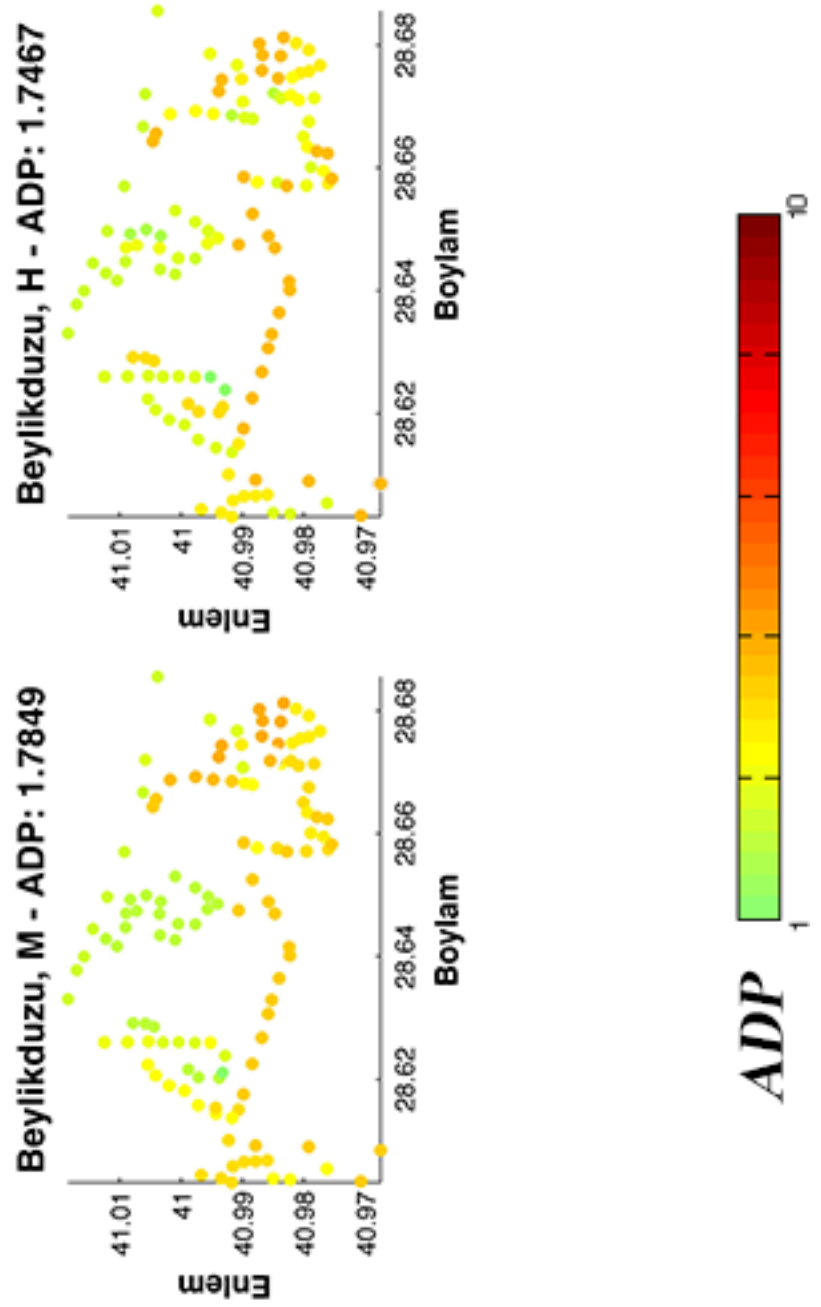
Şekil 7.29. Gaziosmanpaşa ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



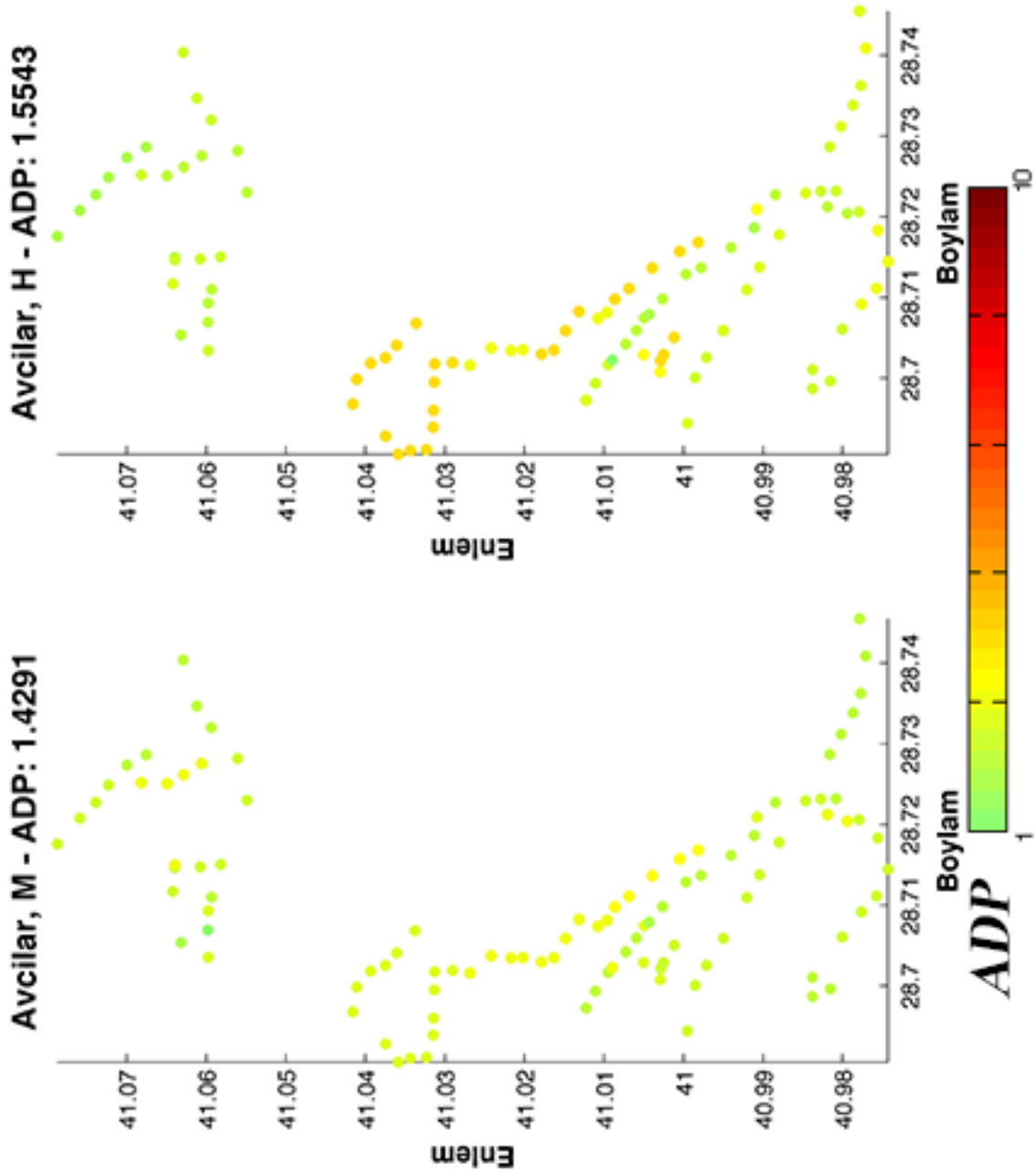
Şekil 7.30. Sultangazi ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



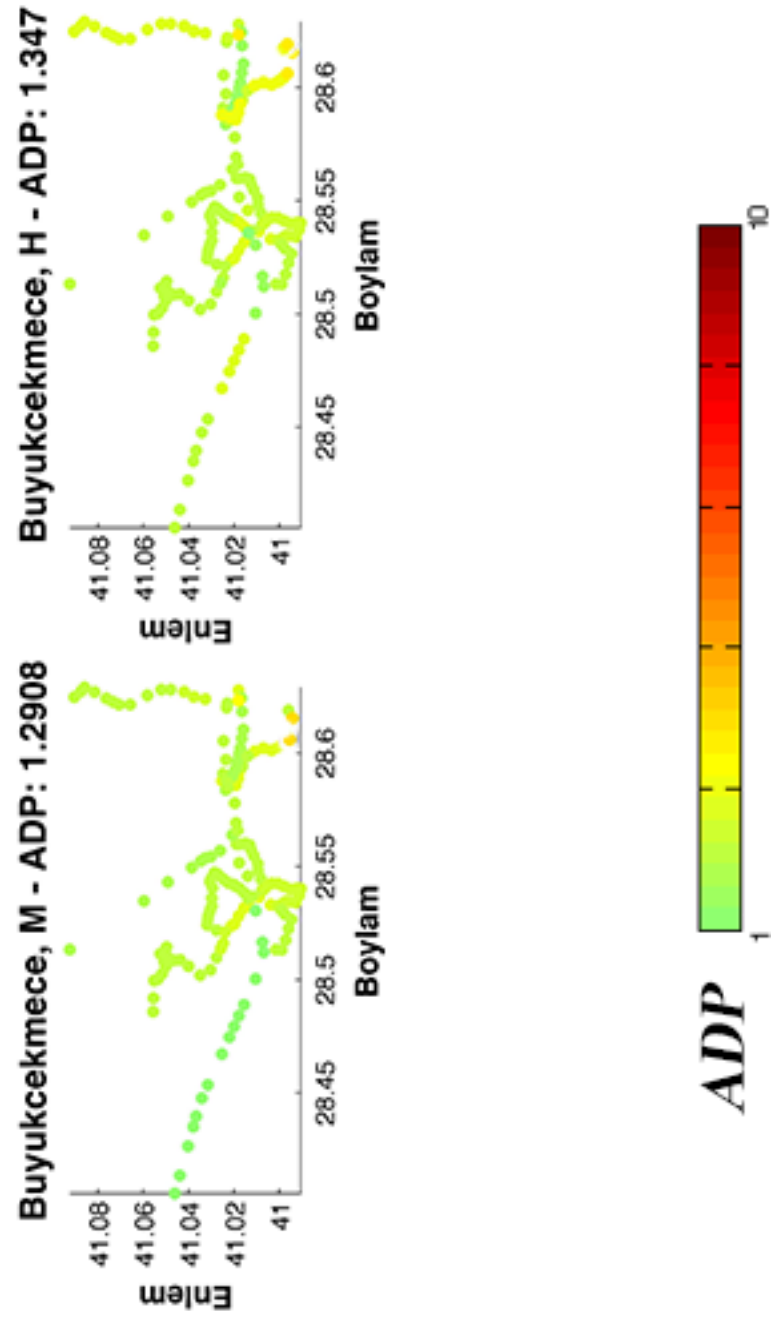
Şekil 7.31. Küçükçekmece ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



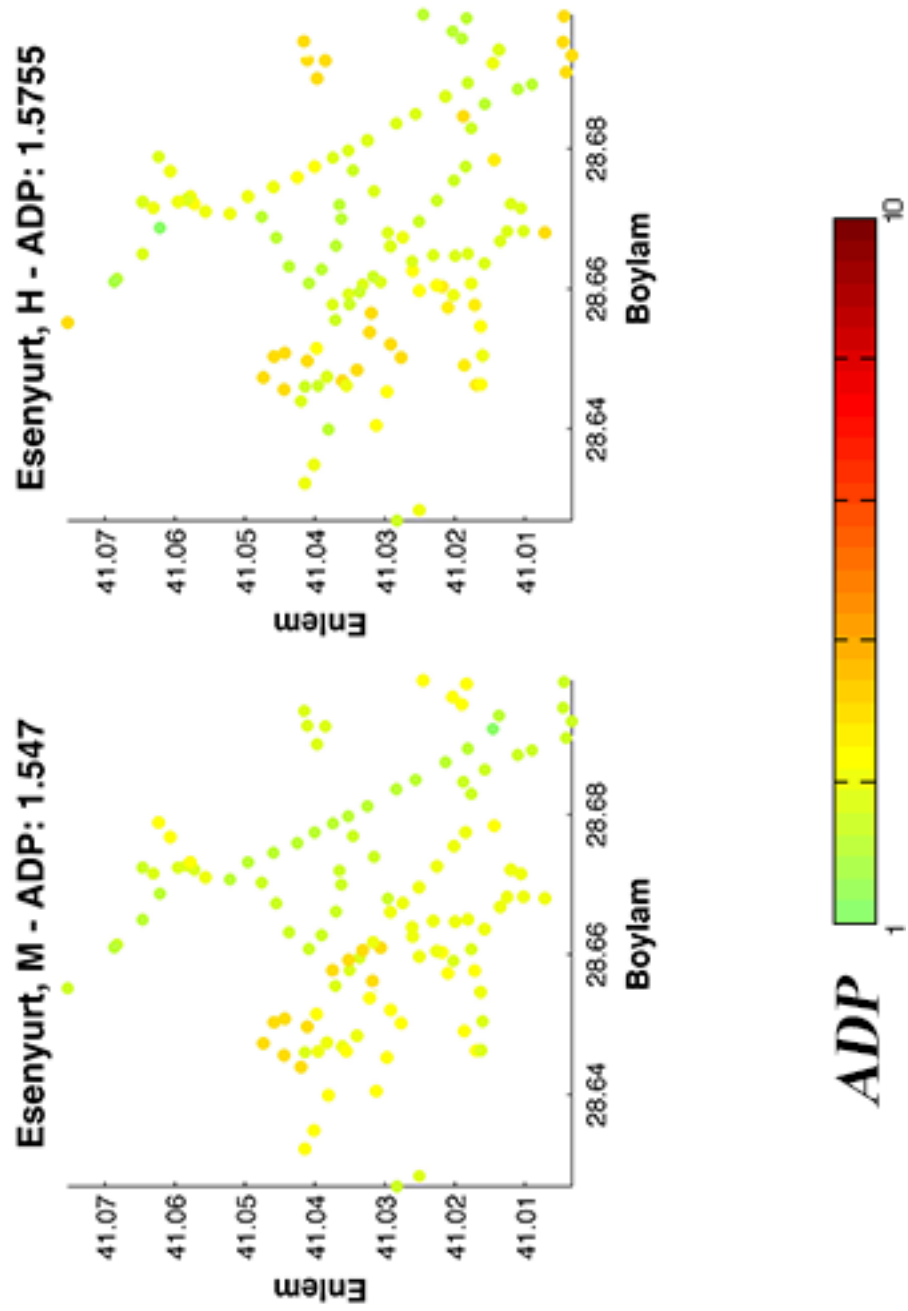
Şekil 7.32. Beylikdüzü ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



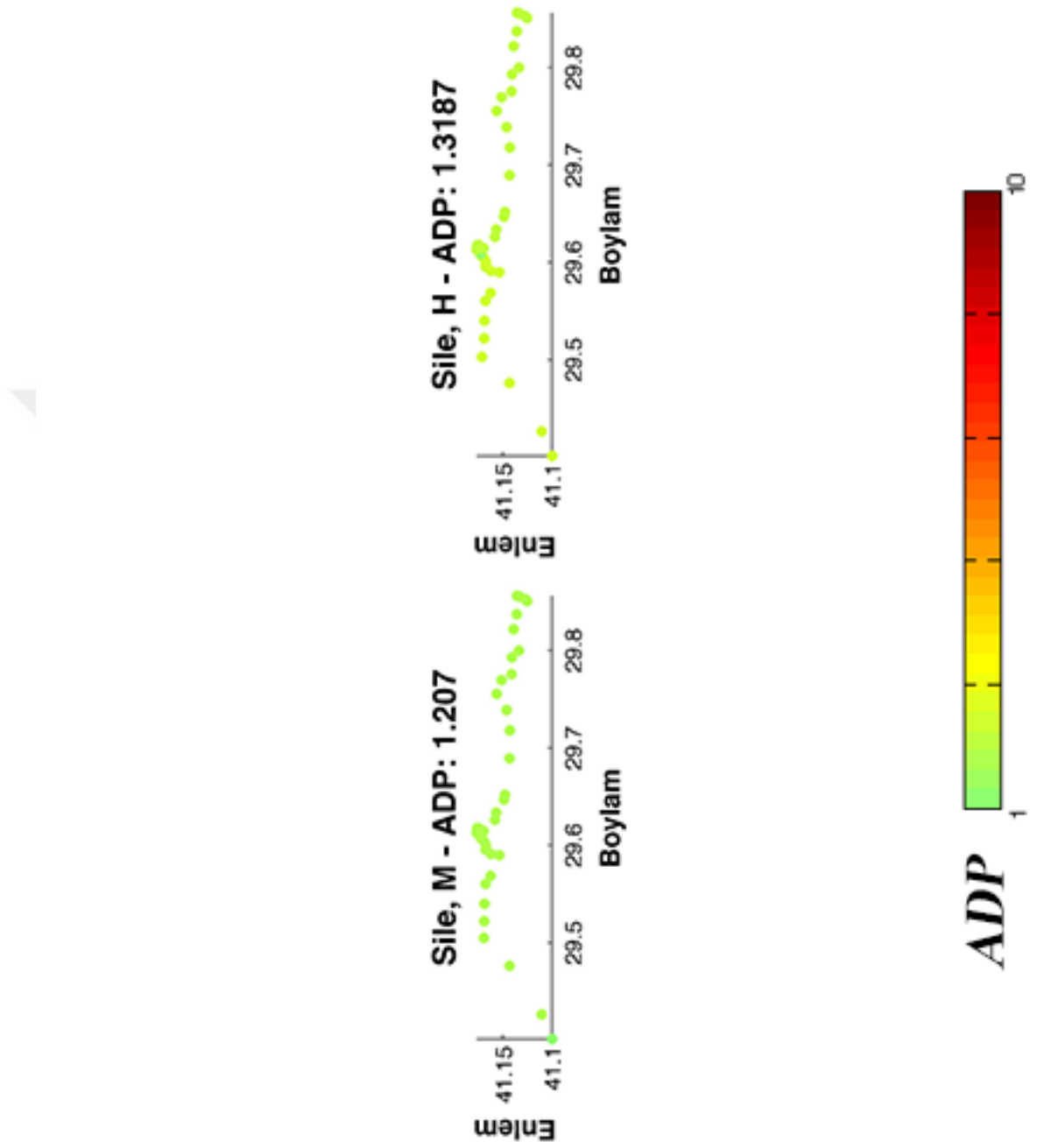
Şekil 7.33. Avcılar ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



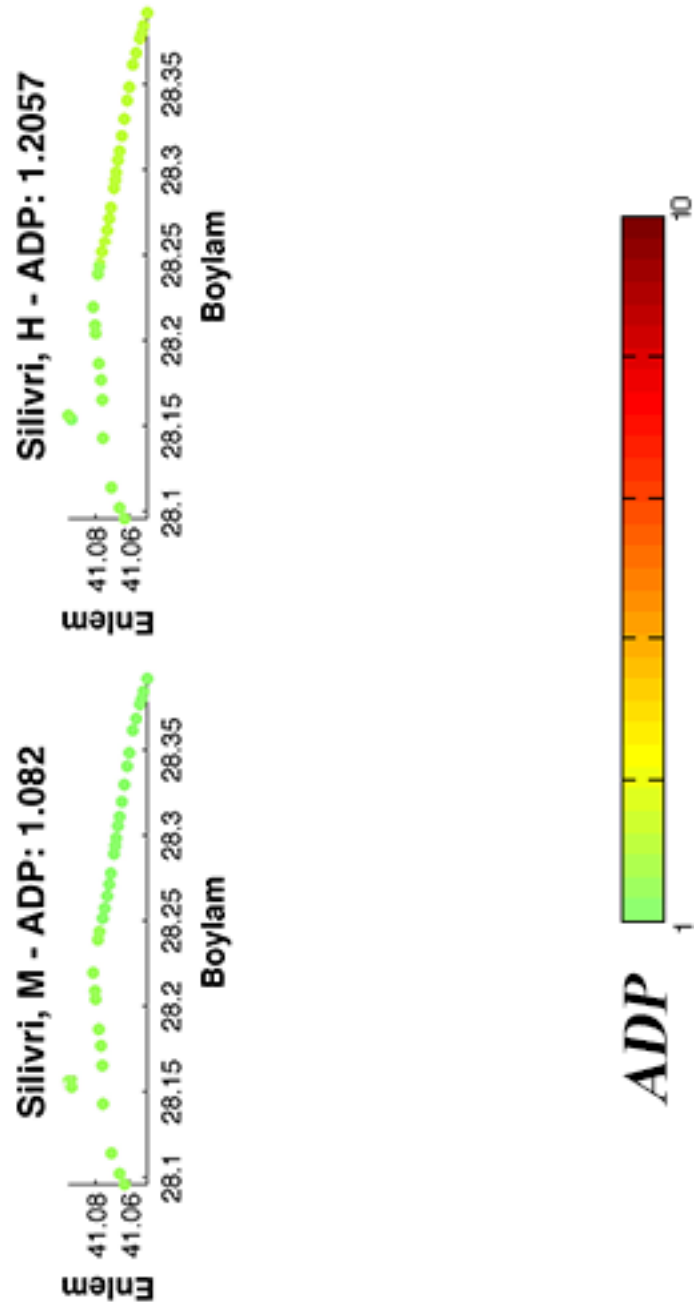
Şekil 7.34. Büyükçekmece ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



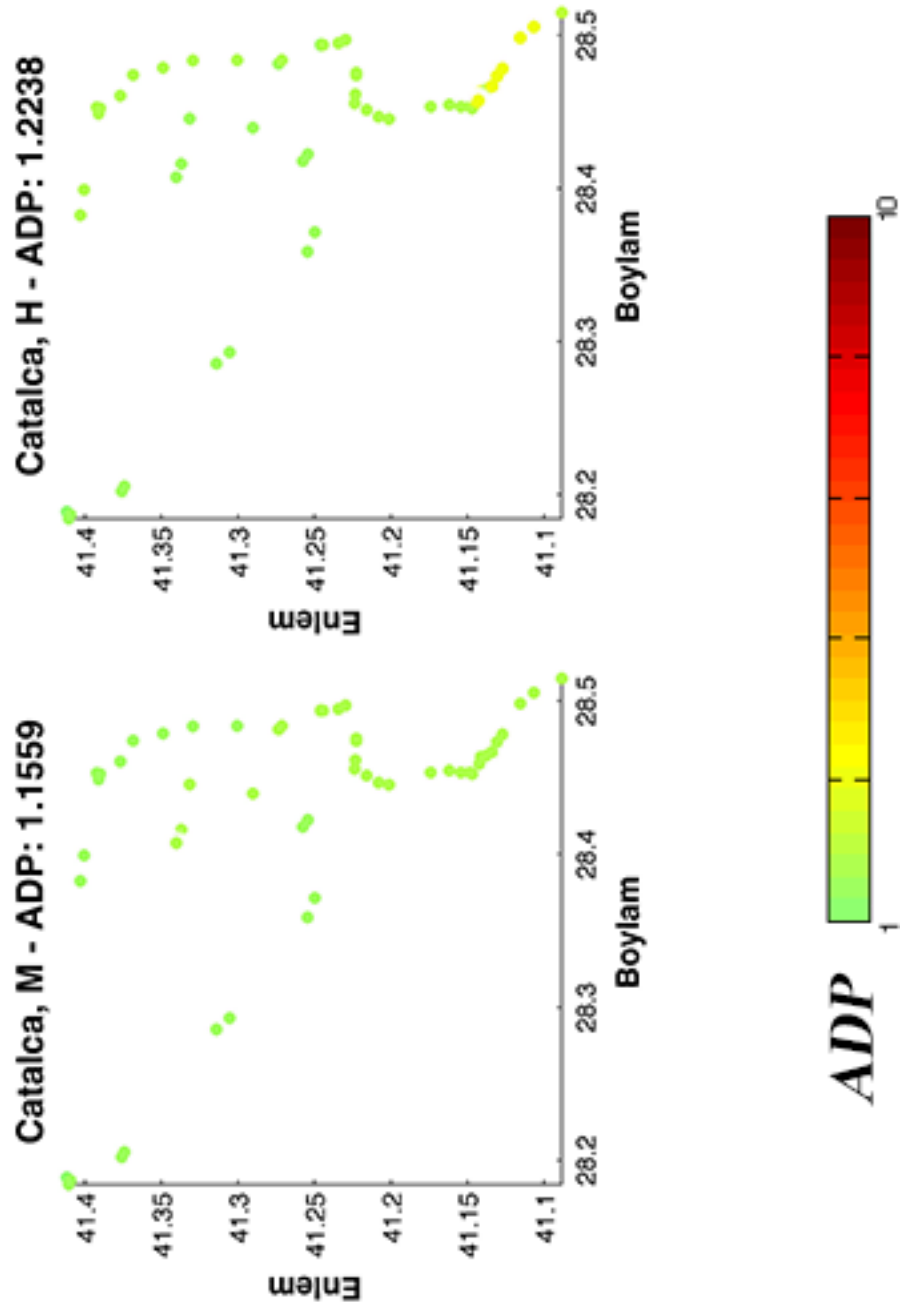
Şekil 7.35. Esenyurt ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



Şekil 7.36. Şile ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



Şekil 7.37. Silivri ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu



Şekil 7.38. Çatalca ilçesi mevcut durum ve hibrit optimizasyon ADP sonucu

ÖZGEÇMİŞ

KADİR AKGÖL

kadirakgol@hotmail.com.tr



ÖĞRENİM BİLGİLERİ

Doktora	Akdeniz Üniversitesi
2013-2018	Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Antalya
Yüksek Lisans	Akdeniz Üniversitesi
2011-2013	Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Antalya
Lisans	Akdeniz Üniversitesi
2005-2010	Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Antalya

MESLEKİ VE İDARİ GÖREVLER

Bölüm Başkanı	Giresun Üniversitesi
2018-Devam Ediyor	Tirebolu MYO, Ulaştırma Hizmetleri Bölümü, Giresun
Öğretim Görevlisi	Giresun Üniversitesi
2018-Devam Ediyor	Tirebolu MYO, Ulaştırma Hizmetleri Bölümü, Giresun
Araştırma Görevlisi	Akdeniz Üniversitesi
2013-2018	Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Antalya

ESERLER

Uluslararası Kitap

Akgöl K., Gunay B., "Rationality of Urban Road Networks", LAP LAMBERT Academic Publishing, 2018

SCI, SSCI, AHCI indekslerine giren dergilerde yayımlanan makaleler:

Akgöl K., Günay B., "Kentiçi Yol Ağlarının Rasyonellik Hesabında Monte Carlo Yönteminin Kullanımı", TEKNİK DERGI, vol.29, pp.8153-8165, 2018

Günay B., Akgöl K., Raveau S., "Rationality of route layouts as a quality indicator of urban road networks", URBAN DESIGN INTERNATIONAL, vol.21, no.3, pp.1-15, 2016

Günay B., Akgöl K., Andréasson I., Terzi S., "The Estimation of Modal Shift Potential For A New Form of Dial-A-Ride Service (in press)", JOURNAL OF PUBLIC TRANSPORTATION, vol.19, pp.1-11, 2016

Uluslararası dergilerde yayımlanan makaleler:

Aydin M.M., Köfteci S., Akgöl K., Yıldırım M.S., "Utilization of a New Methodology on Performance Measurements of Red Light Violations Detection Systems", International Journal of Engineering & Applied Sciences (IJEAS), vol.9, pp.32-41, 2017

Akgöl K., Aydın M.M., Asilkan O., Gunay B., "Prediction Of Modal Shift Using Artificial Neural Networks", Journal of Technology, Education, Management, Informatics, vol.3, pp.223-229, 2014

Ulusal bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler:

Günay, B., Aydın M.M. ve Akgöl K., Modern Dönel Kavşaklara Sorunsuz Bir Dönüşümün Bilimsel Metodolojisi, Karayolu 3. Ulusal Kongresi, 25-27 Kasım 2014, ss.1-12, Ankara.

Günay, B., Akgöl, K., Aydın, M.M. ve Terzi, S., Akıllı Servis Araçları Konseptinin Bilgisayar Modellemesi; Antalya Örneği, 1. Akıllı Ulaşım Sistemleri Kongre ve Sergisi, 26-28 Mayıs 2014, İstanbul.

Akgöl, K. ve Günay, B., Şehiriçi Yol Ağlarında Bir Kalite Göstergesi Olarak Güzergah Rasyonellik Skalaları, 10. Ulaştırma Kongresi, 25-27 Eylül 2013, İzmir.

Uluslararası bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler:

Günay B., Akgöl K., Terzi S., "Discovering the Potential for Intelligent Subscription Bus Services (Poster Presentation)", Public transport passenger flows in the era of ITS, Paris, FRANSA, 11-12 Mayıs 2015

Günay, B., Akgöl, K., Aydın, M.M. ve Terzi, S., Akıllı Servis Araçları Konseptindeki Türel Değişim Potansiyelinin Belirlenmesi, Transist 2013, 6. Uluslararası Ulaşım Sempozyumu ve Fuarı, 25-26 Aralık 2013, İstanbul.

Diğer Yayınlar

Günay B., Aydın M.M., Akgöl K., "TRT Haber", 2015

Akgöl K., Aydın M.M., Asilkan Ö., Günay B., "Prediction of Modal Shift Using Artificial Neural Networks", 4th World Conference on Innovation and Computer Sciences – Insoide-2014, Roma, İtalya, 11-13 Nisan 2014, Sunum, 2014