



**ZAMAN-BAĞIMLI HİYERARŞİK ÇİNLİ  
POSTACI PROBLEMİ VE ÇÖZÜM ÖNERİLERİ**

**Merve KAYACI ÇODUR**

**Doktora Tezi  
Endüstri Mühendisliği Anabilim Dalı  
Endüstri Mühendisliği Bilim Dalı  
Dr. Öğr. Üyesi Mustafa YILMAZ**

**2018**

**Her hakkı saklıdır**

**ATATÜRK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DOKTORA TEZİ**

**ZAMAN-BAĞIMLI HİYERARŞİK ÇİNLİ POSTACI PROBLEMİ  
VE ÇÖZÜM ÖNERİLERİ**

**Merve KAYACI ÇODUR**

**ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI  
Endüstri Mühendisliği Bilim Dalı**

**ERZURUM  
2018**

**Her hakkı saklıdır**



TEZ ONAY FORMU

ZAMAN-BAĞIMLI HİYERARŞİK ÇİNLİ POSTACI PROBLEMİ VE ÇÖZÜM  
ÖNERİLERİ

Dr. Öğr. Üyesi Mustafa YILMAZ danışmanlığında, Merve KAYACI ÇODUR tarafından hazırlanan bu çalışma, 14/12/2018 tarihinde aşağıdaki jüri tarafından Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Bilim Dalı'nda Doktora tezi olarak oybirliği /oy çokluğu (.../...) ile kabul edilmiştir.

Başkan: Prof. Dr. Cafer ÇELİK

İmza :

Üye : Prof. Dr. Ömer GÜNDOĞDU

İmza :

Üye : Doç. Dr. Selçuk Kürşat İŞLEYEN

İmza :

Üye : Doç. Dr. Hakan ÇERÇİOĞLU

İmza :

Üye : Dr. Öğr. Üyesi Mustafa YILMAZ

İmza :

Yukarıdaki sonuç;

Enstitü Yönetim Kurulu'nun 27/12/2018 tarih ve ...52.../...29..... nolu kararı ile onaylanmıştır.

Prof. Dr. Mehmet KARAKAN  
Enstitü Müdürü

Bu çalışma BAP projesi kapsamında desteklenmiştir.  
Proje No: 2015/143

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

## ÖZET

Doktora Tezi

### ZAMAN-BAĞIMLI HİYERARŞİK ÇİNLİ POSTACI PROBLEMİ VE ÇÖZÜM ÖNERİLERİ

Merve KAYACI ÇODUR

Atatürk Üniversitesi  
Fen Bilimler Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Mustafa YILMAZ

Hiyerarşik Çinli Postacı Problemi (HÇPP), ayrıt rotalama problemlerinden biri olan Çinli Postacı Probleminin (ÇPP) bir türüdür. NP-zor problem olan HÇPP ile ilgili yapılan çalışmalarda düğümler arası mesafelerin dikkate alındığı ve seyahat sürelerinin göz ardı edildiği görülmektedir. Ancak, hemen hemen tüm metropol kentlerde seyahat eden araçların hızı ve buna bağlı olarak da ulaşım süreleri trafik yoğunluğu, trafik kazaları ve hava durumu gibi çeşitli faktörlerden dolayı katedilen mesafe ile birlikte gün boyunca değişmektedir. Dolayısıyla, sokaklarda minimum zaman harcayarak seyahat etmek, özellikle kar küreme, çöp toplama ve devriye gezen güvenlik araçlarının rotalanması gibi HÇPP'nin pratikteki birçok uygulama alanında yol güvenliğinin ve trafik akışının sağlanması açısından büyük önem arz etmektedir. Bu çalışmada Zaman-Bağımlı Hiyerarşik Çinli Postacı Problemi (ZB-HÇPP) olarak adlandırılan ve toplam seyahat süresini en aza indirmeyi amaçlayan yeni bir problem türü ele alınmıştır. Problemin çözümüne yönelik ilk olarak, toplam katedilen mesafe ve toplam harcanan seyahat süresinin en küçüklenmesini amaçlayan karma tamsayı matematiksel programlama modelleri geliştirilmiştir. Ayrıca tanımlanan problemin büyük boyutlu problemlerde çözülebilmesi için iki meta-sezgisel algoritma önerilmiştir. Bunlar: Genetik Algoritma (GA) ve hibrit Tavlama Benzetimidir (hTB). Önerilen matematiksel modeller ve meta-sezgisel algoritmalar rasgele oluşturulmuş test problemleri üzerinde incelenerek etkinlikleri gösterilmiştir. Bu çalışma kapsamında matematiksel modeller GAMS 24.2.3'te Cplex 12 karma tam sayılı programlama çözücüsü kullanılarak çözülmüş, meta-sezgisel algoritmalar ise Matlab 2016a programlama dili kullanılarak kodlanmış ve sonuçları raporlanmıştır.

**2018, 122 sayfa**

**Anahtar Kelimeler:** Ayrıt rotalama problemi, Çinli postacı problemi, hiyerarşik Çinli postacı problemi, meta-sezgisel algoritmalar, zaman-bağımlı rotalama problemi

## ABSTRACT

Ph.D. Thesis

### A TIME-DEPENDENT HIERARCHICAL CHINESE POSTMAN PROBLEM AND SOLUTION PROPOSALS

Merve KAYACI ÇODUR

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

Supervisor: Assist. Prof. Dr. Mustafa YILMAZ

The Hierarchical Chinese Postman Problem (HCPP) is a type of the Chinese Postman Problem (CPP) which is one of the arc routing problems. It is seen that the distances between the nodes are taken into consideration and the travel times are ignored in the studies about HCPP which is NP-hard problem. However the speed (and time) of the vehicles traversing in almost all metropolitan areas changes during the day due to a variety of different factors such as peak traffic hours, traffic accidents and weather condition along with the distance. For this reason, travelling on the streets by spending minimum time is of great importance in terms of ensuring road safety and traffic flow particularly in many practical implementation areas of HCPP, such as routing of snow plowing, garbage collection and security patrol vehicles. In this study, a new problem type called the Time-Dependent Hierarchical Chinese Postman Problem (TD-HCPP) that aims to minimize the total travel time is introduced. Firstly, mixed integer mathematical models have been developed to minimize total traveled distance and total traveled time. For the large scale problems two meta-heuristics were proposed. These are: Genetic Algorithm (GA) and hybrid Simulated Annealing (hSA). The proposed models and meta-heuristics were tested on randomly generated problem instances and the effectiveness of these models and algorithms were shown. In this study, mathematical models are solved by Cplex 12 mixed integer programming solver in GAMS 24.2.3, meta-heuristic algorithms are coded in Matlab 2016a programming language and results are reported.

**2018, 122 pages**

**Keywords:** Arc routing problem, Chinese postman problem, hierarchical Chinese postman problem, meta-heuristic algorithms, time-dependent routing problem

## TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren, doktora eęitimim boyunca ilminden faydalandıęım, her konuda büyük yardımda bulunan deęerli hocam Sayın Dr. Öğr. Üyesi Mustafa YILMAZ'a,

Deęerli zamanlarımı ayırarak tezi inceleyen, öneri ve yönlendirmeleri ile katkı saęlayan tez izleme komitesindeki hocalarım Sayın Prof. Dr. Cafer ÇELİK'e ve Sayın Prof. Dr. Ömer GÜNDOęDU'ya

Bu zorlu yolculuęun her anında yanımda olan, bu günlere eriőmemdeki en önemli kiőiler olan deęerli anneme, babama, ablama, eőime ve canım kızım Zeynep Erva'ya en içten teőekkürlerimi sunarım.

**Merve KAYACI ÇODUR**

**Aralık, 2018**

## İÇİNDEKİLER

ÖZET.....	i
ABSTRACT .....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ .....	vi
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ .....	xi
1. GİRİŞ.....	1
<b>2. KAYNAK ÖZETLERİ .....</b>	<b>5</b>
2.1. Şebeke Problemleri.....	5
2.2. Ayrıt Rotalama Problemleri .....	9
2.2.1. Euler tur.....	9
2.2.2. Çinli Postacı problemi ve çeşitleri.....	11
2.2.3. Hiyerarşik Çinli postacı problemi .....	14
2.3. Zaman-Bağımlı Rotalama Problemleri .....	17
2.4. Ayrıt Rotalama Problemleri için Çözüm Yöntemleri.....	23
2.4.1. Kesin ve kesin olmayan yöntemler .....	23
2.4.2. Meta-sezgisel yöntemler.....	24
<b>3. MATERYAL ve YÖNTEM.....</b>	<b>27</b>
3.1. Ön Çalışma: ÇPP Yaklaşımı ile TCDD Bakım Çalışmalarının Planlanması.....	27
3.1.1. Problemin tanımlanması.....	29
3.1.2. ÇPP matematiksel modeli .....	35
3.2. Ön Çalışma: k-ÇPP Yaklaşımı ile Kar Küreme Araçlarının Rotalanması.....	36
3.2.1. Dk-ÇPP'nin tanımlanması.....	37
3.2.2. Dk-ÇPP matematiksel modeli .....	39
3.3. Hiyerarşik Çinli Postacı Problemi ve Matematiksel Modeli.....	42
3.4. Zaman-Bağımlı Hiyerarşik Çinli Postacı Problemi ve Matematiksel Modeli ....	49
3.5. ZB-HÇPP Çözümü için Önerilen Meta-sezgisel Yaklaşımlar .....	58
3.5.1. Çözüm gösterimi .....	59
3.5.2. Uygunluk fonksiyonu değerinin hesaplanması .....	63

3.5.2.a. Rota oluřturma.....	65
3.5.3. ZB-HÇPP için önerilen meta-sezgisel yaklaşım: genetik algoritma .....	70
3.5.3.a. Başlangıç popülasyonunun oluřturulması .....	72
3.5.3.b. Seçme operatörü .....	73
3.5.3.c. Çaprazlama operatörü.....	75
3.5.3.d. Mutasyon operatörü.....	76
3.5.4. ZB-HÇPP için önerilen meta-sezgisel yaklaşım: hibrit tavlama benzetimi .....	79
3.5.4.a. Açgözlü arama algoritması .....	79
3.5.4.b. Tavlama benzetimi algoritması .....	81
<b>4. ARAŐTIRMA BULGULARI ve TARTIŐMA.....</b>	<b>87</b>
4.1. Uygulama Çalışması.....	87
4.1.1. TCDD ray bakım çalışmalarında kullanılan yol ve katener ölçüm aracının optimum rotasının bulunması .....	87
4.1.2. Atatürk Üniversitesi Kampüsü kar ile mücadele araçlarının rotalanması .....	89
4.2. Teorik Çalışma .....	94
4.2.1. Test problemlerinin oluřturulması.....	94
4.2.2. HÇPP matematiksel model çözümü ile elde edilen sonuçlar .....	95
4.2.3. ZB-HÇPP çözümü ve sonuçların deęerlendirilmesi .....	98
<b>5. SONUÇ ve ÖNERİLER.....</b>	<b>108</b>
KAYNAKLAR .....	113
EKLER .....	120
EK 1.....	120
EK 2.....	121
ÖZGEÇMİŐ .....	123



## SİMGELER ve KISALTMALAR DİZİNİ

$\ E_h\ $	$h$ . hiyerarşik sınıfa ait toplam yol sayısı
$B_{ij}$	$(i,j)$ ayrıtının bağlantı matrisi
$C_{ij}$	$(i,j)$ ayrıtının mesafe matrisi
$E$	Yönsüz ayrıtlar kümesi
$E_h$	$h$ . hiyerarşik sınıfa ait ayrıtlar kümesi
$G(V,E)$	$V$ düğümler, $E$ ayrıtlardan oluşan şebeke
$h$	Hiyerarşik seviye
$h(p_{son,i})$	$(p_{son,i})$ ayrıtının hiyerarşik seviyesi
$h(p_{son,j})$	$(p_{son,j})$ ayrıtının hiyerarşik seviyesi
$h_{mevcut}$	$R_{mevcut}$ dizisinin hiyerarşik seviyesi
$I$	İlişki matrisi
$i,j$	Düğümmler
$k$	Zaman dilimi
$K$	Zaman dilimleri kümesi
$L$	Her sıcaklıktaki iterasyon sayısı
$l_k$	$k$ zaman diliminin başlangıç zamanı,
$M$	Büyük pozitif bir sayı
$n$	Şebekedeki toplam düğüm sayısı
$n_{son}$	$R$ rota dizisinin en son düğümü
$O_{ij}$	$(i,j)$ ayrıtının öncelik matrisi
$P$	Kromozom dizisi
$p_c$	Çaprazlama oranı
$p_m$	Mutasyon oranı
$Pop$	Popülasyon
$Pop_{boyut}$	Popülasyon büyüklüğü
$p_{son}$	$R_{mevcut}$ dizisinin en son elemanı
$P_{uzunluk}$	$P$ kromozomunun uzunluğu
$R$	Nihai rota
$R_{dönüş}$	$R_{mevcut}$ dizisinin en son elemanından başlangıç düğümüne dönüş rotası

$R_{mevcut}$	Tamamlanmamış R rota dizisi
$R_T$	Geçici rota dizisi
$S_{başlangıç}$	Tavlama benzetimi algoritmasının başlangıç çözümü
$S_C$	P kromozomunun son elemanı ile direkt bağlantılı işler kümesi
$S_{komşu\_çözüm}$	Tavlama benzetimi algoritmasında üretilen yeni komşu çözüm
$S_{mevcut\_çözüm}$	Tavlama benzetimi algoritmasının geçerli çözümü
$S_R$	Geçilmesi gerekli yollar kümesi
$S_T$	Öncülü olmayan işler kümesi
$t$	Adım sayısı
$t(p_{son}, i)$	$p_{son}$ düğümünden $i$ düğümüne seyahat süresi
$t(p_{son}, j)$	$p_{son}$ düğümünden $j$ düğümüne seyahat süresi
$t^{(R)}$	R rotasının toplam seyahat süresi
$T_0$	Başlangıç sıcaklığı
$T_{son}$	Bitiş sıcaklığı
$u_k$	$k$ zaman diliminin bitiş zamanı
$V$	Düğüm kümesi
$v_{ijk}$	$(i, j)$ yolundan $k$ . zaman aralığında geçen aracın seyahat hızı
$w_0$	Seyahat başlangıç zamanı
$Z$	İşler kümesi
$Z(pt)$	Öncül işler kümesi
$\alpha$	Soğutma oranı
$\delta(i)$	$i$ düğümünden çıkan bütün yollar kümesi
$\delta_h(i)$	$i$ düğümünden çıkan $E_h$ kümesine ait yollar kümesi
$\tau^{(ij)}$	$(i, j)$ ayırıtına atanmış iş numarası

### Kısaltmalar

ARP	Araç Rotalama Problemi
ÇPP	Çinli Postacı Problemi
Dk-ÇPP	Dengeli k Çinli Postacı Problemi
FIFO	İlk Giren İlk Çıkar (First-In-First-Out)

GA	Genetik Algoritma
GSP	Gezgin Satıcı Problemi
HÇPP	Hiyerarşik Çinli Postacı Problemi
hTB	Hibrit Tavlama Benzetimi
k-ÇPP	k Çinli Postacı problemi
KPP	Kırsal Postacı Problemi
NP	Non-polinomial
TB	Tavlama Benzetimi
TCDD	Türkiye Cumhuriyeti Devlet Demiryolları
ZB-ARP	Zaman-Bağımlı Araç Rotalama Problemi
ZB-GSP	Zaman-Bağımlı Gezgin Satıcı Problemi
ZB-HÇPP	Zaman-Bağımlı Hiyerarşik Çinli Postacı Problemi
ZBHF	Zaman Bağımlı Hız Fonksiyonu
ZBKUSF	Zaman Bağımlı Kesikli Ulaşım Süresi Fonksiyonu

## ŞEKİLLER DİZİNİ

Şekil 2.1. Örnek bir şebeke gösterimi.....	5
Şekil 2.2. Şebeke türleri .....	6
Şekil 2.3. Yol ve çevrim gösterimi .....	6
Şekil 2.4. Örnek bir şebeke ve minimum yayılan ağacı .....	7
Şekil 2.5. Königsberg köprü problemi .....	9
Şekil 2.6. Königsberg'deki yedi köprünün şebeke gösterimi .....	10
Şekil 2.7. Euler tur içeren şebeke gösterimi .....	10
Şekil 2.8. Euler tur içermeyen şebeke gösterimi .....	11
Şekil 2.9. $(i,j)$ ayrıtı için kesikli seyahat süresi fonksiyonu .....	19
Şekil 2.10. Örnek bir seyahat hız fonksiyonu .....	20
Şekil 2.11. Örnek bir seyahat süresi fonksiyonu .....	21
Şekil 3.1. Türkiye demiryolu haritası .....	31
Şekil 3.2. Bölgelerin semboller ile gösterimi.....	32
Şekil 3.3. Türkiye demiryolu konvansiyonel ray hattının şebeke gösterimi.....	33
Şekil 3.4. Atatürk Üniversitesi doğu kampüsünün haritası.....	38
Şekil 3.5. Atatürk Üniversitesi kampüs haritasının şebeke gösterimi .....	39
Şekil 3.6. Hiyerarşik şebeke örneği-1 .....	42
Şekil 3.7. Hiyerarşik şebeke örneği-2 .....	43
Şekil 3.8. Seyahat süresi hesaplama algoritması .....	53
Şekil 3.9. Örnek bir çözüm gösterimi .....	63
Şekil 3.10. Uygunluk değeri hesaplama algoritması .....	64
Şekil 3.11. Rota oluşturma algoritması.....	69
Şekil 3.12. GA akış diyagramı .....	71
Şekil 3.13. Başlangıç popülasyonu oluşturma algoritması .....	72
Şekil 3.14. ZB-HÇPP için örnek bir başlangıç popülasyon gösterimi.....	73
Şekil 3.15. Çaprazlama işlemi .....	76
Şekil 3.16. Bir popülasyonun evrimleşme süreci algoritması.....	78
Şekil 3.17. Komşu çözümlerin üretilmesi.....	84
Şekil 3.18. hTB algoritmasının adımları.....	86

<b>Şekil 4.1.</b> k-ÇPP modelinin farklı test problemlerine ilişkin çözüm sonuçları.....	92
<b>Şekil 4.2.</b> Dk-ÇPP modelinin farklı test problemlerine ilişkin çözüm sonuçları.....	92
<b>Şekil 4.3.</b> Araçlar arası gidilen mesafe farkları (k-ÇPP/Dk-ÇPP).....	93



## ÇİZELGELER DİZİNİ

Çizelge 2.1. ÇPP ve çeşitleri ile ilgili yapılan çalışmalar .....	24
Çizelge 3.1. 2015 yılı ilkbahar dönemi bölgelere ait uygulanan araç güzergâhı .....	34
Çizelge 3.2. Dk-ÇPP matematiksel model parametre ve değişkenler .....	40
Çizelge 3.3. Kısıt (3.21) açık gösterimi .....	48
Çizelge 3.4. ZB-HÇPP için kullanılan hız matrisi .....	51
Çizelge 3.5. ZB-HÇPP modelinde kullanılan kümeler, parametreler ve karar değişkenleri .....	54
Çizelge 3.6. Şebekenin iş çizelgeleme problemine dönüştürülmesi .....	61
Çizelge 3.7. Tavlama sürecinin kombinatoriyal optimizasyon problemlerinde karşılığı .....	82
Çizelge 4.1. Bölgelere ait optimum rota uzunlukları .....	88
Çizelge 4.2. Problemin çözümünde k-ÇPP matematiksel modeli ile elde edilen optimum rotalar .....	90
Çizelge 4.3. Problemin çözümünde Dk-ÇPP matematiksel modeli ile elde edilen optimum rotalar .....	90
Çizelge 4.4. k-ÇPP ve Dk-ÇPP modeli ile elde edilen test problem sonuçları .....	91
Çizelge 4.5. Her iki model sonuçlarına göre araçlar arası katedilen mesafe farkları .....	93
Çizelge 4.6. Geliştirilen HÇPP matematiksel modeli ile elde edilen sonuçlar .....	96
Çizelge 4.7. 10 düğümlü HÇPP test problemlerinin optimum rotaları .....	97
Çizelge 4.8. GA parametre değerleri .....	99
Çizelge 4.9. hTB parametre değerleri .....	99
Çizelge 4.10. ZB-HÇPP matematiksel model, GA ve hTB çözüm sonuçları .....	101

## 1. GİRİŞ

Ulaşım sektörü, hem gelişmiş hem de gelişmekte olan ülkeler için yüksek hızla artan ulaşım talebini karşılaması bakımından ekonomik faaliyetlerin en önemli unsurlarından biri haline gelmiştir. İster kamu ister özel sektör olsun hemen hemen her organizasyonun taşıma ile ilgili işleri vardır. Bir şehrin belediyesi tarafından çöplerin toplanması, caddelerdeki karların temizlenmesi gibi faaliyetler kamu sektörü için verilebilecek taşıma örneklerindedir. Üretim yapan bir şirketin hammadde tedarikiğinden son ürünün kullanıcıya ulaştırılmasına kadar geçen süreçte de yine birçok taşıma faaliyeti vardır. Şirketlerin daha iyi rekabet edebilmesini sağlamak ve kamu sektöründeki verimi artırmak için taşıma yönetiminin iyi bir şekilde organize edilmesi gerekmektedir.

Bu tür problemlerde araçların görevlerini etkin bir şekilde yerine getirmesi için hangi rotayı izlemeleri gerektiğinin belirlenmesi maliyet ve zaman tasarrufu açısından büyük önem arz etmektedir. Çeşitli amaçlarla görevli kişilerin veya araçların izlemeleri gereken iş sırasını tanımlayan problemlere genel olarak rotalama problemi denilmektedir. Rotalama birçok üretim ve servis sektöründe önemli bir karar verme sürecidir. Kamu ve özel işletmelerin bu alanlardaki harcamaları her geçen gün artmakta ve büyük miktarlara ulaşmaktadır. Yetersiz planlamalar ve hatalı yatırımlar neticesinde önemli miktarlarda kaynak ve zaman bu alanlarda israf olmaktadır. Bu sebeple, bu tip problemlerin önemi giderek artmakta ve birçok araştırmaya konu olmaktadır. Bu alanlarda yapılan çalışmalar, daha etkin çözümler bulunarak uygulama imkânının artmasına ve böylece önemli miktarda tasarrufların sağlanmasına yardımcı olmaktadır (Emel vd 2004).

Son elli yılda yöneylem araştırması alanında yapılan çalışmalarda rotalama problemlerine olan ilginin sürekli arttığı görülmektedir. Rotalama problemleri düğüm ve ayrıt rotalama problemleri olarak iki sınıfa ayrılmaktadır. Düğüm rotalamada çeşitli kısıtlar altında bir şebekenin düğümleri ile ilgilenilirken, ayrıt rotalama problemlerinde ise şebekenin ayrıtları dikkate alınır. Ayrıt rotalama problemleri ile ilgili yapılan en yaygın çalışmalar Çinli Postacı Problemi (ÇPP) ve Kırsal Postacı Problemidir (KPP). ÇPP, verilen bir şebekede bütün yollardan en az bir kez geçecek şekilde en kısa tur/turların bulunmasını

amaçlar (Corberan *et al.* 2002). KPP’de ise amaç, verilen bir şebekede sadece gerekli görülen yollardan en az bir kez geçecek şekilde en kısa tur/turların bulunmasıdır. ÇPP; mektupların dağıtımı, çöplerin toplanması, cadde ve otobanlarda kar ve buz ile mücadele çalışmaları, sokakların temizlenmesi çalışmaları, okul servisleri ve polis devriye araçların rotalarının çizelgelenmesi, su ve gazete dağıtımı ve etkili web sitesi kullanılabilirliğinin tespiti gibi birçok alanda kullanılabilmektedir (Thimbleby 2003).

Kar küreme, çöp toplama ve tuz serpmeye gibi bazı uygulamalarda ise yollar, öncelik ilişkilerine göre kısıtlandırılabilir. Bu problem, literatürde ÇPP’nin bir türü olan Hiyerarşik Çinli Postacı Problemi (HÇPP) olarak adlandırılır. HÇPP’de amaç, verilen bir hiyerarşik şebekede yolların öncelik ilişkileri göz ardı edilmeden, her bir yoldan en az bir kez geçecek şekilde en kısa tur/turların bulunmasıdır.

Bu doktora tezi kapsamında HÇPP’ye odaklanılmıştır. HÇPP ile ilgili yapılan çalışmalar, gerçek hayat uygulamalarında sıkça karşılaşılmamasına rağmen literatürde oldukça sınırlı sayıdadır. Dahası, gerçek yaşam koşullarını yakından temsil eden çözüm yaklaşımlarının eksikliği ise halen vardır. İlk kez 1987 yılında Dror *et al.* tarafından önerilen HÇPP için yapılan çalışmaların tamamında düğümler arası mesafelerin dikkate alındığı görülmektedir. Ancak, şehir içi ulaşımda gün içerisinde seyahat eden araçların hızı ve buna bağlı olarak da ulaşım süreleri aracın yola çıkış zamanına ve kullandığı yola bağlı olarak değişiklik göstermektedir. Çalışmalarda nadiren ele alınan bu husus, düğümler arası seyahat zamanlarının, günün zamanına bağlı olduğu “zaman bağımlılığı” kavramı olarak adlandırılmaktadır.

Son yıllarda günün farklı zamanlarında seyahat zamanı ya da seyahat hızı bilgisinin bulunması, araştırmacıların bu ek bilgidен yararlanarak farklı rotalama modelleri geliştirmelerine yol açmıştır. Zamana bağlı seyahat sürelerinin modellenmesi konusu, zaman bağımlı rotalama literatüründe önemli oranda dikkat çeken bir konu haline gelmiştir. Zira tüm metropol bölgelerdeki seyahat hızları (ve süre) gün boyunca hava koşulları, trafik sıkışıklığı, trafik kazaları gibi çeşitli faktörler tarafından büyük ölçüde değişmektedir. Özellikle kış aylarında kar küreme işlemi, çöp toplama işlemi, polis



devriye araçlarının rotalanması gibi HÇPP'nin birçok pratik uygulama alanında, caddelerden geçişlerin en kısa zamanda yapılması, yakıt ekonomisi, trafik akışı ve yol güvenliğinin sağlanması açısından büyük önem taşımaktadır.

Literatürde HÇPP için ağırlıklı olarak sezgisel yöntemlere ve 2000'li yıllardan sonra ise kesin çözüm veren yöntemlere başvurulduğu görülmektedir. Ancak rotalama problemlerinde sıklıkla başvuru yapılan tam sayılı programlama temelli tekniklere ve meta-sezgisel yöntemlere ise HÇPP çözümü için başvurulmadığı görülmüştür. Tam sayılı programlama, esnek yapısından dolayı modele kısıt ekleyip çıkartılarak veya mevcut kısıtlar üzerinde ağırlıklandırma gibi düzenlemeler yapılarak, problemde meydana gelebilecek değişikliklere çok rahat uyum sağlayabilmektedir. Kesin çözüm veren bu yaklaşım NP-zor problemlerde sadece çok küçük boyutlar için çözüm verebilmekte olup problem boyutu arttıkça çözüm kalitesi ve zamanı bakımından yetersiz kalmaktadır. Klasik yöntemlerin yetemediği bu NP-zor yapıdaki optimizasyon problemlerinin çözümü için büyüyen çözüm uzayını daha hızlı tarayarak uygun çözümler üreten meta-sezgisel yöntemlere sıklıkla başvurulmaktadır. Bu yöntemler kesin çözümü garanti etmezler, ancak kabul edilebilir zamanda kaliteli çözümler verirler.

Bu çalışma kapsamında zaman bağımlı seyahat süresi kısıtları altında HÇPP ele alınarak çözüm önerileri sunulmuş ve literatürde tespit edilen boşluk belirli ölçüde doldurulmaya çalışılmıştır.

Çalışma 5 bölümden meydana gelmektedir. Takip eden bölümde konunun anlaşılabilirliğini artırmak üzere önce şebeke problemleri ile ilgili temel kavramlara açıklık getirilmiştir. Ayrıt rotalama problemleri sınıflandırılarak ele alınmış ve literatürde sıklıkla çalışılan ÇPP ve bunun bir çeşidi olan HÇPP'ye değinilmiştir. Daha sonra seyahat süreli yaklaşımlar içeren rotalama problemleri ile ilgili çalışmalar taranmış, içerdikleri kısıtlar ve kullandıkları seyahat süresi fonksiyonlarına göre sınıflandırılarak sunulmuştur. Son olarak ayrıt rotalama problem çeşitlerinde ele alınan çözüm yöntemleri ile ilgili literatür bilgisi sunulmuştur.

Tezin omurgasını oluşturan üçüncü bölümde alt başlıklar halinde sırasıyla; ÇPP'nin gerçek hayat problemlerine uygulanabilirliğini belirlemek ve ele alınan probleme etkin çözümler geliştirmek üzere iki ön çalışma yapılmış, HÇPP için önerilen tam sayılı programlama yaklaşımının işleyişinden bahsedilmiş, zaman-bağımlı varyasyonu ile genişletilmiş HÇPP için geliştirilmiş tam sayılı model sunulmuş ve son olarak tasarlanan Genetik Algoritma (GA) hibrit Tavlama Benzetimi (hTB) detaylı bir şekilde açıklanmıştır.

Çalışmanın dördüncü bölümünde, araştırma bulguları ve tartışmalar yer almaktadır. Öncelikle literatürde sunulan ÇPP modeli ile Türkiye'deki tren ray hatlarında bakım yapan aracın en kısa rotası belirlenmiş ve mevcut uygulanan durum ile karşılaştırılarak etkinliği gösterilmiştir. Birden fazla aracın olduğu uygulamalar için araçların katettikleri mesafelerin dengelenmesi amacıyla dengeli k-ÇPP önerilmiş ve modelin etkinliği gerçek hayat problemleri üzerinde gösterilmiştir. Yine aynı bölümde, tezin omurgasını oluşturan HÇPP ele alınarak, önerilen matematiksel model ile küçük boyutlu test problemleri üzerinde sonuçları verilmiştir. Çalışmanın devamı olarak günün farklı zaman dilimlerinde değişen seyahat hızlarının (dolayısıyla sürelerin) dikkate alındığı HÇPP'nin bir uzantısı olan Zaman-Bağımlı Hiyerarşik Çinli Postacı Problemi (ZB-HÇPP) matematiksel modeli sunulmuştur. Matematiksel modelin orta ve büyük boyutlu test problemleri üzerinde etkinlik gösterememesinden dolayı probleme iki farklı meta-sezgisel yöntem ile çözüm aranmıştır. Önerilen GA ve hTB meta-sezgisel yöntemler ile küçük, orta ve büyük test problemlerinin tamamı için çözüm sonuçları raporlanmıştır. Sonuçlar çözüm kalitesi ve bilgisayar zamanı açısından değerlendirilmiştir.

Beşinci bölümde ise yapılan çalışma ve elde edilen sonuçlar ile ilgili değerlendirmelere yer verilmiştir. Ulaşılan bilgi ve deneyimler esas alınarak gelecekte yapılabilecek çalışmalardan bahsedilmiş ve öneriler yapılmıştır.

## 2. KAYNAK ÖZETLERİ

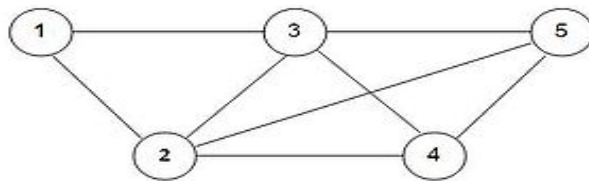
Bu bölüm dört alt başlık altında toplanmıştır. İlk alt başlıkta şebeke problemleri ve türleri ile ilgili genel bilgiler verilmiştir. İkinci alt başlıkta ayrıt rotalama problemlerinden biri olan ÇPP, temel kavramları, çeşitleri ve tez kapsamından ele alınan HÇPP ile ilgili genel bilgiler ve literatür incelemesi verilmiştir. Rotalama problemleri çeşitli sınıflandırmalar altında çok sayıda araştırmacı tarafından incelenmiştir. Üçüncü alt başlıkta zaman-bağımlı seyahat sürelerinin dikkate alındığı rotalama problemleri ve çeşitleri ile ilgili yapılmış çalışmaların ana fikirleri, metotları ve sonuçları sentezlenerek özetleri sunulmuştur. Daha sonra, tez ile ilişkili olarak içeriğinde ayrıt rotalama problemi barındıran çalışmalar taranarak kullanılan çözüm yöntemleri hakkında detaylı bilgiler verilmiştir.

### 2.1. Şebeke Problemleri

Şebeke, düğümler ve bu düğümleri birbirine bağlayan kenarlardan oluşan bir tür ağ yapısıdır. Matematik ve bilgisayar biliminde kullanılan bu teori, bir toplulukta bulunan nesnelere arasındaki ilişkileri modelleyen matematiksel yapıları inceler. Şebeke problemleri, yöneylem araştırmalarında belirlenmiş bir sıra problemin dallarla birbirine bağlanan düğümler olarak bir şebeke halinde modellenip çözülebilen problem türleridir (Taha 2002). Bir  $G$  şebekesi  $(V,E)$  notasyonlarıyla ifade edilir. Burada  $V$  düğümler kümesini,  $E$  ise yönsüz ayrıtlar kümesini temsil etmektedir. Şekil 2.1’de verilen şebeke aşağıdaki gibi tanımlanır:

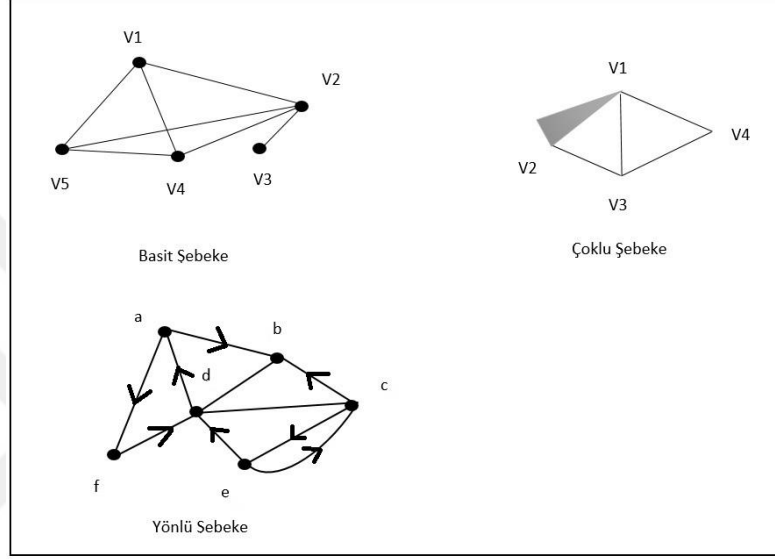
$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{(1, 3), (1, 2), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)\}$$



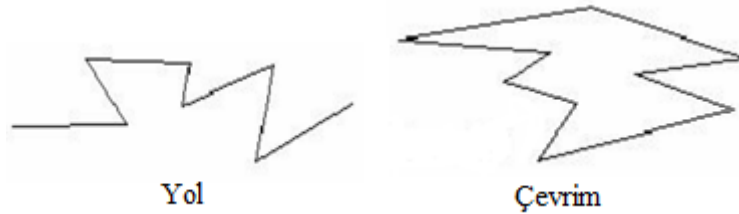
Şekil 2.1. Örnek bir şebeke gösterimi

Her şebekenin kendine özgü bir akış tipi vardır. Şebekedeki akış, bağlantıların sonlu veya sonsuz kapasiteli olması ve bağlantıların yönsüz (çift yönlü) ya da yönlü şebeke olmasına göre ayrılırlar. Şekil 2.2’de basit şebeke, yönlü şebeke ve çoklu şebeke örnekleri verilmiştir.



**Şekil 2.2.** Şebeke türleri

Yol, her bir daldaki akışın yönüne bakılmaksızın iki düğümü birleştiren ayrı dalların sırasidir. Bir yol  $n$  uzunluğunda ise,  $n+1$  adet düğümü ve  $n$  adet ardışık kenarı vardır. Çevrim, bir döngü içeren yolun başladığı düğümde son bulmasıdır. Uzunluğu  $n$  olan bir döngüde  $n$  adet düğüm vardır. Şekil 2.3’te örnek bir yol ve çevrim gösterilmiştir.

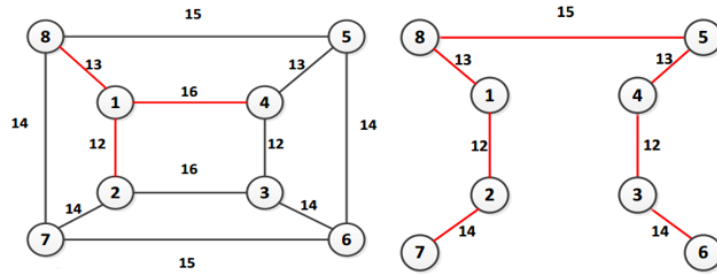


**Şekil 2.3.** Yol ve çevrim gösterimi

Şebeke problemlerinin çözümü için ortaya çıkarılan özel şebeke problemi algoritmaları vardır. Bu tür çalışmalarda önce problemin öğeleri ve amacı tarif edilir, daha sonra problemin şekil veya matris olarak dallar ile birbirlerine bağlı düğümler halinde yapılandırılıp tanımlanması gerekir. En çok kullanılan ve iyi geliştirilmiş özel şebeke modeli algoritması bulunan problemler aşağıda verilmiştir (Emel vd. 2004).

1. Minimum kapsayan ağaç problemi ve algoritmaları
2. En kısa yol problemi ve algoritmaları
3. Maksimum akış problemi ve algoritmaları
4. Minimum maliyet kapasiteli akış problemi ve algoritmaları
5. Kritik yol problemi ve algoritmaları

Minimum kapsayan ağaç problemi, doğrudan veya dolaylı olarak dalların en kısa bağlantılısını kullanarak şebekenin dallarının birbiriyle ilişkilendirilmesini ele alır. Ağaç yönsüz bir şebekede tüm düğümlerin oluşturduğu çevrimsiz bağlantı kümesi olup minimum kapsayan ağaç probleminde şebeke üzerindeki tüm düğümleri birbirine bağlayan her kenarın bir ağırlığa sahip olması (mesafe veya maliyet olarak düşünülebilir) gerekir. Minimum kapsayan ağaç probleminde amaç, minimum toplam ağırlık ile tüm düğümleri birbirine bağlayan yol ağacını belirlemektir. Şekil 2.4'te verilen bir şebekenin yol ağacı gösterilmiştir.



**Şekil 2.4.** Örnek bir şebeke ve minimum yayılan ağacı

Körfezdeki kuyuları iç kesimlerdeki teslim noktalarına bağlayan kıyıdaki doğalgaz boru hattı projesinin tasarımı bu probleme örnek verilebilir. Modelin amacı boru hattı inşaat

maliyetlerinin minimum kılınmasıdır. İki şehri birbirine bağlayan yolların yapımının gerçekleştirilmesi de yine verilebilecek örneklerdendir. Zira yol sisteminin ekonomik tasarımlarından biri yolların toplam uzunluğunun minimum kılınmasıdır ve minimum kapsayan ağaç algoritmasının uyarlanmasıyla istenen sonuç elde edilebilir.

En kısa yol probleminde ulaştırma şebekesinde istenilen, kaynak nokta ile varış nokta arasındaki en kısa yolun bulunmasıdır. Amaç, şebeke üzerindeki iki düğüm noktası arasındaki en kısa uzunluğu ve rotayı bulmaktır. Bir şebeke üzerinde bir noktadan başka bir noktaya ulaşmak için birden fazla yol mevcut olabilir. En kısa yol problemlerinde amaç, bu alternatif yolların içerisinde maliyetçe (genelde mesafe dikkate alınır) en düşük olan yolları bulmaktır.

Maksimum akış problemi, herhangi bir ok üzerindeki kapasiteyi aşmadan başlangıç düğümünden çıkan ve düğüm n'e gelen mümkün olan toplam maksimum akış miktarının tespitidir. Kömür madenini enerji santraline bağlayan, kömürü su içerisinde taşıyarak nakleden boru hattı ağının maksimum kapasitesinin belirlenmesi (bu boru hattı, kömürü özel tasarlanmış borular içerisinde su pompalayarak taşır) problemi maksimum akış problemi örnek verilebilir.

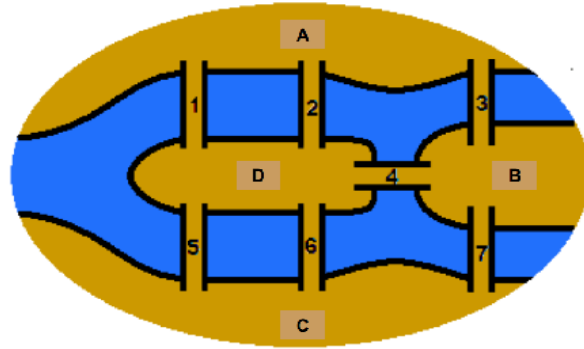
Minimum maliyet kapasiteli akış problemi maksimum akış modelinin genel halidir. Bu problemde amaç, bağlantılardaki akış sınırlamalarını ve düğümlerdeki üretim ve talepleri karşılayacak farklı bağlantılardaki akışları belirlemektir. Petrol kuyularından rafinerilere ham petrol taşıyan petrol boru hattı şebekesi örnek verilebilir. Her boru parçası sonlu maksimum hızda petrol akışına sahiptir.

Bu şebeke problemlerinin birçoğu doğrusal programlama, tamsayı programlama, dinamik programlama problemi olarak da formüle edilip çözülebilirler. Bununla birlikte, önerilen şebeke esaslı algoritmalar simpleks vb. yöntemlerden daha etkilidir.

## 2.2. Ayrıt Rotalama Problemleri

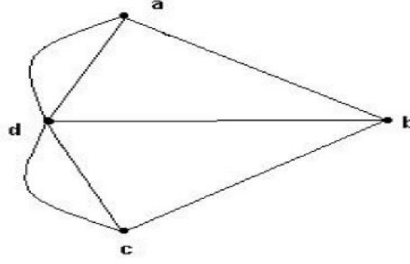
### 2.2.1. Euler tur

Euler tur, İsviçreli bir matematikçi olan Leonhard Euler tarafından 1736 yılında ortaya atılmıştır. Euler çevrimi basit bir çevrim olup bir şebeke içerisindeki her kenardan sadece bir kez geçilmesine izin verilir. Aslında Euler turun ortaya çıkmasında “Königsberg Köprü Problemi” temel oluşturmuştur. Bu probleme göre Königsberg şehrinin tam ortasından eski ve yeni Pregel nehirlerinin birleştiği Pregel (Pregolya) nehri geçmektedir. Bu nehir, Şekil 2.5’te görüldüğü gibi şehri A, B, C ve D olmak üzere dört bölgeye ayırmakta ve bu dört bölge ise yedi köprü ile birbirine bağlanmaktadır (Harary 1960).



**Şekil 2.5.** Königsberg köprü problemi

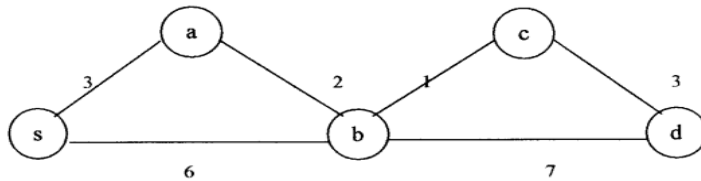
Buradaki problem, herhangi bir noktadan başlayarak yedi köprüünün her birinden sadece bir kez geçmek koşuluyla bütün noktaları ziyaret ettikten sonra başlangıç noktasına geri dönülüp dönülemeyeceğidir. Bu problem Euler tarafından araştırılmış ve Şekil 2.6’daki gibi problem bir şebeke ile gösterilmiştir (Emel vd 2004). Şebekede şehrin dört bölümü düğümlerle ve köprüler ise ayrıtlarla temsil edilmiştir.



**Şekil 2.6.** Köningsberg'deki yedi köprünün şebeke gösterimi

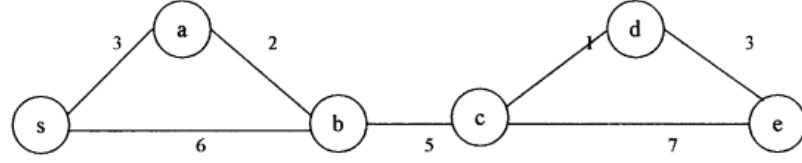
Euler incelemelerinde böyle bir yolun oluşmasının mümkün olmadığını göstermiştir. Euler tur ile bir şebekedeki kenarların tamamından yalnızca bir kez geçilir. Yönsüz bir grafta bir düğümün derecesi düğüme gelen kenarların toplamıdır.

Euler, böyle bir turun oluşabilmesi için her bir bölgeye diğer bölgelerden ulaşılabilirliğin olması ve her bir düğümün çift dereceli olması gerektiğini ortaya koymuştur (Ahuja and Orlin 2002). Her düğümün derecesi o düğüme giren ve çıkan yolların sayısı ile hesaplanır. Çift dereceli bir düğümden o düğüme bağlı yol sayısı çifttir. Zira bir düğüm, başlangıç ya da bitiş düğümü değilse o düğüme gelen kişinin turu devam ettirebilmesi için oradan ayrılması gerekecektir. Dolayısıyla bu tip düğümlerin dereceleri çift olmalıdır. Özetle, bir şebekede tüm köşe noktalarının derecesi çift ise, bu şebeke bir Euler şebekedir ve çizilebilir. Oysa tek dereceli bir düğüme ikinci kez gelen bir kişi çıkış yolu bulamayacaktır. Dolayısıyla tek dereceli düğümler gezintinin başlangıç ve bitiş düğümleri olarak seçilmelidir. Bir grafta iki köşe noktasının derecesi tek sayı ise bu graf çizilebilir ancak başlangıç ve bitiş noktaları farklılık gösterir (Yarı-Euler graf). Eğer bir graftaki tüm köşe noktalarının derecesi tek sayı ise bu graf çizilememektedir. Şekil 2.7 ve Şekil 2.8'de sırasıyla Euler tur içeren ve Euler tur içermeyen şebeke örnekleri verilmiştir.



**Şekil 2.7.** Euler tur içeren şebeke gösterimi





**Şekil 2.8.** Euler tur içermeyen şebeke gösterimi

### 2.2.2. Çinli Postacı problemi ve çeşitleri

Ayrıt rotalama problemlerinden biri olan ÇPP, ilk olarak 1962 yılında Çinli bir matematikçi olan Mei-Ko Kwan tarafından ortaya atılmıştır. Problem, bir postacının postaneden aldığı mektupları şehirdeki tüm sokaklara uğrayacak şekilde mümkün olan en kısa yoldan dağıtmak istemesiyle ortaya çıkmıştır. Postacı mektupların dağıtımından sonra başladığı nokta olan postaneye geri dönmek zorundadır (Ahuja *et al.* 1993).

Çinli postacı probleminin çözümünde Euler çevrimi temel oluşturmaktadır. Edmonds and Johnson (1973) ÇPP için optimum çözüm bulan bir algoritma geliştirmişlerdir. Geliştirdikleri algoritma minimum eşleştirme algoritması (minimum matching algorithm) olup bir şebekedeki bütün tek derece köşeleri çift dereceye dönüştürür. Bu aşamada, bütün tek dereceli köşelere yapay yollar eklenir. Modifiye edilmiş yeni şebeke Euler çevrimi oluşturup ÇPP için optimum çözümü vermektedir. Ayrıca yaptıkları çalışmada ÇPP için alt sınır değerinin Euler tur olduğunu göstermişlerdir. Minieka (1978) ise yaptığı çalışmada bir optimum postacı problem turunda toplam uzunluğun her başlangıç düğümü için aynı olduğunu göstermiştir.

ÇPP, kombinatoryal eniyileme problemlerinden biri olan Gezgin Satıcı Problemine (GSP) benzerlik göstermesine rağmen önemli farklılıklara da sahiptir. GSP, şebekedeki her bir düğüme yalnızca bir kez uğramak koşuluyla en kısa turun (Hamilton turunun) bulunması ilkesine dayanan bir düğüm rotalama problemidir. ÇPP ve GSP arasındaki temel farklılık; ÇPP’de bir şebekenin düğümleri yerine bu düğümleri birbirine bağlayan ayrıtlardan geçilmesi ve bunun da en az bir kez gerçekleştirilmesi koşuludur (Eiselt *et al.*

1995). ÇPP’de eğer bir Euler tur elde edilemiyorsa turun tamamlanabilmesi için ayrıtlardan birden fazla geçilmesi gerekmektedir.

Bu tip problemler ile günlük hayatta otobüs çizelgeleme, yol bakımı, mektup dağıtımı, elektrik, su, doğalgaz sayaçlarının okunması, atık veya çöp toplama işlemleri, okul servisleri, süt ve gazete dağıtımı, kar temizleme ve devriye gezen araçların rotalarının belirlenmesi gibi pek çok uygulamasıyla sık sık karşılaşılabilmektedir.

ÇPP temel olarak, bir şebekedeki ayrıtların yönlerine göre yönlü, yönsüz ve karma ÇPP olarak üçe ayrılmaktadır. Yönlü ve yönsüz ÇPP polinom zamanlı algoritmalar ile çözülebilmektedirler. Bu sebeple her ikisinde P sınıfına ait problem türleridir. Karma ÇPP’de ise yolların bazıları yönlü, bazıları ise yönsüz olup bu problem NP-zor problem sınıfına aittir (Florian 1984). Yönsüz, yönlü ve karma ÇPP’lere başka kısıtların eklenmesi sonucu yeni ÇPP çeşitleri ortaya çıkmıştır. Bunlar aşağıdaki gibi verilebilir (Ahr and Reinelt 2002):

- Yönlü ÇPP (Directed CPP)
- Yönsüz ÇPP (Undirected CPP)
- Karma ÇPP (Mixed CPP)
- k-ÇPP (k-CPP)
- Min-Max k-ÇPP ( Min-Max k-CPP)
- Hiyerarşik ÇPP (Hierarchical CPP)
- Kapasite Kısıtlı ÇPP (Capacitated CPP)
- Hızlı ÇPP (Windy CPP)

Yönsüz ÇPP’de yönsüz ayrıtlardan oluşan bir şebekedeki her bir ayrıta en az bir kez uğrayacak şekilde en kısa turun bulunması amaçlanır. P sınıfına ait bu problem yönlü ÇPP’ye göre daha karmaşıktır. Ele alınan şebeke bir Euler şebeke ise problem Euler tur bulunarak çözülebilir ve tur, ayrıt tekrarı olmadan tamamlanabilir. Ancak Euler şebeke değil ise o zaman tekrarlanan ayrıtların toplam uzunluğu en küçüklenmeye çalışılır (Avriel and Golany 1996). Tekrarlanan ayrıtların toplam uzunluğunun en küçüklenmesi

problemin en iyi sonucunu verir. Problemin çözümünde kullanılan algoritma ve yöntemler; euler turu, dijkstra algoritması, floyd algoritması ve en kısa mesafeli eşleştirme olarak sayılabilir. Ayrıca diğer kombinatoryal en iyileme problemlerinde (gezgin satıcı problemi, araç rotalama problemi vb.) olduğu gibi, ÇPP'de tamsayılı doğrusal programlama yöntemleriyle çözülebilmektedir.

Yönlü ÇPP'de yönlü bir şebeke üzerindeki her ayrıta en az bir kere uğranılarak başlangıç düğümüne geri dönmek koşuluyla en kısa yolun bulunması istenmektedir (Ahuja *et al.* 1993). Problemin çözümünde kullanılan algoritma ve yöntemler, Edmonds and Johnson (1973) serim akış yöntemleri ve tamsayılı doğrusal programlama yöntemleri olarak sayılabilir.

Yönlü ve yönsüz ÇPP'nin birleşmesinden oluşan karma ÇPP'de, hem yönlü hem de yönsüz ayrıtlardan oluşan bir  $G=(V,E,A)$  şebeke üzerindeki tüm ayrıtlara en az bir kez uğrayarak ve başlangıç düğümüne geri dönmek koşuluyla en kısa turun bulunması amaçlanmaktadır (Corberan *et al.* 2002).

k-Çinli postacı için k tane (en az 2) postacı için rota belirlenmeye çalışılmaktadır. Bu problemlerde amaç k tane rotanın toplam uzunluğunu en küçük yapmaktır.

Min-max k-ÇPP, k-ÇPP ile hemen hemen aynıdır. Belirlenen k tane rota arasından en uzun olanı en küçüklenmek istenmektedir. Bu şekilde rotaların daha dengeli olması için yaklaşık eşit uzunluklu rotalar oluşturulmaya çalışılmaktadır (Ahr and Reinelt 2002).

Hızlı ÇPP'de yönsüz bir şebeke üzerindeki ayrıtların uzunluğu, bu ayrıtlardan geçiş yönüne göre değişebilmektedir. Burada da amaç, şebekedeki her bir ayrıttan en az bir kez geçmek koşuluyla en kısa turun bulunmasıdır (Eiselt *et al.* 1995).

Yönsüz kapasite kısıtlı ÇPP'de araçların kapasite kısıtlarında dikkate alınarak en kısa mesafeli turun bulunması amaçlanmıştır (Eglese and Li 1992). Yoğun kar yağışı nedeniyle kar küreme işlemi veya soğuk nedeniyle buzlanmanın yüksek olduğu yollara

tuz serpilmesi işleminin bir rota dâhilinde belli kapasitedeki araçlarla yapılması kapasite kısıtlı ÇPP'ye örnek olarak verilebilir.

### 2.2.3. Hiyerarşik Çinli postacı problemi

Çinli postacı probleminin özel bir hali olan HÇPP'de yollar (ayrıntlar) önceliklerine göre hiyerarşik bir şekilde sınıflandırılır. HÇPP'de amaç verilen bir şebekedeki yolların (ayrıntların) öncelik ilişkilerinin dikkate alınarak, yüksek öncelikli yollardan başlanarak düşük öncelikli yollara doğru, şebekedeki her bir ayrıttan en az bir kez geçmek koşuluyla optimum turu bulmaktır (Dror *et al.* 1987). Gerçek hayatta HÇPP'ye verilebilecek en güzel örneklerden biri kar küreme araçlarının rotalanmasıdır. Zira kar küreme işlemlerinde öncelik ana yolların olup ana yollarda temizleme işlemi bittikten sonra yan yolların vs. temizlenmesine geçilmesi söz konusudur.

Bir hiyerarşik şebeke, farklı öncelik seviyelerini kapsayan ayrıtlardan (yollardan) oluşmaktadır.  $V$ ; düğümler kümesini,  $E$ ; yönsüz ayrıtlar (yollar) kümesini gösteren bir  $G(V,E)$  şebekesi dikkate alındığında, şebeke  $m$  tane farklı alt şebekelere ayrılır ve her bir alt şebeke bir hiyerarşik seviyeyi gösterir. Dolayısıyla bir hiyerarşik şebeke gösterimi;  $G(V,E)=G_1(V_1,E_1) \cup G_2(V_2,E_2) \cup \dots \cup G_m(V_m,E_m)$  şeklinde ifade edilir. Burada  $G_m$ ;  $m$ . alt şebeke olup  $m$ . sırada öncelikli bütün yolları ve bu yollara karşı gelen düğümleri kapsamaktadır.

HÇPP ilk olarak Dror *et al.* (1987) tarafından tanımlanmıştır. Dror *et al.* yaptıkları çalışmada, yolların bölündüğü ve öncelik ilişkilerinin tanımlandığı bir şebekede optimum postacı turunu bulan bir algoritma geliştirmişlerdir. HÇPP'nin NP-zor sınıfında olduğunu ancak problemin bazı özel durumları için polinom zamanlı algoritmalarla da çözülebileceğini göstermişlerdir. Bu özel durumlar; bütün alt hiyerarşiklerin birbirleriyle bağlantılı olduğu ve öncelik ilişkilerinin lineer olduğu durumlardır. Geliştirdikleri algoritma bu bahsedilen özel durumların olmaması halinde sonuç vermemektedir.

Lemieux and Campagna (1984), Alfa and Liu (1988), Krishnamurthi and Damodaran (1998) yaptıkları çalışmalarda HÇPP'nin çözümüne yönelik sezgisel algoritmalar geliştirmişlerdir.

Alfa and Liu (1988) yönlü hiyerarşik bir şebekede daha iyi çözümler veren bir yöntem sunmuşlardır. Yaptıkları çalışmada birbirleriyle bağlantılı olmayan hiyerarşik sınıfları birleştirmek için bağlantısı olmayan hiyerarşik şebekeden yol seçen bir sezgisel yöntem geliştirmişlerdir.

Birbirleriyle bağlantısı olmayan alt şebekeleri birleştiren farklı bir sezgisel metod Damodaran (1997) tarafından sunulmuş olup geliştirilen sezgiselin önceden Alfa and Liu (1988) tarafından aynı amaç için geliştirilen sezgiselden daha iyi performans sergilediğini göstermiştir.

HÇPP için kesin çözüm veren algoritma (exact algorithm) ilk kez Ghiani and Improva (2000) tarafından oluşturulmuştur. Problemi  $O(p|V|)$  düğümlü bir yardımcı grafik üzerinde eşleştirme olarak çözen daha düşük hesaplama karmaşıklık algoritma geliştirmişlerdir. Geliştirdikleri algoritma ile önceden geliştirilen yöntemlerden daha kısa bilgisayar zamanında çözüme erişilmiştir.

Cabral *et al.* (2004) HÇPP'yi kırsal postacı problemine dönüştürecek bir metod sunmuşlardır. Öncelik ilişkilerinin lineer olarak ele alındığı problemin çözümü için dal-kesme algoritmasını kullanarak optimum çözüm elde etmişlerdir.

Korteweg and Volgenant (2006) öncelik derecelerine göre sınıflandırılmış şebekelerde, öncelik sırasına göre minimum maliyetli rotayı bulan farklı bir algoritma ortaya koymuşlardır. Algoritma ile ilk olarak, en yüksek öncelikli alt şebekenin minimum rotası bulunur. Daha sonra öncelik sırası göz ardı edilmeksizin her bir alt şebekenin ayrı ayrı minimum maliyetli rotalarının hesaplanmasıyla işlem tamamlanır.

Genelde CPP'nin bir tur çözümü, HCPP'nin alt sınırı kabul edilmektedir. Damodaran *et al.* (2008) yaptıkları çalışmada, CPP'nin çözümleri ile karşılaştırıldığında HCPP için daha iyi alt sınır değerleri veren yeni bir sezgisel algoritma geliştirmişlerdir. Bu değerlerle kesin arama algoritmaları (dal-sınır gibi) kullanılarak daha kısa zamanda optimum çözüme ulaşıldığını göstermişlerdir. Aynı zamanda geliştirilen sezgisel ile çeşitli problem boyutları ele alınmış ve daha önce yapılan çalışmalardan daha iyi sonuçlar elde edilmiştir.

Sayata and Desai (2015) yaptıkları çalışmada,  $O(k^2n^2)$  zaman karmaşıklığına sahip grafikteki kenar sayısını azaltmak için Kruskal'ın minimum yayılan ağaç yöntemini kullanarak alternatif bir yaklaşım geliştirmişlerdir. Kısacası Dror *et al.* (1987) algoritmasının 2. adımında grafiğin kenar sayısını azaltmak için Kruskal yöntemini uygulamışlar ve Dror *et al.* (1987) tarafından daha önce önerilen zaman karmaşıklığını iyileştirmek için alternatif bir yöntem kullanmışlardır. Önerilen Kruskal tabanlı HCPP sonuçlarının basit HCPP ile kıyaslandığında ortalama %21,64 oranında iyileşme sağlandığını ve hiyerarşinin sayısının 3'ten az, kenar sayısının 10'dan az olduğu durumlarda CPP'ye göre ortalama %13,35 iyileşme sağlandığını göstermişlerdir.

HCPP ile ilgili literatürde yapılan en son çalışma 2017 yılında Colombi *et al.* tarafından ortaya konulmuştur. Yapılan çalışmada hiyerarşik karma kırsal postacı problemini ele almışlardır. Problem çözümü için kesin çözümler üreten dal ve kesme algoritması geliştirmişlerdir. Geliştirdikleri algoritmanın çözümlerini literatürde var olan test problemleri ile karşılaştırmışlar ve problem çözüm sürelerinde bir saate yakın iyileştirme sağlamışlardır.

Literatürün büyük kısmı incelendiği zaman HCPP'nin gerçek hayat problemlerine uygulanabilir olmasına rağmen son derece az sayıda çalışma mevcuttur. Uygulama alanlarının çeşitliliğine ve önemine rağmen HCPP, GSP ve Araç Rotalama Problemleri (ARP) kadar yeterli ilgi görmemiştir. Kombinatoryal eniyileme problemlerinin bu alanı pek çok araştırmacı için hala araştırılmaya açık olduğu aşikârdır.

Şu ana kadarki yapılan çalışmalarda kesin ve kesin olmayan yöntemler kullanılmış olup henüz problemin çözümü için matematiksel model geliştirilmemiştir. Gerçek hayat problemleri üzerine uygulanabilir olması açısından daha fazla kıstası göz önüne alarak yapılan bir modellemenin gereksinimi kaçınılmaz olmaktadır.

HÇPP, NP-zor problem sınıfına aittir (Dror *et al.* 1987). Problemin kombinatoryal yapısı problem boyutu büyüdükçe en iyi sonuçların elde edilmesini güçleştirmektedir. Sonuç olarak problem boyutu arttıkça elde edilen çözüm alternatifleri de artacağından makul zamanda en iyi çözümlere ulaşmak mümkün olamamaktadır.

Bununla beraber mektup ve posta dağıtımı, polis, devriye araçları ve kar temizleme araçlarının rotalanması, otobüs çizelgelenmesi gibi karşılaşılan pek çok problemin HÇPP'ye uyarlanması ve çözüme gidilmesi gerçek hayat ile örtüşmesi açısından büyük önem arz etmektedir.

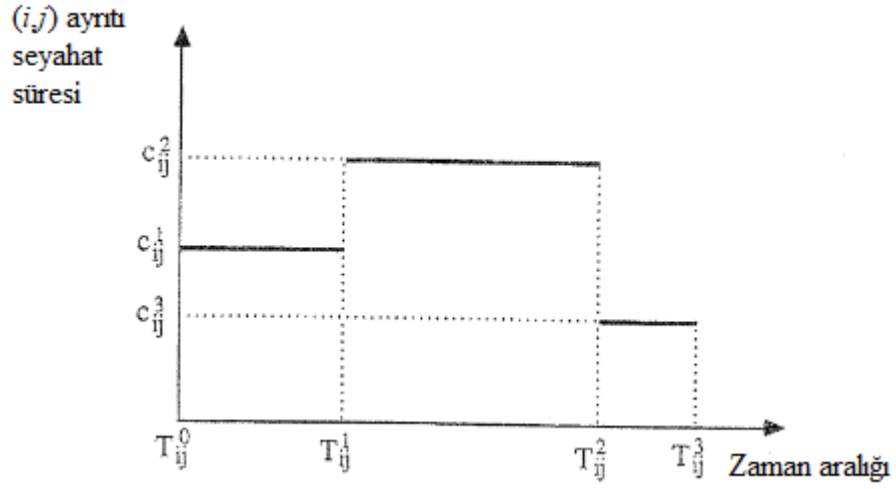
### **2.3. Zaman-Bağımlı Rotalama Problemleri**

Son yıllarda, günün farklı saatlerindeki seyahat süresi ya da seyahat hızı verilerinin kullanılabilirliği, araştırmacıların bu ek bilgiyi kullanarak farklı rotalama problemleri geliştirmelerine yol açmıştır. Rotalama problemlerinin büyük çoğunluğu, problemlerin çözülmesi için gerekli olan tüm bilgilerin zamana göre değişmediği varsayımı altında yürütülmüştür. Düğümler arası ulaşım sürelerinin sabit alındığı çalışmalar, gerçek hayat uygulamalarında zamanla değişen seyahat sürelerinin varlığı göz önüne alındığında pek doğrulanmaz. Zira seyahat süreleri; trafik tıkanıklığı, hava koşulları veya sürücünün araç hızını ayarlayabilmesi gibi durumlara bağlı olarak değişebilir.

Çalışmalarda nadiren ele alınan pratik bir husus, düğümler arası ulaşım sürelerinin günün saatine bağlı olarak değiştiği zaman bağımlılığı kavramıdır. Zaman bağımlı seyahat sürelerinin modellenmesi, zaman bağımlı rotalama problemleri literatüründe önemli bir konu haline gelmiştir. Zaman bağımlı ulaşım süreleri ilk kez Malandraki and Daskin (1989, 1992) tarafından Zaman-Bağımlı Gezgin Satıcı Problemi (ZB-GSP) ve Zaman-

Bağımlı Araç Rotalama Problemi (ZB-ARP) üzerinde ele alınmıştır. Araçlara ait rotalarda harcanan toplam zamanın en aza indirilmesi amaçlanmıştır. Dğümler arası seyahat süreleri, gün içerisindeki farklı zaman dilimlerine ve dğümler arasındaki mesafeye bağlı olarak tanımlanmıştır. Çalışmada, “Zaman Bağımlı Kesikli Ulaşım Süresi Fonksiyonu (ZBKUSF), (time-dependent discrete travel time function)” kullanılmıştır. Literatürde, ZB-ARP için yapılmış çeşitli çalışmalar mevcuttur. Malandraki and Daskin (1992) yaptıkları çalışmada ZBKUSF fonksiyonu kullanarak problemin çözümüne yönelik kesin çözüm veren algoritma geliştirmelerine karşın, Ahn and Shin (1991), Hill and Benton (1992), Malandraki and Dial (1996), Park (2000), Jung and Haghani (2001) çalışmalarında ZBKUSF fonksiyonu kullanarak sezgisel yöntemler geliştirmişlerdir. ZBKUSF’de gün, farklı zaman dilimlerine ( $t_i$ ) ayrılmıştır.  $(i,j)$  düğüm çiftinde,  $i$  başlangıç noktasını  $j$  varış noktasını temsil etmektedir.  $C_{ij}(t_i)$ ,  $(i,j)$  ayrımının seyahat süresini vermekte olup,  $i$  düğümünden çıkış zamanının ait olduğu zaman diliminin bir fonksiyonu olarak temsil edilmiştir.  $(i,j)$  düğümü için üç farklı zaman diliminde tanımlanan kesikli seyahat süreleri Şekil 2.9’da verilmiştir. ZBKUSF’de iki düğüm arasında tanımlanan seyahat süresi, her zaman aralığı için başlangıç zamanına bağlı olarak aralık boyunca sabittir. Ancak trafiğin yoğun olduğu bir kentte, iki nokta arasındaki seyahat süresi, araç hızı sabit olmadığından dolayı kesikli bir fonksiyon olarak göz önüne alınması gerçeklikten uzak bir yaklaşımdır. Zira, gün içerisinde trafik yoğunluğunda meydana gelen değişiklikler, araçların seyahat hızındaki değişmelere bu da seyahat süresinde değişimlere neden olur.



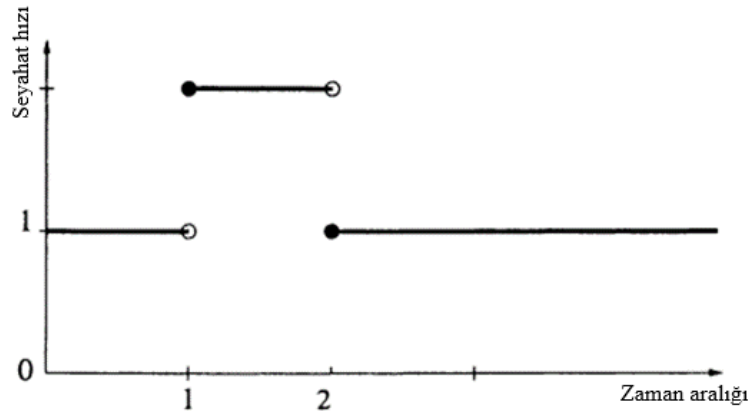


**Şekil 2.9.**  $(i,j)$  ayrıtı için kesikli seyahat süresi fonksiyonu

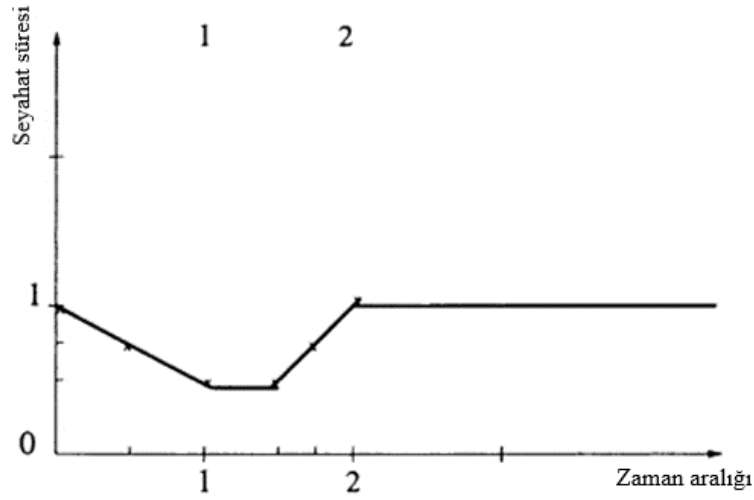
Bu modelin en büyük zayıflığı, "ilk giren ilk çıkar" (FIFO, First-In-First-Out) özelliğini karşılamamasıdır. FIFO özelliği, belirli bir zamanda  $i$  düğümünden  $j$  düğümüne varmak için çıkan bir aracın, daha sonraki bir zamanda  $i$  den  $j$  ye varmak için ayrılan başka bir araçtan daha önce hedefe varmasını garanti eder. Şekil 2.9'da verilen çıkış zamanına bağlı seyahat süresi fonksiyonu bir düğümünden çıkan aracın çıkış zamanına göre bir sonraki düğümüne ulaşım süresini göstermektedir. Bu fonksiyona göre, 2. zaman diliminde  $i$  düğümünden yola çıkış yapan bir araç, kendisinden daha sonraki bir zaman dilimi olan 3. zaman diliminde  $i$  düğümünden çıkış yapan bir araçtan daha sonra  $j$  düğümüne varacaktır. Yani bu duruma göre aracın önce çıkış yapması varacağı yere daha geç ulaşmasına neden olacaktır ki bu durum pratikte karşılaşılan duruma tamamen terstir.

Malandraki (1989) ve Malandraki and Daskin (1992) bu zafiyetin üstesinden gelmek için araçların bir düğümde hareket zamanı fonksiyonunun düzelmesini beklemelerine izin vermeyi önermişlerdir. Ancak, bu öneri sadece düğümlerde gereksiz beklemeye neden olmaktadır. Gerçek hayattaki uygulamalarda (örneğin, endüstriyel ürünlerin dağıtımı), bir aracın/dağıtıcının, yola çıkmaya hazır olsa bile bir müşteri bölgesinde beklemesi fikri gerçeklikten oldukça uzak bir yaklaşımdır.

Ichoua *et al.* (2003) çalışmasına kadarki yapılan çalışmaların tamamında bu teknik kullanılmış ve gerçekçi olmayan bir durum olan, aracın yolu daha kısa sürede kullanacağı zamana kadar müşteride beklemesi varsayımı dikkate alınmıştır. Literatürde FIFO özelliğini en iyi sağlayan ve pratiğe oldukça yakın bir yaklaşım olan “Zamana Bağlı Hız Fonksiyonu (ZBHF)” ilk kez Ichoua *et al.* (2003) tarafından tanıtılmıştır. İki düğüm arasındaki seyahat süresinin sadece aracın bir düğümden çıkış zamanına bağlı olmadığını, aynı zamanda aracın zaman aralığındaki hızına da bağlı olduğunu göstermişlerdir. Bu hız modeli birçok gerçek hayat problemi için oldukça kullanışlıdır. Bu yaklaşıma göre zaman, ZBAUSF yaklaşımında olduğu gibi dilimlere bölünmekte ve her düğüm için aracın o düğümden çıkış zamanına bağlı olarak adımsal bir hız fonksiyonu tanımlanmaktadır. Modeldeki ana nokta, hızın bir hattın (yolun) tüm uzunluğu boyunca sabit olarak alınmaması ve iki ardışık zaman dilimi arasındaki sınır geçildiğinde hızın da değişmesidir. Bu nedenle, seyahat süresi, basit ve kolay bir şekilde değerlendirilebilen parçalı bir sürekli fonksiyondur. Şekil 2.10 ve Şekil 2.11’de sırasıyla örnek bir seyahat hız fonksiyonu ve bununla bağlantılı seyahat süresi fonksiyonu verilmiştir (Ichoua *et al.* 2003).



**Şekil 2.10.** Örnek bir seyahat hız fonksiyonu



**Şekil 2.11.** Örnek bir seyahat süresi fonksiyonu

Sürüş hızları zaman içerisinde sürekli olarak değişmektedir. Ancak Ichoua *et al.* (2003) tanımladıkları adimsal hız fonksiyonunun (ZBHF) adimsal ulaşım süresi fonksiyonundan (ZBAUSF) daha makul bir varsayım olduğunu belirtmişlerdir. Yani, iki ardışık zaman periyodu arasındaki zaman geçişlerinde hız, bağlantı üzerindeki seyahat süresinden çok daha hızlı değişecektir. Seyahat hızındaki değişimlerden dolayı, bu modelde önerilen seyahat süresi fonksiyonu FIFO özelliğini karşılamaktadır. Yani bir düğümden erken ayrılmanın varılacak yere daha erken gidileceğini garanti eder.

Bu bağlamda, planlama periyodu zaman dilimlerine bölünerek bu zaman dilimlerinde her bir yol için sabit bir hız tanımlanmaktadır. Bir düğümden çıkan araç çıkış zamanına ve gideceği yola bağlı olarak belli bir hız ile bir sonraki düğüme seyahat etmektedir. Bu varsayım pratikte karşılaşılan durumlara oldukça benzerlik göstermekte olup amaç, aracın toplam seyahat süresini en aza indirmektir.

Literatürde, ZB-ARP ve ZB-GSP geniş çapta incelenmesine rağmen (Malandraki 1989; Malandraki and Daskin 1992; Schneider 2002; Helvig *et al.* 2003; Fleischmann *et al.* 2004; Chen *et al.* 2006; Albiach *et al.* 2008; Hashimoto *et al.* 2008; Xin *et al.* 2008; Kuo *et al.* 2009; Maden *et al.* 2009; Balseiro *et al.* 2011; Figliozzi 2012; Koç 2012; Koç ve

Karaođlan 2012; Harwood *et al.* 2013; Cordeau *et al.* 2014), zaman-bađımlı ayırıt rotalama problemleri nadiren ele alınmıřtır.

Zaman-bađımlı ayırıt rotalama problemleri ile yapılan alıřmalarda, Tagmouti *et al.* (2007) kar temizleme alıřmasına uyarlanmıř kapasite kısıtlı ve zaman bađımlı servis maliyetli bir ayırıt rotalama problemini ele almıřlardır. Kapasiteli ayırıt rotalama probleminin yeni bir tr olan zaman-bađımlı paralı dođrusal servis maliyetli ayırıt rotalama problemi Tagmouti *et al.* tarafından 2010 yılında geliřtirilmiřtir. Problem kar temizleme iřlemine uygulanmıř ve en yakın komřu arama sezgiseli ile zme gidilmiřtir. Bir bařka problem tr, araların rotaları boyunca beklemelerine msaade edilmemesidir. Yazarlar, bařlangıta geliřtirilen statik versiyonlu problemin zaman bađımlı trne en yakın komřu arama sezgiselini uyarlamıřlardır.

Kapasiteli ayırıt rotalama probleminin dinamik versiyonu Tagmouti *et al.* (2011) tarafından ele alınmıř ve problem zm iin en yakın komřu arama sezgiseli geliřtirilmiřtir.

Black *et al.* (2013) zaman bađımlı dl-toplama ayırıt rotalama problemi isimli yeni bir problem tr tanımlamıřlardır. Problem, seyahat srelerinin gn ierisinde deđiřtiđi bir řebeke hattında toplama ve dađıtım iřlemi yapan bir dizi ykl kamyonlar arasından seim yapma esasına dayanır. Problem zm iin deđiřken komřu arama sezgiseli ve tabu arama metasezgiseli geliřtirilmiřtir.

Zaman-bađımlı PP ile ilgili yapılan alıřmalarda olduka sınırlıdır. Wang and Wen (2002) alıřmalarında ynl PP'yi ele almıřlar ve bu problemi hem kesin hemde bulanık zaman penceresi kısıtları altında incelemiřlerdir. Yazarlar, problemin ilk tamsayılı matematiksel formlasyonunu geliřtirmiřlerdir.

Sun *et al.* (2011a) zaman-bađımlı seyahat sreli PP'nin zmne ynelik yeni bir tamsayılı matematiksel formlasyon geliřtirmiřlerdir.

Sun *et al.* (2011b) zaman-bağımlı seyahat süreli ÇPP'nin NP-zor olduğunu göstermişler ve FIFO özelliğinin dikkate alındığı problem için dinamik programlama algoritması önermişlerdir.

Sun *et al.* (2015) zaman-bağımlı seyahat süreli ÇPP için Wang and Wen (2002) tarafından verilen formülasyonun genişletilmiş bir versiyonu olan lineer tamsayı programlama formülasyonunu geliştirmişlerdir.

## **2.4. Ayırıt Rotalama Problemleri için Çözüm Yöntemleri**

Ayırıt rotalama problemleri literatürde önemli bir araştırma konusu olup çeşitli çözüm yöntemleri geliştirilmiştir. Bu bölümde ayırıt rotalama problemleri ve alt bir problem olan ÇPP için geliştirilen çözüm yöntemleri ele alınmıştır.

### **2.4.1. Kesin ve kesin olmayan yöntemler**

En iyileme problemlerinin çözüm yöntemleri temel olarak 2 sınıfa ayrılır. Bunlar kesin (exact) ve kesin olmayan yöntemler olarak adlandırılırlar. Kesin yöntemlerle en iyi (optimum) çözümlere ulaşılır. Fakat problem boyutu arttıkça, problemin çözüme kavuşturulması oldukça vakit alabilmektedir. Kesin olmayan yöntemler ise yaklaşık ve buluşsal (sezgisel) yöntemler olmak üzere ikiye ayrılır. Bu yöntemler en iyi çözümü garanti etmemekle beraber optimuma yakın kabul edilebilir bir çözüme polinomial zaman içerisinde ulaşmaktadırlar. Ancak yaklaşık yöntemlerle elde edilen çözümlerin en iyi çözüme ne kadar yakın olduğu bellidir (% x gibi belirli bir oran kadar en iyi çözüme yaklaşırlar). Buluşsal yöntemlerin ise en iyi çözüme yaklaşma oranı belli değildir (Ignizio 1980).

ÇPP ile ilgili olarak ele alınan problem tipine ve çözüm yöntemine göre literatürde yer alan bazı önemli çalışmalar ve kullandıkları çözüm yöntemleri Çizelge 2.1'de sunulmuştur.

**Çizelge 2.1.** ÇPP ve çeşitleri ile ilgili yapılan çalışmalar

Yıl	Yazar/Yazarlar	Çözüm Yöntemi	Problem Çeşidi
1993	Malandraki and Daskin	Dal-sınır algoritması	Maksimum faydalı ÇPP
2002	Ahr and Reinelt	Sezgisel algoritma	Min-mak k-ÇPP
2003	Pearn and Wang	Minimum yayılan ağaç algoritması	Maksimum faydalı ÇPP
2005	Feillet <i>et al.</i>	Dal-fiyat algoritması	Karlı ayrıt tur problemi
2005	Pearn and Chiu	Sezgisel algoritma	Maksimum faydalı ÇPP
2006	Ahr and Reinelt	Tabu arama algoritması	Min-mak k-ÇPP
2006	Aminu and Eglese	Kısıt programlama	Zaman pencereli ÇPP
2008	Irnich	Kesme Düzlemi Algoritması	Yönsüz ÇPP
2009	Kramberger <i>et al.</i>	Sezgisel algoritma	ÇPP
2009	Prakash <i>et al.</i>	Sezgisel algoritma	Çok amaçlı ÇPP
2011	Euchi <i>et al.</i>	Sezgisel algoritma	Karlı ayrıt tur problemi
2012	Willemse and Joubert	Tabu Arama Algoritması	Min-mak k-postacı problemi
2013	Corberan <i>et al.</i>	Tam sayı programlama, Dal-kesme algoritması	Maksimum faydalı ÇPP
2015	Yu and Lin	Sezgisel algoritma	Zaman-bağımlı getiri toplamalı ayrıt rotalama problemi

#### 2.4.2. Meta-sezgisel yöntemler

1990'lı yıllardan sonra araştırmacılar modern sezgisel yöntemlere başvurmuşlardır. Modern sezgisel yöntemler literatürde meta-sezgiseller olarak da geçmektedir. Problem boyutu arttıkça makul sürede çözüm veremeyen NP-zor yapıdaki problem türleri için meta-sezgisel algoritmalar tercih edilmektedir. Bu algoritmalar, birçok karmaşık optimizasyon probleminin çözümünde önemli ölçüde yetenekli ve etkili yöntemlere sahip genel algoritmik yapılarıdır (Glover and Kochenberger 2003). Yöneylem araştırması

alanındaki birçok karmaşık kombinatoriyal optimizasyon problemlerinin çözümünde bu algoritmalar başarıyla uygulanmış ve halen uygulanmaya devam edilmektedir.

Meta-sezgiseller farklı özelliklere sahip bir dizi algoritma içerirler. Genel olarak etkin bir meta-sezgiselin belirlenmesi, farklı meta-sezgiseller, bunların temel parametreleri ve operatörleri üzerinde kapsamlı analizler gerektiren deneyime bağlı bir süreçtir. Literatürde meta-sezgisel yaklaşımların ayrıt rotalama problemlerine uygulandığı çeşitli çalışmalar mevcuttur. Bu çalışmada, NP-zor yapıdaki ZB-HÇPP çözümü için GA ve hTB ele alındığından dolayı bu bölümde GA ve Tavlama Benzetimi (TB) ile yapılmış ayrıt rotalama çalışmalarına yer verilmiştir.

Lacomme *et al.* (2006) yaptıkları çalışmada kapasite kısıtlı ayrıt rotalama problemini ele almışlar ve problemin çözümüne yönelik çok amaçlı GA geliştirmişlerdir. Aynı problem türü için ağırlıklandırılmış kodlama tekniğine dayalı GA ise Morgan and Mumford (2009) tarafından geliştirilmiştir.

Dhein *et al.* (2018) iki amaçlı ayrıt rotalama problemi için genetik yerel arama algoritması geliştirmişlerdir. Açık kapasiteli ayrıt rotalama probleminin çözümü için hibrit GA ise Arakaki and Usberti (2018) tarafından geliştirilmiştir.

Wøhlk (2015), Tirkolae *et al.* (2016) çalışmalarında kapasiteli ayrıt rotalama problemini ele almışlar ve tavlama benzetimi temelli bir meta-sezgisel sunmuşlardır.

ÇPP'ye GA yaklaşımının uygulandığı çalışmalar ise oldukça sınırlıdır. Hua and Li-shan (2003), yönlü ve yönsüz çoklu şebekelerde ÇPP için genetik algoritma yaklaşımını kullanarak büyük ve karmaşık ağlarda etkin sonuçlar elde ettiklerini göstermişlerdir. Çalışmalarının NP-tam olarak bilinen karma ÇPP'nin nasıl çözüleceğine dair ışık tutmasını amaçlamışlardır.

Jiang *et al.* (2010) çalışmalarında rotaların optimizasyonu, interaktif sistemlerin dizaynı ve akış tasarımı gibi çeşitli uygulama alanları olan karma ÇPP'yi ele alarak çözüm için

GA önermişlerdir. Çözüm uzayının arama verimliliğini artırmak ve popülasyondaki her bir bireyi makul seviyelerde tutmak için probleme özgü genetik operatörler tasarlanmıştır.

Ma *et al.* (2015) postacının yük kapasitesi ve çalışma zamanı kısıtları altında ÇPP'yi ele almışlardır. GA yapısı ayrıt rotalama problemini düğüm rotalama problemine dönüştürmeden tasarlanmış ve öncelik tabanlı kromozom kodlama tekniği geliştirilmiştir. GA'nın karma şebekelerde her türlü kapasiteli ÇPP'yi etkin bir şekilde çözebildiğini göstermişlerdir.

Bütün bu çalışmalar ışığında, HÇPP için literatürde karşılaşılmayan veya detaylı olarak ortaya konmayan çeşitli durumların söz konusu olduğu görülmektedir. Bunlar, HÇPP ile ilgili yapılan çalışmaların çok az olup, matematiksel modelinin var olmadığı, zaman bağımlılığı kavramının HÇPP için ele alınıp incelenmediği ve problem boyutu arttıkça geliştirilecek meta-sezgisel yaklaşımlara başvurulmadığı gibi unsurlar olduğu yukarıdaki literatür taramasında gösterilmeye çalışılmıştır.

Bu tez kapsamında literatürde bahsedilen eksiklikler göz önüne alınarak HÇPP'nin gerçek hayat problemleri ile örtüşür şekilde matematiksel modeli geliştirilmiştir. Yollarda katedilen toplam mesafenin yanında araçların harcadıkları sürelerin de dikkate alınması amacıyla literatürde var olmayan ve ZB-HÇPP olarak adlandırılan yeni bir HÇPP çeşidi oluşturulmuştur. Problem çözümü için önerilen yaklaşımlar, tam sayılı programlama yaklaşımlarının esnekliğini ve meta-sezgisel algoritmaların çözüm hızını beraber barındırmaktadır. Dolayısıyla problemin tam sayılı matematiksel formülasyonu geliştirilmiş ve problemin NP-zor yapısından dolayı makul zamanda en iyi çözümlere ulaşmak için literatürde daha önce hiç başvurulmamış iki farklı meta-sezgisel yaklaşım ele alınmıştır. Geliştirilen problem türünün ve önerilen çözüm yaklaşımlarının mümkün olduğunca gerçek hayat problemlerine uygulanabilirliğinin artırılması ve literatürde yapılacak ilk çalışma olması amaçlanmıştır.



### 3. MATERYAL ve YÖNTEM

Bu bölüm, uygulama ve teorik olarak iki alt başlık altında hazırlanmıştır. İlk alt başlık altında ÇPP'nin gerçek hayat problemlerine uygulanabilirliğini analiz etmek üzere iki farklı ön çalışma yapılmıştır. İkinci alt başlıkta, ÇPP'nin bir türü olan HÇPP zaman kısıtları varyasyonu altında incelenerek gerçek hayat problemlerine uyarlanabilirliği artırılmıştır. Problem çözümü için önerilen tamsayı matematiksel model ve geliştirilen meta-sezgisel yaklaşımlar detaylı bir şekilde anlatılmıştır.

#### 3.1. Ön Çalışma: ÇPP Yaklaşımı ile TCDD Bakım Çalışmalarının Planlanması

Demiryolu altyapı bakım çalışmaları, tren faaliyetlerinin güvenliğinin sağlanması ve daha iyi bir ulaşım sisteminin elde edilmesi açısından oldukça önemlidir. Her yıl büyük miktarlarda para ray bakım çalışmaları için harcanmaktadır. Bakım maliyetlerinin yanı sıra, tren kaza maliyetleri, yük ve yolcu gecikme maliyetleri gibi durumlar da tren bakım faaliyetleriyle oldukça ilişkilidir. Schlake *et al.* (2009) yalnızca bir trenin gecikme maliyetinin saatte 200 \$'a kadar artabileceğini göstermiştir.

Türkiye'de demiryolu şebekesinin güvenliğini sağlamak ve raylarda meydana gelen bozulmaları tespit etmek için her yıl periyodik olarak bakım ve muayeneler yapılmaktadır. Bu bakım ve muayene çalışmaları için büyük ve pahalı donanımlara sahip özel amaçlı araç ve makineler kullanılmaktadır. Bunlardan bazıları; yol ve katener ölçüm aracı, ultrasonik cihazlar ve taşlama makineleridir. Bunların pahalı olması ve sayılarının oldukça az olması nedeniyle etkin kullanılması büyük önem arz etmektedir. Aynı zamanda bu araçların şebeke üzerinde bir noktadan diğer bir noktaya hareketi günlerce hatta aylarca zaman alabilmektedir. Bütün demiryolu hattı boyunca aracın dolaşması toplam maliyetin büyük bölümünü oluşturduğu için bu araçların en iyi rotalarının belirlenmesi maliyet ve mesafe açısından kritik bir öneme sahiptir. Dolayısıyla ray bakım aracının optimum turunun bulunması problemi, bir rotalama problemi olarak göz önüne

alınabilir. Gerçek hayatta bu araçların güzergâhları bölge müdürlüklerine bağlı uzman kişiler tarafından belirlenmektedir.

Literatürde demiryolu şebeke sisteminin geliştirilmesi açısından en iyi rota ve takım çizelgelerinin elde edilmesini ve demiryolu analizlerini içeren çeşitli çalışmalar yapılmıştır (Hong *et al.* 2009; Fu *et al.* 2015; Murali *et al.* 2016). Liden (2015) demiryolu alt yapısı bakım ve muayene alanında yapılan çalışmaları kapsamlı bir şekilde göstermiştir. Yapılan çalışmada demiryolu şebeke dizaynı, takım çizelgeleme, araç ve takım rotalama ile ilgili literatürde yapılan çalışmalar derlenmiştir.

Peng *et al.* (2011) ray muayene çizelgeleme problemini ele almıştır. Yaptığı çalışmada bir demiryolu şebekesinde ray muayenesi için görev alan ekiplerin çizelgelenmesini toplam mesafe ve maliyet açısından değerlendirmiştir. Bu çalışmaya ilave olarak Peng and Ouyang (2012) daha fazla kısıt ekleyerek yeni bir çözüm metodu geliştirmişlerdir. Lannez *et al.* (2015) ray bakım çizelgeleme problemini incelemiştir. Yapılan çalışmada demiryolu şebekesinin bakımı için tanımlanan görevlerin (işlerin) çizelgelenmesi problemi bir ayrıt rotalama problemi olarak modellenmiştir. Bir yıl içindeki görevler binlerce düğüm ve ayrıttan oluşan bir şebeke üzerinde gösterilerek çizelgelenmiştir. Rotalama açısından bakıldığında ray bakım çizelgeleme problemi, ayrıt rotalama problemlerine benzerlik göstermektedir. Bu problemde tanımlanan her bir iş (görev) şebeke üzerinde bir ayrıt olarak gösterilmiştir. Ayrıt rotalama problemi yaklaşımıyla şebeke üzerinde tanımlanan bu işlerin yapılış sıraları yani çizelgeleri belirlenmiştir.

Çalışmanın bu aşamasında, Türkiye Cumhuriyeti Devlet Demiryolları (TCDD) konvansiyonel hatların bakımı için kullanılan araçlardan biri olan yol ve katener ölçüm aracının optimum rotası ÇPP yaklaşımı ile belirlenerek, mevcut uygulanan durum ile karşılaştırılmıştır.

### 3.1.1. Problemin tanımlanması

2003 yılından bu yana demiryolu sektörü Türkiye'nin en dinamik sektörlerinden biri haline gelmiştir. Türkiye'de konvansiyonel hatların büyük kısmının yenilenmesi ve araç filosunun geliştirilmesiyle birlikte daha hızlı, daha konforlu ve daha güvenli bir demiryolu işletmeciliği yapılmaktadır. Bu yatırımlarla, 2003 yılında 77 milyon olan yolcu sayısı 2014 yılında 153,6 milyona ulaşarak %99'luk bir artış gerçekleştirilmiştir (TCDD Yıllık Rapor 2014).

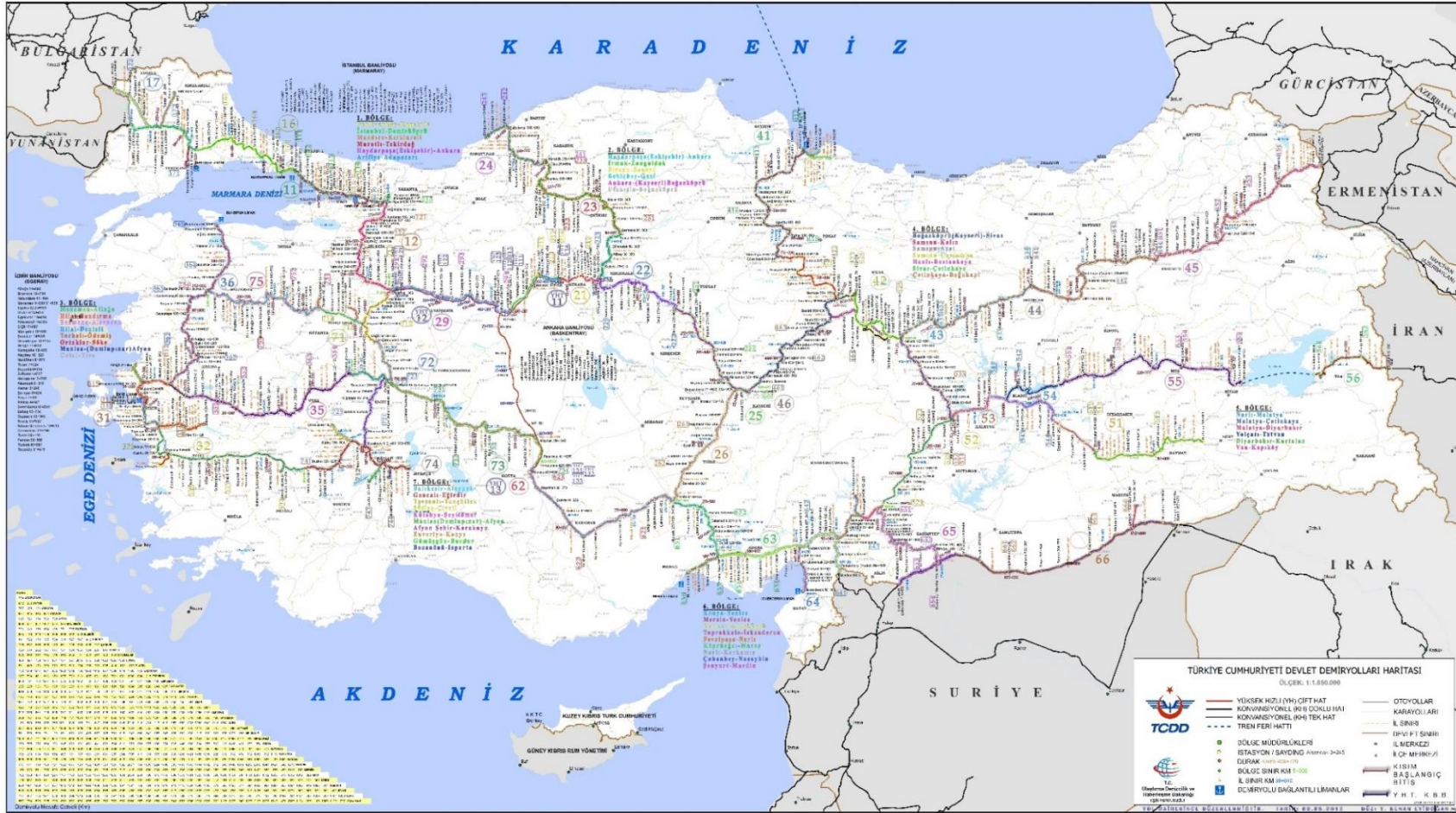
TCDD 8 334 km ana hat, 569 km ikinci hat olmak üzere 8 903 km toplam konvansiyonel ana hat uzunluğuna sahiptir. Bu uzunluğa 2 369 km tali hatları da eklendiğinde toplam konvansiyonel hat uzunluğu 11 272 km'ye ulaşmaktadır (TCDD Yıllık Rapor 2014).

Ulaşımında önemli bir yere gelen tren hatları, var olan eski hatlar ve eklenen yenileri ile beraber büyük bir ulaşım ağı meydana getirmektedir. Bu ulaşım ağının büyüklüğü getirdiği birçok avantajla beraber, hatlarda dolaşan trenlerin çizelgelenmesi, rotalanması ve bakımları gibi çeşitli zorlukları da beraberinde getirmektedir.

Pratikte Türkiye demiryolu ağı, ray bakım ve ölçüm makineleri kullanılarak belirli bir frekansta periyodik olarak kontrol edilmektedir. Türkiye'de ray bakımı için kullanılan farklı özelliklerde çeşitli araçlar bulunmaktadır. Bu araçlardan bir kısmı belirli bölgelere günlük, haftalık veya aylık periyodik bakımlar yapmalarının yanı sıra yılda sadece bir veya iki kere tüm hattı dolaşarak kontrol etmek üzere bakım çalışmalarını yapan araçlarda bulunmaktadır.

Bazı bakım faaliyetleri büyük ve pahalı donanımlara sahip araçlar ile yapılmaktadır. Dolayısıyla tüm hattı dolaşmak suretiyle bakım yapan bu araçların optimum rotasının çıkarılması maliyet ve mesafe açısından kritik bir öneme sahiptir. Çünkü büyük boyutlu ve kompleks şebekelerde kısıtlı sayıda aracın bütün şebekeyi dolaşması için yüzlerce veya binlerce alternatif rota oluşturulabilmektedir. Türkiye'de 6 ayda bir tüm şebekeyi dolaşarak periyodik bakım yapan araçlardan biri yol ve katener ölçüm aracı olup ülkede

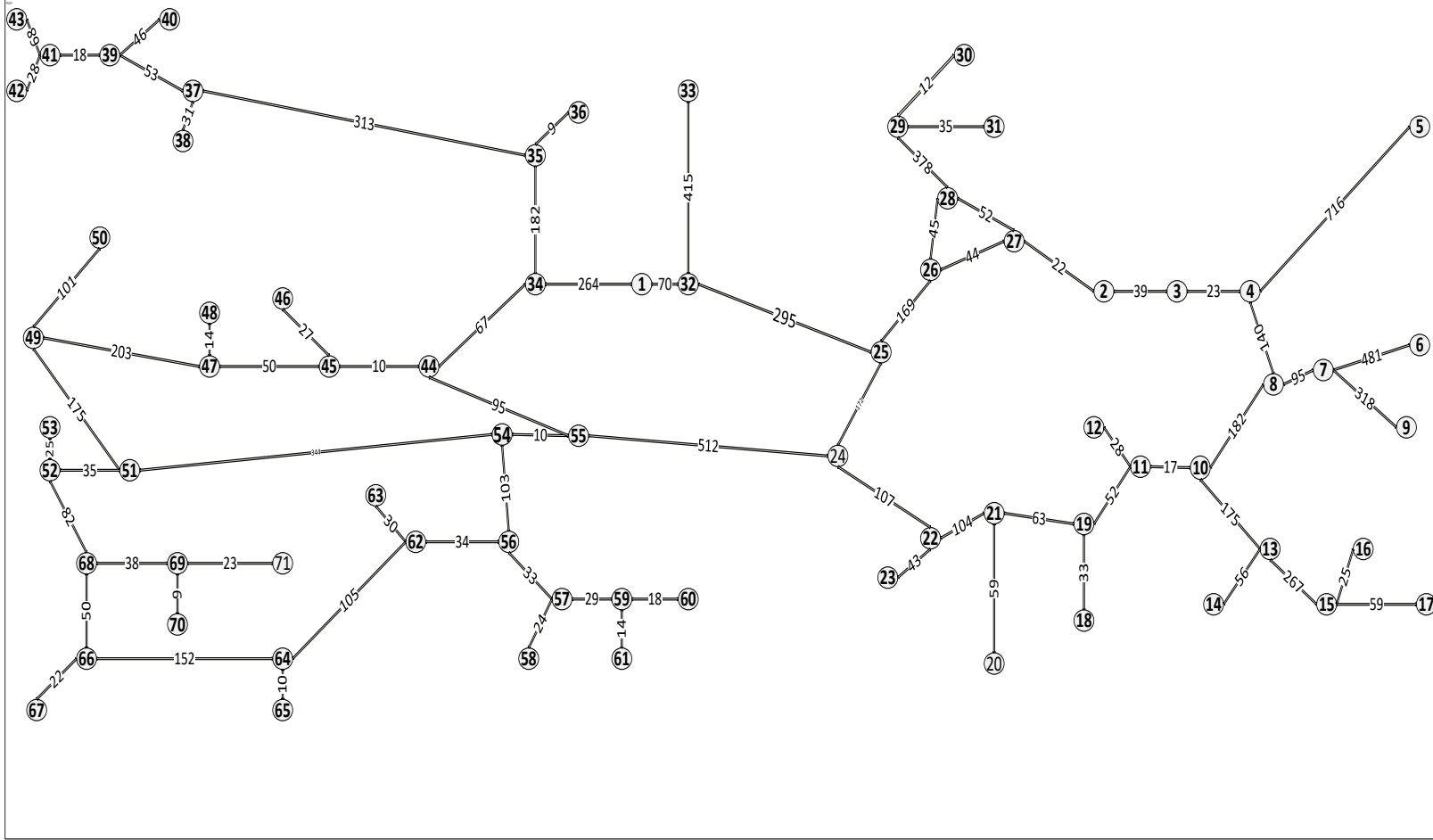
sadece 1 adet bulunmaktadır. Çalışmanın bu bölümünde tüm ray hattını dolaşarak ölçüm yapan yol ve katener ölçüm aracının optimum rotasının bulunması amaçlanmaktadır. Şekil 3.1’de Türkiye demiryolu haritası verilmiştir. Türkiye’deki ray hatları 7 bölgeye ayrılmış olup her bir bölge farklı semboller kullanılarak Şekil 3.2’de gösterilmiştir. Bölgelerin birbirleri ile yakınlık durumu dikkate alınarak bölgeler arası geçiş sağlanmaktadır. Problemin ÇPP’ye uyarlanarak çözülebilmesi için Türkiye demiryolu haritası şebeke haline dönüştürülerek Şekil 3.3’de gösterilmiştir. Elde edilen  $G=(V,E)$  şebekesinde V, düğümler kümesi olup bağlantı noktalarını E, yönsüz ayrıtlar kümesi olup ana ray hatları ve yan hatları temsil etmektedir. Şekil 3.3’de görüldüğü üzere şebeke toplam 71 düğümden oluşmaktadır. Düğümler arası mesafeler km cinsinden şebeke üzerinde gösterilmiştir. Tüm bağlantı noktaları çift yönlü olup eşit mesafededirler.



Şekil 3.1. Türkiye demiryolu haritası



Şekil 3.2. Bölgelerin semboller ile gösterimi



Şekil 3.3. Türkiye demiryolu konvansiyonel ray hattının şebeke gösterimi

Pratikte bölgelerin bakım tarihleri, bölgeler arası bakım sırası ve aracın güzergâhı her bir bölge müdürlüğüne bağlı uzman kişiler tarafından belirlenir. TCDD'den 2015 yılı ilkbahar dönemi bakım ve ölçüm programları temin edilmiş ve Çizelge 3.1'de yol ve katener ölçüm aracının mevcut uygulanmakta olan bakım planı ve her bir bölge için aracın rotası verilmiştir. Bu araç Türkiye'nin başkenti Ankara'da bulunmaktadır. Dolayısıyla aracın başlangıç noktası şebekede 1 nolu düğüm ile belirtilen Ankara olup, tüm bölgelerdeki rayların ölçüm işlemi bittikten sonra araç tekrardan Ankara'ya dönmektedir.

**Çizelge 3.1.** 2015 yılı ilkbahar dönemi bölgelere ait uygulanan araç güzergâhı

Bölgelerin bakım sıralaması	Pratikte uygulanmakta olan mevcut rota	Toplam Mesafe (km)
1	1-34-35-37-38-37-39-40-39-41-43-41-42-41-39-37-35-36-35-34	1 762
3,7	34-44-45-46-45-47-48-47-49-50-49-51-52-53-52-68-69-70-69-71-69-68-66-67-66-64-65-64-66-68-52-51-54-56-62-63-62-64-62-56-57-58-57-59-64-59-60-59-57-56-54-55-24-55-44-34	4 003
2	34-1-32-25-24-25-32-33-32-1	2 169
4	1-32-25-26-27-26-28-29-30-29-31-29-28-27-2-3-4-5-4	3 086
5	4-8-7-6-7-9-7-8-10	2 110
6	10-13-10-11-12-11-19-18-19-21-20-21-22-23-22-24-55-24-25-32-1	2 580
Toplam Mesafe		15 710

Yol ve katener ölçüm aracı 1. bölgeden başlamak üzere sırasıyla 3-7-2-4-5-6 bölgelerinin ölçüm işlemlerini yapmaktadır. 1. bölgede 1 762 km, 3. ve 7. bölgeler coğrafi konum sebebiyle birlikte rota planı yapılmış olup toplamda 4 003 km, 2. bölge 2 169 km, 4. bölgede 3 086 km, 5. bölgede 2 110 km ve son olarak 6. bölgede 2 580 km olmak üzere toplamda 15 710 km kat ederek tekrardan Ankara'ya dönmektedir.

Problem aşağıdaki varsayımlar altında incelenmiştir.



- Makinenin yakıt ihtiyacı hat üzerindeki gerekli depo müdürlüklerinden temin edilmektedir.
- Yollarda geçen süre ihmal edilmektedir.
- Yolcu, yük trenlerinin geçişi ve olumsuz hava şartları ihmal edilmekte olup rotalama işlemine etki etmemektedir.
- Ölçüm aracında ortaya çıkan arızalanma vb. gibi durumlar ihmal edilmiştir.
- Makinenin başlangıç ve bitiş noktası Ankara'dır (1 nolu düğüm).
- Suriye'de yaşanan iç savaş ve terör nedeniyle 14, 15, 16 ve 17 nolu düğümler şebekede gösterilmesine rağmen bakım işlemi geçici süreliğine yapılmamaktadır.
- Göl ve boğaz üzerindeki geçişler feribotla sağlanmakta olup mesafeler ihmal edilmiştir.

### 3.1.2. ÇPP matematiksel modeli

Çinli postacı problemi tamsayılı doğrusal programlama modeli aşağıda verildiği gibidir. Matematiksel model ile verilen şebekedeki bütün yollardan en az bir kere geçecek şekilde minimum tur uzunluğu belirlenmektedir (Eiselt *et al.* 1995).

Değişkenler;

$x_{ij}$  :  $i$  düğümünden  $j$  düğümüne giderken  $(i,j)$  ayrıtımdan geçilme sayısını belirten tamsayı karar değişkenidir.

$x_{ji}$  :  $j$  düğümünden  $i$  düğümüne giderken  $(j,i)$  ayrıtımdan geçilme sayısını belirten tamsayı karar değişkenidir.

Parametreler;

$C_{ij}$  :  $(i,j)$  ayrıtının uzunluğu (mesafesi).

$V$  : Şebekedeki tüm düğümler kümesi.

$E$  : Şebekedeki tüm yönsüz ayrıtlar kümesi.

$n$  : Şebekedeki toplam düğüm sayısı.

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = 0 \quad i = 1, 2, 3, \dots, n \quad \forall i \in V \quad (3.1)$$

$$x_{ij} + x_{ji} \geq 1 \quad \forall (i, j) \in E \quad (3.2)$$

$$x_{ij}, x_{ji} \geq 0 \text{ tamsayı değişkenler.} \quad (3.3)$$

Kısıtları altında,

$$\text{Minimum} \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \quad (3.4)$$

Kısıt (3.1), düğümler arası akışı (sürekliliği) sağlar. (3.2) numaralı kısıt her bir ayrıttan herhangi bir yönde en az bir kez geçilmesi gerektiğini göstermektedir. (3.3) numaralı kısıt ise tüm değişkenlerin pozitif tamsayı olması gerektiğini belirtmektedir. Denklem (3.4), en kısa uzunluğu hedefleyen amaç fonksiyonunu açıklamaktadır.

### 3.2. Ön Çalışma: k-ÇPP Yaklaşımı ile Kar Küreme Araçlarının Rotalanması

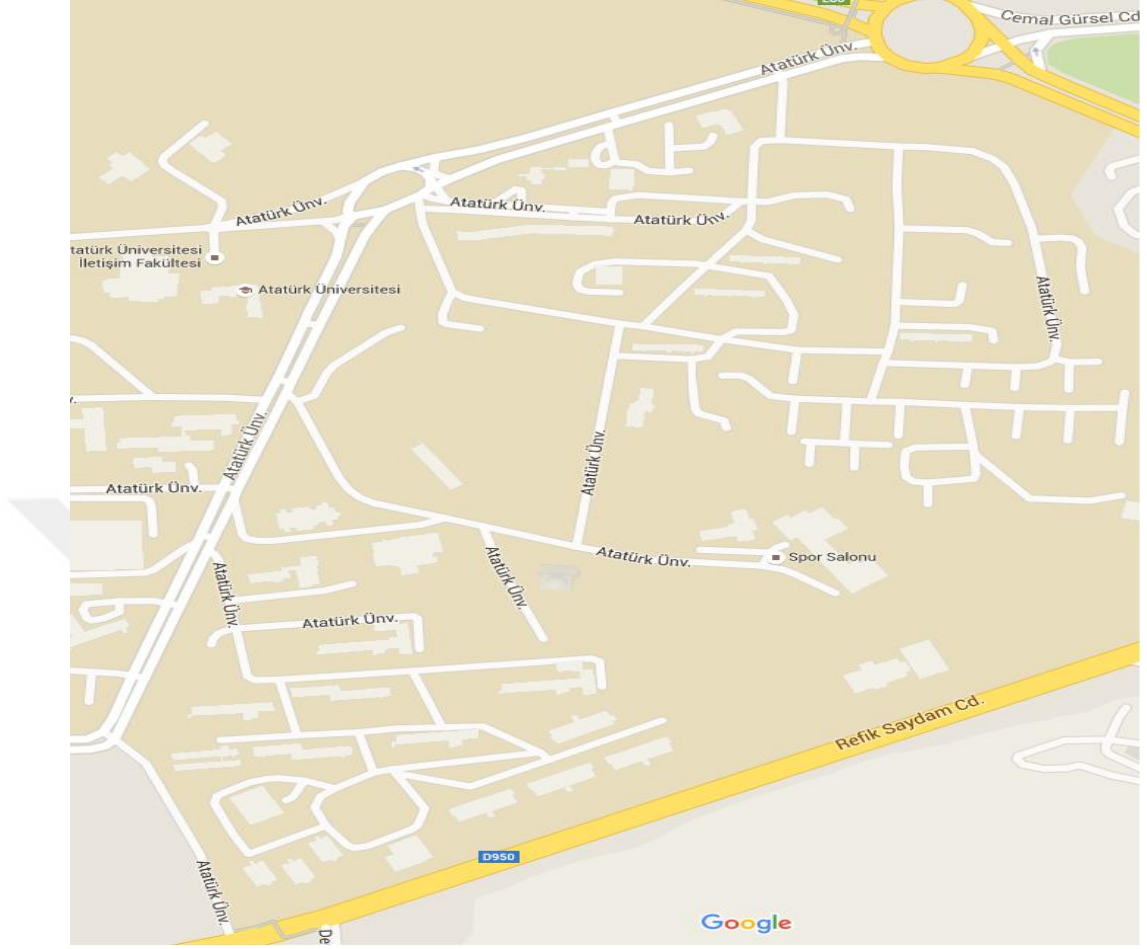
Bir takım gerçek hayat uygulama çalışmalarında tek bir araç yerine birden fazla aracın kullanılması ihtiyacı doğmuştur. ÇPP'nin başka bir türü olan k-Çinli Postacı Probleminde (k-ÇPP), k tane (en az 2) araç için rota belirlenmeye çalışılmaktadır. Yine aynı şekilde her bir araç başlangıç noktasından çıkıp yine başlangıç noktasına dönmek zorundadır. Bu problemde amaç k tane rotanın toplam uzunluğunu en küçükmektir.

Çalışmanın bu aşamasında, yoğun kış şartlarının yaşandığı Erzurum ili için kar küreme araçlarının optimum rotalarının belirlenmesi problemi ele alınmıştır. Uygulama çalışması için Atatürk Üniversitesi doğu kampüsü dikkate alınarak iki adet kar küreme aracı için optimum turlar k-ÇPP yaklaşımı ile elde edilmiştir. Ancak elde edilen sonuçlara bakıldığında araçların katettikleri yollar açısından dengeli dağılmadığı gözlenmiştir. Araçların gittikleri mesafelerin dengelenmesi açısından Dengeli k-Çinli Postacı Problemi (Dk-ÇPP) ele alınarak, matematiksel model geliştirilmiş ve her iki modelden elde edilen sonuçlar karşılaştırılmıştır.

### **3.2.1. Dk-ÇPP'nin tanımlanması**

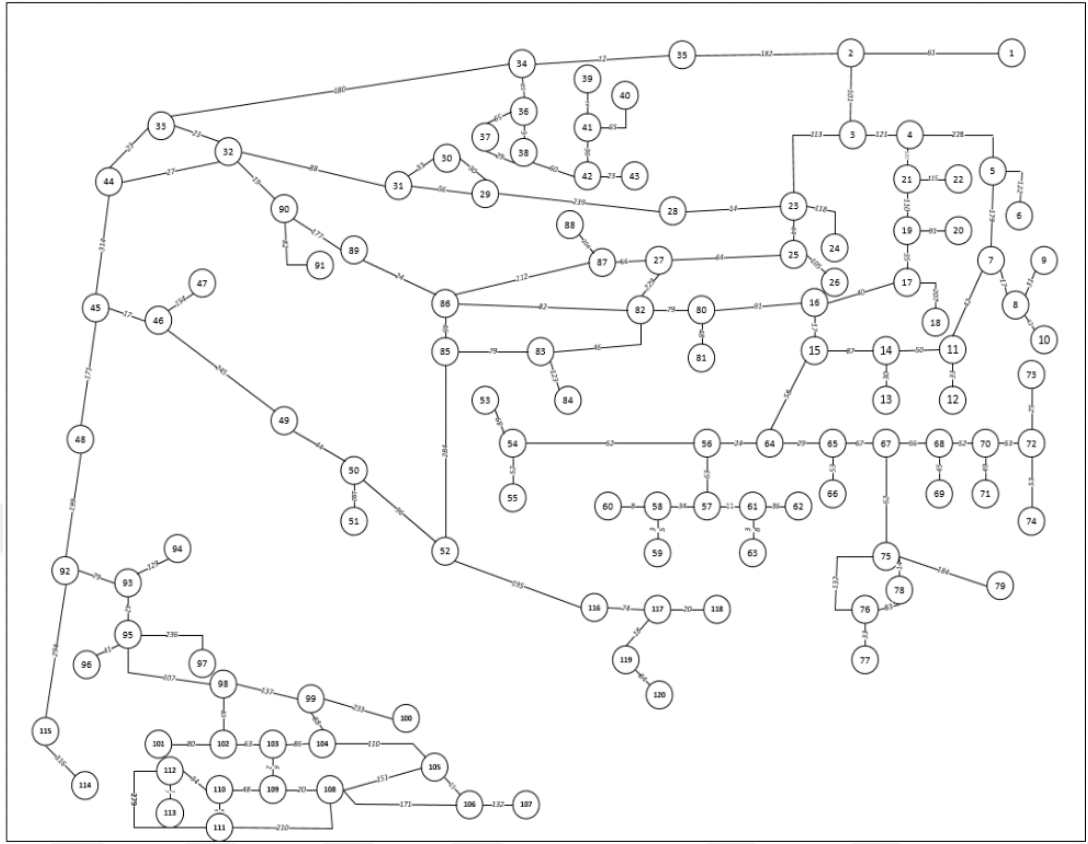
Yoğun kış aylarının yaşandığı Erzurum ilinde kar ve buz ile mücadele çalışmaları kaçınılmazdır. Bu çalışmalardan biri kar küreme işlemi olup karın yağış hızına bağlı olarak kar kalınlığı yaklaşık 2,5 cm kalınlığına gelindiğinde küreme işlemine başlanılmaktadır. Kar küreme işleminin birden fazla araçla yapılması ve yolların tamamının temizlenmesi gerektiğinden dolayı, bu araçların rotalanması probleminde k-ÇPP yaklaşımı ele alınarak çözüme ulaşılabilir.

Bu çalışmada, Atatürk Üniversitesi doğu kampüsü ele alınarak iki adet kar küreme aracı için optimum rotaların belirlenmesi amaçlanmıştır. Şekil 3.4'te Atatürk Üniversitesi doğu kampüsünün haritası verilmiştir.



**Şekil 3.4.** Atatürk Üniversitesi doğu kampüsünün haritası

Problem, k-ÇPP'ye uyarlanarak çözülebilmesi için, ele alınan kampüs haritası Şekil 3.5'te görüldüğü üzere şebeke haline dönüştürülmüştür. Şebeke; ana yollar, yan yollar ve bağlantı noktalarından (kavşaklardan) oluşmaktadır. Şebekede 120 düğüm vardır ve düğümler arası mesafeler (yollar) metre (m) olarak verilmiştir. Düğümleri birbirine bağlayan bütün yollar için gidiş-dönüş eşit mesafededir.



**Şekil 3.5.** Atatürk Üniversitesi kampüs haritasının şebeke gösterimi

### 3.2.2. Dk-ÇPP matematiksel modeli

k-ÇPP modeli ile araçların optimum rotalarının belirlenmesi amaçlanır. Ancak araçların birbirleri arasındaki yük, iş ya da mesafe gibi dağılımlarına bakılmamaktadır.

Bu çalışmada, k-ÇPP tamsayılı programlama formülasyonu dikkate alınarak Dk-ÇPP için matematiksel model geliştirilmiştir (Shafahi and Haghani 2015). Modelin birincil amacı turu en kısa mesafe katederek gerçekleştirmek, ikincil amacı ise araçlar arasındaki katedilen mesafe farkını minimuma düşürmektir. İkincil amaç ile araçların birbirleri arasındaki iş yükünün dengelenmesi amaçlanmıştır.

Dk-ÇPP birtakım varsayımlar içermektedir. Her bir araç için başlangıç noktası 1 nolu düğüm olup, araçlar turlarını tamamladıktan sonra başlangıç noktasına döneceklerdir. Bütün yollar eşit önceliğe ve ağırlığa sahiptir.

Kolaylık sağlamak açısından modelin değişkenleri ve parametreleri Çizelge 3.2’de verilmiştir.

**Çizelge 3.2.** Dk-ÇPP matematiksel model parametre ve değişkenler

Parametre ve değişkenler	
$x_{ijk}$	$i$ 'den $j$ 'ye giderken $(i,j)$ ayrıtından $k$ aracı ile geçilme sayısı
$x_{jik}$	$j$ 'den $i$ 'ye giderken $(j,i)$ ayrıtından $k$ aracı ile geçilme sayısı
$C_{ij}$	$(i,j)$ ayrıtının uzunluğu
$y_{ijk}$	Alt turların elimine edilmesi için yapay değişkendir. Bu değişken $i$ düğümünden $j$ düğümüne $k$ aracı ile geçişin sağlanmasından sorumludur.
$t_{ik}$	0-1 tamsayı değişken. $i$ düğümüne $k$ aracı gelirse 1, değilse 0.
$e_{k,k1}$	$k$ ve $k1$ araçlarının toplam katettikleri mesafelerin farkı
$V$	Şebekedeki tüm düğümlerin kümesi
$E$	Şebekedeki tüm ayrıtların kümesi
$K$	Araçların kümesi
$n$	Toplam düğüm sayısı
$M$	Büyük pozitif sayı
$O_k$	$k$ aracının başlangıç noktası

Yukarıda verilen bilgiler doğrultusunda Dk-ÇPP'nin tamsayılı doğrusal programlama modeli aşağıda verildiği gibi yazılabilir:

$$\sum_j x_{ijk} - \sum_j x_{jik} = 0 \quad i = 1, 2, 3, \dots, n \quad \forall i \in V, k \quad (3.5)$$

$$\sum_k x_{ijk} + x_{jik} \geq 1 \quad \forall (i, j) \in E \quad (3.6)$$

$$\sum_j x_{ijk} + \sum_j x_{jik} \leq 2 * M * t_{ik} \quad \forall i, k \quad (3.7)$$

$$\sum_j y_{ijk} - \sum_j y_{jik} + t_{ik} = 0 \quad \forall i \in (V/O_k), k \quad (3.8)$$

$$y_{ijk} \leq M * x_{ijk} \quad \forall i, j, k \quad (3.9)$$

$$\sum_k \sum_i \sum_j C_{ij} x_{ijk} - \sum_{k1} \sum_i \sum_j C_{ij} x_{ijk1} \leq e_{k,k1} \quad k, k1 \in K \quad (3.10)$$

$$-\sum_k \sum_i \sum_j C_{ij} x_{ijk} + \sum_{k1} \sum_i \sum_j C_{ij} x_{ijk1} \leq e_{k,k1} \quad k, k1 \in K \quad (3.11)$$

$$x_{ijk}, x_{jik}, y_{ijk} \geq 0 \text{ ve tamsayı} \quad (3.12)$$

$$t_{ik} \in \{0,1\} \quad (3.13)$$

Kısıtları altında,

$$\text{Minimum} \sum_i \sum_j \sum_k C_{ij} x_{ijk} + \frac{1}{M} \sum_{k=1}^{K-1} \sum_{k1=k+1}^K e_{k,k1} \quad (3.14)$$

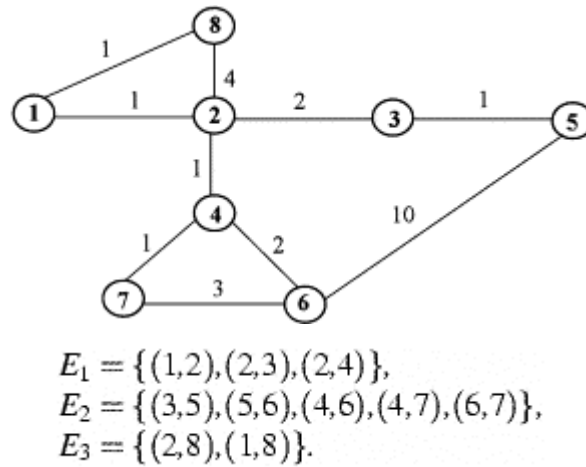
(3.5), akışı sağlayan süreklilik kısıtlarını, (3.6) nolu kısıt ise her bir ayrıttan herhangi bir yönde en az bir kez geçilmesi gerektiğini göstermektedir. (3.7)-(3.9) nolu kısıtlar alt-tur oluşumunu ve süreksizliği engellemektedir. (3.10)-(3.11) nolu kısıtlar ile araçların toplam katettikleri mesafe farklarının mutlak değerleri lineer hale getirilmiştir. Eğer  $k$  aracının toplam tur mesafesi  $k1$ 'den büyükse (3.10) nolu kısıt aktifleşir, aksi durumda (3.11) nolu kısıt aktif olur. (3.12) nolu kısıt tüm değişkenlerin pozitif tamsayı olması gerektiğini

(3.13) nolu kısıt ise değişkenin 0 ya da 1 değerini alması gerektiğini belirtmektedir. (3.14) nolu denklem birincil amacı en kısa uzunluğu hedefleyen, ikincil amacı ise araçlar arasındaki katedilen mesafe farkını minimuma düşürmeyi hedefleyen amaç fonksiyonunu açıklamaktadır.

### 3.3. Hiyerarşik Çinli Postacı Problemi ve Matematiksel Modeli

Hiyerarşik bir şebeke farklı öncelik seviyelerine ait ayrıtlardan (yollardan) oluşmaktadır.  $G=(V,E)$  şebekesi ele alındığında,  $V$ , düğümler kümesini,  $E$  ise yönsüz ayrıtlar (yollar) kümesini oluşturmaktadır. Hiyerarşik bir şebeke,  $k$  tane farklı alt-şebekelere bölünür ve her bir alt-şebeke bir hiyerarşik seviyeyi temsil eder. Şöyleki,  $G=(V,E)=G^1(V^1,E^1) \cup G^2(V^2,E^2) \cup \dots \cup G^m(V^m,E^m)$  olup  $G^m$ ,  $m$  inci hiyerarşik sınıfa karşılık gelen ayrıtlardan ve düğümlerden oluşan bir alt-şebekedir. Eğer bir hiyerarşik şebekede  $p$  seviyesine ait olan yollar ( $E_p$ ),  $q$  seviyesine ait olan yollardan ( $E_q$ ) yüksek öncelikli ise, HÇPP'nin amacı,  $q$  öncelikli yollara geçmeden önce  $p$  öncelikli yollardaki turu en kısa mesafe ile tamamlamasıdır.

Literatürde HÇPP ile ilgili iki adet şebeke örneği mevcuttur. Şekil 3.6'da Dror *et al.* (1987) tarafından geliştirilen küçük boyutlu bir hiyerarşik şebeke ve her bir hiyerarşik sınıfa ait yollar kümesi verilmiştir.

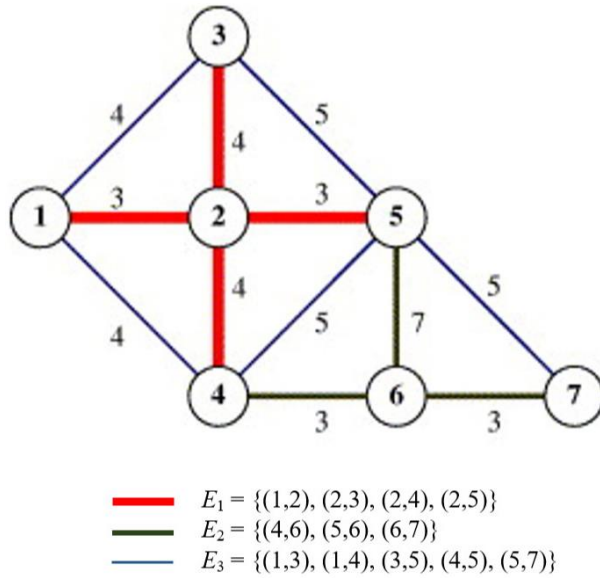


Şekil 3.6. Hiyerarşik şebeke örneği-1



Verilen yönsüz şebeke,  $V=\{1, 2, 3, 4, 5, 6, 7, 8\}$  düğümler kümesi ve  $E=\{E_1 \cup E_2 \cup E_3\}$  ayrıtlar kümesinden oluşmaktadır. Dror *et al.* (1987) yaptıkları çalışmada optimum tur mesafesini 30 ve optimum tur güzergahını, 1-2-4-2-3-5-6-7-4-6-4-2-8-1 olarak bulmuşlardır.

Şekil 3.7’de ise Korteweg and Volgenant (2006) tarafından geliştirilen küçük boyutlu bir hiyerarşik şebeke ve her bir hiyerarşik sınıfa ait yollar kümesi verilmiştir.



**Şekil 3.7.** Hiyerarşik şebeke örneği-2

Verilen yönsüz şebeke,  $V=\{1, 2, 3, 4, 5, 6, 7\}$  düğümler kümesi ve  $E=\{E_1 \cup E_2 \cup E_3\}$  ayrıtlar kümesinden oluşmaktadır. Korteweg and Volgenant (2006) yaptıkları çalışmada, optimum tur mesafesini 67 ve optimum tur güzergahını, 1-2-3-2-4-2-5-6-4-6-7-5-4-1-3-5-2-1 olarak bulmuşlardır.

Yukarıda da bahsedildiği üzere, HÇPP ile ilgili yapılan çalışmalar kesin ve/veya kesin olmayan sonuçlar veren algoritmalar kullanılarak yapılmıştır. Bu çalışmada aşağıdaki varsayımlar altında HÇPP incelenerek matematiksel modeli geliştirilmiştir.

- Hiyerarşik sınıflar içerisindeki tüm yollar birbirleri ile bağlantılıdır.
- Birbirini takip eden her bir hiyerarşik seviye birbiriyle bağlantılıdır.

HÇPP matematiksel modeli ile verilen şebekedeki bütün yolların öncelik ilişkileri dikkate alınarak her bir yoldan en az bir kez geçecek şekilde minimum tur uzunluğu belirlenmektedir.

### Notasyonlar

Modelde kullanılan indisler, kümeler, parametreler ve karar değişkenleri aşağıda başlıklar halinde sunulmuştur.

İndisler;

- $i$  : başlangıç düğümleri
- $j$  : varış düğümleri
- $t$  : rotaya ait yollardan geçiş sıralarını (adımları)
- $h$  : hiyerarşik seviyeleri temsil eder.

Kümeler;

- $i, j / V$  : düğümler  $i, j = \{1, 2, \dots, n\}$ ,
- $h / H$  : hiyerarşikler  $\{1, 2, \dots, h\}$ ,
- $t \leq L$  : adım sayısı  $L = 2 \|E\|$
- $E_h$  :  $h$ . hiyerarşik sınıfa ait ayrıtlar (yollar) kümesi,
- $E$  : Şebekedeki tüm ayrıtların kümesi,  $E = \{E_1 \cup E_2 \cup \dots \cup E_h\}$
- $\|E_h\|$  :  $h$ . hiyerarşik sınıfa ait toplam ayrıt sayısı,

$\delta_h(i)$  :  $i$  düğümünden çıkan  $E_h$  kümesine ait yollar,  $\delta_h(i) = \{j \mid (i, j) \in E_h\}$ .

$\delta(i)$  :  $i$  düğümünden çıkan bütün yollar kümesi,  $\delta(i) = \bigcup_{h \in H} \delta_h(i) = \{\delta_1(i) \cup \delta_2(i) \cup \dots \cup \delta_h(i)\}$

Parametreler;

$C_{ij}$  :  $(i,j)$  ayrıtının mesafe matrisi,

$B_{ij}$  :  $(i,j)$  ayrıtının bağlantı matrisi,

$O_{ij}$  :  $(i,j)$  ayrıtının öncelik matrisi,

$M$  : Büyük pozitif bir sayı,

$n$  : Şebekedeki toplam düğüm sayısı.

0-1 tam sayı değişkenler;

$x_{ij}^t = \begin{cases} 1 & t. \text{ adımda } (i,j) \text{ yolundan geçilmişse} \\ 0 & \text{değilse} \end{cases}$

$y_{ij}^t = \begin{cases} 1 & t. \text{ adımda } (i,j) \text{ yolundan ilk kez geçilmişse} \\ 0 & \text{değilse} \end{cases}$

$\phi_{ht} = \begin{cases} 1 & t. \text{ adımda } h. \text{ hiyerarşiye ait bütün yollardan geçilmişse} \\ 0 & \text{değilse} \end{cases}$

Yukarıda verilen bilgiler doğrultusunda HÇPP'nin karma tamsayılı doğrusal programlama modeli aşağıda verildiği gibi yazılabilir:

$$\sum_{t=1}^L (x_{ij}^t + x_{ji}^t) \geq 1 \quad \forall (i, j) \in E \quad (3.15)$$

$$\sum_i x_{ij}^{t-1} = \sum_s x_{js}^t \quad \forall j, t \quad t > 1 \quad i, j, s \in V \quad (3.16)$$

$$\sum_{(i,j)} x_{ij}^1 \geq 1 \quad (i,j) \in \delta_1(i_0) \quad (3.17)$$

$$\sum_j x_{ji}^t \geq 1 \quad t = t_{son}, i = i_0 \quad (3.18)$$

$$\sum_{(i,j) \in E} x_{ij}^t = 1 \quad \forall t \quad (3.19)$$

$$x_{ij}^t + x_{ji}^t \geq y_{ij}^t \quad \forall i, j, t \quad i < j \quad (3.20)$$

$$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) \leq M * (1 - y_{ij}^t) \quad \forall i, j, t \quad i < j \quad (3.21)$$

$$\sum_{t=1}^L y_{ij}^t = 1 \quad \forall (i, j) \in E \quad i < j \quad (3.22)$$

$$\sum_{t' \leq t} \sum_{(i,j) \in E_h} y_{ij}^{t'} \geq \|E_h\| * \phi_{ht} \quad \forall h, t \quad (3.23)$$

$$\sum_{t' \leq t} \sum_{(i,j) \in E_{h+1}} (x_{ij}^{t'} + x_{ji}^{t'}) \leq M * \phi_{ht} \quad \forall t, h = 1, \dots, H-1 \quad (3.24)$$

$$x_{ij}^t, y_{ij}^t, \phi_{ht} \in \{0,1\} \quad \forall i, j, t, h \quad (3.25)$$

Kısıtları altında,

$$\text{Minimum} \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^L c_{ij} x_{ijt} \quad (3.26)$$

Kısıt (3.15), şebekedeki tüm yollardan herhangi bir yönde en az bir kez geçilir. Kısıt (3.16), düğümler arası akışı (sürekliliği) sağlar. Eğer bir rotada  $(t-1)$ . adımda  $j$ . düğüme gidilmişse bir sonraki  $t$ . adımda  $j$  düğümünden çıkılır. (3.17) nolu kısıt, ilk adımda  $(t=1)$ ,  $i_0$  başlangıç düğümünden gidilebilecek 1. hiyerarşiye ait yollardan herhangi birinden geçilir. Kısıt (3.18), en son adımda başlangıç düğümüne ( $i_0=1$ ) geri dönülür. Kısıt (3.19), her adımda yalnız bir yoldan geçilir. (3.20)-(3.22) nolu kısıtlar,  $(i,j)$  ayrıtından ilk geçilen adımı tanımlayan  $y_{ij}^t$  değişkenine değer ataması yapar. Şöyle ki, herhangi bir  $(i,j)$  ayrıtı ele alındığında  $\{x_{ij}^t\}_t$  kümesi,  $t$  indisi ile elemanların sıralandığı bir vektör olarak verilsin:

$$1 \leq t \leq L, \quad L=10$$

$$\left[ x_{ij}^t \right] = [0010001001] \quad (3.27)$$

$$\left[ x_{ji}^t \right] = [0001000100] \quad (3.28)$$

$$\left[ x_{ij}^t + x_{ji}^t \right] = [0011001101] \quad (3.29)$$

Eşitlik (3.27),  $(i,j)$  ayrıtından 3., 7. ve 10. adımlarda, Eşitlik (3.28), 4. ve 8. adımlarda geçildiğini gösterir. Kısacası yön bakılmaksızın  $(i,j)$  ayrıtından toplamda 5 kez geçilmiş olunup, geçilen adımlar Eşitlik 3.29'da görüldüğü gibi 3, 4, 7, 8 ve 10 dur. Kısıt (3.20),  $x_{ij}^t + x_{ji}^t = 0$  ise  $y_{ij}^t = 0$  olur. Ancak,  $x_{ij}^t + x_{ji}^t = 1$  olduğu durumlarda  $t=3, 4, 7, 8, 10$  nolu adımlar için  $y_{ij}^t = 1$  değerini alabilir. Bu durumda  $(i,j)$  ayrıtından ilk geçilme anını tanımlayan  $y_{ij}^t$  değişkeninin değer alması için kısıt (3.21) aktifleşir.

**Çizelge 3.3.** Kısıt (3.21) açık gösterimi

Kısıt (3.21)		
$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) \leq M * (1 - y_{ij}^t)$		
t=2	$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) = 0$	$y_{ij}^t \in \{0,1\}$
t=3	$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) = 0$	$y_{ij}^t \in \{0,1\}$
t=4	$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) = 1$	$y_{ij}^t = 0$
t=5	$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) = 2$	$y_{ij}^t = 0$
·	·	·
·	·	·
·	·	·
t=10	$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) = 4$	$y_{ij}^t = 0$

Çizelge 3.3'e bakıldığında  $t=4$ . adımda  $(i,j)$  ayrıtından daha önce 1 kez, 5. adımda 2 kez ve en son 10. adımda 4 kez geçilmiş olup (3.21) nolu kısıttaki eşitsizliğin sol tarafı  $\geq 1$  olduğundan  $y_{ij}^t = 0$  olur. Bu durumda sadece  $t=2$  ya da  $t=3$  adımlarında ilk kez  $(i,j)$  ayrıtından geçilme durumu vardır. Kısıt (3.21),  $t$  adımından daha önceki adımlara bakmakta olup daha önceden  $(i,j)$  yolundan geçilip geçilmediğine göre  $y_{ij}^t$  ikili değişkenine değer biçmektedir. Kısıt (3.20) ise sadece  $t$  adımında  $(i,j)$  yolundan geçiş durumunu değerlendirir. Şöyle ki Çizelge 3.3'de 2. ve 3. adımların kendilerinden daha önce  $(i,j)$  yolundan hiç geçilmemesi ve  $t \geq 4$  adımlarının her birinde daha önce en az 1 kez geçilmiş olması,  $(i,j)$  ayrıtından ilk kez 2. ya da 3. adımda geçildiği sonucunu doğurur. Bu durumda Kısıt (3.20),  $t=2$  için  $x_{ij}^t + x_{ji}^t = 0$ ,  $y_{ij}^t = 0$ ,  $t=3$  için  $x_{ij}^t + x_{ji}^t = 1$ ,  $y_{ij}^t = \{0,1\}$  değer almasını sağlar. (3.22) nolu kısıt ise sadece tek bir  $t$  adımı için  $y_{ij}^t = 1$  yaptığından  $t=3$  için  $y_{ij}^t = 1$  değerini alır.

Özetle, (3.20)-(3.22) nolu kısıtlar birlikte çalışmaktadır.  $t \geq 4$  için Kısıt (3.21)'de eşitsizliğin sol tarafı  $\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) \geq 1$  olup, tüm  $t$  ler için ( $t=2, 3$  hariç)  $y_{ij}^t = 0$  olur. Kalan  $t$  adımlarının her biri için  $(i,j)$  ayrıtından geçiş yapıp yapılmadığı Kısıt (3.20) ile, herhangi bir  $(i,j)$  ayrıtı tek bir  $t$  adımı için  $y_{ij}^t = 1$  değerini alabilmekte olup, kalan adımlarda 0 değerini alması Kısıt (3.22) ile kontrol edilir. Kısıt (3.23),  $t$ . adımda  $h$  hiyerarşisine ait yolların tamamından geçilip geçilmediğini kontrol eder. (3.24) nolu kısıt,  $h$ . hiyerarşideki yollar tamamlanmadan  $(h+1)$ . hiyerarşiye ait yollardan geçilemez. Kısıt (3.25), değişkenler  $\{0,1\}$  tamsayı değerlerini alması gerektiğini belirtmektedir. Denklem (3.26), en kısa uzunluğu hedefleyen amaç fonksiyonunu açıklamaktadır.

### 3.4. Zaman-Bağımlı Hiyerarşik Çinli Postacı Problemi ve Matematiksel Modeli

Zaman-bağımlı rotalama problemlerinde hız modelleri, günün herhangi bir zaman diliminde aracın iki düğüm arasındaki seyahat süresinin belirlenmesi için kullanılabilir. Bu bölümde ZB-HÇPP için kullanılan hız fonksiyonu, gerekli veriler ve varsayımlar tartışılacaktır.

ZB-HÇPP için kullanılan hız modeli, Ichoua *et al.* (2003) ve Verbeeck *et al.* (2014) çalışmalarında kullandıkları hız modellerine dayanmaktadır. Ichoua hız modelinde yollar üç farklı kategoriye ayrılmış olup yolların kategorilendirilmesi işlemi rasgele olarak yapılmıştır. ZB-HÇPP'de yolların hiyerarşik sınıflandırılması nedeniyle rasgele atama ile kategorilendirilmesi çok doğru bir yaklaşım olmayacaktır. Dolayısıyla çalışmada, Verbeeck *et al.* (2014) tarafından sunulan yol kategorileri ve farklı zaman aralıklarını içeren hız modeli ZB-HÇPP problemine uyarlanmıştır.

Oluşturulan ZB-HÇPP hız modelinde bir aracın seyahat hızı (buna bağlı olarak seyahat süresi) günün farklı zaman aralıkları ve yol kategorileri ile ilişkilendirilmiştir. Çizelge 3.4'te verilen  $5 \times 4$  boyutlu hız matrisinde satırlar ayrıt kategorilerini, sütunlar zaman aralıklarını (periyotları) temsil etmektedir ( $C_{ayrıt} = 5, K = 4$ ). Eşit olmayan zaman aralıkları, sabah ve akşamları trafiğin yoğunluğunu kalan zamanlar ise normal seyrettiğini

yansıtmaktadır. Hız matrisinden her bir  $k \in \{T_1, T_2, T_3, \dots, T_K\}$  zaman aralığı ve ayrıt kategorisi  $C_{ayrıt}$  için  $v_{ijk}$  hız değeri belirlenir. HÇPP’de sıkça karşılaşılan kar küreme, çöp toplama vb. uygulamalar günün 24 saatinde yapılabildiği için, zaman aralıklarında bir sınırlama getirilmemiştir. Dolayısıyla gerçek hayat uygulamalarında olduğu gibi, araç ihtiyaç duyulduğu günün herhangi bir saatinde depo yerinden çıkarak, operasyon bitince geri dönebileceği varsayılmıştır.

Yolların kategorilendirilmesi işlemi ise hiyerarşik seviyeler dikkate alınarak yapılmıştır. Çalışmada hiyerarşik yolların, Verbeeck *et al.* (2008)’in tanımladığı ayrıt kategorileri ile ilişkilendirilmesinde kullanılan üç temel husus şunlardır:

- Birinci önceliğe sahip yollar sıklıkla şehir merkezindeki yollar olup her daim trafik yoğunluğu vardır. Dolayısıyla Verbeeck hız modelinde ilk satır olan ‘tüm gün trafik yoğun (always busy)’ ayrıt kategorisine birinci öncelikli yollar atanmıştır.
- En son hiyerarşik seviyeye ait yollar ise trafikte çok tercih edilmeyen yollardır. Bu yollar nadir kullanıldığından dolayı trafik yok denecek kadar az olup hız modelinde tanımlanan hız değerleri daha yüksektir. Dolayısıyla Verbeeck hız modelinde en son satır olan ‘nadiren seyahat edilen (seldom traveled) ayrıt kategorisine en son hiyerarşik seviyeye ait yollar atanmıştır.
- Arada kalan hiyerarşik seviyeli yollar ise  $(E_m, m = 2, \dots, H-1)$ , Verbeeck hız modelinde ‘sabah saatlerinde yoğunluk (morning peak)’, ‘akşam saatlerinde yoğunluk (evening peak)’ veya ‘sabah ve akşam saatlerinde yoğunluk (two peaks both morning and evening)’ kategorilerine rasgele olarak atanmıştır. Yani arada kalan hiyerarşik yollar, sabah yoğunluğu, akşam yoğunluğu ya da her iki zamandaki yoğunluğu temsil edebilirler.

Özetle, hiyerarşik sınıf sayısı her zaman sabit olmayıp problem türüne göre değişmektedir. Bu sebeple,  $G(V,E)$  şebekesi için tanımlı toplam hiyerarşi sayısı kadar ayrıt kategori sayısı dikkate alınır. Her problem tipi için ilk ve son satırlara sırasıyla birincil öncelikli ve en son öncelikli yolların ataması yapılır.



Çizelge 3.4'den  $v_{ijk}$  hız bilgisine şu şekilde ulaşılır.  $i$  düğümünden çıkan araç  $j$  düğümüne giderken ihtiyaç duyulan hız bilgisi için;

- $(i,j)$  yolunun hiyerarşik seviyesine bakılır. Eğer 1. öncelikli yol ise tabloda 1. satır, en son öncelikli yol ise tabloda 5. satır dikkate alınır.
- $i$  düğümünden kalkış zamanı ( $wd_i$ ) bize aracın hangi zaman diliminde olduğu bilgisini verir. Örneğin aracın kalkış zamanı  $7^{00}$ – $9^{00}$  saatleri arasında ise araç,  $k=1$  zaman dilimindedir ve 1. sütun dikkate alınır.

Bu şekilde  $(i,j)$  yolundan geçerken bu yolun hiyerarşik seviye bilgisi (satırlar) ve  $i$  yolundan çıkış zamanı bilgisi ile belirlenen zaman dilimi (sütunlar) kesişimi (satır ve sütun) bize hız bilgisini verir.

3 hiyerarşili bir test problemi örneğinde;

- $i$  den  $j$  ye giderken  $(i,j)$  1. hiyerarşik yol ise ve  $i$  düğümünden kalkış zamanı  $17^{30}$  ise;

$v_{ijk} = 0,5$ 'tir (1. satır ve 3. sütun kesişimi).

- $i$  den  $j$  ye giderken  $(i,j)$  2. hiyerarşik yol ise ve  $i$  düğümünden kalkış zamanı  $20^{30}$  ise;

$v_{ijk} = 0,5$ 'tir (3. satır (çünkü ara hiyerarşi) ve 1. sütun kesişimi).

- $i$  den  $j$  ye giderken  $(i,j)$  3. hiyerarşik yol ise ve  $i$  düğümünden kalkış zamanı  $16^{00}$  ise;

$v_{ijk} = 1,5$ 'tir (5. satır (son hiyerarşi olduğu için en son satır) ve 2. sütun kesişimi).

**EK 1**'de 2, 3, 4 ve 5 hiyerarşili test problemleri için ele alınan hız matrisleri verilmiştir.

**Çizelge 3.4.** ZB-HÇPP için kullanılan hız matrisi

Ayrıt kategorileri	Zaman aralıkları			
	Sabah Yoğunluk 7 <sup>00</sup> – 9 <sup>00</sup>	Normal 9 <sup>00</sup> – 17 <sup>00</sup>	Akşam Yoğunluk 17 <sup>00</sup> – 19 <sup>00</sup>	Normal 19 <sup>00</sup> – 7 <sup>00</sup>
Birincil öncelikli	0,5	0,81	0,5	0,81
Ara öncelikli	0,5	0,7	1,0	1,5
	0,5	1,5	0,5	1,5
	1,0	1,5	0,5	0,7
Son öncelikli	1,5	1,5	1,5	1,5

$i$  ve  $j$  düğümü arasındaki seyahat zamanının hesaplanması için, bu iki düğüm arasındaki mesafe matrisi ( $c_{ij}$ ), seyahat hız matrisi ( $v_{ijk}$ ) ve  $i$  düğümünden kalkış zamanı ( $w_{di}$ ) verilerine ihtiyaç duyulur. Kalkış zamanı bilgisi o ayrıtın hangi zaman diliminde olduğu bilgisini verir. Dolayısıyla ( $i,j$ ) yolunun ait olduğu ayrıt kategorisi ve zaman aralığı bilgileri ile ( $i,j$ ) yolunda seyahat eden aracın hız bilgisi Çizelge 3.4'den elde edilir. Rotalama boyunca araç bir zaman diliminden başlayarak başka bir zaman diliminde seyahatine devam eder. Yolların ve zaman dilimlerinin değişmesi aracın farklı hızlarla seyahat etmesine olanak sağlar. İki düğüm arasındaki seyahat zamanının hesaplanması iki düğüm arasındaki mesafe ve hız ile doğrudan ilişkilidir. Bu yöntem Ichoua *et al.* (2003) çalışmasında anlatılmıştır. Araç,  $i$  düğümünden  $t_0 \in [t_k^-, \bar{t}_k]$  zamanında ayrılır ve ( $i,j$ )  $\in C_{ayrıt}$  yolu boyunca seyahat ederek  $j$  varış noktasına varır.  $t$ , iki düğüm arasında geçen süreyi,  $t'$  ise  $j$  düğümüne varış zamanını temsil etmektedir. Şekil 3.8'de Ichoua *et al.* (2003) tanımladığı seyahat süresi hesaplama algoritması verilmiştir.

---

```

 $t_0$  tanımla
 $c_{ij}$  tanımla
 $t' = t + (c/v_{ijk})$ 
While  $t' > \bar{t}_k$  do
 $c \leftarrow c - v_{ijk}(\bar{t}_k - t)$ 
 $t \leftarrow \bar{t}_k$ 
 $t' \leftarrow t + (c/v_{ijk+1})$ 
 $k \leftarrow k+1$ 
end
seyahat süresi =  $(t' - t_0)$ 

```

---

### Şekil 3.8. Seyahat süresi hesaplama algoritması

Bu çalışmada aracın bulunduğu düğümden çıkış zamanına göre tanımlanan hız değerinin, aracın  $j$  inci düğüme seyahati boyunca değişmediği varsayılmıştır. Bu varsayım altında geliştirilen araç seyahat süresi hesaplama algoritması ilerleyen bölümlerde detaylı anlatılmıştır.

Bu kısımda ZB-HÇPP için geliştirilen karma tam sayılı matematiksel formülasyonu verilmiştir. Model, yönsüz bir  $G(V,E)$  şebekesi için oluşturulmuştur.  $V = \{i,j=1,2, \dots, n\}$ , düğümler kümesini vermektedir.  $i_0$  başlangıç noktası olup araç bu noktada seyahate başlayıp işi bittikten sonra aynı noktaya dönüş yapar.  $E = \{(i,j): i,j \in V \text{ and } i \neq j\}$ , yönsüz ayrıtlar kümesini,  $C = \{c_{ij} : i, j \in V\}$  düğümler arası mesafeleri vermektedir ( $c_{ij} = c_{ji}$ ). Hiyerarşik bir şebekede, ayrıtlar  $E^1, E^2, \dots, E^h$  hiyerarşik önceliklerine göre ayrılırlar ( $h = 1, 2, \dots, H$ ). Her  $i \neq j$  için  $E_i \cap E_j = \emptyset$ .

Modelde kullanılan indisler, kümeler, parametreler ve karar değişkenleri Çizelge 3.5’de verilmiştir.

**Çizelge 3.5.** ZB-HÇPP modelinde kullanılan kümeler, parametreler ve karar değişkenleri

<b>İndisler</b>	
$i$	Başlangıç düğümleri,
$j$	Bitiş düğümleri,
$t$	Rotaya ait yollardan geçiş sıralarını (adımları),
$k$	Ayrıtlardan geçilen zaman dilimlerini,
$h$	Ayrıtların ait oldukları hiyerarşik seviyeleri temsil eder.
<b>Kümeler</b>	
$V$	Düğümler kümesi $i,j=\{1, 2, \dots, n\}$ ,
$H$	Hiyerarşik seviyeler kümesi $\forall h \in H$ ,
$K$	Zaman dilimleri kümesi $\forall k \in K$ ,
$L$	Adım sayısı $\forall t \in L$ ,
$E_h$	$h$ . hiyerarşik sınıfa ait ayrıtlar (yollar) kümesi,
$E$	Şebekedeki tüm ayrıtların kümesi $E = \{E_1 \cup E_2 \cup \dots \cup E_h\}$ ,
$\ E_h\ $	$h$ . hiyerarşik sınıfa ait toplam yol sayısı,
$\delta_h(i)$	$i$ düğümünden çıkan $E_h$ kümesine ait yollar, $\delta_h(i) = \{j \mid (i,j) \in E_h\}$ .
$\delta(i)$	$i$ düğümünden çıkan bütün yollar kümesi.
<b>Parametreler</b>	
$C_{ij}$	$(i,j)$ ayrıtlarının mesafe matrisi,
$B_{ij}$	$(i,j)$ ayrıtlarının bağlantı matrisi,
$O_{ij}$	$(i,j)$ ayrıtlarının öncelik matrisi,
$n$	Toplam düğüm sayısı,
$l_k$	$k$ zaman diliminin başlangıç zamanı,
$u_k$	$k$ zaman diliminin bitiş zamanı,
$v_{ijk}$	$k$ . zaman aralığında $(i,j)$ yolundan geçen aracın seyahat hızı,
$w_0$	Seyahat başlangıç zamanı
$M$	Büyük pozitif sayı.
<b>Karar Değişkenleri</b>	
$x_{ij}^t$	$\begin{cases} 1 & \text{Eğer } t. \text{ adımda } (i,j) \text{ ayrıtlarından geçilmişse} \\ 0 & \text{diğer,} \end{cases}$
$y_{ij}^t$	$\begin{cases} 1 & \text{Eğer } t. \text{ adımda } (i,j) \text{ ayrıtlarından ilk kez geçilmişse} \\ 0 & \text{diğer,} \end{cases}$
$\phi_{ht}$	$\begin{cases} 1 & \text{Eğer } t. \text{ adımda } h. \text{ hiyerarşiye ait bütün yollardan geçilmişse} \\ 0 & \text{diğer,} \end{cases}$
$z_{ijt}^k$	$\begin{cases} 1 & \text{ } i \text{ düğümünden } j \text{ düğümüne } t. \text{ adımda } k. \text{ zaman diliminde geçilmişse} \\ 0 & \text{diğer,} \end{cases}$
$wd_i^t$	: sürekli karar değişkeni, $i$ düğümünden $t$ . adımda hareket zamanı,
$wa_i^t$	: sürekli karar değişkeni, $i$ düğümüne $t$ . adımda varış zamanı,
$T_{ij}^t$	: $t$ . adımda $i$ düğümünden $j$ düğümüne seyahat süresi.

Yukarıda verilen bilgiler doğrultusunda ZB-HÇPP'nin tamsayılı doğrusal programlama modeli aşağıda verildiği gibidir:

$$\sum_{t=1}^L (x_{ij}^t + x_{ji}^t) \geq 1 \quad \forall (i, j) \in E \quad (3.30)$$

$$\sum_i x_{ij}^{t-1} = \sum_s x_{js}^t \quad \forall j, t \quad t > 1 \quad i, j, s \in V \quad (3.31)$$

$$\sum_{(i,j)} x_{ij}^1 \geq 1 \quad (i, j) \in \delta_1(i_0) \quad (3.32)$$

$$\sum_j x_{ji}^t \geq 1 \quad t = t_{son}, \quad i = i_0 \quad (3.33)$$

$$\sum_{(i,j) \in E} x_{ij}^t = 1 \quad \forall t \quad (3.34)$$

$$x_{ij}^t + x_{ji}^t \geq y_{ij}^t \quad \forall i, j, t \quad i < j \quad (3.35)$$

$$\sum_{t' < t} (x_{ij}^{t'} + x_{ji}^{t'}) \leq M * (1 - y_{ij}^t) \quad \forall i, j, t \quad i < j \quad (3.36)$$

$$\sum_{t=1}^L y_{ij}^t = 1 \quad \forall (i, j) \in E \quad i < j \quad (3.37)$$

$$\sum_{t' \leq t} \sum_{(i,j) \in E_h} y_{ij}^{t'} \geq \|E_h\|^* \phi_{ht} \quad \forall h, t \quad (3.38)$$

$$\sum_{t' \leq t} \sum_{(i,j) \in E_{h+1}} \left( x_{ij}^{t'} + x_{ji}^{t'} \right) \leq M^* \phi_{ht} \quad \forall t, h = 1, \dots, H-1 \quad (3.39)$$

$$wd_i^{1'} = w_0 \quad i = i_0 \quad (3.40)$$

$$v_{ijk} * (wa_j^t - wd_i^t) \leq c_{ij} + 2 * M - M * (x_{ij}^t + z_{ijt}^k) \quad \forall i, j, t, k \quad (3.41)$$

$$v_{ijk} * (wa_j^t - wd_i^t) \geq c_{ij} * z_{ijt}^k \quad \forall i, j, t, k \quad (3.42)$$

$$wd_j^t = wa_j^{t-1} \quad \forall j, t \quad t > 1 \quad (3.43)$$

$$T_{ij}^t \leq M * x_{ij}^t \quad \forall i, j, t \quad (3.44)$$

$$T_{ij}^t \leq (wa_j^t - wd_i^t) \quad \forall i, j, t \quad (3.45)$$

$$T_{ij}^t \geq (wa_j^t - wd_i^t) - M * (1 - x_{ij}^t) \quad \forall i, j, t \quad (3.46)$$

$$wd_i^t + M * z_{ijt}^k \leq u_k + M \quad \forall i, j, t, k \quad (3.47)$$

$$wd_i^t \geq z_{ijt}^k * l_k \quad \forall i, j, t, k \quad (3.48)$$

$$\sum_k z_{ijt}^k = x_{ij}^t \quad \forall i, j, t \quad (3.49)$$

$$x_{ij}^t, y_{ij}^t, \phi_{ht}, z_{ijt}^k \in \{0,1\}, T_{ij}^t, wd_i^t, wa_i^t \geq 0 \quad \forall i, j, t, h, k \quad (3.50)$$

Kısıtları altında

$$\text{Minimum} \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^L T_{ij}^t \quad (3.51)$$

(3.30)-(3.39) nolu kısıtlar HÇPP'nin temel kısıtları olup Bölüm 3.3.1'de detaylı bir şekilde anlatılmıştır.

Sabit uzaklık yerine zaman bağımlı seyahat sürelerinin yer aldığı temel zaman bağımlılığı kavramları (3.40)-(3.49) nolu kısıtlarda verilmiştir. Kısıt (3.40), aracın  $i_0$  başlangıç düğümünden hareket zamanını (kalkış zamanını) verir.  $t=1$ . adımda aracın tanımlı kalkış zamanında ( $w_0$ ) yola çıkmasını sağlar. (3.41) ve (3.42) nolu kısıtlar, iki düğüm arasındaki geçiş süresinin ( $wa_j^t - wd_i^t$ ), hız ve katedilen mesafeyle bağlantılı olmasını sağlar. Kısıt (3.43),  $t$ . adımda ( $t>1$ )  $j$  düğümünden kalkış zamanının bir önceki ( $t-1$ ). adımda  $j$ . düğümüne varış zamanına eşit olmasını garanti eder.

$T_{ij}^t$  her bir  $t$ . adımda ( $i,j$ ) düğümü arasında seyahat süresini temsil eder. (3.44)-(3.46) nolu kısıtlar düğümlere giriş ve çıkış zamanları arasındaki farkların düğümler arası seyahat süreleri ile bağlantılı olmasını sağlar. Şöyle ki, her bir  $t$ . adımda ki seyahat süresi  $T_{ij}^t$ ,  $x_{ij}^t$ , 0-1 binary değişkeninin, düğümlerdeki varış ve kalkış zamanlarının farkları ile çarpılarak hesaplanır. Yani,  $T_{ij}^t = (wa_j^t - wd_i^t) * x_{ij}^t$  olarak ifade edilir. Her  $t$  adımında sadece bir ( $i,j$ ) ayrıttandan geçildiğinden dolayı,  $x_{ij}^t = 0$  olduğunda, doğal olarak seyahat süresi

de 0 olur. Amaç fonksiyonu toplam seyahat süresini en küçükleme olduğundan formülasyon gösterimi şu şekilde ifade edilir:

$$\text{Minimum} \sum_{i=1}^n \sum_{j=1}^n \sum_{t=1}^L (wa_j^t - wd_i^t) * x_{ij}^t \quad (3.52)$$

(3.52) nolu ifade iki karar değişkeni çarpımı olduğundan model non-lineer olmaktadır.

(3.44)-(3.46) nolu kısıtlar,  $(wa_j^t - wd_i^t) * x_{ij}^t$  çarpım ifadesinin  $T_{ij}^t$  pozitif sürekli değişkenine atanması ile modeli lineerleştirir.

(3.47) ve (3.48) nolu kısıtlar, zaman dilimlerinin alt ve üst sınırlarının aracın düğümlerden kalkış zamanlarıyla bağlantılı olmasını sağlar.

(3.49) nolu kısıt, eğer  $(i,j)$  yolundan  $t$ . adımda geçilmişse, aracın mutlaka bir zaman diliminde başladığını garantilemektedir. Kısıt (3.50) karar değişkenlerini açıklar. Denklem (3.51), en kısa süreyi hedefleyen amaç fonksiyonunu açıklamaktadır.

### 3.5. ZB-HÇPP Çözümü için Önerilen Meta-sezgisel Yaklaşımlar

Kombinatoriyal problemler, problemin boyutu arttıkça çözüm zamanı üstel artan bir yapıya sahiptir. Bu durum araştırmacıları sezgisel ve meta-sezgisel algoritmalara odaklanmaya yöneltmiştir. Önerdiğimiz matematiksel model literatürde şimdiye kadar hiç olmamasına rağmen, gerçek hayat problemleri açısından çok küçük sayılacak problemlere çözüm üretememiştir. Dolayısıyla, problemin boyutu büyüdükçe geliştirilen model ile herhangi bir uygun çözüm dâhi elde edilememektedir. Bu durum, optimum çözümü her zaman vermese de optimuma yakın çözümler üretebilen sezgisel veya meta-sezgisel algoritmaların geliştirilmesini kaçınılmaz kılmaktadır.



### 3.5.1. Çözüm gösterimi

Meta-sezgisel algoritmelerde çözümlerin kodlanması işlemi, sonuçların kalitesi açısından büyük önem taşımaktadır. Zira kullanılacak kodlama tekniği ile problem tanıtılmış olacaktır. Sezgisel algoritmelerde her bir kromozom bir uygun çözümü temsil etmektedir. Başlangıçta doğru ve tam ifade edilemeyen aday çözümler ile ya kalitesi düşük ya da uygun olmayan sonuçlar elde edilmektedir. ZB-HÇPP karmaşık bir yapıya sahip olduğundan dolayı meta-sezgisel yaklaşımların kullanılarak yeni çözümlerin üretilmesi işlemleri de karmaşık bir hal almaktadır. ZB-HÇPP’de uygun çözümler rotalar olup çözüm gösterimleri doğrudan bu elde edilen rotalar ile yapılabilir. Ancak bu gösterim, kenarlardan birden fazla kez geçilebilme esnekliğinden dolayı farklı uzunluklarda kromozomların oluşmasına sebep olabilir. Ayrıca, rotalar hiyerarşik önceliklere göre belirlendiğinden, bu yapının bozulması olursuz çözümlere neden olmaktadır. Tüm bu nedenlerden dolayı sezgisel algoritmelerin en temel ve önemli basamağı olan çözüm gösterimlerinin rotalar ile yapılması; ilerleyen bölümlerde anlatılan çaprazlama, mutasyon ve komşu çözüm üretme gibi operatörlerin tasarlanması açısından uygun ve pratik bir yaklaşım olmadığı yapılan çalışmalar neticesinde gözlenmiştir.

GA ve TB meta-sezgiselleri literatürde birçok çizelgeleme ve düğüm rotalama problemlerine uyarlanmış ve problemlerinin birçoğunda çözüm gösterimleri permütasyon kodlama tekniği kullanılarak yapılmıştır (Jung and Haghani (2001); Moon *et al.* (2002); Wang and Zheng (2003); Gaafar and Masoud (2005); Hart *et al.* (2005); Jiang *et al.* (2005); Razali (2015); Kumar and Panneerselvam (2015); Mukhairez and Maghari (2015); Mohammed *et al.* (2017); Samadi-Dana *et al.* (2017)).

Permütasyon kodlama ile  $n$  adet şehir/iş sayısı verildiğinde (gen);

- Her gen bir şehri/işi temsil eder.
- Genlerin dizilimi (permütasyon) o anki “turu/iş sırasını” temsil eder.
- $n=5$  genli örnek çözüm gösterimleri şu şekildedir;

1. kromozom: (2 3 0 1 4)
2. kromozom: (0 2 3 1 4)
3. kromozom: (4 3 1 0 2)

Gezgin satıcı ve araç rotalama problemleri birer düğüm rotalama problemi olduklarından dolayı düğüm sayısı kadar gen tanımlanıp bunların farklı dizilimleri ile kromozomlar oluşturulabilir. Bu problemlerde kromozom uzunlukları sabit olup problemdeki düğüm sayısına eşittir.

Literatürde yapılan bir takım ayrıt rotalama problemlerinde de çözüm gösterimleri,  $n$  adet kenarın permütasyonunu içeren bir sayı dizisi şeklinde ifade edilmiştir. Moreira and Ferreira (2010) GSP'ye dönüştürülen KPP'nin çözümüne yönelik yeni bir GA yaklaşımı tanımlamışlardır. Çalışmada tüm geçilmesi gerekli kenarlar numaralandırılmış ve her bir kromozomun genleri bu numaralar ile temsil edilmiştir. Yu and Lin (2015) zaman-bağımlı ödül toplama ayrıt rotalama problemi için açgözlü arama sezgiseli geliştirmişlerdir. Çözümler  $\{1, 2, \dots, n\}$  kümesi ile belirtilen  $n$  adet ödüllü ayrıtların permütasyonunu içeren bir sayı dizisi ile temsil edilmiştir. Rabbani *et al.* (2016) bir ayrıt rotalama problemi olan kapasiteli  $k$  araçlı kırsal postacı probleminin çözümü için çok amaçlı TB algoritması geliştirmişlerdir. Kromozom uzunluğunu, geçilmesi gerekli yolların sayısı ve araçların sayısının toplamından 1 (bir) eksik olarak tanımlamışlardır. Her bir çözüm tüm araçlar ve rotalarının rassal permütasyon dizilimi ile gösterilmiştir. Tirkolae *et al.* (2016) kapasiteli ayrıt rotalama problemi için geliştirdikleri hibrit meta-sezgisel algoritmada geçilecek yollar sıralı bir dizi şeklinde gösterilmiştir.

Literatür çalışmalarına bakıldığında son yıllarda ayrıt rotalama problemlerinde permütasyon kodlama tekniğinin kullanıldığı görülmektedir. ZB-HÇPP bir ayrıt rotalama problemi olup çözüm gösterimi çalışmanın ilk ve en önemli aşaması olmuştur. Bu çalışmada, ZB-HÇPP birer iş çizelgeleme problemi olarak ele alınmış ve aday çözümler permütasyon kodlama tekniği ile gösterilmiştir. Bu teknik ile tüm ayrıtlar numaralandırılarak yapılması gerekli birer "iş" olarak tanımlanmıştır. Aşağıda şebeke probleminin iş çizelgeleme problemine dönüştürülme adımları detaylı olarak verilmiştir:

**Adım 1.** Verilen  $G(V,E)$  şebekesinde,  $\forall(i,j) \in E$  ayrıtı yapılması gerekli birer iş ya da görev olarak tanımlanır ve  $\tau^{(i,j)}$  sembolü ile gösterilir. İşlerin sayısı toplam kenar sayısına eşittir. ( $\tau=1,2,\dots,Z$ .  $Z=||E||$ ).

**Adım 2.**  $(i,j)$  ayrıtının hiyerarşik seviyesine bakılarak, bu kenara karşı gelen  $\tau^{(i,j)}$  işinin tüm öncül işleri belirlenir ve  $Z(pt)$  kümesine atanır. Eğer  $(i,j) \in E_1$  ise, bu kenara karşılık gelen  $\tau^{(i,j)}$  işinin öncülleri yoktur.  $(i,j) \in E_h$ ,  $h > 1$  olması halinde,  $\tau^{(i,j)}$  işinin öncül işler kümesi,  $(i,k) \in E_{h-1}$  ayrıtılarına atanan görevlerin tamamından oluşur. Korteweg and Volgenant (2006) çalışmasında sunulan şebekenin (Şekil 3.7’de verilen) iş çizelgeleme problemine dönüştürülmüş hali Çizelge 3.6’da verilmiştir.

**Çizelge 3.6.** Şebekenin iş çizelgeleme problemine dönüştürülmesi

İş/görev no	Ayrıt $(i,j)$	Hiyerarşik seviye $h(i,j)$	Öncül işler kümesi
1	(1,2)	1	-
2	(2,3)	1	-
3	(2,4)	1	-
4	(2,5)	1	-
5	(4,6)	2	{1,2,3,4}
6	(5,6)	2	{1,2,3,4}
7	(6,7)	2	{1,2,3,4}
8	(1,3)	3	{5,6,7}
9	(1,4)	3	{5,6,7}
10	(3,5)	3	{5,6,7}
11	(4,5)	3	{5,6,7}
12	(5,7)	3	{5,6,7}

Şekil 3.7’de verilen şebekede toplam 12 adet geçilmesi gereken yol mevcuttur. Her yol bir iş olarak tanımlandığından, toplam yol sayısı kadar iş sayısı vardır. Bir çözüm gösterimi, E kenar sayısı uzunluğunda bir tam sayı dizilimi şeklindedir. Çözümü

oluşturan her bir eleman bir işi ( $\tau$ ) temsil etmektedir. Çözümlerdeki elemanların dizilimi, önceliklerine göre atanan işlerin sırasından oluşmaktadır. Şöyleki; {1,2,3,4} nolu işlerin atanması en başta yapılır. Zira bu işler 1. hiyerarşik seviyeye ait yolları temsil etmekte olup öncülleri yoktur. Bu işlerin tamamı atandıktan sonra (tamamının atanması 1. hiyerarşik yolların hepsinden geçildiği anlamına gelmektedir) {5,6,7} numaralı işlerin atanması yapılır. Bu işlerin atanması tamamlandıktan sonra artık {8,9,10,11,12} nolu işlerin öncülleri kalkacağından dolayı, bunların atanma işlemi gerçekleştirilir. Sonuç olarak, oluşturulan çözümler ZB-HÇPP'nin hiyerarşik yapısını sağlayan sabit uzunluktaki yapılarıdır.

İşlerin öncelikleri dikkate alınarak rasgele atanmaları ile farklı çözümler oluşturulabilir. Ancak tamamen rassal atama, olurlu fakat kötü çözümler üretilmesine neden olur. Şöyle ki, bir  $P=[1\ 2\ 3\ 4\ 5\ 7\ 6\ 9\ 8\ 11\ 10\ 12]$  kromozom diziliminde işler önceliklerine göre dizilmiştir yani çözüm uygun çözümdür. Fakat ard arda gelen işlerin karşılık geldiği yollar birbirleriyle direkt bağlantılı değildir. Örneğin,  $\tau^{(5,6)} = 6$  nolu işten sonra  $\tau^{(1,4)} = 9$  nolu iş gelmektedir. Bunun anlamı, (5,6) yolundan sonra (1,4) yolundan geçileceğidir. Bu yollar birbirleriyle direkt bağlantılı değildir. Dolayısıyla sonuç olurludur fakat kötü bir çözümdür. Meta-sezgiseller'de aday çözümlerin iyi ifade edilmiş dizilerle oluşturulması, sezgiselin kalitesini ve hızını doğrudan etkilemektedir. Bu sebeple etkin bir başlangıç çözümü için geliştirilen yöntemin adımları şu şekildedir:

Adım 1.  $P = [(\_)]$  çözüm dizisinin ilk elemanına iş atanması yapılır.  $i_0=1$  nolu düğüm, başlangıç noktası olduğundan dolayı sadece bu düğümle bağlantılı yol/yollardan birinin atanması yapılır. Çizelge 3.6'da öncülü olmayan sadece (1,2) ayrıtı olup bu ayrıtı karşılık gelen  $\tau^{(1,2)} = 1$  nolu iş  $P$  dizisine atanır.

Adım 2. Öncülü olmayan işleri içeren  $S_T$  kümesi oluşturularak sıradaki atanacak işler listesi belirlenir.

Adım 3.  $P$  çözüm dizisinin her  $t$  inci elemanına iş atanması yapılmadan önce bir önceki ( $t-1$ ). elemana atanmış işe bakılır. ( $t-1$ ). elemana atanmış işin direkt bağlantılı olduğu diğer işler kümesi  $S_C$  oluşturulur. Şöyleki, her  $t$  adımında  $\tau = P(t-1)$  işi ile direkt bağlantılı işlerden oluşan  $S_C$  kümesi oluşturulur.

**Adım 4.**  $S$  kümesinden rassal bir iş  $(\tau^{(i,j)} \in S)$  seçilerek  $P(t)$  dizisine eklenir. ( $S = \{S_T \cap S_C\}$ ).

Eğer  $S = \emptyset$  ise,  $S_T$  kümesinden rasgele olarak bir iş seçilir.

**Adım 5.**  $P$  çözüm dizilimi tamamlanıncaya kadar 2, 3 ve 4. adımlar tekrar edilir.

12 işli probleme ait örnek bir çözüm gösterimi (kromozom) Şekil 3.9'da verilmiştir.

1	3	2	4	6	5	7	12	11	9	8	10
---	---	---	---	---	---	---	----	----	---	---	----

(1,2) (2,4) (2,3) (2,5) (5,6) (4,6) (6,7) (7,5) (5,4) (4,1) (1,3) (3,5)

**Şekil 3.9.** Örnek bir çözüm gösterimi

Geliştirilen bu yaklaşımın gerekliliği ve avantajları şu şekildedir;

- ZB-HÇPP'nin karmaşık yapısına uygun olarak çok daha kısa sürede ve kolay bir şekilde aday çözümler oluşturulabilmektedir.
- Çözüm dizilerinin (kromozomların) uzunlukları sabit olup kenar sayısı ( $E$ ) kadardır.
- Literatürde rotalama problemlerinde yaygın kullanılan permütasyon kodlama tekniği, ZB-HÇPP'ye uyarlanmıştır.
- Çaprazlama, mutasyon ve komşu çözüm üretme operatörleri bireylerin uygunlukları bozulmadan yapılmıştır.

### 3.5.2. Uygunluk fonksiyonu değerinin hesaplanması

Her bir çözümün uygunluk değeri, denklem (3.51)'de verildiği gibi toplam seyahat süresinin en küçüklenmesidir. Yani amaç fonksiyonu uygunluk fonksiyonu olarak ele alınmıştır.

$(i,j)$  kenarından geçme süresi, iki düğüm arasındaki mesafe ( $c_{ij}$ ) ve hız ( $v_{ijk}$ ) ile doğrudan ilişkilidir. Rota boyunca, araç bir zaman diliminde seyahatine başlar ve yolculuk boyunca başka bir zaman dilimine geçiş yapar. Yolların ve zaman aralıklarının değişmesi aracın hızının değişmesine yol açar.  $i$  düğümünden kalkış zamanı ( $w_{di}$ ) aracın bulunduğu zaman

aralığı bilgisini verir.  $(i,j)$  ayrıtın ait olduğu yol kategorisi bilgisi ile hız matrisinden aracın hız bilgine ulaşılır. Aracın  $i$  düğümünden belirlenen bir başlangıç hızıyla hareket ettiği ve varacağı  $j$  düğümüne seyahati boyunca aynı hızda devam ettiği varsayımı altında bir R rotasının toplam seyahat süresinin  $t^{(R)}$  hesaplama algoritması Şekil 3.10'da verilmiştir.

---

```

R ← rota
 $w_{d0}$  ← Seyahat başlangıç zamanı
 $t^{(R)}$  ← 0, R rotasının toplam seyahat süresi
for n=1:2:(length(R)-1)
    i=R(n)
    j=R(n+1)
    c=c(i,j)
     $wd_i$  ← i düğümünden hareket zamanı
    h ← (i,j) ayrıtının yol kategorisini tanımla,  $h=h(i,j)$ .
    k ←  $wd_i$ 'ye göre zaman aralığını tanımla  $k=1,\dots,4$ 
    v ← HızMatrisi(h,k)
     $t^{(i,j)}$  ← c/v
     $wd_j = wd_i + t^{(i,j)}$ , j düğümüne varış zamanını hesapla.
     $t^{(R)} = t^{(R)} + t^{(i,j)}$ 
end
Çıktı  $t^{(R)}$ 

```

---

Şekil 3.10. Uygunluk değeri hesaplama algoritması

ZB-HÇPP bir rotalama problemi olduğundan dolayı işlerin sıralanmasından oluşan kromozomun doğrudan uygunluk değerini hesaplamak mümkün değildir. Çünkü sıralı dizilen işlerin karşılık geldiği yollar tam bir rota oluşturmamaktadır. Şöyle ki, Şekil 3.9'da verilen kromozomun mevcut rotası,  $(i,j)$  yollarının dizilimi ile oluşturulduğunda düğümler arası akış bozulmaktadır. Kırmızı ile belirtilen yerlerde rotanın akışı sağlanmamaktadır.

Rota = (1,2) (2,4) (2,3) (2,5) (5,6) (4,6) (6,7) (7,5) (5,4) (4,1) (1,3) (3,5)

Dolayısıyla ilk olarak sıralı işlerle ifade edilen kromozomların rotalara dönüştürülmesi gerekir. Daha sonra oluşturulan rotaların toplam seyahat süresi hesaplanır.

Bir çözümün (kromozomun) uygunluk değerinin hesaplanması işlemi 3 adımda gerçekleştirilir.

- 1- Rota oluşturulur.
- 2- Oluşturulan rotanın en son düğümünden başlangıç düğümüne dönüş yolu eklenir.
- 3- Nihai rotanın uygunluk fonksiyonu değeri hesaplanır.

Bir kromozomun uygunluk değerinin hesaplanması için rota oluşturma yöntemi geliştirilmiştir. Tüm aday çözümler, Bölüm 3.5.2.a'da anlatılan bu yöntem kullanılarak uygun rotalara dönüştürülmüştür.

### 3.5.2.a. Rota oluşturma

İşlerin önceliklerine göre sıralanması şeklinde tanımlanan kromozomların uygunluk değerlerinin hesaplanması için ilk adım kromozomların rotalarının oluşturulması işlemidir. Yani ZB-HÇPP bir ayrıt rotalama problemi olduğundan dolayı her bir yoldan geçilme zorunluluğu, yapılması gerekli birer iş olarak tanımlanmıştı. Bu yaklaşım ile hem tamamlanmamış iş kalmamakta (yani tüm yollardan geçilmekte) hem de işlerin öncelikleri dikkate alınarak atama yapıldığından hiyerarşik ilişki göz ardı edilmemektedir.

Ele alınan bir P kromozomunun R rotasına dönüştürülmesi adımları ve bu süreçte kullanılan semboller aşağıda verilmiştir.

Semboller:

$P_{uzunluk}$  : P kromozomunun uzunluğu

$R$  : nihai rota

$R_{mevcut}$  : tamamlanmamış  $R$  rota dizisi

$R_T$  : geçici rota dizisi

$R_{dönüş}$  :  $R_{mevcut}$  dizisinin en son elemanından 1 nolu başlangıç düğümüne dönüş rotası

$\tau^{(i,j)}$  :  $(i,j)$  ayrıtına atanmış iş no,  $\tau^{(i,j)} = 1, \dots, Z$ .

$p_{son}$  :  $R_{mevcut}$  dizisinin en son elemanı, ( $p_{son} \in V$ ).

$h_{mevcut}$  :  $R_{mevcut}$  dizisinin hiyerarşik seviye bilgisini verir.  $R_{mevcut}$  dizisinde  $\forall (i,j) \in E_h$  yollarının tamamından geçilmişse,  $h_{mevcut} = h+1$  dir. Eğer  $R_{mevcut}$  dizisinde  $h$ . hiyerarşik seviyeli eksik yol/yollar ( $(i,j) \in E_h$ ) varsa,  $h$ . hiyerarşik yolların tamamından geçilmediği anlamına gelir ve  $h_{mevcut} = h$  olur. HÇPP'nin temel yapısında hiyerarşik sıralama olup,  $h$ . hiyerarşik yollar tamamlanmadan  $h+1, h+2, \dots, H$  hiyerarşili yollardan geçilemez. O yüzden rota tamamlama işleminde  $h_{mevcut}$  bilgisi gereklidir.

$t(p_{son}, i)$ :  $p_{son}$  düğümünden  $i$  düğümüne seyahat süresi.

$t(p_{son}, j)$ :  $p_{son}$  düğümünden  $j$  düğümüne seyahat süresi.

$t^{(R)}$  : R rotasının toplam seyahat süresi

$h(p_{son}, i)$ :  $(p_{son}, i) \in E$  ayrıtının hiyerarşik seviyesi.

$h(p_{son}, j)$ :  $(p_{son}, j) \in E$  ayrıtının hiyerarşik seviyesi.

$I$  : İlişki matrisi ( $n \times n$ ). Eğer  $i$  ve  $j$  düğümleri direkt bağlantılı ise,  $I(i,j)=1$ , değilse  $I(i,j)=0$ .

P kromozomuna ait her bir iş  $\tau^{(i,j)}$  sırasıyla işleme alınır. Bu işe karşılık gelen  $(i,j)$  yolu, akış bütünlüğünü sağlayacak şekilde  $R_{mevcut}$  rota dizisine dâhil edilir. Geliştirilen rota oluşturma yaklaşımının adımları aşağı verildiği gibidir.

Adım 1. P kromozomunun ilk geni  $P(1)$  seçilir.  $P(1) = \tau^{(i,j)}$  işine karşılık gelen  $(i,j)$  ayrıtı  $R_{mevcut}$  dizisine eklenir. Başlangıçta  $R_{mevcut} = []$ .

Adım 2.  $P(t) = \tau^{(i,j)}$  işine karşılık gelen  $(i,j)$  ayrıtı seçilir, ( $t=2, \dots, P_{uzunluk}$ ).

Adım 2.1.  $p_{son}, h_{mevcut}$ , bilgilerini tanımla.  $R_{mevcut}$  rotası oluşturulurken hem düğümler arası akışın hem de hiyerarşik sıralamanın sağlanması gerekir. Bu sebeple,  $p_{son}$  düğümünden  $i$  ve  $j$  düğümlerine giden yollara ve bu yolların hiyerarşik seviye ( $h$ ) bilgilerine bakılır ( $h=1, \dots, H$ ). Daha sonra hangi yolun  $R_{mevcut}$  rotasına ekleneceğine karar verilir.



Adım 2.1.1. Eğer  $p_{son} = i$  ise, bu durumda akışı sağlamak için ilave yollara gerek kalmadığından,  $R_{mevcut}$  dizisine  $(i,j)$  ayrıtı doğrudan eklenir.

Adım 2.1.2. Eğer  $p_{son} = j$  ise, bu durumda akışı sağlamak için ilave yollara gerek kalmadığından,  $R_{mevcut}$  dizisine  $(j,i)$  ayrıtı doğrudan eklenir.

Adım 2.1.3. Eğer  $p_{son} \neq i$  ve  $p_{son} \neq j$  ise, o halde akış bütünlüğünü sağlamak için  $p_{son}$  düğümünden  $i$  ve  $j$  düğümlerine sırasıyla gidilebilecek yollar bulunur. Bulunan yolların seyahat sürelerine bakılarak, en kısa seyahat süreli yol  $R_{mevcut}$  dizisine eklenir.

Adım 2.1.3.1.  $p_{son}$  düğümünden  $i$  düğümüne gidilebilecek yol bulunur. Bu yol geçici rota dizisine  $(R_{Ti})$  eklenir. Bu aşamada iki durumla karşılaşılır.

*i.*  $p_{son}$  ve  $i$  düğümleri arasında direkt bağlantı vardır yani  $I(p_{son}, i)=1$ . Bu durumda hiyerarşik kontrol yapılarak karar verilir. Eğer  $h(p_{son}, i) \leq h_{mevcut}$  ise,  $(p_{son}, i) \cup (i,j)$  yolu geçici rota dizisi  $R_{Ti}$ 'ye atanır. Eğer  $h(p_{son}, i) > h_{mevcut}$  ise, bulunan  $(p_{son}, i)$  yolu eklenemez. Bu durumda  $p_{son}$  düğümünden  $i$  düğümüne hiyerarşik sıralamayı bozmayacak şekilde alternatif yollar bulunur. Bunun için yeni bir şebeke  $G'=(V',E')$  oluşturulur. Bu yeni şebeke sadece  $h_{mevcut}$  hiyerarşisine eşit yada daha küçük hiyerarşik yolları içerir  $\forall (i,j) \in E', h \leq h_{current}$ . Daha sonra oluşturulan bu yeni  $G'$  şebekede dijkstra algoritması ile,  $p_{son}$  düğümünden  $i$  düğümüne en kısa yol bulunur. Bulunan  $(p_{son}, i) \cup (i,j)$  yolu geçici rota dizisi  $R_{Ti}$ 'ye eklenir.

*ii.*  $p_{son}$  ve  $i$  düğümleri arasında direkt bağlantı yoktur yani  $I(p_{son}, i)=0$ . Bu durumda da yeni graph  $G'=(V',E')$  oluşturulur.  $p_{son}$ , düğümünden  $i$  düğümüne en kısa yol bulunur. Bulunan  $(p_{son}, i) \cup (i,j)$  yolu geçici rota dizisi  $R_{Ti}$ 'ye eklenir.

Elde edilen  $R_{Ti}$  dizisinin seyahat süresi hesaplanır  $t^{(R_{Ti})}$ .

Adım 2.1.3.2.  $p_{son}$  düğümünden  $j$  düğümüne gidilebilecek yol bulunur. Adım 2.1.3.1'deki işlemler  $R_{Tj}$  rota dizisinin bulunması için tekrarlanır. Elde edilen  $R_{Tj}$  dizisinin seyahat süresi hesaplanır  $t^{(RTj)}$ .

Adım 2.2.  $t^{(RTi)}$  ve  $t^{(RTj)}$  değerleri karşılaştırılır. Seyahat süresi az olan yol  $R_{mevcut}$  dizisine eklenir.

Adım 3. Kromozomdaki tüm işler için Adım 2 tekrar edilir.

Adım 4.  $R_{mevcut}$  dizisinin son elemanından,  $i_0=1$  nolu düğüme en kısa süreli yol bulunarak  $R_{dönüş}$  dizisine eklenir.

Adım 5. Nihai rota dizisi tamamlanır,  $R = \{R_{mevcut} \cup R_{dönüş}\}$ .

Şekil 3.11'de bir P kromozomunun R rotasına dönüştürülme algoritması verilmiştir.

---

```

Wd0, P, Puzunluk
R, R_mevcut, Rdönüş ← ∅
R_mevcut(1) = i, R_mevcut(2) = j, (i,j) = τ(i,j) = P(1)
f(i,j) hesapla
Wd ← Wd0 + f(i,j)
for k=2: Puzunluk
    τ(i,j) ← P(k)
    p_son ← R_mevcut dizisinin en son düğümü
    if p_son = i
        R_mevcut ← R_mevcut ∪ {(i,j)}
    elseif p_son = j
        R_mevcut ← R_mevcut ∪ {(j,i)}
    elseif ((p_son ≠ i) & (p_son ≠ j))
        h_mevcut ← R_mevcut dizisinin mevcut en son hiyerarşik seviyesi
        R_Ti, R_Tj ← ∅

        if I(p_son,i) = 1
            h_kontrol ← h, (p_son,i) ∈ E_h
            if h_kontrol ≤ h_mevcut
                R_Ti ← {(p_son,i)}
            elseif h_kontrol > h_mevcut
                G' ← yeni bir şebeke oluştur
                R_Ti ← dijkstra(G', p_son,i)
            end
        elseif I(p_son,i) = 0
            G' ← yeni bir şebeke oluştur
            R_Ti ← dijkstra(G', p_son,i)
        end
        R_Ti ← R_Ti ∪ {(i,j)}
        f(R_Ti) ← UygunlukDeğeriHesapla(R_Ti)
        if f(R_Ti) < f(R_Tj)
            R_mevcut ← R_mevcut ∪ {R_Ti}
            Wd ← Wd + f(R_Ti)
        elseif f(R_Ti) < f(R_Tj)
            R_mevcut ← R_mevcut ∪ {R_Tj}
            Wd ← Wd + f(R_Tj)
        else
            R_mevcut ← R_mevcut ∪ rasgele seç {R_Ti} ya da {R_Tj}
        end
    end

        if I(p_son,j) = 1
            h_kontrol ← h, (p_son,j) ∈ E_h
            if h_kontrol ≤ h_mevcut
                R_Tj ← {(p_son,j)}
            elseif h_kontrol > h_mevcut
                G' ← yeni bir şebeke oluştur
                R_Tj ← dijkstra(G', p_son,j)
            end
        elseif I(p_son,j) = 0
            G' ← yeni bir şebeke oluştur
            R_Tj ← dijkstra(G', p_son,j)
        end
        R_Tj ← R_Tj ∪ {(j,i)}
        f(R_Tj) ← UygunlukDeğeriHesapla(R_Tj)
    end
end

Output;
R_mevcut, Wd
end
p_son ← R_mevcut
R_dönüş ← dijkstra(G, p_son, 1)
R ← R_mevcut ∪ {R_dönüş}

```

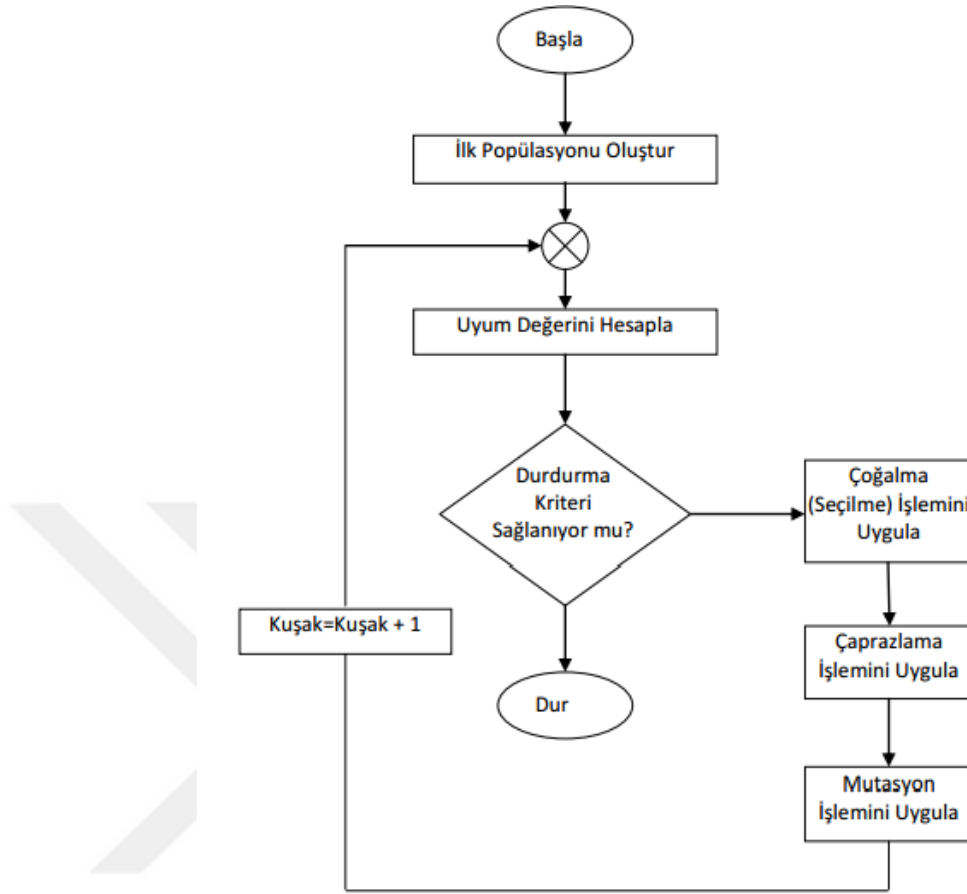
Şekil 3.11. Rota oluşturma algoritması

**EK 2**'de Şekil 3.9'da verilen P kromozomunun rotaya dönüştürülme adımları verilmiştir.

### **3.5.3. ZB-HÇPP için önerilen meta-sezgisel yaklaşım: genetik algoritma**

Çalışmanın bu bölümünde, ZB-HÇPP çözümü için, rotalama problemlerinde sıklıkla kullanılan GA yaklaşımı önerilmiştir. Genetik algoritmalar doğal seçim ve doğal genetik temeline dayanan en iyileme algoritmalar olup bir dizi yapıda en uygun olanların hayatta kalma prensibi üzerine kurulmuştur (Goldberg 1987). Bu algoritma, canlılarda bulunan genetik gelişimi simüle etmektedir. GA popülasyon temelli bir optimizasyon yöntemidir. Bu başlangıç popülasyonu her kuşakta çeşitli evrimsel tabanlı işlemler ile art arda geliştirilir (evrimleşir). En son kuşağın en uygun bireyi problem için en iyi çözüm kabul edilir. Bu en iyi çözüm her zaman optimum olmayabilir.

Şekil 3.12'de GA'nın akış diyagramı verilmiştir. Genetik algoritmaların temel prensibi, olası çözümlerin yapay kromozomu ifade eden diziler şeklinde genler vasıtasıyla kodlanmasına dayanır. Her bir dizi bir değerlendirme fonksiyonu kullanılarak hesaplanan bir uygunluk değerine sahiptir. Bir genetik algoritmanın çalışması için ilk adım, başlangıçta kullanılacak olan popülasyonun oluşturulmasıdır. Bu noktada bireyler rasgele oluşturulur. Ama dikkat edilmesi gereken bir seçenekte başlangıç bireylerin (eğer biliniyorsa) çözüme yakın olduğu bilinen parametre değerleri ile oluşturarak çözümü hızlandırmaktır. Evrim sürecine girmeden bireyler uygunluklarına göre değerlendirilmektedir. Bu, evrim süreci içerisinde bir sonraki nesle aktarılacak olanların belirlenmesi için gereklidir. Daha sonra popülasyon seçme, çaprazlama ve mutasyon gibi bir takım işlemlerin olduğu evrim sürecine geçer. Bu süreç istenilen düzeyde bir birey bulunana, topluluk ortalama başarıda artış sağlayamaz duruma gelene ya da önceden belirlenen evrim sayısı tamamlanana kadar devam eder.



**Şekil 3.12.** GA akış diyagramı

Geliştirilen GA'nın uygulama adımları sırasıyla aşağıda verildiği gibidir.

- Arama uzayındaki tüm olası çözümlerden bir takımını kromozomlarla ifade edilerek başlangıç popülasyonu oluşturulur.
- Her bir kromozom (dizi) için uygunluk değerleri hesaplanır.
- Rasgele olarak seçilen bir grup birey çaprazlama ve mutasyon işlemlerine tabi tutulur.
- Oluşturulan yeni popülasyon ile eskisi yer değiştirilir.
- Durdurma kriteri sağlanana dek yukarıdaki işlemlere devam edilir. İterasyon bitiminde en uygun dizi çözüm olarak seçilir.

### 3.5.3.a. Başlangıç popülasyonunun oluşturulması

İlk aşamada Bölüm 3.5.1’de verilen çözüm gösterimi yaklaşımı ile uygun bireyler (kromozomlar) oluşturulmuştur. GA’da başlangıç popülasyonu rasgele oluşturulmuş bu aday çözümlerden meydana gelmektedir. Popülasyon büyüklüğü, kullanıcı tarafından tanımlanmakta olup problem boyutuna göre değişmektedir. Popülasyon büyüklüğünün çok yüksek tutulması problemin çözüm süresini artırmasından dolayı çok tercih edilmemektedir.

Bu çalışmada önerilen bir P kromozomu ve başlangıç popülasyonu oluşturma algoritması Şekil 3.13’de verilmiştir.

---

```

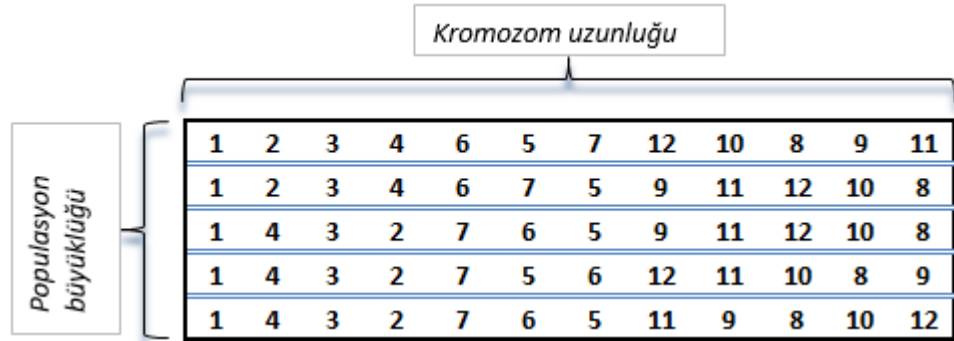
Popboyut ← popülasyon büyüklüğü
Pop ← [ ]
while ||Pop|| < Popboyut
  Yeni bir P kromozomu oluştur
  P(1) ← rasgele  $\tau^{(1,j)}$  işi seç ve ata
  for t=2:E
    ST, SC, S={ST ∩ SC} dizilerini belirle
    S dizisinden rasgele bir iş ( $\tau$ ) seç
    if S =  $\emptyset$ 
      ST dizisinden rasgele bir iş ( $\tau$ ) seç
    end
     $\tau$  işini P(t) kromozom içine yerleştir
    ST ← ST \  $\tau$ 
  end
  Pop ← Pop ∪ {P}
end
Pop ← Nihai popülasyon

```

---

Şekil 3.13. Başlangıç popülasyonu oluşturma algoritması

Şekil 3.14’de ZB-HÇPP için örnek bir başlangıç popülasyonu gösterimi verilmiştir.



Şekil 3.14. ZB-HÇPP için örnek bir başlangıç popülasyon gösterimi

### 3.5.3.b. Seçme operatörü

Popülasyondaki her bireyin (kromozomun) uygunluk değerleri hesaplanarak seçme işlemine tabi tutulur. Genetik Algoritmada uygunluk değerlendirmesi, bir uygunluk fonksiyonu sonucu elde edilen uygunluk değeri ile yapılmaktadır (Paksoy 2007). Toplumunu oluşturan bireylere ait uygunluk değerlerinin belirlenmesi için genellikle bir fonksiyona ihtiyaç duyulmaktadır. Problemin türüne göre uygunluk değeri genellikle problemin amaç fonksiyonu ele alınarak hesaplanmaktadır. Algoritmanın hazırlık aşamasında belirlenen uygunluk fonksiyonu ile popülasyondaki tüm bireylerin uygunluk değerleri hesaplanır. Böylece  $n$  elemanlı popülasyonda,  $f_1 \dots f_n$  olmak üzere  $n$  adet uygunluk değeri hesaplanmaktadır. Bireylerin uygunluk değerlerine göre seçme, çaprazlama ya da mutasyon işlemleri uygulanmaktadır. Çünkü kromozomların kalitesi, uygunluk değerlerine göre ölçülmektedir. Uygunluk değeri en iyi olan kromozom, problemin en iyi çözümünü veren kromozomdur. Uygunluk değeri yüksek bireylerin seçilerek, yeni oluşturulacak topluma aktarılması, genetik algoritmanın gücünü oluşturan başlıca faktörlerden biridir. Çalışmada popülasyondaki her bir kromozomun uygunluk değeri Bölüm 3.5.2’de açıklanan yöntem ile hesaplanmıştır.

Uygunluk değerleri hesaplanan kromozomlardan hangilerinin bir sonraki nesle aktarılacağı kurulan seçim mekanizmaları ile belirlenir. Bu seçimi yapmak için rulet çarkı, elitizm, turnuva seçimi veya doğal seçim gibi farklı seçme yöntemleri vardır.

Bu aşamada iki farklı seçme operatörü kullanılmıştır. Her ikisinde de amaç, yüksek uygunluğa sahip dizilerin çoğalmasını sağlayarak daha iyi bireylerin oluşturulmasıdır.

a) Rulet Çarkı Yöntemi: En sık kullanılan yöntemlerden biridir. Bu yöntemle uygunluk değeri yüksek olan güçlü bireyin doğma olasılığının yüksek olması sağlanır. Bu yöntemin adımları şu şekildedir.

- Tüm fonksiyonlar için uygunluk fonksiyonun aldığı değer diğer bir deyişle uygunluk değeri ( $f_i$ ) hesaplanır.
- Uygunluk değerinin toplamı ( $\sum f$ ) bulunur.
- Normalde bireylerin seçilme olasılığı  $p_i = f_i / \sum f$  olarak hesaplanır. Ancak ele aldığımız problemde amaç fonksiyonu minimizasyon olduğundan dolayı bu şekilde bir hesaplama düşük değerli bireylerin seçilme ihtimalini artırmanın aksine azaltacaktır. Bu durum, popülasyonun iyileştirilmesinden ziyade daha da kötüye gitmesine, uzun mesafelerle turu tamamlayan bireylerden oluşmasına sebep olacaktır. Dolayısıyla bu çalışmada, tüm bireylerin uygunluk değeri  $1/f_i$  olarak hesaplanır. Bu durumda bireylerin seçilme olasılığı ise

$$p_i = \frac{1/f_i}{\sum 1/f_i} \quad (3.53)$$

şeklindedir.

- Bireylerin kümülatif olasılık değerleri hesaplanır.
- Birey sayısı kadar 0-1 aralığında rassal sayı üretilir.
- Rassal sayılara karşılık gelen kromozomlar ebeveyn bireyler olarak alınır.



## b) Doğal seçim

Gelecek nesilde hangi kromozomun belirlenmesi için gerçekleştirilen seçme işleminde kullanılan diğer bir yöntem doğal seçme operatörüdür. Bu yöntemin adımları şu şekildedir.

- Kromozomlar küçükten büyüğe doğru sıralanır (Enbüyükleme problemlerinde büyükten küçüğe doğru alınır) .
- Sıralanmış popülasyonun ilk yarısı  $N_{pop}$  olarak alınır kalan yarısı atılır.
- Oluşturulan yeni  $N_{pop}$  ilk yarısı  $N_{iyi}$  diğer yarısı ise  $N_{kötü}$  olarak 2'ye ayrılır.
- GA çalıştırıldığında, her iterasyonda eşleşme havuzuna sadece  $N_{iyi}$  kromozomlarından alım yapılır ve iterasyon bittiğinde  $N_{pop}$ ,  $N_{iyi}$ ,  $N_{kötü}$  kromozomları güncellenir.

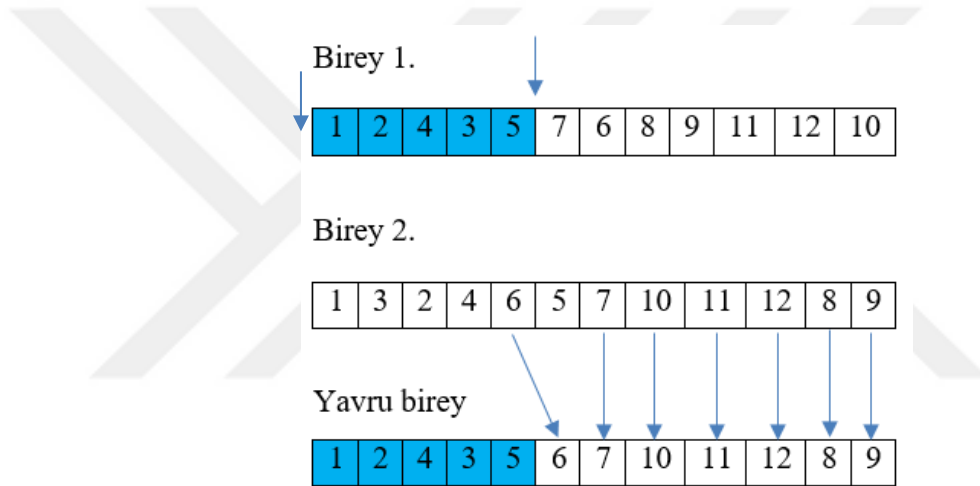
Başlangıç popülasyonu oluşturulduğunda doğal seçim yaklaşımı ile popülasyon güncellenmiştir. Daha sonraki çaprazlama ve mutasyon işlemlerinin gerçekleştirilmesi aşamalarında rulet çarkı yöntemiyle ebeveyn kromozomların seçilme işlemi yapılmıştır.

### 3.5.3.c. Çaprazlama operatörü

Çaprazlama, popülasyondaki çeşitliliği artırmanın çok önemli bir aşamasıdır. Çaprazlama operatörü ile popülasyondan seçilen iki bireyden yeni birey/bireylerin oluşturulması sağlanır. En çok kullanılan çaprazlama yöntemleri; tek noktalı çaprazlama, çift noktalı çaprazlama, PMX çaprazlama, OX çaprazlama, CX çaprazlama ve LOX çaprazlama teknikleridir. Bu çalışmada ZB-HÇPP'nin karmaşık yapısı ve barındırdığı öncelik kısıtları nedeniyle tek nokta çaprazlama tekniği kullanılmıştır. Zira bu teknikte de seçilen bireylerin bir noktadan kesilip karşılıklı yer değiştirmeleri olursuz çözümlere sebep olmaktadır. Olurlu çözüm üretmek için geliştirilen tek nokta çaprazlama tekniği ise şu şekildedir.  $p \neq E$  olmak üzere rasgele bir çaprazlama noktası ( $p$ ) seçilir. İlk bireyin bu noktaya kadar olan tüm genleri yavru bireye aynen kopyalanır. Yavru bireyin kalan kısmı ise şu şekilde tamamlanır: İkinci bireyin tüm genleri 1'den başlanarak sırasıyla kontrol

edilir ve yavru bireyde olmayan genler ( $p+1$ ) noktasından itibaren yavru bireye aktarılır. Yavru bireye atanmamış gen kalıncaya kadar bu işlem devam edilir.

Bu teknik ile çaprazlama oranına göre ( $p_c$ ) seçilen iki bireyden sadece bir yavru birey oluşturulur ve yeni bireyin hiyerarşik sıralaması bozulmaz. Yani olurlu bir çözüm üretilir. Şekil 3.15'te örnek bir çaprazlama işlemi gösterilmiştir. 12 işli bir kromozom için çaprazlama noktası  $p=5$  olarak alındığında ( $1 \leq p < 12$ ) yeni yavru birey aşağıdaki gibi oluşturulur.



Şekil 3.15. Çaprazlama işlemi

### 3.5.3.d. Mutasyon operatörü

Mutasyon işlemi ile çözümün yerel optimuma takılması engellenerek popülasyon çeşitliliğinin artırılması amaçlanır. Mutasyon işlemi çaprazlamadan sonra gerçekleşir ve en yaygın haliyle mevcut kromozomdaki genlerin iki veya daha fazlasının yer değiştirilmesi ile yeni kromozomlar üretilmesi amaçlanır. ZB-HÇPP'de rasgele iki genin seçilip yer değiştirilmesi işlemi, seçilen genlerin farklı hiyerarşilerden olma ihtimaline karşın olursuz çözümlere neden olabilir. Dolayısıyla aynı hiyerarşik seviyeye ait genlerin seçilip yer değiştirilmesi işlemi olurlu çözümler üretmek açısından etkili bir yaklaşım olacaktır.

Çalışmada Bölüm 3.5.4.b’de detaylı anlatılan tavlama benzetimi komşu çözüm üretme yöntemi, mutasyon operatörü ile kombine edilmiştir. ZB-HÇPP’ye uyarlanan yeni komşu çözüm üretme yaklaşımı şu şekildedir. Rasgele bir iş seçilir ve bu işin hiyerarşik seviyesi belirlenir. Seçilen iş, aynı hiyerarşik seviyeli diğer işler ile sırasıyla yer değiştirilir. Bu yöntem ile  $\|E_h\| - 1$  kadar yeni birey üretilir ve aralarından en küçük uygunluk değerine sahip bir birey seçilir. Örneğin mutasyon işlemine uğrayacak bir  $C1 = (1, 3, 2, 4, 5, 7, 6, 9, 10, 8, 11, 12)$  kromozomunu ele alalım. Bu kromozom üç alt-diziden oluşmaktadır;  $C1_1 = \{1, 2, 3, 4\} \in Z_1$ ,  $C1_2 = \{5, 6, 7\} \in Z_2$  ve  $C1_3 = \{8, 9, 10, 11, 12\} \in Z_3$ . C1 dizisinden mutasyona uğratmak için  $p=7$  nolu gen (6 nolu iş) rasgele seçildiğinde oluşturulacak yeni diziler  $\|E_2\| - 1 = 2$  kadardır. Zira 6 nolu iş 2. hiyerarşik seviyeye ait olup sadece bu hiyerarşik seviyeye ait diğer işler (5 ve 7) ile yer değiştirilecektir. Bu şekilde oluşturulan yeni bireyler sırasıyla;

$$C1' = (1, 3, 2, 4, 6, 7, 5, 9, 10, 8, 11, 12)$$

$$C1'' = (1, 3, 2, 4, 5, 6, 7, 9, 10, 8, 11, 12)$$

En sonunda üretilen yeni bireyler arasından en iyi olan birey seçilir.

GA ile yapılan çalışmalarda mutasyon oranı, çaprazlama oranı gibi çok yüksek tutulmaz. Genellikle mutasyon oranı ( $p_m$ ) kullanıcı tarafından 0.01 – 0.15 arasında alınır. Bir popülasyonun seçme, çaprazlama ve mutasyon operasyonları ile gerçekleştirilen evrimleşme süreci algoritması Şekil 3.16’da verilmiştir.

---

```

Pop ← mevcut popülasyon
PopUygunluk ← Popülasyondaki tüm kromozomların uygunluk değerleri kümesi
P1 ← Birey seç(Pop)
P2 ← Birey seç(Pop)
if RassalSayıÜret (∈(0,1)) < pc
    P1,P2 ← çaprazla
    p ← rassal çaprazlama noktası seç, 1<p<E.
    Yavru_birey []1xp ← kopyala P1 []1xp
    for t=(p+1):E
        for i=1:length(P2)
            a ← P2 dizisinden atanmamış iş varsa seç
            break;
        end
        Yavru_birey(t) ← a;
    end
    Yavru_birey
    if RassalSayıÜret (∈ (0,1)) < pm
        Yavru_birey ← mutasyon
        r1 ← rassal sayı üret 1<r1<E
        r2 ← rassal sayı üret 1<r2<E
        while r1==r2 | h(Yavru_birey(r1)) ~ h(Yavru_birey(r2))
            r2 ← tekrar rassal sayı üret
        end
        Yeni_birey ← kopyala Yavru_birey
        Yeni_birey (r1) = Yavru_birey (r2)
        Yeni_birey (r2) = Yavru_birey (r1)
    end
    Yeni_birey
end
Yeni_birey_fonk ← hesapla uygunluk değeri(Yeni_birey)
Popülasyon güncelle
kontrol=0
for i=1:length(PopUygunluk)
    if Yeni_birey_fonk < PopUygunluk(i)
        Pop(i,:) ← Yeni_birey
        indis=i
        kontrol=1
        break;
    end
end
if kontrol =1
    for j=(indis+1): length(PopUygunluk)
        Pop(j,:) ← Pop(indis,:)
        indis = indis +1
    end
end
Çıktı: Pop, PopUygunluk

```

---

Şekil 3.16. Bir popülasyonun evrimleşme süreci algoritması

### 3.5.4. ZB-HÇPP için önerilen meta-sezgisel yaklaşım: hibrit tavlama benzetimi

Bu bölümde, önerilen doğrusal programlama modeli için, GAMS/Cplex çözücüsünün elde ettiği yaklaşık çözümlerden daha iyi çözümler elde edebilmek ve çözüm elde edilemeyen büyük boyutlu problemleri kabul edilebilir sürede çözebilmek amacıyla ikinci bir meta-sezgisel yöntem geliştirilmiştir. TB en iyi bilinen genel meta-sezgisel metodlardan biridir. TB yöntemi durdurma kriteri sağlanıncaya kadar mevcut çözümü geliştirmeye devam eden güçlü bir yerel arama algoritmasıdır. Bu sezgiselin popularitesi literatürde benzerlerinden üstün performans gösteren birçok yayınlanmış çalışma ile daha da artmıştır. Bu bölümde, ZB-HÇPP için TB tabanlı sezgisel algoritma geliştirilmiş ve bu algoritmanın performansının dahada iyileştirilmesi için açgözlü arama sezgiseli TB ile hibritlenerek karar vericiye en uygun çözümler sunulmaya çalışılmıştır. Önerilen hibrit Tavlama Benzetimi (hTB) algoritması temel olarak 2 adımdan oluşmaktadır. Sezgisel yöntemlerde iyi bir başlangıç çözümle başlanması hesaplama zamanının azaltılması açısından büyük önem arz etmekte olup ilk adımda Bölüm 3.5.4.a'da açıklanan açgözlü arama algoritması ile bir başlangıç çözümü üretilmektedir. TB algoritması kullanılarak yeni komşu çözümler oluşturma süreci hTB algoritmasının 2. adımını oluşturmaktadır.

#### 3.5.4.a. Açgözlü arama algoritması

Açgözlü arama algoritması uygulanması çok kolay bir yaklaşımdır. Her adımda o adım için alınabilecek en iyi karar alınır. Bu şekilde yerel optimum seçim yapılarak global optimum çözümün bulunması umut edilir. Bu çalışmada önerilen açgözlü yaklaşım en yakın komşu arama temellidir. Düğüm rotalama problemlerinde en yakın komşu arama metodu ile başlangıç noktasından başlanarak hep en yakın komşunun ziyaret edilmesiyle tur gerçekleştirilir. Çalışmada geliştirilen komşu arama tabanlı açgözlü arama yaklaşımında ise hiyerarşik kısıtlar ihlal edilmeyecek şekilde başlangıç noktasından başlanarak en yakın (ZB-HÇPP için en yakın ifadesi, en kısa süreli yol anlamında kullanılmıştır) ziyaret edilmemiş kenarlara gidilir. Gidilecek bir sonraki en yakın kenarın alt tur oluşturmaya veya hiyerarşik önceliği bozması durumunda en yakın olmayan ve daha önceden ziyaret edilmemiş başka kenara gidilir. Bazı durumlarda ise ziyaret

edilmemiş kenarlar aracın en son bulunduğu düğüm ile direkt bağlantılı olmayabilir. Bu durumda rotadaki akışı sağlamak için en yakın kenardan geçmek yerine özel tanımlanmış yollardan geçilerek asıl ziyaret edilmemiş yollara varış sağlanır. Bu süreç tüm kenarlar en az bir kez ziyaret edilinceye kadar devam edilir. ZB-HÇPP karmaşık yapısından dolayı karşılaşılabilecek tüm bu durumların dikkate alınarak geliştirildiği açgözlü yaklaşım uygulama adımları aşağıda verildiği gibidir:

Adım 1. Başlangıç düğümü  $i_0 = 1$ ,  $\mathbf{R}$  rota dizisine eklenir (başlangıçta  $R = \emptyset$ ).

Adım 2. Geçilmesi gereken yollar (kenarlar) belirlenip  $\mathbf{S}_R$  dizisine eklenir.

Adım 3. Oluşturulan  $R$  rotasının en son düğümü ( $n_{son}$ ) ile bağlantılı yollar (kenarlar) tanımlanır ve  $\mathbf{S}$  dizisine eklenir.

- Eğer  $S \cap S_R \neq \emptyset$  ise, yani sıradaki geçilmesi zorunlu yollar aynı zamanda  $R$  rotasının en son düğümü ile bağlantılıysa, bu yollardan en kısa süreli olanı  $R$  dizisine atanır.
- Eğer  $S \cap S_R = \emptyset$  ise, sıradaki geçilmesi zorunlu kenarların ( $\in S_R$ ),  $n_{son}$  ile doğrudan bağlantılı olmadığı anlamına gelir. Bu sebeple  $n_{son}$  düğümünden gidilebilecek (hiyerarşik kısıtlar göz ardı edilmeden) düğümler kümesi ( $N$ ) tanımlanır. Eğer bu düğüm/düğümler aynı zamanda  $S_R$  kümesinin elemanı ise en kısa süreli olanı seçilir ve  $R$ 'ye eklenir. Eğer  $(N \cap S_R) = \emptyset$  ise,  $N$ 'deki her düğüm  $S_R$ 'deki düğümlerle bağlantılı olup olmadığı kontrol edilir.  $N$  kümesindeki düğümlerle bağlantılı bu düğümlerden biri rasgele seçilir ve  $R$  rotasına eklenir.  $n_{son}$  düğümü ile  $S_R$  dizisindeki herhangi bir düğümün bağlantısı sağlandıktan sonra  $R$  rotasına  $S_R$  dizisinden bağlantı kurulan kenar eklenir.

Adım 4. Tüm kenarlardan geçildiyse Adım 5'e gidilir, aksi takdirde Adım 2 ve Adım 3 tekrar edilir.

Adım 5. Algoritma sonlandırılır ve sonuçlar rapor edilir.

### 3.5.4.b. Tavlama benzetimi algoritması

TB, kombinatoryal eniyileme problemleri için iyi çözümler veren stokastik arama yöntemidir. Bu yaklaşım, metallerin fiziksel olarak tavlama işleminden esinlenilerek geliştirilmiştir. TB yapısal bozukluklar içeren metal kristallerinin yüksek bir sıcaklıktan başlayarak kademeli olarak soğutulması ve böylece yapısal bozukluklarından arındırılarak minimum enerjili kristal yapılara dönüştürülmesi mantığına dayanır. Bu yaklaşımın kombinatoryal optimizasyon problemlerinin çözümünde kullanılabilmesi fikri ilk kez Kirkpatrick *et al.* (1983), Cerny (1985) tarafından ortaya atılmıştır.

Bu yöntem ile TB algoritması çözüm kümesini dolaşmaya önceden belirlenmiş sıcaklıkla ve bir başlangıç çözümler ( $S_{başlangıç}$ ) başlar. Daha sonra komşuluk fonksiyonu yardımıyla bu çözüme komşu çözüm/çözümler ( $S'$ ) üretir. Komşu çözümün uygunluk fonksiyonu değeri eldeki mevcut çözümün uygunluk fonksiyonu değerinden daha iyi ise bu komşu çözüm direkt olarak mevcut çözümle yer değiştirir ve ilerleyen iterasyonlarda yeni komşu çözümler bu çözümden üretilir. Aksi takdirde kötü bir amaç fonksiyonu değerine sahipse amaç fonksiyonunda  $\Delta$  kadar bir yükselmeye yol açan hareketin kabul olasılığına bakılır. Rasgele üretilen bir sayı kabul olasılık değerinden daha küçük ise komşu çözüm kabul edilerek devam edilir, diğer türlü mevcut çözümü terk etmeksizin farklı bir komşu çözüm üretilir ve her seferinde farklı bir rassal sayı ile aynı karşılaştırma yapılır. Belli sayıda iterasyon boyunca aynı sıcaklık korunur ve iterasyon sayısına ulaşıldığında sıcaklık kademeli olarak azaltılır. Sıcaklık, önceden tanımlanmış bir seviyenin altına indiğinde ise algoritma sonlandırılır ve arama esnasında bulunduğu en iyi çözümü rapor eder. TB'de kötü çözümü seçme olasılığı sistemli bir şekilde sıcaklıkla azaltılır. TB sıcaklık yüksek iken (ilk iterasyonlarda) hemen hemen tüm komşu çözümleri kabul ettiği için monte carlo aramaya, sıcaklık düşük iken (son iterasyonlarda) hemen hemen sadece mevcut çözümden daha iyi komşu çözümleri kabul ettiği için yerel aramaya benzer. TB algoritmasının temel amacı, çözüm uzayında aranmadık bölge bırakmamaktır. TB'nin kombinatoryal optimizasyon problemleri için optimuma yakın çözümler veren kullanışlı bir yöntem olarak kullanıldığı söylenebilir. TB, GSP, çizelgeleme, karesel atama problemi, şebeke tasarımı gibi birçok kombinatoryal eniyileme probleminin

çözümünde kullanılmıştır. Fiziksel tavlama süreci ile kombinatoriyal optimizasyon problemleri arasındaki benzerlik Çizelge 3.7’de verildiği gibi açıklanabilir (Reeves 1993).

**Çizelge 3.7.** Tavlama sürecinin kombinatoriyal optimizasyon problemlerinde karşılığı

Tavlama İşlem Süreci	Kombinatoriyal Eniyileme
Sıcaklık	Kontrol parametresi
Enerji	Amaç fonksiyonu değeri
Durum değişikliği	Komşu çözüm
Kristalleşme	Sezgisel çözüm
Sistemin durumları	Uygun çözüm

Herhangi bir problemin çözümünde TB kullanılması için bazı parametrelerin belirlenmesi gerekir. Bu parametreler: başlangıç sıcaklığı ( $T_0$ ), her sıcaklıktaki iterasyon sayısı ( $L$ ), soğutma fonksiyonu ( $P$ ), soğutma oranı ( $\alpha$ ) ve bitiş sıcaklığıdır ( $T_{son}$ ). Başlangıç sıcaklığı bir girdi parametresidir. Sıcaklık, kötü çözümlerin kabul edilme olasılığını kontrol etmek için kullanılır. İterasyon sayısı, her sıcaklıkta üretilen komşu çözümlerin sayısını ifade eder. Soğutma fonksiyonu, bir önceki iterasyon sıcaklığına bağlı olarak mevcut iterasyon sıcaklığını belirler. Bitiş sıcaklığı algoritmayı durdurma kriteridir. Bu sıcaklığa erişildiğinde algoritma sonlandırılır.

TB algoritması kontrollü bir yaklaşımla hareket eder. Daha kötü çözümlerin kabulüne izin vererek çözümün yerel optimuma takılma sorununa engel olur. Yani TB algoritması, global en iyi değerden daha kötü bir değere geçişe kontrollü bir şekilde izin vererek yerel en iyiden kurtulmayı sağlamaktadır. Böylece yüksek sıcaklıklarda (kötü çözümleri kabul etme olasılığı yüksek olduğu için) diğer algoritmalara göre daha geniş bir alanı taramaya imkân vermektedir. Bu nedenlerden dolayı tez çalışmasında ikinci bir meta-sezgisel yaklaşım olarak TB algoritması önerilmiştir.

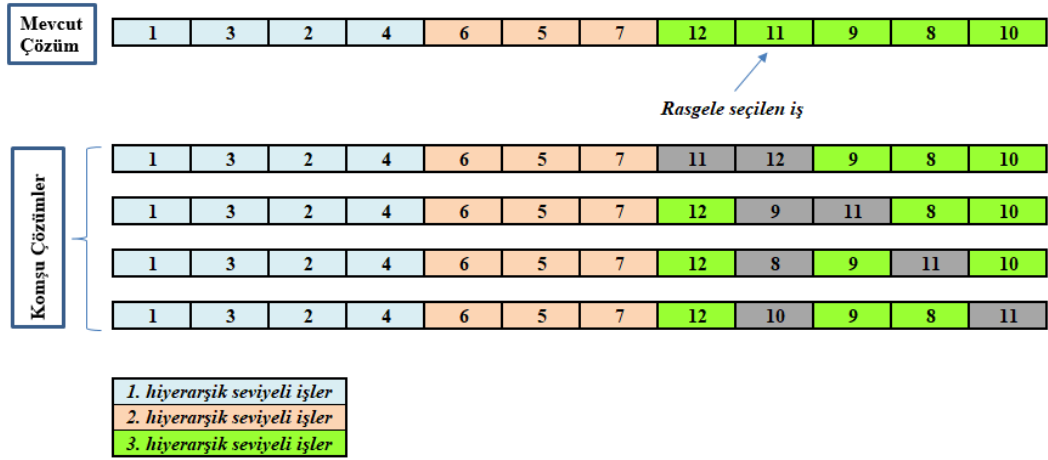


### **Başlangıç çözümünün gösterimi**

Önerilen TB algoritmasının çözüm gösterimi Bölüm 3.5.1’de açıklanan işlerin bir permütasyon dizilimi şeklinde ele alınmıştır. Açgözlü arama metodu ile elde edilen çözümler uygun birer rota şeklindedir. Dolayısıyla tavlama benzetimine başlangıç çözüm olacak bu rotaların permütasyon dizilim şekline dönüştürülmesi gerekmektedir. Zira bu gösterim ile komşu çözümlerin çok daha hızlı ve etkin bir şekilde üretilmesi ve bu sayede çözüm uzayında olabildiğince fazla sayıda çözüm aranması sağlanmıştır.

### **Komşu çözüm üretme**

Çeşitli komşu çözüm üretme yöntemleri vardır. Bu çalışmada kullanılan komşu çözüm üretme metodu, Ishibuchi *et al.* (1995) tarafından geliştirilen modifiye edilmiş bir üretim mekanizmasına dayanmaktadır. Bu teknikte, mevcut bir çözümden çeşitli komşular oluşturulmakta ve bu komşuların en iyisi incelenmektedir. Bu yöntemin ZB-HÇPP’ye uyarlanması problemin hiyerarşik seviyeleri ve her seviyedeki toplam yol sayısı dikkate alınmıştır. Mevcut çözümden komşu çözümler üretmek için rasgele seçilen bir işin ait olduğu hiyerarşik sınıfa bakılır. Daha sonra bu iş sırasıyla kendisiyle aynı hiyerarşik sınıfa ait işlerle yer değiştirir. Bu şekilde her bir iterasyonda  $M = \|E_n\| - 1$  kadar komşu çözüm üretilir. Ishibuchi *et al.* (1995) yöntemine göre bu üretilen  $M$  komşu çözüm arasından en iyi uygunluk değerine sahip çözüm işleme alınır. Aşağıda şekilsel olarak komşu çözüm üretimi gösterilmiştir.



**Şekil 3.17.** Komşu çözümlerin üretilmesi

Şekil 3.17’de rasgele seçilen 11 nolu iş, 3. hiyerarşik sınıfa ait yolu temsil etmektedir. Bu hiyerarşik seviyeye ait toplam iş (yol) sayısı 5’tir. Seçilen iş sırasıyla aynı hiyerarşik seviyeye ait diğer işler ile yer değiştirerek  $M=5-1=4$  farklı komşu çözüm üretilir.

Başlangıç parametreleri belirlendikten sonra her bir sıcaklık değeri için tanımlanan iterasyon sayısı kadar komşu çözümler üretilir ve amaç fonksiyon değerleri hesaplanır. Maksimum iterasyon sayısı  $L$ 'ye ulaşıldığında ( $i \geq L$ ) sıcaklık değeri güncellenir. Çalışmada standart tavlama benzetiminden farklı olarak her iterasyonda  $M$  adet komşu çözüm üretildiği için iterasyon sayısı  $L$  düşük olarak alınmıştır. Bunun sebebi, problem boyutu arttıkça her iterasyonda üretilecek komşu çözüm sayısı  $M$ 'nin artmasıdır. Aksi takdirde yüksek  $L$  değeri bilgisayar hesaplama zamanının artmasına neden olmaktadır.

TB algoritmasının adımları aşağıdaki gibi özetlenebilir:

Adım 1. TB parametre değerleri tanımlanır ( $T_0, \alpha, L, T_{son}$ ),  $i = 0$ .

Adım 2. Açgözlü arama algoritmasından elde edilen çözüm TB'nin başlangıç çözümü ( $S_{başlangıç}$ ) olarak alınır.

Adım 3. Mevcut çözümden ( $S$ ) rasgele ve bağımsız olarak  $M$  farklı komşu çözüm üretilir. Bölüm 3.5.2’de açıklanan uygunluk fonksiyonu hesaplama yöntemi ile  $M$  komşu

çözümün uygunlukları hesaplanır. En iyi uygunluk değerine sahip komşu çözüm (S') değerlendirmeye alınır,  $i=i+M$ .

- S' uygunluk değeri S den daha iyi ise S' komşu çözümü mevcut çözümle yer değiştirir.
- Eğer S' uygunluk değeri daha kötü ise kabul olasılığına bakılır. Kabul olasılığında iki parametre söz konusudur. İlki algoritmanın mevcut sıcaklık (T) değeri, ikinci ise mevcut ve komşu çözümlerin uygunluk değerleri arasındaki fark ( $\Delta$ )'tır. Kabul olasılık formülü  $P = (\exp^{-\Delta/T})$  dir. (0,1) aralığında rasgele bir  $p$  olasılık değeri üretilir. Eğer bu  $p$  değeri kabul olasılık değerinden küçük ise ( $p < P = (\exp^{-\Delta/T})$ ) üretilen yeni komşu çözüm kabul edilerek iterasyonlar devam ettirilir. Aksi halde mevcut çözüm değişmeyecektir.

Adım 4. Maksimum iterasyon sayısına ulaşıldığında soğutma çizelgesi uygulanarak mevcut sıcaklık,  $\alpha$  soğutma oranına göre düşürülür  $\alpha T \rightarrow T$ . Sıcaklığın yavaş soğutulması önerilir. Bu nedenle,  $\alpha$  değeri genellikle 0,80–0,99 arasında seçilir.

Adım 5. Sayaçlar ve parametre değerleri güncellenir. Durdurma kriteri sağlanana kadar 3. ve 4. adımlar devam ettirilir.

Önerilen TB algoritması için durdurma kriteri olarak önceden belirlenmiş bir minimum sıcaklık değeri ( $T_{son}$ ) dikkate alınmıştır. Mevcut sıcaklık değeri soğutma çizelgesi ile bu minimum değere ulaştığında algoritma sonlanmaktadır. ZB-HÇPP için önerilen hTB algoritmasının adımları Şekil 3.18'de verilmiştir.

---

```

Sbaşlangıç ← açgözlü arama algoritmasından başlangıç çözümü ata
T0, α, L, Tson ← parametre değerlerini tanımla
T ← T0
Smevcut çözüm ← Sbaşlangıç dizisini mevcut çözüm olarak belirle
while T < Tson
    x ← rasgele sayı üret ([1,e])
    is ← Smevcut çözüm(x)
    h ← Seçilen işin ait olduğu hiyerarşik seviyeyi tanımla
    isler ← Seçilen iş ile aynı hiyerarşik seviyeye ait işleri bu diziye ata

    % komşu çözüm üretme

    for i=1:L
        for j=1:length(isler)
            Syeni çözüm ← 'is' ile isler(j) elemanını Smevcut çözüm dizisinde yer değiştir
            f(Suygunluk) ← UygunlukDeğeriHesapla(Syeni çözüm)
        end
        Skomşu çözüm ← |Eh-1| kadar Syeni çözüm'den min f(Suygunluk) değerli çözümü ata

        % komşu çözüm ile mevcut çözüm karşılaştırma

        f(S) ← UygunlukDeğeriHesapla(Smevcut çözüm)
        f(S') ← UygunlukDeğeriHesapla(Skomşu çözüm)
        Δ ← f(S') - f(S)
        if Δ ≤ 0
            Smevcut çözüm ← Skomşu çözüm
        elseif (rasgele_sayı_üret(0,1)) < (e-Δ/τ)
            Smevcut çözüm ← Skomşu çözüm
        else
            Smevcut çözüm
        end
    end
    T ← T*α
end

```

---

Şekil 3.18. hTB algoritmasının adımları

#### 4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Bu bölümde ÇPP ve HÇPP için Bölüm 3 kapsamında sunulan durumlar ve çözüm önerileri 2 alt başlık altında test edilmiştir. İlk kısımda, gerçek hayat problemleri ÇPP yaklaşımı ile çözülerek sonuçların etkinliği sunulmuştur. İkinci kısımda, tez çalışmasının ana çatisını oluşturan HÇPP'nin matematiksel modeli test edilmiştir. Daha sonra zaman-bağımlı varyasyonu ile genişletilen HÇPP matematiksel model sonuçları ile meta-sezgisel yöntemlerin sonuçları değerlendirilmiştir. Son olarak orta ve büyük boyutlu test problemleri üzerinde geliştirilen meta-sezgisel yöntemlerin sonuçları çözüm kalitesi ve bilgisayar zamanı açısından karşılaştırılmıştır.

Matematiksel modeller, modelleme ve optimizasyon problemlerinin çözümü için kullanılan yüksek seviye bir programlama dili olan Gams 24.2.3 programı ara yüzü kullanılarak Cplex 12 karma tam sayılı programlama çözücüsü ile sonuçlar elde edilmiştir. Bununla birlikte sezgisel yöntemler Matlab 2016a programlama dili ile kodlanmıştır. Her iki yöntemde intel i7 3.40 GHz işlemcili 8 GB belleğe sahip bilgisayarlar çalıştırılmıştır.

##### 4.1. Uygulama Çalışması

###### 4.1.1. TCDD ray bakım çalışmalarında kullanılan yol ve katener ölçüm aracının optimum rotasının bulunması

Bu çalışmada Türkiye'deki ray hattı bakımı için mevcut durum incelenerek optimum rotayı amaçlayan ÇPP matematiksel model yaklaşımı önerilmiştir. Önerilen matematiksel model ile bakım faaliyetlerinde katedilen toplam mesafenin minimize edilmesi amaçlanmıştır.

Aracın bulunduğu Ankara ili (1 nolu düğüm) ilk hareket noktası seçilmiştir. ÇPP matematiksel modelin çözülmesi ile elde edilen en iyi rota, 1-32-33-32-25-24-25-26-27-

26-28-29-30-29-31-29-28-27-2-3-4-5-4-8-7-6-7-9-7-8-10-13-10-11-12-11-19-18-19-21-20-21-22-23-22-24-55-54-56-57-58-57-59-60-59-61-59-57-56-62-63-62-64-65-64-66-67-66-68-69-70-69-71-69-68-52-53-52-51-54-51-49-50-49-47-48-47-45-46-45-44-55-44-34-35-36-35-37-38-37-39-40-39-41-42-41-43-41-39-37-35-34-1.

**Çizelge 4.1.** Bölgelere ait optimum rota uzunlukları

Bölgelerin bakım sıralaması	Bölgelere ait optimum rota	Toplam mesafe (km)	Pratikte uygulanan toplam mesafe (km)	İyileştirme (%)
2	1-32-33-32-25-24-25	1 539	2 169	29,04
4	25-26-27-26-28-29-30-29-31-29-28-27-2-3-4-5-4	2 720	3 086	11,86
5	4-8-7-6-7-9-7-8-10	2 110	2 110	0,0
6	10-13-10-11-12-11-19-18-19-21-20-21-22-23-22-24	1 019	2 580	60,50
3,7	24-55-54-56-57-58-57-59-60-59-61-59-57-56-62-63-62-64-65-64-66-67-66-68-69-70-69-71-69-68-52-53-52-51-54-51-49-50-49-47-48-47-45-46-45-44-55-44-34	3 300	4 003	17,56
1	34-35-36-35-37-38-37-39-40-39-41-42-41-43-41-39-37-35-34-1	1 760	1 762	0,11
Toplam mesafe		12 448	15 710	20,76

Çizelge 4.1’de görüldüğü üzere sunulan optimum rota 2 nolu bölgeden başlayıp sırasıyla 4, 5, 6, 3, 7, 1 bölgelerini ziyaret etmektedir. Bölgelerde % iyileştirmeler dikkate alındığında en büyük iyileştirmenin 6. bölgede olduğu ve 5. bölgede ise herhangi bir iyileştirme yapılamadığı görülmektedir. Matematiksel model sonucunda elde edilen optimum rotada 2. bölgede 1 539 km, 4. bölgede 2 720 km, 5. bölgede 2 110 km, 6. bölgede 1 019 km, 3. ve 7. bölgelerde 3 300 km ve son olarak 1. bölgede 1 760 km olup toplamda 12 448 km yol katedildiği görülmektedir. Çizelge 4.1’de sunulan mevcut duruma göre hat üzerinde katedilen toplam mesafede %20,76’lık bir iyileştirme gerçekleştirilmiştir.

Optimum mesafedeki iyileştirmenin ana sebebi bölgeler arası geçiş güzergâhlarından kaynaklandığı görülmektedir. Örneğin, var olan uygulamada bakıma 1. bölgeden başlanılıp 3. ve 7. bölgenin bakımından sonra sırasıyla 2., 4. ve 5. bölgelere geçiş yaparak 6. bölgede bakımı tamamlayarak başlangıç noktasına dönmektedir. Oysaki Çizelge 4.1’de gösterildiği üzere elde edilen optimum rotada bakım çalışması 2. bölgeden başlayıp sırasıyla 2., 4., 5., 6., 3-7. bölgelerinden sonra 1. bölgeye geçiş yaparak başlangıç noktasına dönmektedir.

Bu güzergâhta giden aracın bir düğümden diğer bir düğüme tekrar yapmadan sadece bir kez geçerek tüm şebekeyi dolaştığı görülmektedir. Örneğin, Çizelge 3.1’de 24 nolu düğümden 55 nolu düğüme geçiş hem 3-7 nolu bölgelerin bakımında hem de 6. bölgenin bakımında yapılmıştır. Oysaki sunulan optimum bakım planında bu düğümler arasındaki yol bir kez kullanılmıştır. Dolayısıyla elde edilen sonuç hem bölgeler arası geçişlerin sıralanması hem de şebekedeki noktalar arası geçiş tekrarlanmaması ile gerçek hayat uygulamasından oldukça farklılık göstermekte olup her iki durum açısından optimum sonucu vermiştir.

#### **4.1.2. Atatürk Üniversitesi Kampüsü kar ile mücadele araçlarının rotalanması**

İkinci uygulama çalışmasında, birden fazla aracın rotalanması problemi üzerinde durulmuştur. Bu aşamada hem araçlar için en kısa mesafeli turları bulunması hem de araçların katettikleri mesafelerin dengelenmesini amaçlanmıştır. Atatürk Üniversitesi doğu kampüs şebekesi ele alınmıştır. İki adet kar küreme aracının rotalarının belirlenmesi için önerilen k-ÇPP ve Dk-ÇPP matematiksel model sonuçları sırasıyla Çizelge 4.2 ve Çizelge 4.3’te verilmiştir.

**Çizelge 4.2.** Problemin çözümünde k-ÇPP matematiksel modeli ile elde edilen optimum rotalar

Araç No	Optimum Rota	Toplam Tur Mesafesi (m)
1	1-2-3-23-24-23-25-26-25-27-82-83-84-83-85-52-116-117-118-117-119-120-119-117-116-52-50-51-50-49-46-47-46-45-48-92-93-94-93-95-96-95-97-95-98-99-100-99-104-105-106-107-106-108-111-110-112-113-112-111-110-109-103-104-105-108-109-103-102-101-102-98-95-93-92-115-114-115-92-48-49-46-45-44-33-44-32-90-91-90-89-86-82-80-16-17-18-17-19-20-19-21-22-21-4-3-2-1.	12 409
2	1-2-35-34-36-37-38-42-43-42-41-40-41-39-41-42-38-36-34-33-32-31-30-29-31-29-28-23-25-27-87-88-87-86-85-83-82-80-81-80-16-15-64-56-54-53-54-55-54-56-57-61-62-61-63-61-57-58-59-58-60-58-57-56-64-65-66-65-67-75-79-75-78-76-77-76-75-67-68-69-68-70-71-70-72-73-72-74-72-70-68-67-65-64-15-14-13-14-11-12-11-7-8-9-8-10-8-7-5-6-5-4-3-2-1.	6 851
Toplam Mesafe		19 260
Rotalar arası mesafe farkı		5 558

**Çizelge 4.3.** Problemin çözümünde Dk-ÇPP matematiksel modeli ile elde edilen optimum rotalar

Araç No	Optimum Rota	Toplam tur mesafesi (m)
1	1-2-35-34-36-37-38-42-43-42-41-40-41-39-41-42-38-36-34-33-44-45-46-47-46-49-46-45-48-92-93-94-93-95-96-95-97-95-98-99-100-99-104-105-106-107-106-108-105-104-103-109-110-112-113-112-111-110-111-108-109-103-102-101-102-98-95-93-92-115-114-115-92-48-49-50-51-50-52-116-117-118-117-119-120-119-117-116-52-85-83-82-27-25-23-3-2-1.	10 612
2	1-2-3-4-21-22-21-19-20-19-17-18-17-16-80-81-80-82-86-89-90-91-90-32-33-44-32-31-30-29-31-29-28-23-24-23-25-26-25-27-87-88-87-86-85-83-84-83-82-80-16-15-64-65-66-65-67-75-76-77-76-78-75-79-75-67-68-69-68-70-71-70-72-73-72-74-72-70-68-67-65-64-56-57-61-62-61-63-61-57-58-59-58-60-58-57-56-54-55-54-53-54-56-64-15-14-13-14-11-12-11-7-8-9-8-10-8-7-5-6-5-4-3-2-1.	8 648
Toplam Mesafe		19 260
Rotalar arası mesafe farkı		1 964



1 nolu düğüm, araçlar için başlangıç noktası olarak alınmıştır. Çizelge 4.2 ve Çizelge 4.3'te görüldüğü üzere, araçlar 1 nolu düğümden başlar ve kar küreme işlemi tamamlandıktan sonra tekrar başlangıç noktasına geri dönerler. Çizelge 4.2'de ilk araç 12 409 m ikinci araç ise 6 851 m yol katederek turlarını tamamlamaktadırlar. Toplam tur mesafesi 19 260 metredir. Bunlara ilave olarak birinci ve ikinci aracın katettikleri mesafe farkı ise 5 558 metre olarak bulunmuştur. Oysaki Çizelge 4.3'te Dk-ÇPP modeli ile elde edilen sonuçlara bakıldığında, birinci aracın 10 612 m ikinci aracın 8 648 m yol katettiği görülmektedir. Ayrıca bu iki aracın katettikleri mesafe farkı ise 1 964 m olarak bulunmuştur.

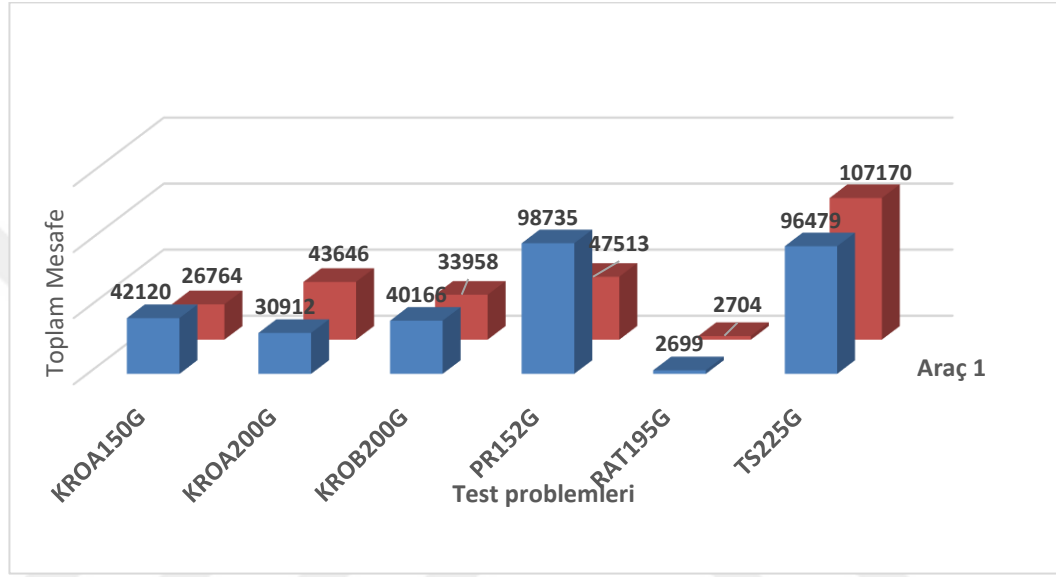
Çizelge 4.2 ve Çizelge 4.3'e bakıldığında her iki modelde toplam katedilen mesafenin aynı olduğu görülmektedir. Bunlara ilave olarak, Dk-ÇPP modelinde araçların hem rotalarının hem de katettikleri mesafelerin k-ÇPP modeli ile elde edilen sonuçlardan farklılık gösterdiği görülmektedir. Sonuç olarak, Dk-ÇPP modeli ile araçlar arasında katedilen mesafe açısından daha dengeli rotalar elde edilmiştir.

Ayrıca bu çalışmada geliştirilen Dk-ÇPP modeli literatürde var olan 6 test problemi için çözülmüş ve sonuçlar her iki durum için karşılaştırılmıştır. Bu test problemlerine ait veri dosyaları <http://www.uv.es/corberan/instancias.html>. sayfasından temin edilmiştir.

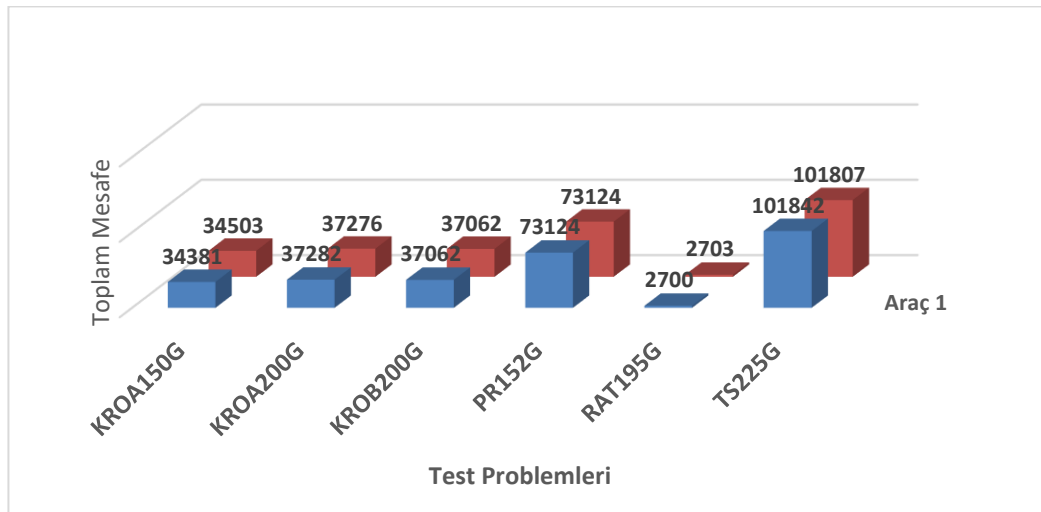
**Çizelge 4.4.** k-ÇPP ve Dk-ÇPP modeli ile elde edilen test problem sonuçları

Test Problemleri	k-ÇPP			Dk-ÇPP		
	Araç 1	Araç 2	Toplam Mesafe	Araç 1	Araç 2	Toplam Mesafe
KROA150G	42120	26764	68884	34381	34503	68884
KROA200G	30912	43646	74558	37282	37276	74558
KROB200G	40166	33958	74124	37062	37062	74124
PR152G	98735	47513	146248	73124	73124	146248
RAT195G	2699	2704	5403	2700	2703	5403
TS225G	96479	107170	203649	101842	101807	203649

Çizelge 4.4'te iki modelden elde edilen araç rotalarının farklı olduğu görülmektedir. Aynı zamanda Şekil 4.1 ve Şekil 4.2'de k-ÇPP ve Dk-ÇPP model performansları gösterilmiştir. Çizelge ve şekillerden görüldüğü üzere, geliştirilen matematiksel model, ele alınan test problemlerinin tamamında büyük oranlarda iyileştirmeler sağlamıştır.



Şekil 4.1. k-ÇPP modelinin farklı test problemlerine ilişkin çözüm sonuçları



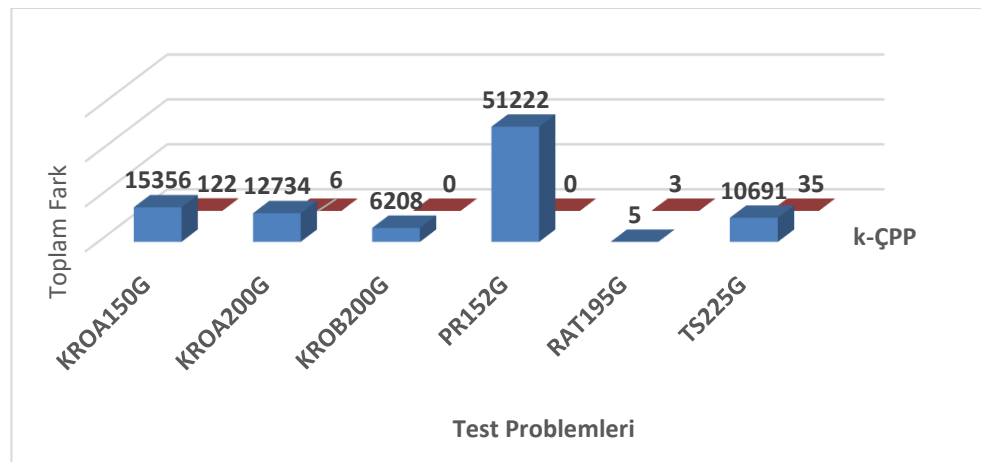
Şekil 4.2. Dk-ÇPP modelinin farklı test problemlerine ilişkin çözüm sonuçları

Her iki model sonuçlarına göre araçların katettikleri mesafeler arası farklar Çizelge 4.5’de verilmiştir.

**Çizelge 4.5.** Her iki model sonuçlarına göre araçlar arası katedilen mesafe farkları

Test Problemleri	Araçların katettikleri mesafe farkları (k-ÇPP) (br)	Araçların katettikleri mesafe farkları (Dk-ÇPP) (br)
KROA150G	15356	122
KROA200G	12734	6
KROB200G	6208	0
PR152G	51222	0
RAT195G	5	3
TS225G	10691	35
<b>TOPLAM</b>	<b>96216</b>	<b>166</b>

Ele alınan test problemlerinin tamamında toplam farklılık 96216 birimden 166 birime düştüğü görülmektedir. k-ÇPP ve Dk-ÇPP modelleri ile bulunan rotalara göre araçlar arası gidilen mesafe farkları Şekil 4.3’te gösterilmiştir.



**Şekil 4.3.** Araçlar arası gidilen mesafe farkları (k-ÇPP/Dk-ÇPP)

En büyük iyileşmenin PR152G test probleminde olduğu görülmektedir. Zira k-ÇPP modelinde iki araç arasındaki gidilen mesafe farkı 51 222 iken, Dk-ÇPP modelde bu fark 0'a düşerek %100'lük bir iyileşme sağlanmıştır.

## 4.2. Teorik Çalışma

### 4.2.1. Test problemlerinin oluşturulması

Önerilen matematiksel modeller ve meta-sezgisel algoritmaların test edilmesi için yeterli test problemleri ve uygun performans ölçütleri gereklidir. Şimdiye kadar HÇPP için literatürde karşılaştırma test örnekleri geliştirilmemiştir. Bu çalışma kapsamında iki Sınıf HÇPP örneği ele alınmıştır:

- Mevcut test örnekleri: Literatürde HÇPP'ye ait iki adet küçük boyutlu test örneği vardır. Bunlar; Dror *et al.* (1987) ve Korteweg and Volgenant (2006)'ın çalışmalarında kullandıkları Şekil 3.6 ve Şekil 3.7'de verilen test örnekleridir. Bu test örneklerindeki düğüm sayıları sırasıyla 8 ve 7 olup her iki örnekte de toplam hiyerarşi sayısı  $h=3$ 'tür.
- Rasgele test örnekleri: İkinci sınıf test örnekleri; düğüm, kenar ve hiyerarşik seviye sayılarına göre rasgele oluşturulmuştur. Test problemleri  $G=(V,E)$  şebekesi ele alındığında düğümler kümesi  $V=\{7,8,9,10,20,30,40,50\}$ , hiyerarşiler kümesi  $H=\{2,3,4,5\}$  olarak belirtilmiştir. Her bir düğüme karşılık gelen yollar kümesi yol sayıları ile orantılı şekilde kolay ve orta olmak üzere 2 farklı biçimde verilmiştir. Örneğin 10 düğümlü test problemi şebekesinde ayrıtlar seti  $E^{10}=\{13,18\}$ , 20 düğümlüde  $E^{20}=\{55,76\}$ .

Bu örnekler, farklı zaman aralıklarında yol kategorilerine göre değişen sürüş hızları ile uyarlanmıştır. Bir tam gün (24 saat)  $k=4$  farklı zaman aralığına bölünmüştür. Tüm test örneklerinde başlangıç düğümü  $i_0=1$ , hareket saati  $8^{30}$  olarak kabul edilmiştir. Oluşturulan test problemlerinin karakteristik özellikleri ayrıntılı bir şekilde Çizelge 4.10'da verilmiştir.

#### 4.2.2. HÇPP matematiksel model çözümü ile elde edilen sonuçlar

Geliştirilen HÇPP matematiksel modeli ile küçük boyutlu problemlerde modelin etkinliği ve geçerliliği test edilmiştir. Bu sebeple 10 ve 20 düğümlü test problemleri ele alınmış ve problemlerin çözümü için maksimum süre 3 saat (10 800 sn) olarak belirlenmiştir. Verilen bu süre içerisinde 10 düğümlü 13 ve 18 kenarlı problemlerin tamamında saniyeler/dakikalar içerisinde optimum çözümlere ulaşılmıştır. Çözücü, kenar sayısı artırılarak tekrar koşurulduğunda (10 düğüm 30 kenar) ise 3 saatte belirli GAP değerleri ile optimuma yakın çözümler elde edilmiştir. Çizelge 4.6'da ele alınan test problemlerinin çözümünde elde edilen sonuçlar, problemin bilgisayar çözüm zamanı ve GAP değerleri verilmiştir. Optimum çözümü bulunan test problemlerine ait rotalar ise Çizelge 4.7'de verilmiştir. Rotaların her birinde görüldüğü üzere, hiyerarşik sınıfa ait yollardan sırasıyla geçilmekte olup  $h$ . hiyerarşik sınıfa ait yolların tamamından geçilmeden  $h+1, h+2, \dots, H$  sınıflarına ait yollardan geçilmemektedir.

Model, 10 düğümlü problemlerin hepsinde optimum yada optimuma yakın çözümler üretmesine karşın, 20 düğümlü problemlerin sadece ikisinde sonuç verebilmiştir. Çalışmanın bu aşaması tezin gidişatına yön verme niteliğinde olmuştur. Zira ZB-HÇPP matematiksel modelinin çatisını oluşturan HÇPP modelinin etkinliği incelenmiştir. Problemin boyutu büyüdükçe geliştirilen model ile herhangi bir uygun çözüm dâhi elde edilememesinden dolayı optimum çözümü her zaman vermese de optimuma yakın çözümler üretebilen meta-sezgisel algoritmanın geliştirilmesi fikri doğmuştur.

**Çizelge 4.6.** Geliştirilen HÇPP matematiksel modeli ile elde edilen sonuçlar

Test Problemi	h	V	E	Amaç Fonk. Değeri	CPU zamanı (sn)	GAP (%)
Testprob25	2	10	13	<b>1281</b>	0,94	0
Testprob26		10	18	<b>1417</b>	483,28	0
Testprob.ek1		10	30	2244	10800,00	7,37
Testprob27	3	10	13	<b>1386</b>	32,14	0
Testprob28		10	18	<b>1516</b>	124,25	0
Testprob.ek2		10	30	2362	10800,00	11,62
Testprob29	4	10	13	<b>1249</b>	2,02	0
Testprob30		10	18	<b>1582</b>	65,34	0
Testprob.ek3		10	30	2598	10800,00	17,45
Testprob31	5	10	13	<b>1013</b>	0,77	0
Testprob32		10	18	<b>2006</b>	182,53	0
Testprob.ek4		10	30	2768	10800,00	18,88
Testprob33	2	20	55	4151	10800,00	18,55
Testprob34		20	76	5726	10800,00	14,91
Testprob.ek5		20	127	-	-	-
Testprob35	3	20	55	-	-	-
Testprob36		20	76	-	-	-
Testprob.ek6		20	127	-	-	-
Testprob37	4	20	55	-	-	-
Testprob38		20	76	-	-	-
Testprob.ek7		20	127	-	-	-
Testprob39	5	20	55	-	-	-
Testprob40		20	76	-	-	-
Testprob.ek8		20	127	-	-	-

**Kalın:** Optimum çözüm

**Çizelge 4.7.** 10 düğümlü HÇPP test problemlerinin optimum rotaları

Test Problemi	$h$	Hiyerarşik sınıflara ayrılmış yollar kümesi	Optimum rota	Optimum tur uzunluğu
Testprob25	2	$E_1(i,j)$ /1.5, 2.3, 2.5, 2.6, 3.9, 4.9, 5.7/ $E_2(i,j)$ /1.6, 3.10, 4.8, 6.10, 7.8, 8.10/	1-5-7-5-2-6-2-3-9-4-8-7-8-10-3-10-6-1	1281
Testprob26	2	$E_1(i,j)$ /1.2, 1.5, 1.9, 2.5, 2.7, 2.8, 3.8, 3.9, 4.5, 4.6, 4.8, 4.9, 7.9/ $E_2(i,j)$ /1.6, 1.8, 5.10, 6.9, 9.10/	1-2-8-4-8-3-9-4-6-4-5-2-7-9-1-5-10-9-6-1-8-1	1417
Testprob27	3	$E_1(i,j)$ /1.5, 1.6, 4.5, 6.9/ $E_2(i,j)$ /2.6, 4.6, 4.7, 4.8, 5.8/ $E_3(i,j)$ /1.4, 4.9, 4.10, 8.9/	1-6-9-6-1-5-4-7-4-6-2-6-4-8-5-8-9-4-10-4-1	1386
Testprob28	3	$E_1(i,j)$ /1.3, 1.5, 1.6, 1.7, 2.9, 5.7, 6.7, 6.9/ $E_2(i,j)$ /2.4, 4.5, 4.8, 4.9, 4.10, 5.8, 8.9/ $E_3(i,j)$ /3.4, 3.8, 3.10/	1-3-1-7-6-7-5-1-6-9-2-4-8-9-4-10-4-5-8-3-4-10-3-1	1516
Testprob29	4	$E_1(i,j)$ /1.8, 1.9, 8.9/ $E_2(i,j)$ /1.2, 1.4, 2.5, 2.6, 2.7, 4.7/ $E_3(i,j)$ /1.6, 1.10/ $E_4(i,j)$ /7.9, 9.10/	1-9-8-1-2-5-2-7-4-1-2-6-1-10-9-7-4-1	1249
Testprob30	4	$E_1(i,j)$ /1.2, 1.7, 2.5, 3.7, 5.7/ $E_2(i,j)$ /1.4, 1.10, 2.10, 4.7, 7.10, 9.10/ $E_3(i,j)$ /4.5, 4.10, 5.6, 6.10/ $E_4(i,j)$ /6.7, 7.8, 8.9/	1-7-3-7-5-2-1-10-9-10-7-4-1-2-10-4-5-6-10-9-8-7-6-7-1	1582
Testprob31	5	$E_1(i,j)$ /1.2, 1.7, 2.6, 5.10, 6.10/ $E_2(i,j)$ /5.7, 6.7/ $E_3(i,j)$ /5.9, 6.9/ $E_4(i,j)$ /2.4, 4.5/ $E_5(i,j)$ /1.3, 3.5/	1-7-1-2-6-10-5-7-6-9-5-4-2-4-5-3-1	1013
Testprob32	5	$E_1(i,j)$ /1.4, 1.5, 1.6, 4.6, 6.8, 6.9/ $E_2(i,j)$ /2.4, 3.4, 3.6, 3.10, 4.10/ $E_3(i,j)$ /2.3, 2.8, 2.10/ $E_4(i,j)$ /5.6, 6.10/ $E_5(i,j)$ /3.5, 3.7/	1-4-6-8-6-1-5-1-6-9-6-3-10-4-2-4-3-2-8-2-10-6-5-3-7-3-4-1	2006

### 4.2.3. ZB-HÇPP çözümü ve sonuçların değerlendirilmesi

Bu bölümde Cplex çözücü, GA ve hTB'nin performansları karşılaştırmalı olarak değerlendirilmiştir. Bütün test problemlerine ait sonuçlar Çizelge 4.10'da sunulmuştur. Tablonun ilk sütununda test probleminin adı ikinci, üçüncü ve dördüncü sütunlarda ise sırasıyla o probleme ait düğüm sayısı, hiyerarşi sayısı ve kenar sayısı ( $|V|$ ,  $|h|$  ve  $|E|$ ) bilgileri verilmiştir. Sonraki üç sütun Cplex çözücünün koşturulmasıyla elde edilen sonuçları içermektedir. Bu sütunlar sırasıyla bulunan eniyi amaç fonksiyonu değerini, bilgisayar çözüm zamanını ve Cplex tarafından bulunan en iyi çözüm ile Cplex tarafından bildirilen en iyi amaç fonksiyonu değeri arasındaki sapmanın yüzdesini göstermektedir ( $O^{Cplex}$ ,  $T^{Cplex}$ , Gap). Problemlerin çözümü için verilen maksimum süre 3 saat (10800 saniye) olarak belirlenmiştir. 7, 8, 9 düğümlü test problemlerinin tamamı ve 10 düğümlü 13 kenarlı test problemlerinde saniyeler içerisinde optimum çözümler elde edilmiştir (HÇPP1, HÇPP2, (Testprob1-Testprob24), Testprob25, Testprob27, Testprob29, Testprob31). Bu test problemlerinde optimal çözümlere ulaşılmasına rağmen kenar sayısı ve hiyerarşi sayısının artmasıyla bilgisayar çözüm zamanının belirgin bir şekilde arttığı görülmüştür. 10 düğüm, 18 kenarlı üç problemde 3 saat içerisinde %50'nin üzerinde GAP değerli uygun çözümler elde edilmiştir. 10 düğüm ve 18 kenarlı son test örneğinde (Testprob32) ise 3 saat içerisinde herhangi bir uygun çözüm bulunamamıştır. 20 düğüm ve daha fazlası problemler için model uygun çözümler üretememiştir.

Problemin NP-zor yapısından dolayı sunulan matematiksel model orta ve büyük boyutlu test problemlerinde uygun zaman aralığında çözümler bulamamıştır. Bu nedenle çeşitli büyüklükteki test problemleri için kısa zamanda olurlu çözümler üretmek amacı ile meta-sezgisel yöntemler geliştirilmiştir. Geliştirilen GA ve hTB algoritmalarının sonuçları Çizelge 4.10'da verilmiştir. 8-10 nolu sütunlar sırasıyla GA tarafından bulunan ortalama uygunluk fonksiyonu değeri, eniyi uygunluk fonksiyonu değeri ve eniyi çözümün bilgisayar CPU zamanı (saniye) bilgisini vermektedir ( $M^{GA}$ ,  $O^{GA}$  and  $T^{GA}$ ). hTB tarafından bulunan ortalama uygunluk fonksiyonu değeri, eniyi uygunluk fonksiyonu değeri ve eniyi çözümün bilgisayar CPU zamanı (saniye) bilgisi ise sırasıyla 11., 12. ve 13. sütunlarda gösterilmiştir ( $M^{hTB}$ ,  $O^{hTB}$  and  $T^{hTB}$ ). Her iki algoritmada eniyi uygunluk



fonksiyonu deęerlerini veren parametrelerin belirlenmesi için test problemleri üzerinde birtakım rassal denemeler yapılmıştır. Bu denemeler sonucu uygun olduğuna kanaat getirilen aynı zamanda literatürde de yaygın olarak kullanılan GA ve hTB parametre deęerleri Çizelge 4.8 ve Çizelge 4.9’da verilmiştir. Her bir test problemi aynı koşullar altında 5 defadan fazla çalıştırılmıştır.

**Çizelge 4.8.** GA parametre deęerleri

Popülasyon Büyüklüğü	Çaprazlama Oranı	Mutasyon Oranı	İterayon Sayısı	Seçme Operatörü	Çaprazlama yapısı
50	0,80	0,05	500	Rulet tekeri	Tek-nokta çaprazlama

**Çizelge 4.9.** hTB parametre deęerleri

Başlangıç Sıcaklığı ( $T_0$ )	Soğutma Oranı ( $\alpha$ )	İterayon Sayısı (L)	Bitiş Sıcaklığı ( $T_{son}$ )
50	0,90	10	1

GA ve hTB çalıştırılırken kullanılan durdurma kriterleri şu şekildedir.

- Küçük boyutlu problemlerde optimum çözüme ulaşıldığında,
- Orta ve büyük boyutlu problemlerde sabit iterasyon sayısına ulaşıldığında algoritma sonlandırma işlemi gerçekleştirilir.

Geliştirilen meta-sezgisel yöntemlerin etkinlikleri, küçük boyutlu test problemleri üzerinde matematiksel model çözümü ile büyük boyutlarda çözüm kalitesi ve bilgisayar zamanı açısından karşılaştırılmıştır.

7, 8 ve 9 düğümlü problemlerin tamamında Matematiksel model, GA ve hTB ile optimum çözümlere ulaşılmıştır. Ancak Cplex çözücüsünün kullandığı ortalama bilgisayar cpu

zamanının hTB'den 4,65 kat, hTB algoritmasının kullandığı ortalama bilgisayar cpu zamanının ise GA'dan 83,5 kat daha fazla olduğu gözlenmiştir. Dolayısıyla üç yönteminde bu test problemleri üzerinde %100 başarılı olduğu söylenebilir ancak GA en etkin olanıdır.

10 düğüm ve 13 kenarlı örneklerde Cplex, GA ve hTB'nin performansı, 10 düğümden küçük örneklerin performansları ile benzerlik göstermektedir. Cplex çözücü ile eniyi çözümlerin bulunamadığı 10 düğüm 18 kenarlı üç test örneğinin (Testprob26, Testprob28, Testprob30) tamamında GA daha iyi sonuçlar bulmuştur.

Kenar sayısının artmasıyla birlikte farklı sonuçlar elde edilmiştir. Cplex'in uygun çözümler üretilmediği 20 düğümlü 8 adet test örneğinin 6 tanesi için hTB daha iyi sonuçlar vermiştir. Bununla birlikte, GA sonuçları 2 test örneği için hTB'den daha iyidir.

**Çizelge 4.10.** ZB-HÇPP matematiksel model, GA ve hTB çözüm sonuçları

Test Problemleri	V	h	E	$O^{Cplex}$	$T^{Cplex}$ (sn)	Gap (%)	$M^{GA}$	$O^{GA}$	$T^{GA}$ (sn)	$M^{hTB}$	$O^{hTB}$	$T^{hTB}$ (sn)
HÇPP1	8	3	10	<b>41,90</b>	1,17	0	41,90	<b>41,90</b>	0	41,90	<b>41,90</b>	0
HÇPP2	7	3	12	<b>70,64</b>	17,06	0	70,64	<b>70,64</b>	0	70,64	<b>70,64</b>	1,15
Rnd_inst1	7	2	7	<b>93,65</b>	0,13	0	93,65	<b>93,65</b>	0	93,65	<b>93,65</b>	0
Rnd_inst2			10	<b>100,72</b>	1,52	0	100,72	<b>100,72</b>	0	100,72	<b>100,72</b>	0
Rnd_inst3		3	7	<b>102,74</b>	0,09	0	102,74	<b>102,74</b>	0	102,74	<b>102,74</b>	1,23
Rnd_inst4			10	<b>83,31</b>	0,19	0	83,31	<b>83,31</b>	0	83,31	<b>83,31</b>	0
Rnd_inst5		4	7	<b>108,13</b>	0,09	0	108,13	<b>108,13</b>	0	108,13	<b>108,13</b>	0
Rnd_inst6			10	<b>96,16</b>	0,08	0	96,16	<b>96,16</b>	0	96,16	<b>96,16</b>	1,03
Rnd_inst7		5	7	<b>140,83</b>	0,09	0	140,83	<b>140,83</b>	0	140,83	<b>140,83</b>	3,19
Rnd_inst8			10	<b>100,46</b>	0,17	0	100,46	<b>100,46</b>	1,81	100,46	<b>100,46</b>	0

**Çizelge 4.10.** (devam)

Test Problemleri	$ V $	$ h $	$ E $	$O^{Cplex}$	$T^{Cplex}$ (sn)	Gap (%)	$M^{GA}$	$O^{GA}$	$T^{GA}$ (sn)	$M^{hTB}$	$O^{hTB}$	$T^{hTB}$ (sn)
Rnd_inst9	8	2	8	<b>122,57</b>	0,11	0	122,57	<b>122,57</b>	0	122,57	<b>122,57</b>	1,25
Rnd_inst10			10	<b>113,09</b>	0,05	0	113,09	<b>113,09</b>	0	113,09	<b>113,09</b>	100,0
Rnd_inst11		3	8	<b>125,26</b>	0,03	0	125,26	<b>125,26</b>	0	125,26	<b>125,26</b>	3,0
Rnd_inst12			10	<b>102,64</b>	0,03	0	102,64	<b>102,64</b>	0	102,64	<b>102,64</b>	6,64
Rnd_inst13		4	8	<b>158,86</b>	0,08	0	158,86	<b>158,86</b>	0	158,86	<b>158,86</b>	0
Rnd_inst14			10	<b>168,95</b>	1,38	0	168,95	<b>168,95</b>	0	168,95	<b>168,95</b>	0
Rnd_inst15		5	8	<b>108,17</b>	0,16	0	108,17	<b>108,17</b>	0	108,17	<b>108,17</b>	0
Rnd_inst16			10	<b>108,37</b>	0,17	0	108,37	<b>108,37</b>	0	108,37	<b>108,37</b>	0
Rnd_inst17	9	2	10	<b>113,185</b>	0,47	0	113,185	<b>113,185</b>	0	113,185	<b>113,185</b>	1,34
Rnd_inst18			12	<b>131,98</b>	501,42	0	131,98	<b>131,98</b>	0	131,98	<b>131,98</b>	7,89
Rnd_inst19		3	10	<b>95,58</b>	0,11	0	95,58	<b>95,58</b>	0	95,58	<b>95,58</b>	0
Rnd_inst20			12	<b>128,22</b>	174,0	0	128,22	<b>128,22</b>	0	128,22	<b>128,22</b>	1,18
Rnd_inst21		4	10	<b>141,25</b>	0,42	0	141,25	<b>141,25</b>	0	141,25	<b>141,25</b>	0
Rnd_inst22			12	<b>115,20</b>	0,09	0	115,20	<b>115,20</b>	0	115,20	<b>115,20</b>	0
Rnd_inst23		5	10	<b>132,27</b>	2,53	0	132,27	<b>132,27</b>	0	132,27	<b>132,27</b>	0
Rnd_inst24			12	<b>129,02</b>	0,55	0	129,02	<b>129,02</b>	0	129,02	<b>129,02</b>	23,23

Çizelge 4.10. (devam)

Test Problemleri	V	h	E	$O^{Cplex}$	$T^{Cplex}$ (sn)	Gap (%)	$M^{GA}$	$O^{GA}$	$T^{GA}$ (sn)	$M^{hTB}$	$O^{hTB}$	$T^{hTB}$ (sn)
Rnd_inst25	10	2	13	<b>1405,88</b>	14,75	0	1405,88	<b>1405,88</b>	0	1405,88	<b>1405,88</b>	0
Rnd_inst26			18	1749,73	10800	87	1757,6	<b>1738,25</b>	500,38	1855,98	1832,3	451,45
Rnd_inst27		3	13	<b>1330,77</b>	21,0	0	1330,77	<b>1330,77</b>	0	1373,43	1373,43	223,55
Rnd_inst28			18	1652,62	10800	93	1631,03	<b>1610,20</b>	412,42	1666,47	1666,47	384,67
Rnd_inst29		4	13	<b>1154,25</b>	22,0	0	1154,26	<b>1154,25</b>	0	1154,25	<b>1154,25</b>	10,69
Rnd_inst30			18	1452,63	10800	61	1446,14	<b>1439,22</b>	51,41	1469,60	1447,50	303,95
Rnd_inst31		5	13	<b>1147,60</b>	0,53	0	1147,60	<b>1147,60</b>	0	1147,60	<b>1147,60</b>	0
Rnd_inst32			18	-	10800	-	2037,89	<b>2014,59</b>	403,23	2039,85	<b>2014,59</b>	390,26
Rnd_inst33	20	2	55	-	-	-	5090,23	4955,06	1534,3	4547,41	<b>4531,80</b>	1532,0
Rnd_inst34			76	-	-	-	6379,91	6178,44	1505,0	5830,50	<b>5770,90</b>	2736,4
Rnd_inst35		3	55	-	-	-	4384,66	4162,59	389,3	4082,95	<b>3891,70</b>	755,1
Rnd_inst36			76	-	-	-	5849,10	<b>5663,70</b>	1517,0	6009,81	5787,10	1606,4
Rnd_inst37		4	55	-	-	-	5526,38	5376,51	1354,0	5272,82	<b>5150,20</b>	1401,1
Rnd_inst38			76	-	-	-	7163,84	7108,46	1716,9	7008,27	<b>6824,48</b>	1469,2
Rnd_inst39		5	55	-	-	-	5253,84	<b>5054,89</b>	425,2	5835,96	5694,00	847,2
Rnd_inst40			76	-	-	-	6337,77	6284,85	1168,8	6139,39	<b>5861,46</b>	750,6

Çizelge 4.10. (devam)

Test Problemleri	$ V $	$ h $	$ E $	$O^{Cplex}$	$T^{Cplex}$ (sn)	Gap (%)	$M^{GA}$	$O^{GA}$	$T^{GA}$ (sn)	$M^{hTB}$	$O^{hTB}$	$T^{hTB}$ (sn)
Rnd_inst41	30	2	125	-	-	-	10441,95	10381,20	5796,1	9055,50	<b><u>8996,30</u></b>	8297,9
Rnd_inst42			174	-	-	-	14324,43	13606,70	11967,0	12163,67	<b><u>12148,00</u></b>	16181,0
Rnd_inst43		3	125	-	-	-	10943,48	10788,10	2625,8	10248,45	<b><u>10138,27</u></b>	3569,3
Rnd_inst44			174	-	-	-	13756,43	13624,10	4253,5	12213,83	<b><u>12146,35</u></b>	7652,7
Rnd_inst45		4	125	-	-	-	12332,20	12125,50	3980,2	10707,38	<b><u>10603,00</u></b>	3349,1
Rnd_inst46			174	-	-	-	16285,28	16082,60	4099,2	14573,51	<b><u>14233,00</u></b>	3743,7
Rnd_inst47		5	125	-	-	-	11526,68	11386,90	2760,8	10950,81	<b><u>10779,86</u></b>	1713,8
Rnd_inst48			174	-	-	-	16864,38	16786,30	3827,5	15341,61	<b><u>15209,00</u></b>	4809,5
Rnd_inst49	40	2	223	-	-	-	19853,23	19731,60	11104,0	16822,00	<b><u>16758,00</u></b>	24526,0
Rnd_inst50			312	-	-	-	29242,13	29196,20	29493,0	24654,00	<b><u>24595,00</u></b>	52740,0
Rnd_inst51		3	223	-	-	-	19464,18	19192,70	10401,0	17060,30	<b><u>17042,47</u></b>	14118,8
Rnd_inst52			312	-	-	-	25926,50	25788,90	14188,0	22687,23	<b><u>22560,42</u></b>	21330,1
Rnd_inst53		4	223	-	-	-	20366,95	20095,80	8466,3	18447,78	<b><u>18400,42</u></b>	8052,0
Rnd_inst54			312	-	-	-	25252,33	25049,00	10186,0	22139,97	<b><u>22057,40</u></b>	14629,7
Rnd_inst55		5	223	-	-	-	19751,18	19466,00	4031,8	19405,99	<b><u>19346,34</u></b>	7265,2
Rnd_inst56			312	-	-	-	27236,65	27079,00	9982,3	24437,71	<b><u>24222,05</u></b>	17325,4

**Çizelge 4.10.** (devam)

Test Problemleri	$ V $	$ h $	$ E $	$O^{\text{Cplex}}$	$T^{\text{Cplex}}$ (sn)	Gap (%)	$M^{\text{GA}}$	$O^{\text{GA}}$	$T^{\text{GA}}$ (sn)	$M^{\text{hTB}}$	$O^{\text{hTB}}$	$T^{\text{hTB}}$ (sn)
Rnd_inst57	50	2	350	-	-	-	31447,65	31333,70	43548,0	27233,27	<b><u>27099,58</u></b>	62880,9
Rnd_inst58			490	-	-	-	37923,37	37869,20	42089,0	32041,33	<b><u>31978,00</u></b>	130625,0
Rnd_inst59		3	350	-	-	-	34763,28	34282,40	28752,0	30203,92	<b><u>30013,36</u></b>	43064,6
Rnd_inst60			490	-	-	-	38762,50	38420,00	31194,0	34026,94	<b><u>33999,68</u></b>	64734,9
Rnd_inst61		4	350	-	-	-	32437,18	32336,50	31584,0	29816,30	<b><u>29772,36</u></b>	25974,2
Rnd_inst62			490	-	-	-	48155,73	47741,10	56180,0	41343,29	<b><u>41099,54</u></b>	53704,3
Rnd_inst63		5	350	-	-	-	34935,15	34802,00	10326,0	31856,22	<b><u>31733,51</u></b>	16233,4
Rnd_inst64			490	-	-	-	49027,53	48620,20	34854,0	43498,50	<b><u>43479,00</u></b>	46781,0

**Kalın:** Cplex, GA ve hTB sonuçları birbirine eşittir. **Kalın ve alt çizgi:** Bulunan en iyi çözüm.

Problem boyutu arttıkça hTB'nin GA'dan daha iyi performans sergilediği görülmüştür. 30, 40 ve 50 düğümlü örneklerin tamamında GA tarafından bulunan ortalama amaç fonksiyonu değeri, hTB tarafından bulunan ortalama amaç fonksiyonu değerinden %12,75 daha büyüktür. hTB algoritması bu test örneklerinin tamamında daha iyi sonuçlar bulmasına karşın algoritmanın harcadığı ortalama bilgisayar zamanı GA'dan 1,57 kat daha büyüktür. Genel olarak, hTB algoritması birçok test problemi için GA'dan daha iyi performans sergilemiştir. Şöyleki, GA sadece 5 test probleminde hTB'den daha iyi sonuçlar vermesine karşın hTB, GA'dan 30 test problemi için daha iyi sonuçlar vererek üstünlüğünü kanıtlamıştır.

Sonuç olarak, hTB'nin küçük ve orta büyüklükteki örnekler için GA ile rekabetçi olduğu ve problem boyutunun artmasıyla daha üstün performans sergilediği gözlenmiştir. Elde edilen sonuçlar iki meta-sezgisel açıdan şu şekilde değerlendirilebilir:

Genetik algoritmaların en büyük avantajlarından biri çözüm uzayının tamamını aramalarıdır. Ancak problem boyutu arttıkça olası permütasyonların sayısında hızla artmakta olup popülasyondaki mevcut permütasyonlar bu büyüme oranını takip edemeyebilirler. Bundan dolayı GA yerel optimuma takılabilir. Popülasyon sayısının artırılması ile bu sorunun çözülebilmeye karşın bilgisayar hesaplama zamanında da ciddi artış görülür. GA'lar çaprazlama ve mutasyon operatörleri ile evrimleşme süreçlerini tamamlayarak yerel optimuma takılmadan global optimuma ulaşmayı hedeflerler. Çaprazlama işlemi ile sağlam bireylerden yeni çocuklar oluşturularak popülasyondaki çeşitliliğin sağlanması amaçlanır. Ancak daha önceki bölümlerde de bahsedildiği gibi ZB-HÇPP karmaşık yapıya sahip bir problem olup çok fazla kısıt içermektedir. Bu nedenle, olurlu çözümler üretecek şekilde bireylerin çaprazlanmasıyla ortaya çıkan yeni bireylerde çok fazla farklılık gözlenmemiştir. Çünkü çaprazlama operatörü problemin olurluluğunun çok kolay bozulmasından dolayı etkin bir şekilde kullanılamamıştır. Bu sebeplerden dolayı GA'nın çözüm uzayını çok fazla araştıramadığı ve yerel optimumda takıldığı düşünülmektedir. GA'nın daha kısa hesaplama zamanında çözümlere ulaşmasının da yine bu nedenlerden kaynaklandığı muhtemeldir.



Hibrit tavlama benzetiminde sadece mevcut çözümde komşu arama metodları ile düzeltmeler yapılarak yeni çözümlere gidilir. Bu durum, çaprazlamadaki gibi kompleks yöntemlerin kullanılmasını yahut aynı ya da benzer bireylerin üretilmesini engeller. Ayrıca önerilen komşu çözüm üretme stratejisi her zaman uygun çözümler üreterek yerel optimumda takılmadan aramaya daha iyi bir çözümle devam edebilmek için çeşitlendirmeyi sağlar. Dolayısıyla ZB-HÇPP için hTB algoritmasının çok daha kolay uyarlanabildiği ve yeni olurlu çözümlerin etkin bir şekilde üretilebildiği gözlenmiştir.



## 5. SONUÇ ve ÖNERİLER

Rotalama birçok üretim ve hizmet sektörü için önemli bir karar verme sürecidir. Devletler ve müteşebbisler tarafından her yıl milyonlarca dolar bu operasyonlar için harcanmaktadır. Ayrıca yanlış planlama neticesinde çok büyük miktarda paralar israf olmaktadır. Dolayısıyla bu bağlamda yapılan çalışmalar maliyet açısından büyük önem arz etmektedir.

Son yıllarda rotalama problemlerinden biri olan ÇPP'ye ait çalışmalar ele alınarak çeşitli çözüm önerileri sunulmuştur. Bu tez çalışmasında ÇPP kavramı, çeşitleri genel olarak açıklanmış ve uygulama çalışmalarıyla etkinliği gösterilmiştir. Daha önceden HÇPP için ele alınmayan ancak literatürde ve gerçek hayat uygulamalarında karşımıza çıkan çeşitli durumlar dikkate alınarak kapsamlı çözüm önerileri sunulmuştur.

Çalışma kapsamında ilk olarak, TCDD'de rutin aralıklarla yapılan ray bakım işlemleri için tüm ray hattını dolaşarak bakım yapan yol ve katener ölçüm aracının yönsüz hatlar üzerinde en kısa uzunluktaki rotasının belirlenmesi ele alınmıştır. Problem, ÇPP yaklaşımı ile çözülmüştür. Sunulan bu uygulamada, mevcut Türkiye ray hattı bakım planları ve önerilen optimum bakım planı karşılaştırılmıştır. Elde edilen sonuçların etkinliği mevcut uygulanmakta olan durum ile karşılaştırmalar yapılarak gösterilmiştir.

Günümüz rotalama çalışmalarında bir araç yerine birden fazla araçla da yapılan işlemler olmaktadır. Bunlardan bir tanesi de kış aylarında yapılmakta olan kar küreme çalışmalarıdır. Atatürk üniversitesi doğu kampüsünde yapılan kar küreme işlemleri için k tane aracın rotası k-ÇPP yaklaşımı ile belirlenmiştir. Araçların katettikleri mesafelerin birbirleri ile dengeli olması açısından Dk-ÇPP modeli önerilmiştir. Hem uygulama çalışması hem de literatürden alınan farklı test problemi için her iki model çalıştırılarak sonuçlar karşılaştırılmıştır. Elde edilen sonuçlara bakıldığında, geliştirilen model ile ciddi oranlarda iyileştirmeler yapıldığı gözlenmiştir.

Bununla beraber, teknolojinin de getirdiği imkanlar sayesinde büyük ve kompleks yapıya sahip ağlar üzerinde rotalama işlemi yapılırken çeşitli varsayımlar ve özellikler altında kısıtların olması gerçek hayat uygulamalarında karşılaşılan durumlardandır. Bu durumlardan bir tanesi yolların öncelik ilişkilerinin dikkate alındığı HÇPP olup çalışmada karma tam sayılı matematiksel modeli önerilmiştir. Önerilen modelin etkinliği küçük boyutlu test problemleri üzerinde sınanarak ürettiği sonuçlar raporlanmıştır.

Literatürde sunulan HÇPP çalışmalarının tamamında rotalama esnasında araçların katettikleri mesafeler dikkate alınmıştır. Bu tür yaklaşımlar yoğun trafik koşullarına sahip kentsel alanlarda ki servis uygulamalarında zayıf kalmaktadır. Artan nüfus yoğunluğu ve hızlı kentselleşme sonucu ortaya çıkan yoğun trafik koşulları neticesinde araçların seyahat süreleri göz ardı edilemeyecek bir hal almıştır. Dolayısıyla bu çalışmada HÇPP'nin yeni bir varyasyonu ZB-HÇPP ele alınmıştır. Böylelikle, seyahat boyunca hava durumu, trafik yoğunluğu ve kazaları gibi çeşitli nedenlerden dolayı oluşabilecek süre değişimleri göz önünde bulundurularak, gerçek hayat problemleri için daha uygulanabilir hali ile ortaya konmuştur. Literatürde ilk olarak, zamana bağlı hız fonksiyonu özelliğini göz önüne alarak “ilk giren ilk çıkar (FIFO)” yaklaşımını sağlayan, doğrusal yapıya sahip ve kesin hesaplama gerçekleştiren yeni bir karma tamsayılı matematiksel model geliştirilmiştir.

Dror *et al.* (1987), Korteweg and Volgenant (2006) ortaya koydukları iki küçük test örneğine ilaveten rassal olarak 64 adet test problemi türetilmiştir. Verbeeck *et al.* (2014) tarafından kullanılan seyahat hızı matrisi hiyerarşik yollar ile ilişkilendirilerek HÇPP'ye uyarlanmıştır.

Önerilen ZB-HÇPP matematiksel modeli ile bu problemler üzerinde gerçekleştirilen deneysel çalışmaların sonuçları sunulmuş ve yorumlanmıştır. Model; dağıtım, nakliye ve servis firmalarının her zaman her türlü trafik durumunu dikkate alarak en kısa süreli rotalarının belirlenmesine yardımcı olabilir. Nitekim HÇPP ve ZB-HÇPP gerçek hayatta kar ve buzlanma ile mücadele işlemleri, devriye gezen güvenlik araçları, çöp toplama faaliyetleri ve önceliklere dayalı geçiş yollarının önemli olduğu yol bakım faaliyetleri gibi

çeşitli uygulamalar için genişletilebilir. Bu tür uygulamalarda aracın trafikte harcadığı süre, ulaşımda güvenliğin sağlanması ve kazaların önlenmesi açısından kilit bir öneme sahiptir. Bununla birlikte önerilen matematiksel modelin gerçek hayat problemleri için çok küçük sayılabacak test problemlerini çözebildiği, orta ve büyük boyutlu test problemleri üzerinde etkisini kaybettiği görülmüştür.

Son yıllarda optimizasyon kavramı işletmelerin planlama, tasarım, çizelgeleme ve maliyetlendirme faaliyetlerinde oldukça önemli bir konu haline gelmiştir. Ancak büyük boyutlu sistemlerde, bu alanlardaki etkinliklerin optimize edilmesi zor olduğundan, bu tür işlemlerin algoritmalar vasıtasıyla bilgisayar programları ile gerçekleştirilmesi mümkündür. Optimizasyon uygulamaları için en etkili sonuç veren yöntemlerden biri meta-sezgisel algoritmalarıdır. ZB-HÇPP için sonuç elde edilemeyen büyük boyutlu problemlerde meta-sezgisel algoritmalara ihtiyaç duyulmuş ve iki farklı meta-sezgisel algoritma geliştirilmiştir. Bunlar sırasıyla genetik algoritma ve hibrit tavlama benzetimidir. GA'da çözüm uzayının tamamı rassal olarak aranırken TB'de komşu arama yaklaşımı ile çözüm uzayında eniyi çözümün bulunması amaçlanır. Her iki sezgiselde de yeni çözümlerin hızlı ve doğru oluşturulması çözüm uzayının daha hızlı aranmasını ve bilgisayar çözüm zamanının kısalsmasını sağlar. Dolayısıyla tüm bu işlemlerin etkin ve hızlı bir şekilde yapılabilmesi için sezgisel yöntemlerin en temel basamağı olan çözüm gösteriminin doğru ve probleme uygun bir şekilde yapılması gerekmektedir. HÇPP Np-zor problem olup onun bir uzantısı olan ZB-HÇPP'de Np-zor problemdir. Problemde düğümler yerine ayrıtların dikkate alınması, her bir ayrıttan en az bir kez (birden fazla kez geçilebilme izninin olması) geçilmesi ve problemin temel kısıtları olan hiyerarşik önceliklerinin bozulmaması gibi çeşitli kısıtlar altında incelenmesi, problemi karmaşık bir yapıya sokmuştur. Bu sebeple yeni çözümler üretme aşamasında aksaklıklar yaşanmaması için rotalama problemlerinde sıkça kullanılan permütasyon çözüm gösterimi sunulmuştur. Bunun için şebeke problemi iş çizelgeleme problemi gibi düşünülmüş, geçilmesi gerekli ayrıtlar yapılacak işler halinde tanımlanmıştır. Yollar arasındaki hiyerarşik öncelikler de yine aynı mantıkla o yola karşı gelen işin öncülü gibi tanımlanmıştır. Önerilen çözüm gösterimi her iki meta-sezgisel için ele alınmıştır. Meta-sezgisel yöntemlerin etkinliği, küçük boyutlarda matematiksel model ile elde edilen en

iyi çözümler ile kıyaslanarak gösterilmiş, orta ve büyük boyutlu problemlerde ise alınan sonuçlar raporlanarak tartışılmıştır.

İşletmeler açısından, lojistik alanında küçük iyileştirmelerin dahi büyük kazanımlar getirebileceği bilinmektedir. Bundan dolayı, sunulan çalışmanın gerek üretim gerekse hizmet sektöründe ekonomik anlamda katkı sağlayacağı dolayısı ile ülke ekonomisi içinde olumlu yönde etkisi olacağı düşünülmektedir. Bu çalışma kapsamında, gerçek hayat uygulamalarında sıklıkla karşılaşılan HÇPP ve ZB-HÇPP çözümleri için çeşitli önerilerde bulunulmuştur. Bir araç yerine birden fazla aracın kullanılması, araçların belirli kapasiteye sahip olması gibi farklı durumlar rotalama ve ulaştırma planını etkileyecektir. Bu varsayımlar altında HÇPP'nin tekrardan ele alınarak çözüm yolları araştırılabilir.

Gerçek hayat problemlerinde bazı amaçlar karar vericiler tarafından net bir şekilde belirlenebilirken, bazı amaçlar bu kadar rahat tanımlanamazlar. Bundan dolayı HÇPP'ye çeşitli amaçlar için bulanık veya gri modelleme ile çözüm önerileri sunulabilir.

Bununla beraber yapılan çalışma kapsamında HÇPP için tanımlanan seyahat süreleri deterministik yapıdadır. Oysaki gerçek hayat uygulamalarında bu verilerin belirli bir ortalama ve standart sapma değerine bağlı olarak da ele alınabileceği bilinmektedir. Bu amaçla problemin stokastik programlama ile modellenebileceği literatürde görülen boşluklardandır.

Bunlara ilaveten ZB-HÇPP için çözüm önerileri aşağıdaki gibi özetlenebilir:

- Küçük test problemlerini makul bir süre içerisinde çözebilen kesin çözüm algoritmaları (Dal-kesme algoritması, lagrange gevşetmesi, sütun türetme vb.) geliştirilebilir.
- Geliştirilen meta-sezgisel algoritmalar farklı parametre değerleri (başlangıç sıcaklığı, soğutma oranı, popülasyon büyüklüğü, çaprazlama ve mutasyon oranı gibi) altında incelenebilir.

- GA ve hTB’de evrimleşme sürecinde farklı yeni bireyler oluşturma teknikleri kullanılabilir.
- ZB-HÇPP için farklı meta-sezgisel yaklaşımlar geliştirilebilir.



**KAYNAKLAR**

- Ahn, B.H. and Shin J.Y., 1991. Vehicle-routing with time windows and time-varying congestion. *Journal of the Operational Research Society*, 42(5), 393-400.
- Ahr, D. and Reinelt G., 2002. New heuristics and lower bounds for the min-max k-Chinese postman problem. 10th Annual European Symposium on Algorithms, Rome, Italy.
- Ahr, D. and Reinelt G., 2006. A tabu search algorithm for the min-max k chinese postman problem. *Computers & Operations Research*, 33(12), 3403-3422.
- Ahuja, R.K. and Orlin J.B., 2002. Combinatorial algorithms for inverse network flow problems. *Networks*, 40(4), 181-187.
- Ahuja, R.K., Magnanti, T.L. and Orlin, J.B., 1993. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 863 p, New Jersey, USA.
- Albiach, J., Sanchis J. and Soler D., 2008. An asymmetric TSP with time windows and with time-dependent travel times and costs: an exact solution through a graph transformation. *European Journal Operational Research*, 189:789-802.
- Alfa, A.S. and Liu D.Q., 1988. Postman routing problem in a hierarchical network. *Engineering Optimization*, 14: 127-138.
- Aminu, U.F. and Eglese R.W., 2006. A constraint programming approach to the Chinese postman problem with time windows. *Computers & Operations Research*, 33(12), 3423-3431.
- Arakaki, R.K. and Usberti F.L., 2018. Hybrid genetic algorithm for the open capacitated arc routing problem. *Computers & Operations Research*, 90:221-231.
- Avriel, M. and Golany B., 1996. *Mathematical Programming for Industrial Engineers*, Marcel Dekker Inc., 656 p, New York, USA.
- Balseiro, S.R., Loiseau I. and Ramonet J., 2011. An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers & Operations Research*, 38 (6), 954-966.
- Black, D., Eglese R. and Wøhlk S., 2013. The time-dependent prize-collection arc routing problem. *Computers & Operations Research*, 40, 526-535.
- Cabral, E.A., Gendreau M., Ghiani G. and Laporte G., 2004. Solving the hierarchical Chinese postman problem as a rural postman problem. *European Journal of Operational Research*, 155(2), 44-50.
- Cerny, V., 1985. Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1), 41-51.
- Chen, H.K., Hsueh C.F. and Chang M.S., 2006. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(5), 383-408.
- Colombi, M., Corberan A., Mansini R., Plana I. and Sanchis J.M., 2017. The hierarchical mixed rural postman problem: polyhedral analysis and a branch-and-cut algorithm. *European Journal of Operational Research*, 257, 1-12.
- Corberan, A., Marti R. and Sanchis J.M., 2002. A Grasp heuristic for the mixed Chinese postman problem. *European Journal Of Operational Research*, 142(1), 70-80.

- Corberan, A., Plana I., Rodriguez-Chia A.M. and Sanchis J.M., 2013. A branch and cut algorithm for the maximum benefit Chinese postman problem. *Mathematical Programming*, 141(1-2), 21-48.
- Cordeau, J.F., Ghiani G. and Guerriero E., 2014. Analysis and branch-and-cut algorithm for the time-dependent travelling salesman problem. *Transportation Science*, 48, 46-58.
- Damodaran, P., 1997. A methodology for dynamic planning of road service during a snow fall. MS. Thesis, Northern Illinois University, DeKalb, IL.
- Damodaran, P., Krishnamurthi M. and Srihari K., 2008. Lower bounds for hierarchical Chinese postman problem. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 15(1), 36-44.
- Dhein, G., Araújo O.C.B. and Cardoso G., 2018. Genetic local search algorithm for a new bi-objective arc routing problem with profit collection and dispersion of vehicles. *Expert Systems with Applications*, 92, 276-288.
- Dror, M., Stern H. and Trudeau P., 1987. Postman tour on a graph with precedence relation on arcs. *Networks*, 17, 283-294.
- Edmonds, J. and Johnson E.L., 1973. Matching euler tours and the Chinese postman problem. *Mathematical Programming*, 5, 88-124.
- Eglese, R.W. and Li L.Y.O., 1992. Efficient routing for winter gritting. *Journal Operational Research Society*, 43, 1031-1034.
- Eiselt, H.A., Gendreau M. and Laporte G., 1995. Arc Routing Problems, Part 1: The Chinese Postman Problem. *Operations Research*, 43(2), 231-242.
- Emel, G.G., Taşkın Ç. and Dinç E., 2004. Yönsüz Çinli postacı problemi: polis devriye araçları için bir uygulama. *Anadolu Üniversitesi Sosyal Bilimler Dergisi*, 3(1), 121-140.
- Euchi, J., Chabchoub H. And Yassine A., 2010. Metaheuristics optimization via memory to solve the profitable arc tour problem. 8th International Conference of Modeling and Simulation, Hammamet, Tunisia.
- Feillet, D., Dejax P. and Gendreau M., 2005. The profitable arc tour problem: solution with a branch-and-price algorithm. *Transportation Science*, 39(4), 539-552.
- Figliozzi, M.A., 2012. The time-dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E-Logistics and Transportation Review*, 48 (3), 616-636
- Fleischmann, B., Gietz M. and Gnutzmann S., 2004. Time-varying travel times in vehicle routing. *Transportation Science*, 38 (2), 160-173.
- Florian, M. (1984). *Transportation Planning Models*, Elsevier Science Publishers B.V., 499 p, North Holland.
- Fu, H., Nie L., Meng L., Sperry B.R. and He Z., 2015. A hierarchical line planning approach for a large-scale high speed rail network: The China Case. *Transportation Research Part A*, 75, 61-83.
- Gaafar, L.K. and Masoud S.A., 2005. Genetic algorithms and simulated annealing for scheduling in agile manufacturing. *International Journal of Production Research*, 43(14), 3069-3085.
- Ghiani, G. and Improta G., 2000. An algorithm for the hierarchical Chinese postman problem. *Operations Research Letters*, 26, 27-32.



- Glover, F. and Kochenberger G.A., 2003. Handbook of Metaheuristics, US: Kluwer Academic Publishers, chapter 10, pp. 287-291.
- Goldberg, D.E., 1987. Computer-Aided Pipeline Operation Using a Genetic Algorithms and Rule Learning-Part I-Genetic Algorithms in Pipeline Optimization. *Engineering with Computers*, 3, 35-45.
- Harary, F., 1960. Some historical and intuitive aspects of graph theory. *SIAM Review*, 2(2), 123-131.
- Hart, E., Ross P. and Corne D., 2005. Evolutionary scheduling: A Review. *Genetic Programming and Evolvable Machines*, 6, 191-220.
- Harwood, K., Mumford C. and Eglese R., 2013. Investigating the use of metaheuristics for solving single vehicle routing problems with time varying traversal costs. *Journal of Operational Research Society*, 64, 34-47.
- Hashimoto, H., Yagiura M. and Ibaraki T., 2008. An iterated local search algorithm for the time-dependent vehicle routing problem with time Windows. *Discrete Optimization*, 5 (2), 434-456.
- Helvig, C., Robins G. and Zelikovsky A., 2003. The moving-target travelling salesman problem. *Journal of Algorithms*, 49, 153-174.
- Hill, A.V. and Benton W.C., 1992. Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society*, 43(4), 343-351.
- Hong, S.P., Kim K.M., Lee K. and Park B.H., 2009. A pragmatic algorithm for the train-set routing: the case of Korea high-speed railway. *The International Journal of Management Science*, 37(3), 637-645.
- Hua, J. and Li-shan K., 2003. Genetic algorithm for Chinese postman problems. *Wuhan University Journal of Natural Sciences*, 8(1), 316-318.
- Ichoua, S., Gendreau M. and Potvin J.Y., 2003. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2), 379-396.
- Ignizio, P. J., 1980. Solving Large-Scale Problems: A Venture Into a New Dimension. *The Journal of the Operational Research Society*, 31(3), 217-225.
- Irnich, S., 2008. Undirected postman problems with zigzagging option: A cutting-plane approach. *Computers & Operations Research*, 35(12), 3998-4009.
- Ishibuchi, H., Misaki S. and Tanaka H., 1995. Modified simulated annealing algorithms for the flow shop sequencing problem. *European Journal of Operational Research*, 81, 388-398.
- Jiang, H., Kang L., Zhang S. and Zhu F., 2010. Genetic algorithm for mixed Chinese postman problem. 5th International Symposium ISICA, Wuhan, China.
- Jiang, Q., Sarker R. and Abbass H., 2005. Tracking moving targets and the non-stationary traveling salesman problem. *Complexity*, 11, 171-179.
- Jung, S. and Haghani A., 2001. Genetic algorithm for the time-dependent vehicle routing problem. *Transportation Research Record: Journal of the Transportation Research Board*, 1771, 164-71.
- Kirkpatrick, S., Gelatt Jr C.D. and Vecchi M.P., 1983. Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Koç, Ç. ve Karaoğlan İ., 2012. Çok kullanımlı ve zaman pencereli araç rotalama problemi için bir matematiksel model. *Journal of the Faculty of Engineering & Architecture of Gazi University*, 27(3), 569-576.

- Koç, Ç., 2012. Zaman-Bağımlı Araç Rotalama Problemi. Y.Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Korteweg, P. and Volgenant T., 2006. On the hierarchical Chinese postman problem with linear ordered classes. *European Journal of Operational Research*, 169, 41-52.
- Kramberger, T., Štrubelj G. and Žerovnik J., 2009. Chinese postman problem with priority nodes. *Foundations of Computing and Decision Sciences*, 34(4), 233-264.
- Krishnamurthi, M. and Damodaran P., 1998. A modified postman tour heuristic for efficient snow removal planning, *Proceedings of 7th Industrial Engineering Research Conference, Banff, Canada*.
- Kumar, S.N. and Panneerselvam R., 2015. A time-dependent vehicle routing problem with time windows for e-commerce supplier site pickups using genetic algorithm. *Intelligent Information Management*, 7, 181-198.
- Kuo, Y., Wang C.C. and Chuang P.Y., 2009. Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds. *Computers & Industrial Engineering*, 57 (4), 1385-1392.
- Kwan, Mei-Ko, 1962. Chinese Postman Problem, *Graphic Programming Using Odd or Even Points, Chinese Math.*, 1, p. 273-277.
- Lacomme, P., Prins C. and Sevaux M., 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers & Operations Research*, 33(12), 3473-3493.
- Lannez, S., Artigues C., Damay J. and Gendreau M., 2015. A railroad maintenance problem solved with a cut and column generation matheuristic. *Networks, Wiley*, 66(1), 40-56.
- Lemieux, P.F. and Campagna L., 1984. The snow ploughing problem solved by a graph theory algorithm. *Civil Engineering Systems*, 1, 337-341.
- Liden, T., 2015. Railway infrastructure maintenance – a survey of planning problems and conducted research. *Transportation Research Procedia*, 10, 574-583.
- Ma, Y.H. and Tian G.L. and Li X., 2015. Genetic algorithm for the capacitated Chinese postman problem on mixed networks. *Applied Mechanics and Materials*, 701, 44-49.
- Maden, W., Eglese R. and Black D., 2009. Vehicle routing and scheduling with time-varying data: a case study. *Journal of Operational Research Society*, 61, 515-522.
- Malandraki, C. and Daskin M.S., 1992. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science*, 26(3), 185-200.
- Malandraki, C. and Daskin M.S., 1993. The maximum benefit Chinese postman problem and the maximum benefit traveling salesman problem. *European Journal of Operational Research*, 65(2), 218-234.
- Malandraki, C. and Dial R.B., 1996. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research*, 90(1), 45-55.
- Malandraki, C., 1989. Time dependent vehicle routing problems: formulations, solution algorithms and computational experiments. MS Thesis, Northwestern University, Evanston, Illinois.
- Minieka, E., 1978. *Optimization algorithms for networks and graphs*, Marcel Dekker, Inc, 356 p, Newyork.

- Mohammed, M.A., Ghani M.K.A., Hamed R.I., Mostafa S.A., Ahmad M.S. and Ibrahim D.A., 2017. Solving vehicle routing problem by using improved genetic algorithm for optimal solution. *Journal of Computational Science*, 21, 255-262.
- Moon, C., Kim J., Choi G. and Seo Y., 2002. An efficient genetic algorithm for the travelling salesman problem with precedence constraints. *European Journal of Operational Research*, 140, 606-617.
- Moreira, M.R. and Ferreira J.S., 2010. A genetic algorithm for the undirected rural postman problem. <https://www.semanticscholar.org>, (10.05.2017).
- Morgan, M.J.W. and Mumford C.L., 2009. A weight-coded genetic algorithm for the capacitated arc routing problem. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, Montreal, Québec, Canada,
- Mukhairez, H.H.A. and Maghari A.Y.A., 2015. Performance comparison of simulated annealing, GA and ACO applied to TSP. *International Journal of Intelligent Computing Research*, 6(4), 647-654.
- Murali, P., Ordóñez, F., Dessouky, M.M., 2016. Modelling strategies for effectively routing freight trains through complex networks. *Transportation Research Part C*, 70, 197-213.
- Paksoy., S., 2007. Genetik algoritma ile proje çizelgeleme. *Doktora Tezi, Çukurova Üniversitesi, Adana, Türkiye.*
- Park, Y.B., 2000. A solution of the bicriteria vehicle scheduling problems with time and area-depend travel speeds. *Computers & Industrial Engineering*, 38, 173–187.
- Pearn, W.L. and Chiu W.C., 2005. Approximate solutions for the maximum benefit Chinese postman problem. *International Journal of Systems Science*, 36(13), 815-822.
- Pearn, W.L. and Wang K.H., 2003. On the maximum benefit Chinese postman problem. *Omega*, 31(4), 269-273.
- Peng, F. and Ouyang Y., 2012. Track maintenance production team scheduling in railroad networks. *Transportation Research Part B: Methodological*, 46(10), 1474-1488.
- Peng, F., Kang S., Li X., Ouyang Y., Somani K. and Acharya D., 2011. A heuristic approach to the railroad track maintenance scheduling problem. *Computer – Aided Civil and Infrastructure Engineering*, 26(2), 129-145.
- Prakash, S., Sharma M.K. and Singh A., 2009. A heuristic for multi-objective Chinese postman problem. *International Conference on Computers & Industrial Engineering*, Troyes, France.
- Rabbani, M., Alamdar S.F. and Farrokhi-Asl H., 2016. Capacitated windy rural postman problem with several vehicles: A hybrid multi-objective simulated annealing algorithm. *International Journal of Supply and Operations Management*, 2(4),1003-1020.
- Razali, N.M., 2015. An efficient genetic algorithm for large scale vehicle routing problem subject to precedence constraints. *Procedia – Social and Behavioral Sciences*, 195, 1922-1931.
- Reeves, C.R., 1993, *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc., 320 p, New York, NY, USA.
- Samadi-Dana S., Paydar M.M. and Jouzdani J., 2017. A simulated annealing solution method for robust school bus routing. *International Journal of Operational Research*, 28(3), 307-326.

- Sayata, U.B. and Desai N.P., 2015. An algorithm for Hierarchical Chinese postman problem using minimum spanning tree approach based on Kruskal's algorithm. IEEE International Advance Computing Conference (IACC), Bangalore, India.
- Schlake, B.W., Edwards J.R., Hart J.M., Barkan C.P.L., Todorovic S. and Ahuja N., 2009. Automated inspection of railcar underbody structural components using machine vision technology. In proceedings of the TRB 88<sup>th</sup> Annual Meeting, Washington, DC.
- Schneider, J., 2002. The time-dependent travelling salesman problem. *Physica A*, 314, 151-155.
- Shafahi, A. and Haghani A., 2015. Generalized maximum benefit multiple Chinese postman problem. *Transportation Research Part C: Emerging Technologies*, 55, 261-272.
- Sun, J., Meng Y. and Tan G., 2015. An integer programming approach for the Chinese postman problem with time-dependent travel time. *Journal of Combinatorial Optimization*, 29(3), 565-588.
- Sun, J., Tan G. and Hou G., 2011a. A new integer programming formulation for the Chinese postman problem with time dependent travel times. *World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering*, 5(4), 410-414.
- Sun, J., Tan G. and Qu H., 2011b. Dynamic programming algorithm for the time dependent Chinese Postman Problem. *Journal of Information and Computational Science*, 8(5), 833-841.
- T.C. Devlet Demiryolları İşletmesi Genel Müdürlüğü 2014 Yılı Faaliyet Raporu. <http://www.tcdd.gov.tr/files/istatistik/2014faaliyetraporu.pdf> (10.10.2015)
- Tagmouti, M., Gendreau M. and Potvin J.Y., 2007. Arc routing problems with time-dependent service costs. *European Journal of Operational Research*, 181, 30-39.
- Tagmouti, M., Gendreau M. and Potvin J.Y., 2010. A variable neighborhood descent heuristic for arc routing problems with time-dependent service costs. *Computer Industrial Engineering*, 59, 954-963.
- Tagmouti, M., Gendreau M. and Potvin J.Y., 2011. A dynamic capacitated arc routing problems with time dependent service costs. *Transportation Research Part C: Emergence Technology*, 19, 20-28.
- Taha, H. A., 2002. *Operations Research: An Introduction*. 832 p, Prentice Hall.
- Thimbleby, H., 2003. Explaining code for publication. *Software-Practice and Experience*, 33(10), 975-1001.
- Tirkolaee, E.B., Alinaghian M., Sasi M.B. and Esfahani M.S., 2016. Solving a robust capacitated arc routing problem using a hybrid simulated annealing algorithm: A waste collection application. *Journal of Industrial Engineering and Management Studies*, 3(1), 61-76.
- Verbeeck, C., Sörensen K., Aghezzaf E.H. and Vansteenwegen P., 2014. A fast solution method for the time dependent orienteering problem. *European Journal of Operational Research*, 236, 419-432.
- Wang, H.F. and Wen Y.P., 2002. Time-constrained Chinese postman problems. *Computers and Mathematics with Applications*, 44, 375-387.
- Wang, L. and Zheng D.Z., 2003. An effective hybrid heuristic for flow shop scheduling. *Int. J. Adv. Manuf. Tech*, 21, 38-44.

- Willemse, E. and Joubert J.W., 2012. Applying min–max k postmen problems to the routing of security guards. *Journal of the Operational Research Society*, 63(2), 1-19.
- Wøhlk, S., 2015. Simulated annealing for the capacitated arc routing problem, using an online formulation. <https://www.researchgate.net/publication/242091823> (20.07.2018).
- Yu, V.F. and Lin S.W., 2015. Iterated greedy heuristic for the time-dependent prize-collection arc routing problem. *Computers & Industrial Engineering*, 90, 54-66.
- Zhao, X., Goncalves G. and Dupas R., 2008. A genetic approach to solving the vehicle routing problem with time-dependent travel times. 16th Mediterranean Conference on Control and Automation, Ajaccio, France.



## ÖZGEÇMİŞ

1987 yılında Erzurum’da doğdu. İlk, orta ve lise öğrenimini Erzurum’da tamamladı. 2009 yılında Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümünde lisans eğitimini tamamladı. 2009 yılında Atatürk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı. 2010 yılında Atatürk Üniversitesi’nde araştırma görevlisi olarak göreve başladı. Evli ve bir çocuk annesi olan Merve KAYACI ÇODUR, halen Atatürk Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümünde araştırma görevlisi olarak çalışmaktadır.