

ATATÜRK ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
İŞLETME ANABİLİM DALI

Tuba YAKICI AYAN

SEZGİSEL ARAŞTIRMANIN ARAÇ ROTALAMA PROBLEMLERİNDE  
KULLANILMASI VE BİR SEZGİSEL METOD DENEMESİ

89896

DOKTORA TEZİ

TEZ YÖNETİCİSİ  
Prof. Dr. Şule ÖZKAN

ERZURUM, 1999

**T.C. YÜKSEKÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ**

## ÖZET

## DOKTORA TEZİ

**SEZGİSEL ARAŞTIRMANIN ARAÇ ROTALAMA PROBLEMLERİNDE  
KULLANILMASI VE BİR SEZGİSEL METOD DENEMESİ**

Tuba YAKICI AYAN

Danışman : Prof. Dr. Şule ÖZKAN  
1999, Sayfa 262

Jüri : Prof. Dr. Şule ÖZKAN

.....

.....

Çalışmada sezgisel araştırma tanımlandıktan sonra araç rotalama problemleri ve bu problemlere ilişkin sezgisel metodlar açıklanmıştır.

Uygulama olarak, bir işyeri servis araçları rotalama problemi için sezgisel bir çözüm metodu geliştirilmiş ve söz konusu metod ile mevcut durumdakinden daha iyi bir çözüm üretilmiştir.

ABSTRACT

Ph. D. Thesis

**HEURISTIC SEARCH ON VEHICLE ROUTING PROBLEMS AND A  
HEURISTIC METHOD TRAIL**

Tuba YAKICI AYAN

Supervisor : Prof. Dr. Şule ÖZKAN  
1999, Page 262

Juri : Prof. Dr. Şule ÖZKAN

.....  
.....

In this study; firstly, heuristic search had been described. Then, vehicle routing problems and heuristic methods for that problems had been described.

As an application, a heuristic solution method had been developed for a daily transportation of employees problem. Then a solution better than existing practice has been produced.

**TEŞEKKÜR**

Doktora öğrenimim boyunca gerek bilgi ve görüşleri ile ve gerekse vermiş olduğu olumlu yönde motivasyonla bana destek olan hocam Sayın Prof. Dr. Şule ÖZKAN'a teşekkür ederim.



## TABLOLAR DİZİNİ

## Sayfa No

|                                                                                               |     |
|-----------------------------------------------------------------------------------------------|-----|
| Tablo 2.1. Rotalama Problemlerinin Karakteristikleri .....                                    | 20  |
| Tablo 5.1. Önerilen Metod ile $P=1000$ İçin Oluşturulan Rotalar .....                         | 186 |
| Tablo 5.2. Önerilen Metod ile $P=1500$ İçin Oluşturulan Rotalar .....                         | 187 |
| Tablo 5.3. Önerilen Metod ile $P=2000$ İçin Oluşturulan Rotalar .....                         | 188 |
| Tablo 5.4. Önerilen Metod ile $P=2500$ İçin Oluşturulan Rotalar .....                         | 189 |
| Tablo 5.5. Önerilen Metod ile $P=3000$ İçin Oluşturulan Rotalar .....                         | 190 |
| Tablo 5.6. Önerilen Metod ile Sağlanan Sonuçlar ve Mevcut<br>Durumunun Karşılaştırılması..... | 190 |



## ŞEKİLLER DİZİNİ

### Sayfa No

|             |                                                                                |     |
|-------------|--------------------------------------------------------------------------------|-----|
| Şekil 1.1.  | Standart bir metodun uygulanması durumu .....                                  | 6   |
| Şekil 1.2.  | Olası bir sezgisel çözüm süreci.....                                           | 8   |
| Şekil 1.3.  | AC nin uzunluğu nedir?.....                                                    | 8   |
| Şekil 2.1.  | Standart VRP rotaları .....                                                    | 16  |
| Şekil 3.1.  | TSP turu.....                                                                  | 25  |
| Şekil 3.2.  | Alt- turlar .....                                                              | 26  |
| Şekil 3.3.  | Bir düğümler setinin dışbükey kabuğu .....                                     | 35  |
| Şekil 3.4.  | Minimal tarama ağacı.....                                                      | 40  |
| Şekil 3.5.  | Euler dairesi .....                                                            | 40  |
| Şekil 3.6.  | İki-satıcı problemi.....                                                       | 57  |
| Şekil 3.7.  | Tur Birleştirme.....                                                           | 61  |
| Şekil 3.8.  | Süpürme prosedürü ile oluşturulan kümeler.....                                 | 70  |
| Şekil 3.9.  | Rotaların karşılaştırılmaları, 3-opt algoritması.....                          | 74  |
| Şekil 3.10. | $\tilde{c}_{ik}$ 'nin hesaplanması .....                                       | 77  |
| Şekil 3.11. | İki rota arasında tekrar katar yerleştirme örneği.....                         | 82  |
| Şekil 3.12. | İki rota arasında karşılıklı katar değiştirme örneği.....                      | 82  |
| Şekil 3.13. | Denklik değişimi algoritması, Adım 3'de oluşturulan<br>bir grafik örneği ..... | 96  |
| Şekil 3.14. | TGT algoritması .....                                                          | 112 |
| Şekil 3.15. | FSMVRP için Salhi, Rand (1993) Sezgiselinin akış<br>diyagramı .....            | 118 |
| Şekil 3.16. | Zaman pencereli bir rotalama problemi.....                                     | 126 |
| Şekil 3.17. | Bir 2-opt değişimi .....                                                       | 144 |
| Şekil 3.18. | Bir 2-opt* değişimi.....                                                       | 144 |
| Şekil 3.19. | İki müşterili bir seri üzerinde Or-opt .....                                   | 145 |
| Şekil 4.1.  | İzin verilen seyahat ve dönüş yönleri .....                                    | 151 |
| Şekil 4.2.  | Simetrik olmayan fakat Euler olan bir grafik örneği.....                       | 153 |

|            |                                                         |     |
|------------|---------------------------------------------------------|-----|
| Şekil 4.3. | MIXED1 sezgiseli ile elde edilen çözüm.....             | 154 |
| Şekil 4.4. | Yönsüz RPP için grafik dönüşümü.....                    | 157 |
| Şekil 4.5. | BÜYÜK KANALLAR algoritmasının gösterimi.....            | 161 |
| Şekil 4.6. | KÜÇÜK KANALLAR algoritmasının gösterimi.....            | 163 |
| Şekil 5.1. | Maksimum sapma kısıtı .....                             | 179 |
| Şekil 5.2. | Tek depo-çok araç rotalama problemi için akış diyagramı | 182 |



## İÇİNDEKİLER

|                                                                         | Sayfa No |
|-------------------------------------------------------------------------|----------|
| ÖZET .....                                                              | I        |
| ABSTRACT .....                                                          | II       |
| TEŞEKKÜR .....                                                          | III      |
| TABLolar DİZİNİ .....                                                   | IV       |
| ŞEKİLLER DİZİNİ .....                                                   | V        |
| <br>GİRİŞ .....                                                         | <br>1    |
| BİRİNCİ BÖLÜM : SEZGİSEL ARAŞTIRMA .....                                | 4        |
| 1.1. Kötü Yapılı Problemler (ISP) .....                                 | 5        |
| 1.2. Sezgisel Metodların Amacı Tatmin Etme Özelliği .....               | 9        |
| 1.3. Sezgisel Çözümün Uygunluğu .....                                   | 10       |
| 1.4. Sezgisel Yöntemin Gücü .....                                       | 10       |
| 1.5. Sezgisel Algoritmalar ve Programlar .....                          | 12       |
| 1.5.1. Çözüm-İnşa Stratejileri .....                                    | 12       |
| 1.5.2. Yap-Boz Stratejileri .....                                       | 12       |
| 1.5.3. Çözüm-İyileştirme Stratejileri .....                             | 13       |
| 1.5.4. Araştırma-Öğrenme Stratejileri .....                             | 13       |
| İKİNCİ BÖLÜM : ARAÇ ROTALAMA PROBLEMİ (VRP) .....                       | 15       |
| 2.1. Araç Rotalama Problemlerinin Yan Kısıtları .....                   | 16       |
| 2.2. Araç Rotalama Probleminin Çözümünü Zorlaştıran<br>Özellikler ..... | <br>17   |
| 2.3. Araç Rotalama Problemlerinin Sınıflandırılması .....               | 19       |
| 2.4. Araç Rotalama Problemi İçin Genel Çözüm Stratejileri .....         | 21       |
| ÜÇÜNCÜ BÖLÜM : DÜĞÜM ROTALAMA PROBLEMİ .....                            | 23       |
| 3.1. Seyyar Satıcı Problemi (TSP) .....                                 | 24       |
| 3.1.1. TSP'nin Matematiksel Formülasyonları .....                       | 26       |
| 3.1.2. TSP İçin Sezgisel Çözüm Stratejileri .....                       | 29       |
| 3.1.2.1. Tur-İnşa Algoritmaları .....                                   | 29       |

|                                                          | Sayfa No |
|----------------------------------------------------------|----------|
| 3.1.2.2. Tur İyileştirme Metodları.....                  | 42       |
| 3.1.2.3. Karma Yöntemler .....                           | 49       |
| 3.1.2.4. Yönlü (asimetrik) TSP İçin Bir Yaklaşım.....    | 50       |
| 3.1.2.5. Zaman Kısıtlı TSP İçin Sezgisel Yaklaşımlar. 52 |          |
| 3.1.2.6. TSP Tur Uzunluğunun Tahmin Edilmesi.....        | 54       |
| 3.2. Tek Depo/Çok Araç Rotalama Problemi.....            | 55       |
| 3.2.1. Katlı TSP.....                                    | 55       |
| 3.2.2. Tek Depo/Çok Araç Rotalama Problemi (VRP)         |          |
| Formülasyonları .....                                    | 57       |
| 3.2.3. VRP İçin Seçilen Sezgisel Çözüm Teknikleri .....  | 60       |
| 3.2.3.1. Clarke-Wright Tasarruf Algoritması .....        | 60       |
| 3.2.3.2. Paralel Tasarruf Esaslı Sezgisel Algoritma....  | 62       |
| 3.2.3.3. Süpürme Yaklaşımı .....                         | 65       |
| 3.2.3.4. Lagrangean Gevşetme Yaklaşımı .....             | 71       |
| 3.2.3.5. Matematik Programlama Esaslı Yaklaşık           |          |
| Algoritmalar .....                                       | 74       |
| 3.2.3.6. Faz Dönüşüm Yaklaşımı .....                     | 79       |
| 3.2.3.7. Yasak Araştırması Sezgiseli.....                | 84       |
| 3.2.3.8. Çözüm İyileştirme İçin Doyumsuzluk ve           |          |
| Denklik Değişimi Sezgiselleri.....                       | 92       |
| 3.3. Çok depo/araç Rotalama Problemi (MVRP).....         | 97       |
| 3.3.1. MVRP Formülasyonları.....                         | 97       |
| 3.3.2. Tahsis-Etme-Süpürme Sezgiseli.....                | 98       |
| 3.3.3. Tillman ve Cain (1972) Tasarruf Yaklaşımı.....    | 99       |
| 3.3.4. Golden, Magnanti ve Nguyen (1977) Tasarruf        |          |
| Yaklaşımı .....                                          | 101      |
| 3.4. Filo Hacmi ve Karışımı ile Araç Rotalama Problemi   |          |
| (FSMVRP).....                                            | 102      |
| 3.4.1. FSMVRP İçin Matematiksel Formülasyonlar.....      | 103      |
| 3.4.2. Tasarruf Sezgiselleri.....                        | 105      |

|                                                                                | <b>Sayfa No</b> |
|--------------------------------------------------------------------------------|-----------------|
| 3.4.2.1. Kombine Tasarruf Sezgiseli (CS) .....                                 | 105             |
| 3.4.2.2. İyimser Fırsat Tasarrufu Sezgiseli (OOS) .....                        | 107             |
| 3.4.2.3. Gerçekçi Fırsat Tasarrufu Sezgiseli (ROS) ...                         | 108             |
| 3.4.2.4. Gerçekçi Fırsat Tasarrufu Sezgiseli (ROS-v) .....                     | 109             |
| 3.4.3. Dev Tur (GT) Algoritmaları.....                                         | 110             |
| 3.4.3.1. Tek Bölümlü GT Algoritması (TGT) .....                                | 111             |
| 3.4.3.2. Çok Bölümlü GT Algoritması (ÇGT) .....                                | 112             |
| 3.4.4. İyileştirme Algoritması.....                                            | 113             |
| 3.4.5. Salhi ve Rand (1993) Sezgisel Çözüm Tekniği.....                        | 114             |
| 3.5. Stokastik Araç Rotalama Problemi (SVRP).....                              | 121             |
| 3.6. Zaman Pencereleli Araç Rotalama ve Programlama<br>Problemi (VRSPTW) ..... | 125             |
| 3.6.1. VRSPTW İçin Matematiksel Formülasyonlar .....                           | 127             |
| 3.6.2. VRSPTW İçin Sezgisel Çözüm Algoritmaları .....                          | 130             |
| 3.6.2.1. VRSPTW için Tasarruf Sezgiselleri .....                               | 133             |
| 3.6.2.2. Zaman-Esaslı En Yakın Komşu Sezgiseli .....                           | 134             |
| 3.6.2.3. Ardışık Giriş Sezgiseli .....                                         | 135             |
| 3.6.2.4. Paralel Giriş Sezgiseli .....                                         | 137             |
| 3.6.2.5. Zaman Esaslı Süpürme Sezgiseli.....                                   | 142             |
| 3.6.2.6. Yasak Araştırması Sezgiseli.....                                      | 142             |
| DÖRDÜNCÜ BÖLÜM : KANAL ROTALAMA PROBLEMİ (ARP).....                            | 149             |
| 4.1. Çinli Postacı Problemi (CPP) .....                                        | 150             |
| 4.1.1. Karışık Çinli Postacı Problemi (MCP) .....                              | 151             |
| 4.2. Kırsal Postacı Problemi (RPP) .....                                       | 155             |
| 4.2.1. Yönsüz Kırsal Postacı Problemi.....                                     | 156             |
| 4.2.2. Yönlü Kırsal Postacı Problemi .....                                     | 158             |
| 4.2.3. Karışık RPP (Stacker Crane Problemi) .....                              | 158             |
| 4.3. Kapasite Kısıtlı Kanal Rotalama Problemi (CARP) .....                     | 164             |
| 4.3.1. CARP İçin Sezgisel Algoritmalar.....                                    | 164             |
| 4.3.1.1. Kur-Kır Algoritması .....                                             | 165             |

|                                                          | <b>Sayfa No</b> |
|----------------------------------------------------------|-----------------|
| 4.3.1.2. Yol Tarama Algoritması .....                    | 167             |
| 4.3.1.3. Genişlet-Birleştir Algoritması .....            | 168             |
| 4.3.1.4. Paralel Giriş Algoritması .....                 | 168             |
| 4.3.1.5. Genişlet-Yerleştir Algoritması .....            | 172             |
| <b>BEŞİNCİ BÖLÜM : İŞYERİ ULAŞIM SERVİS ARAÇLARINI</b>   |                 |
| <b>ROTALAMA PROBLEMİ ÖRNEĞİ.....</b>                     | <b>175</b>      |
| 5.1. Problemin Özellikleri .....                         | 175             |
| 5.2. Probleme İlişkin Verilerin Elde Edilmesi .....      | 177             |
| 5.3. İnsan Faktörüne İlişkin Kısıtlamalar .....          | 177             |
| 5.3.1. Toplam Mesafe Kısıtı .....                        | 178             |
| 5.3.2. Yön Kısıtı .....                                  | 178             |
| 5.3.3. Maximum Sapma Kısıtı .....                        | 179             |
| 5.4. Önerilen Sezgisel Metod .....                       | 180             |
| 5.5. Elde Edilen Sonuçların Yorumlanması .....           | 183             |
| <b>SONUÇ.....</b>                                        | <b>191</b>      |
| <b>KAYNAKLAR .....</b>                                   | <b>195</b>      |
| Ek 1. Uygulama Problemine Ait Duraklar ve Talepler.....  | 206             |
| Ek 2. Duraklar-Arası Mesafeler.....                      | 209             |
| Ek 3. İstanbul İli Toplutaşım Haritası .....             | 254             |
| Ek 4. Probleme İlişkin Mevcut Rotalar .....              | 255             |
| Ek 5. Mevcut Rotalara İlişkin Harita.....                | 256             |
| Ek 6. İş Yeri Servis Araçlarının Rotalama Programı ..... | 257             |
| Ek 7. Çözüm Rotalarına İlişkin Harita .....              | 261             |
| <b>ÖZGEÇMİŞ .....</b>                                    | <b>262</b>      |

## GİRİŞ

Gerçek yaşam problemlerinin bir kısmı bilinen yöneylem araştırması-optimizasyon teknikleri ile çözülebilirken, önemli bir kısmı için optimum sonuçları elde etmek, gerek değişken sayısının ve gerekse işlem sayısının aşırı ölçüde fazla oluşu nedeni ile çok zor hatta imkansız olmaktadır. Optimum sonuçların bulunabilmesinin zor olduğu bu tip problemler için optimum yerine tatmin edici ölçüde iyi sonuçların belirlenmesi tercih edilmekte ve bu amaçla sezgisel yöntemler kullanılmaktadır.

Sezgisel çözüm metodları problemlerin optimum sonuçları ile bu sonuçları elde etmek için harcanması gereken zaman ve çaba arasında bir uzlaşma sağlayarak, hızlı ve iyi çözümler üretebilen stratejilerdir. Bu tanımdan da açık olduğu gibi sezgisel çözüm metodları ile elde edilen sonuçların en iyi olması gerekmekte bir diğer deyişle tatmin edici ölçüde iyi olan yaklaşık sonuçlar öneri niteliğinde olmaktadır.

Sezgisel metodları problemin analitik veya simülasyon prosedürleri ile çözülebilmesini engelleyen uzayı daraltarak problemi çözülebilir hale getirmek için insani öngörü, sezme ve öğrenme özelliklerini kullanmakta ve metodun her bir aşaması daha sonraki aşamalarına yol göstermektedir. Bu yönü ile sezgisel çözüm metodları dinamik bir yapıya sahiptirler.

Yukarıda söz edilmiş olan ve bilinen optimizasyon teknikleri ile çözülemeyecek kadar karmaşık olan problemlere iyi bir örnek Araç Rotalama Problemi (VPR) dir. Araç rotalama problemi en basit ifade ile, mal, hizmet veya şahısların bir araçlar filosu ile belirli arz noktalarından gene belirli talep veya teslimat noktalarına ulaştırılmaları amacı ile kapasite ve mesafe kısıtlarına uygun olarak en kısa yolların belirlenmesi problemi olarak tanımlanabilir. Bununla birlikte problemin pek çok versiyonları mevcuttur.

Ulaşım ve dağıtım faaliyetlerinde işletim etkinliğinin giderek daha fazla önem kazanması araç rotalama problemlerine olan ilgiyi arttırmıştır. Literatürde bu problemler için çözüm arama metodlarını ortaya koyan yüzlerce çalışma mevcuttur. Problemin çok çeşitli versiyonlarının varlığı ve bu tip problemler için kesin algoritmaların mevcut olmaması ve en önemlisi gerçek yaşamda bu tür problemlerin hem sayıca çok hem de önemli olmaları nedeni ile araştırma açısından ilginçlikleri sürmektedir.

Bu çalışmada sezgisel metodların araç rotalama problemlerinin çözümleri için nasıl kullanılabileceklerini açıklamak amacı ile problemin çeşitli versiyonları için geliştirilmiş olan sezgisel metodlar ortaya konulmakta ve problemin özel bir versiyonu için bir sezgisel çözüm metodu geliştirilmektedir.

Beş bölümden oluşan çalışmanın birinci bölümünde sezgisel araştırma tanımlanmakta, bazı özellikleri ve türleri açıklanmaktadır.

İkinci bölümde klasik araç rotalama problemi (VRP) tanımlanmakta, sınıflandırılmakta ve özellikleri açıklanmaktadır.

Üçüncü bölümde Araç Rotalama Probleminin iki temel versiyonundan birincisi olan ve çalışmamızın ana konusunu oluşturan düğüm rotalama problemi ele alınarak, seyyar satıcı problemi (TSP) etraflıca incelenmektedir. Bu başlık altında TSP'nin matematiksel formülasyonları ve bunların çözümleri için geliştirilmiş olan sezgisel çözüm stratejileri incelendikten sonra tek depo / çok araç rotalama problemi, çok depo / araç rotalama problemi, stokastik araç rotalama problemi ve zaman pencereli araç rotalama ve programlama problemi ayrı ayrı ele alınarak açıklanmakta ve bunların herbiri için matematiksel formülasyonlar ve sezgisel çözüm metodları sunulmaktadır.

Dördüncü bölümde kanal rotalama problemi (ARP), çinli postacı problemi (CPP), kırsal postacı problemi (RPP) ve kapasite kısıtlı kanal rotalama problemi (CARP) olmak üzere üç alt başlık altında ele alınmakta

ve bu problemlerin herbiri yönlü, yönsüz ve karışık kanallar durumlarına göre ayrı ayrı incelenmektedir.

Çalışmanın beşinci ve son bölümünde bir uygulama problemi olarak, T. Garanti Bankası Genel Müdürlüğünde (Güneşli-İstanbul) görev yapan personelin Avrupa yakasındaki ulaşım hizmetlerine dair rotalama problemi ele alınmakta ve geliştirilen sezgisel bir çözüm metodu ile çözüldükten sonra elde edilen sonuçlar yorumlanmaktadır.



## BİRİNCİ BÖLÜM

### SEZGİSEL ARAŞTIRMA

Sezgisel araştırma, yöneylem araştırması metodlarına alternatif bir yaklaşımın sağlandığı ve kişisel deneyimlere dayanarak daha ileri analiz için metodlar ve prosedürlerin geliştirildiği bir içsel araştırma sürecidir. Sezgisel problem çözme metodu, lineer programlamanın simplex metodu gibi mümkün olan en iyi çözümü elde etmeyi vaat eden bir çözüm metodu değil, özel bir problemin hızlı ve tatmin edici bir çözümünü üretebilecek bir metod veya metodlar arama stratejisidir.

Sezgisel problem çözme, Grek dilindeki "keşfetmek" veya "bulmak" anlamındaki "heuriskein" kelimesinden türetilen ve özel bir probleme bir veya daha fazla tatmin edici çözümün bulunmasında yardımcı olabilen ve sezgiseller olarak adlandırılan bir kurallar setinin geliştirilmesini içermektedir.<sup>1</sup> Daha önce de ifade edilmiş olduğu gibi burada optimallik garantisi bulunmamakta ve bunun yerine tatmin edicilik vurgulanmaktadır.

Pek çok uygulamada, sezgisellerin kullanılması, problem uzayını bir analitik veya simulasyon prosedürünün uygulanabilmesine imkan verecek kadar daraltabilmektedir. Bazı durumlarda, kullanılan sezgisellerin kendileri kabul edilebilir çözümler üretebilmektedirler. Bu ifadeden de anlaşılacağı gibi sezgisel araştırmanın kullanımı, bilinen yöntemlerle çözümleri hem zaman hem de çaba açısından çok zor ve hatta hemen hemen imkansız olan çok büyük yapıları problemlere dair uzayı çok küçülterek bu yöntemlerin kullanılabilmesine imkan sağlamaktadır.

---

<sup>1</sup> C. MOUSTAKAS, "Heuristic Search", Sage Publications inc, Newbury Park, California, 1990, s. 9.

Analitik prosedürler, matematiksel ispatlama ve bilinen özellikler tarafından desteklenen sonuç çıkarımına dayanmakta iken, sezgisel prosedürler yaratıcılık, uzak görüşlülük, sezgi ve öğrenme gibi insani karakteristiklere dair sonuç çıkarımına dayanmaktadırlar.

Bir probleme çözüm aramada özel bir sezgiselin izlenmesinin nedeni, sezgisel olarak veya deneyimlere dayanılarak kabul edilebilir bir çözüm üretmeyi vaat etmesidir. Ancak bu araştırma prosesi esnasında daha iyi bir kuralın keşfedilmesi durumunda eski kural iptal edilmektedir. Bu nedenle, sezgisel araştırma mevcut kabul edilmiş kuralların kullanılmalarının yanı sıra, bu kuralların yerine daha da iyi kuralların araştırılmasını da içermektedir.

Pek çok gerçek yaşam problemi, bilinen matematiksel teknikler ile direkt analitik çözüm bulma açısından uygun yapıda değildir. Böyle tekniklerin uygulanmaları teorik olarak mümkün olsa bile, eğer problem çok büyük ise bu pratik olmayabilir. Bu tip problemlere kötü yapı (ILL Structured, ISP) problemler denilmekte ve çözümlerinde sezgisel metodlar yardımcı olmaktadır.

### 1.1. Kötü Yapılı Problemler (ISP)

Çözülebilmeleri için uygun analitik veya algoritmik çözüm tekniklerinin mevcut olmadığı problemler, genel olarak "kötü yapı problemler" olarak adlandırılmaktadırlar. Bunun tersi, iyi yapı problemler (WSP) dir.

Bütün problemler aşağıda sıralanan genel tanımlayıcı karakteristiklere değişken dereceler ile sahiptirler. Herhangi bir problemin bu karakteristikleri taşıma dereceleri, iyi veya kötü yapı olma niteliğini belirlemektedir<sup>2</sup>

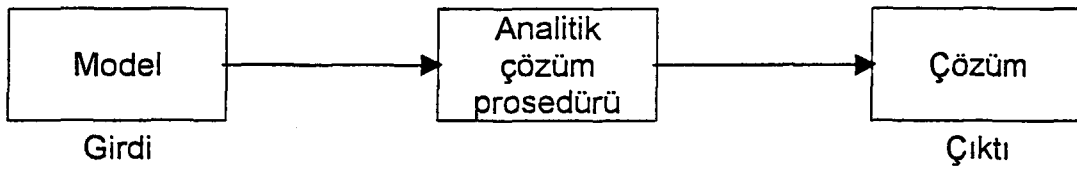
---

<sup>2</sup> H.G. DAELLENBACH, J. A. GEORGE," Introduction to Operations Research Techniques", Allyn an Bucon Inc., Boston, 1984, s. 512.

- (a) Probleme ait herhangi bir bilgi, uygun bir model içerisinde gösterilebilir.
  - (b) Kabul edilebilir bir model, bütün erişilebilir ve kabul edilebilir çözümleri kapsamalıdır.
  - (c) Herhangi bir çözümün geçerliliğine ve kabul edilebilirliğine karar vermek için belirli kriterler mevcuttur.
  - (d) Kabul edilebilir bir modeli çözmek için, programlanabilir bir metod mevcuttur.
  - (e) Çözüm prosedürü, sadece uygulanabilecek miktarda hesaplama içermelidir.
  - (f) Kabul edilebilir modelin ihtiyaç duyduğu bütün bilgi, elverişli veri toplama çabası sınırları içerisinde bulunmalıdır.
- 

ISP'ler ile WSP'ler arasındaki sınır belirsiz ve değişkendir. Ancak basit lineer optimizasyon problemleri gibi belirli problemlerin kesin olarak iyi oldukları söylenebilir. Buna karşılık satranç oyunları, mimari dizayn tipleri vb. çeşitli problemler ancak sezgisel yaklaşımlar kullanılarak çözülebilmektedirler.

Birçok yöneylem araştırması planı, standart bir çözüm metodunun uygulanmasını içermektedir. Yöneylem araştırmasının ana görevi, bilinen bir optimizasyon tekniği ile çözülebilecek uygun bir model kurmaktır. Bir anlamda çözüm metodu doğru girdi verildiğinde gerekli çıktıyı üretecek bir kara kutu olarak görülebilir.



Şekil 1.1. Standart bir metodun uygulanması durumu

Standart metodların aksine bir sezgisel yaklaşım, Şekil 1.2'de gösterildiği gibi bir süreç içerebilir.<sup>3</sup>

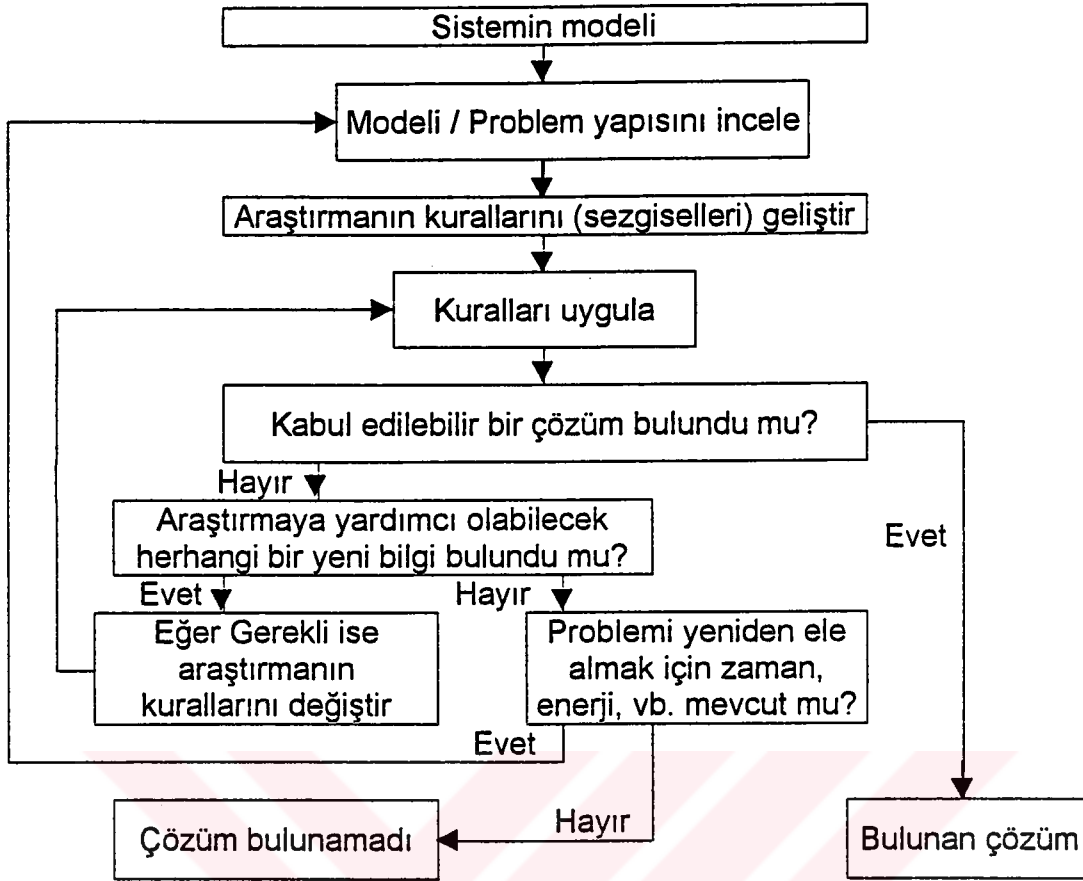
Bu sürecin çeşitli aşamalarında insani problem çözme yetenekleri uygulanmaktadır. İlk olarak, araştırmanın başlangıç kurallarını geliştirmek için, problem yapısının idrak edilmesi yani sezilmesi gerekmektedir. Bu sezgi, keşfetme, anlama, iç yüzünü görebilme ve öğrenme (tecrübe birikimi) ile elde edilebilmektedir. Kuralları uygulama sürecinde, problem yapısına dair ilave bilgi öğrenilebilmekte veya yeni sezgiler elde edilebilmektedir. Bu durum, kuralların daha rasyonel olarak uygulanmasına veya muhtemelen araştırma kurallarının tamamen yeniden tayinine imkan vermektedir.

Bir çözüm üretmede veya süreci yeniden yapılandırmada başarısız olan bir araştırma metodu bile faydalıdır. Çünkü bu başarısızlığın ardından ikinci bir denemede en azından bir olası araştırma yolu (başarısız olan yol) elimine edilebilmekte ve böylece bir çözüm metodunun araştırılması yönlendirilmektedir.

Sezgiseller, analitik tekniklerin kullanılmaları esnasında mevcut olmayan iki temel insani problem çözme davranış karakteristiğine sahiptirler. Bunlardan birincisi, deneme-yanılma yöntemi ile sonuç çıkarma iken ikincisi ise "ani keşfetme" özelliğidir. Ani keşfetme özelliğini açıklamak için Şekil 1.3'deki geometrik soruyu ele almak uygun olacaktır.

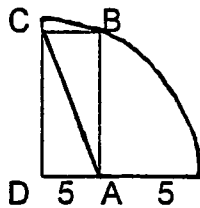
---

<sup>3</sup> Daellenbach, George, Age., S. 514.



Şekil 1.2. Olası bir sezgisel çözüm süreci

Şekil 1.3. de ABCD dörtgeninin bir köşesi D merkezli çeyrek dairenin çevresi üzerindedir. Verilen uzunluklar ile AC köşegeninin uzunluğunun bulunması istenmektedir. Bu problemi çözmek için bazı insanlar alternatif hesaplama yöntemlerini deneyeceklerdir. Halbuki diğer bazı insanlar, AC nin uzunluğunun DB köşegeninin uzunluğuna ve bunun da dairenin yarıçapı olan 10 birime eşit olduğunu derhal keşfedebileceklerdir.



Şekil 1.3. AC nin uzunluğu nedir?

Bu problemdeki gibi, basit çözüm metodunun mevcut olduğu durumlar, "ani keşif" ile karakterize edilmektedirler ve bu durumda deneme-yanılma davranışına gerek kalmamaktadır.

## 1.2. Sezgisel Metodların Amacı Tatmin Etme Özelliği

Sezgisel metodlar normal olarak optimal çözümleri garanti etmemektedirler. Ancak kötü yapıli problemlerin çoğu, gerçek anlamda optimal çözümlere sahip olmadıkları için bu bir dezavantaj değildir. Böyle problemler, zorunlu olarak bir veya daha fazla kabul edilebilir çözümün belirlenmesi ile çözülebilmektedirler. Esasen amacı tatmin etme, insani tercih davranışına amacı optimize etmeden daha iyi uymaktadır. İşletme organizasyonlarında, muhtemelen en yüksek yönetim düzeylerinde, amacı tatmin etme çok yaygındır ve zıt amaçlar arasında denge oluşturma zorunluluğu açıktır. Amacı tatmin etme tipindeki problemler, iki önemli özelliğe sahiptirler. Bunlardan birincisi, tatmin davranışının esnek yapısına dair bir özelliktir. Yani belirli bir zaman noktasında tatmin edici görülebilen bir şey, yeni aydınlatıcı bilgi elde edildiği zaman artık tatmin edici olmayabilir. Veya daha önce tatmin edici görülmeyen bir şey, yeni aydınlatıcı bilgi ile tatmin edici olabilmektedir. Örneğin bir yönetici belirli kâr, pazar payı ve sermaye artış düzeylerini tatmin edici bulabilir. Fakat eğer bütün bu amaçların aynı anda artırılabilceği fark edilirse, önceki düzeyler artık tatmin edici olmayacaktır. Aksine eğer bu amaç düzeylerine aynı anda ulaşılamayacağı fark edilirse hedef düzeylerinin bir veya birkaçı küçültülebilecek ve daha önce tatmin edici görülmeyen düzeyler artık tatmin edici olacaklardır.

Tatmin etme davranışının ikinci ve daha önemli bir özelliği, kabul edilebilir bir çözümün, ölçülebilir olmayan (en azından amaç ölçüsünün elverişli bir çaba miktarı ile elde edilememesi anlamında ölçülebilir olmayan) zayıf tanımlı bir vasıflar setini taşıması olasılığıdır. Örneğin gelişmekte olan bir ülkede, hükümetin küçük bir bölümünü üretebileceği bir projeler serisi mevcut olabilir. Buradaki proje seçimlerinin bazı vasıfları, kültürel organizasyon ve faaliyetlere olumsuz etki yapmamaları, kabul

edilebilir gelir dağılımı yapısına yol açmaları ve bölgesel gelişmeye önemli katkı yapma olasılıkları olabilir. Böyle durumlarda önerilen bir yaklaşım, karar vericiye her bir tercih karakteristiğini değişik derecelerde taşıyan bir çözümler seti sunmaktır. Karar verici bu durumda en fazla tatmin edici bulunan çözümü seçebilecektir. Bu yaklaşımı kullanan araştırmacı, çelişen amaçlar arasında denge fonksiyonları oluşturmak için keyfi varsayımlar yapmak zorunda kalmamakta ve büyük ölçüde sübjektif temelde tahmin edilmiş katsayılar içeren yapay kısıtlar oluşturma ihtiyacını da duymamaktadır.

### 1.3. Sezgisel Çözümün Uygunluğu

Bir sezgisel çözümün kabul edilebilmesi için uygun olması şarttır. Burada "uygunluk" terimi ile ifade edilen şey, elde edilen çözümün modelin yan kısıtları ile uyumlu olmasıdır. Bazı sezgisel yöntemler çözüm prosedürünün her aşamasında uygunluğun korunması zorunluluğu taşıırken, araç rotalama problemini çözmek için kullanılan bir sezgisel yöntem olan yasak araştırması gibi bazı algoritmalar ise ara çözümlerin uygun olmamalarına izin vermekte, ancak bunun için amaç fonksiyonuna bir ceza terimi eklemektedirler. Bir diğer yöntem olarak lagrangean gevşetme algoritması, yan kısıtları amaç fonksiyonuna taşıyarak, uygunluk problemini temelden halletme yolunu seçmektedir.

### 1.4. Sezgisel Yöntemin Gücü

Pek çok gerçek yaşam probleminin, bilinen optimizasyon teknikleri ile çözülemeyecek kadar büyük ve karmaşık yapılara sahip olduklarından ve bunların ancak sezgisel yöntemler ile çözülebileceklerinden daha önce söz edilmişti. Bu tipteki problemlere bir örnek olarak, bizim çalışmamızın odak noktasını oluşturan seyyar satıcı problemi (TSP) ele alınabilir. TSP, bir ana şehirden başlayıp her bir şehri kesin olarak ve sadece birer kere ziyaret etmesi ve ana şehre geri dönmesi gereken bir satıcının minimum maliyetli ziyaret sırasını belirleme problemidir. Ziyaret edilecek şehir sayısının az olması durumunda bütün olası turları tek tek ele almak

mümkündür. Ancak şehir sayısı arttıkça durum karmaşıklaşmaktadır. Örneğin,  $n=25$  şehir söz konusu olduğunda  $n!=1.5 \times 10^{25}$  adet farklı tur mevcut olmaktadır. Bir bilgisayarda saniyede  $10^{16}$  turun değerlendirilebileceği varsayımı ile, bu problem için bütün turların değerlendirilmesi yaklaşık 50 yılımızı alacaktır. Bu gibi problemler matematikte "NP-hard" olarak adlandırılmaktadırlar. Buradaki "NP", polinomik olmamayı ifade etmektedir. Yani böyle problemleri çözmek için gerekli olan zaman ve işlem sayısı, nesne sayısının veya daha genel olarak problem hacminin üstel bir fonksiyonu ile artmaktadır. Bir başka deyişle, bilinen optimizasyon algoritmaları, bu problemleri problem hacmine göre polinomik bir zaman içinde çözememektedirler.

NP-hard problemlerin çözümlerinde izlenen yol ya bir sezgisel kullanarak tam ölçekli bir modeli yaklaşık olarak çözmek ya da bir yaklaşık modeli optimal alarak çözmektir.

Her iki durumda da çözümün optimal olması garanti edilmemekle birlikte bir performans ölçüsü ile optimale yakınlık derecesi belirlenmektedir. Bu performans ölçüsü, "en kötü durum hata performansı" olarak adlandırılmakta ve en kötü durumdaki sezgisel çözüm değerinin optimal çözüm değerine oranının üst sınırı olarak belirlenmektedir. En kötü durum hata performansı, ya matematiksel ispat yöntemleri ile ya da bilinen optimizasyon metodları ile çözülebilecek büyüklükteki pek çok probleme sezgisel metodun uygulanması ve sezgisel çözüm değeri ile optimal çözüm değerinin karşılaştırılması ile belirlenmektedir.

Bir optimal çözüm değerinin  $\alpha$  katından daha büyük olmadığı garanti edilen sezgisel çözüm değeri  $\alpha$ -optimal olarak adlandırılmaktadır.

Bir optimal çözüm değeri ile bir sezgisel çözüm değeri arasındaki nispi hatanın, problem hacmi arttıkça sıfıra yaklaşması ise bu sezgiselin asimptotik olarak optimal olduğu anlamına gelmektedir.

### 1.5. Sezgisel Algoritmalar ve Programlar

Bu bölümde kullanılan "sezgisel program terimi", bir sezgisel yöntemin bilgisayar uygulaması anlamında kullanılmaktadır.

Sezgisel problem çözme tekniklerinin çoğu, aşağıda açıklanmakta olan dört genel stratejiden birine veya bunlardan iki veya daha fazlasının bir karışımına aittir. Söz konusu stratejiler en genel anlamda şu şekilde sınıflanabilmektedirler.

- (a) Çözüm-inşa stratejileri
- (b) Yap-boz stratejileri
- (c) Çözüm-iyileştirme stratejileri
- (d) Araştırma-öğrenme stratejileri

#### 1.5.1. Çözüm-İnşa Stratejileri

Çözüm-inşa stratejilerinde, belirli bir kurallar setine göre, her seferinde bir eleman olmak üzere bir çözüm inşa edilmeye çalışılmaktadır. Örneğin özel bir kritere uygun olacak şekilde  $n$  adet nesnenin her birini  $m$  adet gruptan birine tahsis etme probleminde ( $n > m$ ), ilk olarak nesneler bir öncelik ölçütüne göre sıralanarak işe başlanılabilir. Bu öncelik ölçütü, çözümün kabul edilebilir olup olmadığına karar vermek için kullanılacak kriteri yansıtmaktadır. İkinci adım, bu öncelik sırasına göre nesneleri gruplara tahsis etmektir. Bu stratejinin bir diğer varyasyonu ise, her bir tahsis işleminden sonra kalan tahsis edilmemiş nesnelerin öncelik sıralamasını yeniden değerlendirmeye dayanmaktadır.

#### 1.5.2. Yap-Boz Stratejileri

Yap-boz stratejileri, karmaşık bir problemin bir çok küçük, daha basit alt probleme bölünmesine ve her bir alt problemin bireysel olarak veya hiyerarşik bir sıra içerisinde çözülmesine dayanmaktadırlar. Hiyerarşik sıralı çözümde, en uygun çözüm sırasına göre herhangi bir alt problemin çıktısı daha sonraki bir problem için girdi olarak kullanılmaktadır.

Alt problemlerin çözümleri uygun şekilde birleştirilerek, bütün problemin çözümü elde edilmektedir.

### 1.5.3. Çözüm-İyileştirme Stratejileri

Bir çözüm-inşa stratejisi veya bir yap-boz stratejisi kullanılarak elde edilmiş olan bir başlangıç çözümünün kabul edilebilirlik derecesini artırmak amacı ile bu çözüme özel bir sezgiseller serisi uygulanabilir. Başlangıç çözümü, problemin basitleştirilmiş bir yaklaşık modeline bir standart optimizasyon tekniği uygulanarak da elde edilebilir. Bu durumda, elde edilmiş olan çözümün orijinal problemin çözümüne dönüştürülmesi amacı ile sezgisel iyileştirme kuralları uygulanır.

### 1.5.4. Araştırma-Öğrenme Stratejileri

Bu yaklaşım, çözüm uzayının yönlendirilmiş bir incelemesini içermektedir. Araştırma esnasında yeni bilgilerin kazanılmadığı yönler iptal edilerek, araştırma uzayı küçültülmektedir. Bir başka deyişle, her bir denemenin sonucu, araştırmacıya çözüm uzayının bir veya daha fazla kısmını elimine etme ve çözümü bulmak için daha fazla bilgi sağlayacak denemeleri formüle etmede yol göstermek için kullanılmaktadır.

Özel bir sezgisel program, yukarıdaki stratejilerden iki veya daha fazlasının bir kombinasyonunu içerebilir. Örneğin, çözüm-inşa ve yap-boz stratejileri daha sonra çözüm iyileştirme ve araştırma-öğrenme stratejilerinin bir karışımı ile iyileştirilebilecek olan bir başlangıç çözümü sağlayabilirler.

Bir çözüm iyileştirme veya araştırma-öğrenme stratejisi kullanılacağı zaman, prosedürün başarılı olarak sonuca ulaşacağı veya başarısız olarak ifade edileceği bütün koşullar dikkatli olarak açıklanmalıdır. Örneğin, eğer prosedür uygun bir çözüm bulmayı içeriyorsa, bir kere böyle bir çözüm belirlenir belirlenmez sona erebilir veya k adet deneme veya m dakika içinde uygun bir çözüm oluşmamış ise terk edilebilir. Bazen araştırmacıya prosedürü araştırmanın belirli noktalarında

sona erdirmeye veya müdahale etme imkanı veren bir interaktif program da arzu edilebilir.

Birçok sezgisel program, prosedürün bir veya birkaç aşamasında optimizasyon metodları içerebilir. Örneğin, yap-boz stratejisinde oluşturulan alt problemlerin bazıları lineer programlama gibi bir optimizasyon tekniği ile çözülebilirler. Ayrıca simülasyon metodları da bir sezgisel programın bir parçasını oluşturabilirler. Örneğin, sezgisel bir yöntem ile türetilen bir çözümün uygunluğunu test etmek için bir simülasyon metodu kullanılabilir. Alternatif olarak, karmaşık karar problemlerine iyi çözümler araştırmayı hızlandırmak için sezgisel kurallar, simülasyon ile birlikte kullanılabilirler.

Optimizasyon metodları, modelin genel varsayımlarına uyan herhangi bir probleme uygulanabilmelerine karşın sezgisel programlar, simülasyon modelleri gibi büyük ölçüde probleme özgü olma eğilimindedirler. Bununla birlikte araştırmacıların parametrik programlar geliştirmekten kaçınmaları gerekmektedir. Bir sürecin uygun çözümler bulma yeteneği, özel veya sınırlı bir aralıktaki parametre değerlerine bağlı olmamalıdır. Çünkü bu parametreler örneğin zamanla değiştikleri zaman prosedür artık faydasız olmaktadır. Ayrıca çözümün bu parametrelerdeki değişimler karşısında hassaslığını test edebilme niteliği, sezgisel programlar için çok önemli bir zaman kaybı olmaktadır.

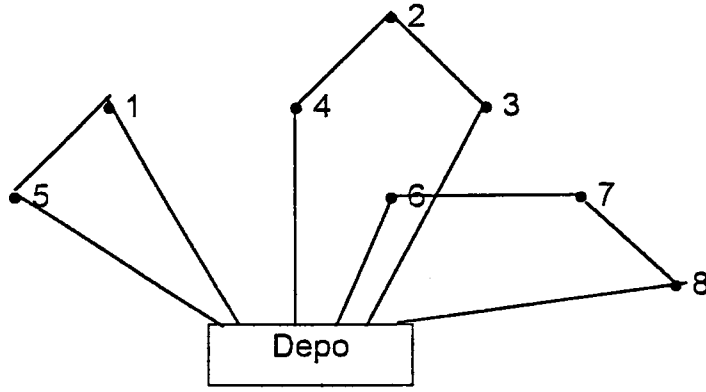
## İKİNCİ BÖLÜM

### ARAÇ ROTALAMA PROBLEMİ (VRP)

Araç rotalama problemi (VRP), en basit şekli ile bir merkezi depoda üslenmiş bulunan ve her biri aynı kapasiteye sahip olan homojen araçlar filosunun, her biri farklı bir yerleşime ve bilinen (deterministik) ve bölünemeyen bir talebe sahip olan bir müşteriler setine, toplam seyahat mesafesini veya süresini minimize edecek şekilde hizmet sunarak depoya geri dönmesi için gerekli rotaların ve her bir rotadaki yolların belirlenmesi problemidir. Bu problem, "standart VRP" olarak adlandırılmaktadır. Şekil 2.1 de bir örnek standart VRP'nin çözülmesi ile elde edilmiş olan rotalar gösterilmektedir.

Araç rotalama problemlerinin karakteristiklerindeki ve yan kısıtlarındaki farklılıklar problemin niteliğini ve çözüm yollarını değiştirerek pek çok varyantlarını ortaya çıkartmaktadır. Problemin söz konusu varyantlarına Bölüm 2.2 'de kısaca değinilecek olmakla birlikte, bizim çalışmamızda araç rotalama problemi iki ana gruba ayrılarak incelenmektedir. Bu gruplardan birincisi düğüm rotalama problemi, ikincisi ise kanal rotalama problemidir.

Bir müşteriler şebekesindeki her bir müşteriye ziyaret ederek depoya geri dönme probleminde, talepler sadece şebekenin düğümlerinde (müşteri yerleşimlerinde) ortaya çıktıkları için bu durumda, problem "düğüm rotalama problemi" olarak adlandırılmaktadır.



Şekil 2.1. Standart VRP rotaları

Bir şebekedeki müşteri sayısının çok fazla olması ve bu müşteriler arasındaki mesafelerin birbirlerine çok yakın olmaları durumunda kanal rotalama problemi ortaya çıkmakta ve bu problem, şebekedeki her bir kanalı katetme ve başlangıç noktasına geri dönme problemi olarak açıklanabilmektedir.

Düğüm rotalama problemi Bölüm 3'de, kanal rotalama problemi ise Bölüm 4'de ayrıntılı olarak ele alınacakları için burada sadece çok basit tanımlarının verilmesi yeterli görülmektedir.

## 2.1. Araç Rotalama Problemlerinin Yan Kısıtları

Araç rotalama problemleri, çok sayıda yan kısıta sahip olabilirler. Burada bunlardan en önemli olan ve en fazla karşılaşılan beş yan kısıttan söz edilmektedir. Aşağıda özetlenen yan kısıtlardan bir veya bir kaçının varlığı problemin çözüm yöntemini etkilemekte, hatta bazen tamamen değiştirebilmektedir.

Yan kısıtlar;

- (1) Kapasite kısıtları: Her bir müşteriye dair bir pozitif ağırlık (veya talep) vardır. Herhangi bir aracın rotası üzerindeki ağırlıkların toplamı aracın kapasitesini aşamaz (kapasite kısıtlı VRP durumu).

- (2) Herhangi bir rota üzerindeki müşteri sayısı bir  $q$  sayısını aşamaz. (Bunun özel bir durumu,  $i > 1$  için  $i$  nin talebi  $d_i = 1$  olmak üzere, araç kapasitesi  $D=q$  olması halidir).
- (3) Toplam zaman kısıtı: Herhangi bir rotanın uzunluğu, belirli bir  $L$  sınırını aşamaz. Bir rotanın uzunluğu, rota üzerindeki müşteriler arasındaki seyahat süreleri ( $c_{ij}$ ) ve her bir müşterideki duraklama süreleri ( $t_i$ ) toplamından oluşmaktadır.
- (4) Zaman pencereleri: Müşteri  $i$ ,  $[a_i, b_i]$  zaman aralığı içerisinde ziyaret edilmelidir.
- (5) Şehirler arasındaki öncelik ilişkileri: Müşteri  $i$ , müşteri  $j$ 'den önce ziyaret edilmelidir.

## 2.2. Araç Rotalama Probleminin Çözümünü Zorlaştıran Özellikler

Bölüm 2'nin başındaki tanımlamamıza göre araç rotalama problemi basit bir problem gibi görünmesine karşın, gerçek yaşamda karşımıza çıkan pek çok rotalama problemi, sanıldığından çok daha karmaşıktır.

Schrage (1981), problemin çözümünü zorlaştıran 10 özelliği sıralamıştır.<sup>4</sup>

- (1) İsteklerin frekansı : Müşterilerin hizmet talepleri belirli bir frekans ile ortaya çıkabileceği gibi, bu frekans müşteriden müşteriye değişebilir. Örneğin, ekmek sipariş eden müşterilere teslimatlar günlük olarak yapılmaktadır. Yani ekmek talebinin frekansı belirlidir ve 1 gündür. Bunun aksine evsel kullanım için yakıt veya tüpgaz teslimatında ise talep frekansı kullanım oranına bağlıdır ve müşteriden müşteriye çok değişebilmektedir.

---

<sup>4</sup> L. SCHRAGE, "Formulation and Structure of More Complex/Realistic Routing and Scheduling Problems", Networks 11, 1981, s. 229-232.

- (2) Zaman Pencereleeri : Zaman pencerelerinin varlığı, müşteri yerleşimlerinin spesifik zamanlarda ziyaret edilmeleri zorunluluğunu ifade etmektedir. Öğrenciler için okul servis araçlarının rotalanması buna iyi bir örnektir.
- (3) Günün hangi zamanında bulunulduğuna bağlı olarak değişen seyahat süresi: Ziyaret edilecek müşterilerin şehir merkezlerinde bulunmaları durumunda trafik sıkışıklığı önemli bir faktör olabilmektedir. Ayrıca işe başlama ve işten çıkış saatlerinde seyahat süresi değişebilmektedir. Bu gibi durumlarda, seyahat süresi gün içindeki zamanlara göre değişmekte ve bunun sonucu olarak bir müşteriler şebekesindeki her bir hatta ait maliyet de gün içindeki zamana bağlı olmaktadır.
- (4) Çok boyutlu kapasite kısıtları : Aynı anda hem hacim hem de ağırlık üzerinde kısıtlar olabilmektedir. Bu durum, özellikle farklı ürünleri taşımak için aynı araçlar kullanıldığı zaman bir sıkıntı yaratabilmektedir.
- (5) Farklı tipte araçların varlığı : Büyük firmalar, pek çok alternatif tipte araçlara sahip olabilirler. Araç tipleri, kapasite, işletim maliyeti ve aracın kapalı turlarla kısıtlanmış (yani aracın işini bitirdikten sonra depoya geri dönmek zorunda) olup olmamasına göre farklılık gösterebilirler. Bir çok farklı tipte aracın varlığı uygun seçeneklerin sayısını büyük ölçüde artırmaktadır.
- (6) Bölünebilir taleplerin varlığı : Eğer bir müşteri özel bir talebe sahip ise, bu müşteriye birden fazla aracın tahsis edilmesi akılcı olabilir.
- (7) Belirsizlik : Araç rotalama algoritmaları hemen hemen her zaman gerekli bilgilerin önceden bilindiğini varsaymaktadırlar. Aslında bir müşteriler şebekesinin belirli kısımlarını katetmek için gerekli olan süre, trafik ve hava koşulları ve araç arızaları gibi faktörlere bağlı olarak büyük ölçüde değişebilmektedir.

### 2.3. Araç Rotalama Problemlerinin Sınıflandırılması

Bütün rotalama sistemlerinin temel çıktısı, zorunlu olarak aynıdır. Yani her bir araç veya sürücü için bir rota elde edilmektedir. Rota genel olarak, müşteri yerleşimlerinin ziyaret sırasını belirlemektedir.

Çeşitli rotalama sistemleri aynı tür çıktıya sahip oldukları için bu problemler, sahip oldukları karakteristiklere ve kendilerini çevreleyen varsayımlara dayanılarak birbirlerinden ayırt edilebilmektedirler. Örneğin, bir depoda üslenen bir araçlar filosu ile çeşitli yerleşimlere mal teslim etme probleminde, teslimat zamanları üzerinde önsel kısıtların mevcut olmaması ve bütün malların kısa bir zaman periyodu içerisinde teslim edilebilmeleri halinde zamansal olgular ihmal edilebilmekte ve bunun sonucu olarak saf araç rotalama problemi ortaya çıkmaktadır. Bu duruma iyi bir örnek olarak evlere şişe suyu teslimatı verilebilir. Genelde çeşitli hizmet faaliyetlerine ilişkin karakteristikler ve kısıtlamalar farklı model varsayımlarına ve farklı problem kategorilerine yol açmaktadır.

Tablo 2.1, Bodin ve Golden (1981) tarafından hazırlanan ve rotalama problemlerinin farklılık gösterebildikleri karakteristiklerin ayrıntılı bir listesini sergilemektedir.<sup>5</sup> Tablodaki karakteristikler, ele alınan bir rotalama probleminin en çabuk ve en iyi şekilde tanınmasını sağlamaktadırlar. Çeşitli karakteristiklerin farklı kombinasyonları, çok fazla sayıda olası problem oluşumuna yol açmaktadır. Bir tek depo, sınırsız kapasiteli bir tek aracın bulunduğu, seyahat süresi veya mesafesinin seyahat yönünden bağımsızlığı şeklinde tanımlanan bir yönsüz şebekenin düğümlerinde ortaya çıkan belirli taleplerin mevcut olduğu, rota süresi üzerinde herhangi bir kısıtın bulunmadığı ve amacın toplam seyahat mesafesine göre rotalama maliyetlerini minimize etmek olduğu bir problem, daha sonra ayrıntılı olarak incelenecek olan ve çok bilinen seyyar satıcı problemi (TSP)'dir. Bu problemdeki diğer bütün karakteristiklerin

---

<sup>5</sup> L. BODIN, B. GOLDEN, "Classification in vehicle routing and scheduling", Networks 11(2), 1981, s. 97-108.

aynı kalması koşulu ile homojen ve sınırlı kapasiteye sahip olan bir araçlar filosunun söz konusu olması halinde ise standart VRP ortaya çıkmaktadır.

Tablo 2.1'deki karakteristiklerin farklı kombinasyonlarına göre çok fazla sayıda ve farklı özelliklere sahip rotalama problemlerinin ortaya çıkabileceği açıktır. Bu problemlerin her birini tek tek ele alarak incelemek çok zor olacağı için bizim çalışmamızda önce müşteri taleplerinin yerleşimlerine göre ikili bir ayrım yapılmakta ve sonra bunlardan düğüm rotalama problemi için depo sayısına göre gene ikili bir ayrım yapılarak tek depo ve çok depo durumları ayrı ayrı incelenmektedir.

Tek depo durumunda araç sayısına göre iki ayrı problem ortaya çıkmaktadır. Bunlardan birincisi TSP ve ikincisi de standart VRP'dir.

Tablo 2.1. Rotalama Problemlerinin Karakteristikleri

| Karakteristikler              | Olası Seçenekler                                                                                                                                                                                                                                                                             |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) Mevcut filonun hacmi      | - Tek araç<br>- Çok araç                                                                                                                                                                                                                                                                     |
| (2) Mevcut filonun tipi       | - Homojen (tek tip)<br>- Heterojen (çok tip)<br>- Özel araç tipleri                                                                                                                                                                                                                          |
| (3) Araçların ana üsleri      | - Tek depo<br>- Çok depo                                                                                                                                                                                                                                                                     |
| (4) Taleplerin yapısı         | - Deterministik talepler<br>- Stokastik talepler<br>- Talebin kısmen karşılanması                                                                                                                                                                                                            |
| (5) Talep yerleşimleri        | - Düğümlerde<br>- Kanallarda<br>- Karışık                                                                                                                                                                                                                                                    |
| (6) Temel şebeke              | - Yönsüz<br>- Yönlü<br>- Karışık                                                                                                                                                                                                                                                             |
| (7) Araç kapasitesi kısıtları | - Var (hepsi aynı)<br>- Var (farklı kapasiteler)<br>- Yok (sınırsız kapasite)                                                                                                                                                                                                                |
| (8) Maximum rota süreleri     | - Var (bütün rotalar için aynı)<br>- Var (farklı rotalar için farklı)<br>- Yok                                                                                                                                                                                                               |
| (9) Operasyonlar              | - Sadece toplama<br>- Sadece dağıtma<br>- Karışık                                                                                                                                                                                                                                            |
| (10) Maliyetler               | - Değişken veya rotalama maliyetleri<br>- Sabit işletim maliyetleri<br>- Ortak taşıyıcı maliyetleri (hizmet verilmeyenler için)                                                                                                                                                              |
| (11) Amaçlar                  | - Toplam rotalama maliyetlerini minimize etme<br>- Sabit ve değişken maliyetler toplamını minimize etme<br>- Gerekli araç sayısını minimize etme.<br>- Hizmete veya rahatlığa dayanan fayda fonksiyonunu maximize etme.<br>- Müşteri önceliklerine dayanan fayda fonksiyonunu maximize etme. |

Kanal rotalama problemi, literatürde düğüm rotalama probleminden çok daha az ele alınmış olduğu ve çok daha dar kapsamlı olduğu için, bu konudaki ayrım daha genel bir yapıdadır. Söz konusu ayrıma göre kanal rotalama problemi bu çalışmada, bütün kanalların ziyaret edildiği Chinese Postman Problemi (CPP) ve bazı kanalların ziyaret edildiği Runal Postman Problemi (RPP) olmak üzere iki alt bölümde ele alınmakta ve önemliliği nedeni ile kapasite kısıtlı kanal rotalama problemi de ilave bir alt bölüm oluşturmaktadır.

#### 2.4. Araç Rotalama Problemi İçin Genel Çözüm Stratejileri

Araç rotalama problemleri, NP-hard problemler sınıfına girmektedirler.<sup>6</sup> Yani bu problemleri optimal olarak çözmek için bilinen yaklaşımların hesaplama süreleri ve çabaları, problem hacmine göre üstel olarak artmakta ve bu durum optimal çözümün elde edilmesini zorlaştırmakta hatta genellikle imkansızlaştırmaktadır. Bu nedenle, bütün NP-hard problemler için olduğu gibi araç rotalama problemi için de, optimal çözüm arama yerine optimele yakın çözümler elde etmeye yarayan birçok farklı sezgisel prosedür geliştirilmiştir ve halen de geliştirilmeye devam edilmektedir.

Bizim çalışmamızın da odak noktası olan sezgisel algoritmalar, uygun ve optimele yakın çözümler elde etmek için, problemin yapısını matematiksel ve genellikle sezgisel olarak işleyen prosedürlerdir. Bir sezgisel metodun elde etmiş olduğu çözümler optimal çözüme yeterince yakın oldukları zaman bu metodun etkin olduğu söylenmektedir. Bir çok durumda sezgisel çözümün optimal çözümünden sapması üzerinde sınırlar (en kötü durum sınırları) elde etmek mümkündür. Bölüm 3'de TSP sezgiselleri için böyle sınırlar sunulmaktadır. Bilinen en kötü durum hata sınırlarına sahip olan etkin algoritmaların dizaynı, önemli bir araştırma konusudur. Bu konuda ayrıntılı bilgiler, Fisher (1980), Graham (1979).

---

<sup>6</sup> C.H. PAPADIMITRIOU, K. STEIGLITZ, "Combinatorial Optimisation", Prentice-Hall, 1982, S. 371.

Klee (1980), Sahni (1977), Silver (1980), Lenstra ve Rinnooy Kan (1979) ve Ball ve Magazine (1981) de mevcuttur.

Araç rotalama problemleri için genel çözüm stratejileri bir çok farklı açıdan farklı şekillerde sınıflanabilmektedirler. Aşağıda VRP çözüm metodları en basit tarzda 7 ana grupta sınıflanmaktadır.

- (1) Önce küme-sonra rota yaklaşımı : Önce talep düğümleri kümelere bölünerek, her bir kümeye ayrı bir araç tahsis edilmekte ve daha sonra bir TSP prosedürü kullanılarak bu kümeler rotalanmakta yani her bir küme için minimum maliyetli bir tur belirlenmektedir.
- (2) Önce rota-sonra küme yaklaşımı : Bütün talep düğümlerini kapsayan büyük ve bir tek TSP turu oluşturulmakta ve sonra bu tur, yan kısıtlara uygun yani feasible olan rotalara bölünmektedir.
- (3) Tasarruf/giriş metodları : Önce bütün düğümler için, uygun olması zorunlu olmayan bir başlangıç turu oluşturulmaktadır. Daha sonraki her bir adımda bu turun maliyeti, alternatif turların maliyetleri ile karşılaştırılmakta ve amaç fonksiyonunda en büyük tasarrufu sağlayan alternatif seçilmektedir. Prosedür, uygun bir tur ile sona ermektedir.
- (4) İyileştirme/değişim metodları : Bir iyileştirme mekanizması vasıtası ile bir başlangıç rotası üzerinde çalışmakta ve kanalları değiştirerek rotayı iyileştirmeyi denemektedir. Bu metodlar, 2-opt, 3-opt, k-opt ve or-opt prosedürlerini içermektedirler.

## ÜÇÜNCÜ BÖLÜM

### DÜĞÜM ROTALAMA PROBLEMİ

Genel rotalama problemi, bir araçlar filosu ile bir düğümler veya kanallar setine hizmet sunabilmek amacı ile, her bir araç için bir adet olmak üzere en düşük maliyetli ve uygun bir rotalar setini oluşturma problemidir.

Tanımdan da anlaşılabilceği gibi rotalama problemi, düğümler setine mi yoksa kanallar setine mi hizmet verileceğine göre iki ana gruba ayrılmaktadır. Bunlardan birincisi "düğüm rotalama" ve ikincisi ise "kanal rotalama" problemi olarak adlandırılmaktadır. Kanal rotalama problemi, Bölüm 4 de ayrıntılı olarak incelenecektir. Düğüm rotalama problemi ve varyantları ise bu bölümde ele alınmaktadır.

$G=[N, A, C]$ , düğümler seti ( $N$ ), kanallar seti ( $A$ ) ve maliyet matrisi ( $C$ ) ile ifade edilen bir şebeke diyagramdır. Bu şebeke üzerindeki düğümler seti ( $N$ ), her biri deterministik veya stokastik bir hizmet talebine sahip olan ve kesinlikle ve yalnız bir kere ziyaret edilmesi gereken müşterileri temsil etmektedir. Kanallar seti ( $A$ ), bu müşterileri birbirlerine bağlayan yolları ve maliyet matrisi ( $C=[c_{ij}]$ ) ise  $i$  müşterisinden ( $i$  düğümünden)  $j$  müşterisine ( $j$  düğüme) seyahat etmenin yani  $i$ - $j$  kanalını katetmenin maliyetini göstermektedir. Bu şebeke üzerindeki hizmet talepleri sadece düğümlerde ortaya çıkmakta ve şebekenin bütün düğümlerini belirli bir maliyet ölçütüne göre en az maliyet ile ziyaret etmek için gerekli rotaların bulunması, "düğüm rotalama problemi" olarak ifade edilmektedir.

Bölüm 3'de, düğüm rotalama probleminin üç temel varyantı olan tek depo-tek araç (TSP), tek depo-çok araç (standart VRP) ve çok depo-çok araç problemleri ve bunlara ait sezgisel çözüm stratejileri açıklanmaktadır.

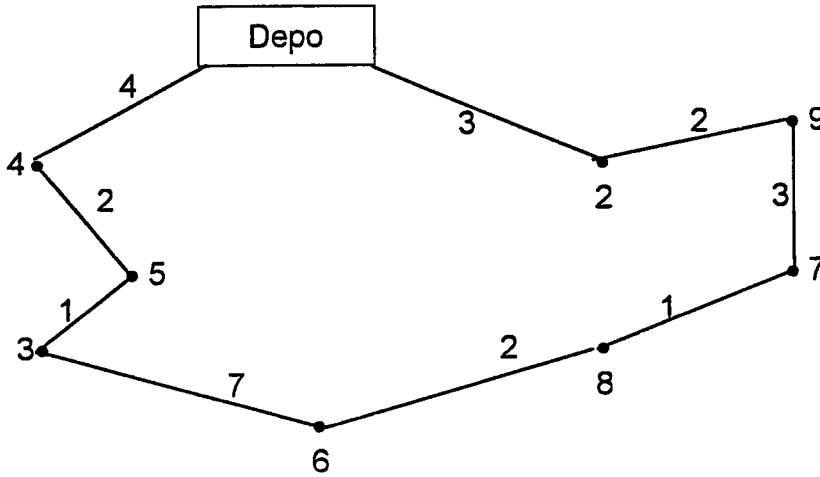
### 3.1. Seyyar Satıcı Problemi (TSP)

Düğüm rotalama problemlerinin temel varyantı olan seyyar satıcı problemi (traveling salesman problem, TSP), bir tek satıcı veya araç için, ana şehirde (tek depoda) başlayıp, bütün müşterileri kesinlikle ve sadece birer kere ziyaret ettikten sonra başlangıç noktasına geri dönen minimum maliyetli bir rota bulma problemidir. Bu tanımdaki maliyet terimi, her bir müşteri çifti arasındaki seyahat süresini veya mesafesini ifade etmekte ve  $C=[c_{ij}]$  matrisi ile gösterilmektedir.  $c_{ij}$ ,  $i$  müşterisinden  $j$  müşterisine seyahatin maliyetidir. Bu maliyetin seyahat yönünden bağımsız yani  $c_{ij} = c_{ji}$  olması durumunda "yönsüz" veya "simetrik" TSP olarak adlandırılan problem ortaya çıkmaktadır. Maliyetin seyahat yönüne bağlı olarak değiştiği yani  $c_{ij} \neq c_{ji}$  olduğu problem ise "yönlü veya "asimetrik" TSP'dir. Şekil 3.1'de, bir TSP tur örneği gösterilmektedir. Bu örnekte, kanallar üzerindeki rakamlar, sözkonusu kanalları katetmenin maliyetleridir ve bu maliyetler simetriktir. Ayrıca üçgen eşitsizliği geçerlidir. Yani,

$$c_{ki} \leq c_{kj} + c_{ji} \quad (3.1)$$

TSP, araç rotalama problemi (VRP) ile yakın ilişkili bir problemdir. Bir rota üzerinde talep edilebilecek hizmet miktarını sınırlandıran kapasite kısıtlarının varlığı, birden fazla satıcının söz konusu olduğu katlı TSP yi (MTSP) araç rotalama problemine (VRP) dönüştürmektedir. Kısıtların gevşetilmesi halinde, yani araçların sınırsız kapasitelere sahip olmaları durumunda ise VRP, MTSP ye indirgenmektedir.

VRP için pek çok sezgisel çözüm prosedürü, herhangi bir aşamasında TSP'leri kullanmaktadır. Örneğin, önce rota-sonra küme yaklaşımları, önce bütün düğümler üzerinde büyük bir TSP turu oluşturmakta ve sonra bu turu uygunluk koşullarını gözeten rotalara bölmektedirler.



Şekil 3.1. TSP turu

Önce küme-sonra rota yaklaşımları ise, önce kümeleme problemini çözerek müşterileri kümelere tahsis etmekte ve sonra TSP sezgisellerini kullanarak her bir küme için bir rota oluşturmaktadırlar.

Bir  $G=[N, A, C]$  şebekesi üzerinde ifade edilen bir TSP,  $G$  üzerinde minimum toplam maliyetli bir Hamilton dairesinin belirlenmesini gerektirmektedir. Bir Hamilton dairesi, her bir  $i \in N$  düğümünden kesinlikle bir kere geçen bir dairedir. TSP yapısal olarak NP-hard problemler sınıfına aittir.<sup>7</sup> Yani TSP için polynomial olarak sınırlı bir algoritma mevcut değildir ve bu nedenle pek çok sezgisel algoritma geliştirilmiştir. Daha önce de ifade edilmiş olduğu gibi sezgisel çözüm algoritmaları, optimal değil fakat  $\alpha$ -optimal olan TSP turları oluşturmaktadırlar. Bir  $\alpha$ -optimal TSP turu, optimal tur maliyetinin  $\alpha$  katından daha yüksek maliyetli olmadığı garanti edilen bir turdur. Bu tanımdan da anlaşıldığı gibi optimal TSP turu için  $\alpha=1$  olmaktadır. Aşağıda Bölüm 3.2 de TSP'nin matematiksel formülasyonları ortaya konulmakta ve Bölüm 3.3'de ise TSP için seçilen çeşitli sezgisel çözüm metodları açıklanmaktadır.

<sup>7</sup> C.H. PAPADIMITRIOU, "The Euclidean travelling salesman problem is NP-Complete", Theory of Computer Science 4(3), 1977, s. 237-244.

### 3.1.1. TSP'nin Matematiksel Formülasyonları

TSP'nin matematiksel formülasyonları, problemin karmaşıklığını ve diğer rotalama ve dağıtım problemleri ile ilişkisini ortaya koymak açısından faydalı olacaktır. Ayrıca bu formülasyonlar, algoritmik stratejilere ve sezgisel çözüm prosedürlerine yol göstermektedirler.

Burada, basitlik amacı ile maliyetlerin simetrik oldukları ( $c_{ij}=c_{ji}$ ) varsayılmaktadır. Problem, bütün düğümler üzerinde, orijinden (düğüm 1) başlayarak aynı yerde sona eren ve her bir düğümden kesinlikle ve yalnız bir kere geçen, minimum toplam uzunluğa veya süreye veya genel olarak minimum maliyete sahip olan bir tur oluşturmaktır. Problemi formüle etmek için,

$$x_{ij} = \begin{cases} 1, & \text{eğer tur üzerinde } i-j \text{ kanalı var ise} \\ 0, & \text{aksi halde} \end{cases}$$

şeklinde bir 0-1 değişkeni tanımlanmaktadır.

TSP'nin tahsis-esaslı bir formülasyonu, kesin olarak her bir düğüm  $i$  den bir  $(i, j)$  kanalı çıkacak ve her bir düğüm  $j$  ye bir  $(i, j)$  kanalı girecek şekilde  $x=(x_{ij})$  karar değişkenleri matrisini seçmektir. Bu seçim, tur üzerindeki her bir düğümün kendisini izleyen bir düğüme tahsisini ifade etmektedir. Tahsis gerekliliklerinin kendi başlarına  $x$  matrisinin bir tura denk olmasını garanti edemeyecekleri açıktır. Tahsis bir tur yerine, şekil 3.2'deki gibi alt - turlara yol açabilmektedir.



Şekil 3.2. Alt- turlar

Alt-turlar oluşması olasılığını yok etmek için, kanal seçim matrisi seçeneklerine ilave kısıtlar (S) yüklenmesi gerekmektedir. Bu durumda TSP için bir formülasyon aşağıdaki gibi ifade edilebilmektedir.<sup>8</sup>

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (3.2)$$

$$\sum_{i=1}^n x_{ij} = b_j = 1 \quad (j=1, 2, \dots, n) \quad (3.3)$$

$$\sum_{j=1}^n x_{ij} = a_i = 1 \quad (i=1, 2, \dots, n) \quad (3.4)$$

$$X = (x_{ij}) \in S \quad (3.5)$$

$$x_{ij} = 0 \text{ veya } 1 \quad (i, j = 1, 2, \dots, n) \quad (3.6)$$

Yukarıdaki formülasyonda S seti, tahsis kısıtları (3.3), (3.4) ve (3.5)'e uygun olan alt-tur çözümlerini engelleyen herhangi bir kısıtlar seti olabilir. Bu kısıtlar, alt-tur eliminasyon kısıtları olarak adlandırılmakta ve aşağıdaki olası seçenekleri içermektedirler.<sup>9</sup>

$$(1) S = \{(x_{ij}) : \sum_{i \in W} \sum_{j \in W} x_{ij} \geq 1, \quad N \text{ nin boş olmayan her uygun alt seti } W \text{ için};$$

$$(2) S = \{(x_{ij}) : \sum_{i \in R} \sum_{j \in R} x_{ij} \leq |R| - 1, \quad \{2, 3, \dots, n\} \text{ nin boş olmayan her uygun alt seti } R \text{ için};$$

$$(3) S = \{(x_{ij}) : Y_i - Y_j + n x_{ij} \leq n - 1, \text{ bazı } Y_i \text{ reel sayıları ve } 2 \leq i \neq j \leq n \text{ için};$$

S seti, seçenek (1) ve (2) de yaklaşık  $2^n$  adet alt-tur eliminasyon kısıtı içerirken, seçenek (3) de sadece  $n^2 - 3n + 2$  adet kısıt içermektedir. Ayrıca şekil 3.2'deki alt-turların kısıt (3.3), (3.4) ve (3.6)'yı tatmin ettiklerine

<sup>8</sup> L. BODIN, B.L. GOLDEN, A. ASSAD, M. BALL, "Routing and Scheduling of vehicles and Crews: The state of the Art", Computers and operations Research 10(2), 1983, s.83

<sup>9</sup> Bodin ve diğerleri, 1983, Agm., s.83

fakat kısıt (3.5)'e uygun olmadıklarına yani bu şeklin bir tur olmadığına dikkat edilmelidir.

Alt-tur eliminasyon kısıtları (1), çözüm  $x$  deki şebekedeki düğümlerin uygun her alt setinin  $(Q)$  diğer düğümlere bağlanması gerekliliğini ifade etmektedir. Bu kısıtlar, Şekil 3.2 deki  $Q=\{1, 2, 3\}$  çözümünü kabul etmemektedirler. Alt-tur eliminasyon kısıtları (2),  $x$  de seçilen kanalların daireler içermeyeceklerini çünkü  $R$  düğümleri üzerindeki bir dairenin  $|R|$  adet kanal içermesi gerektiğini ortaya koymakta ve Şekil 3.2 deki  $R=\{4, 5, 6\}$  çözümünü kabul etmemektedirler. Şekil 3.2 deki 4-5, 5-6 ve 6-4 kanalları için (3) nolu kısıtların ilavesi,  $3n \leq 3(n - 1)$  şeklindeki bir çelişkiyi ortaya koymaktadır.  $X$  matrisi, bir alt-tur çözümü içerdiği zaman da benzer bir çelişki ortaya çıkmaktadır. Böylece kısıt (1), (2) ve (3)'ün her biri alt-turları engellemektedir.

Yukarıdaki alt-tur eliminasyon kısıtlarına göre, her bir durumda turlar uygundur. Kısıt (1) ve (2) de bu kendiliğinden açıkça görülmektedir. Kısıt (3) için durum biraz daha karmaşık olmakla birlikte burada da uygunluk kolayca açıklanabilmektedir. Bu amaçla;

$$Y_i = \begin{cases} t, & \text{eğer bir turda } i \text{ düğümü } t.\text{yinci adımda ziyaret ediliyorsa} \\ 0, & \text{aksi halde} \end{cases}$$

şeklinde bir değişken tanımlanırsa, bir turdaki bir kanala ait her bir kısıt (3),  $t-(t+1)+n \leq n-1$  olduğunu ifade etmektedir. (3) deki diğer kısıtlar ise sadece  $Y_i \leq n$  ve  $Y_j \geq 1$  olması durumunda  $Y_i - Y_j \leq n-1$  olduğunu ortaya koymaktadır.

TSP'nin yukarıda sunulan en basit 0-1 tamsayı programlama formülasyonunda,  $n^2$  adet 0-1 değişken ve  $n^2 - n + 2$  adet kısıt bulunmaktadır. Bu durumda 50 müşterili (düğümlü) bir problemde bile 2500 adet değişken ve 2452 adet kısıt bulunacaktır. Bu örnek, TSP'nin kesin algoritmalar ile optimal olarak çözülmeye çalışılmasının ne kadar zor ve zaman alıcı olacağını göstermektedir. Bu nedenle TSP'yi çözmek için

bir çok sezgisel strateji geliştirilmiştir. Aşağıdaki bölümde bu stratejilerden bizce en önemli görülenleri açıklanmaktadır.

### 3.1.2. TSP İçin Sezgisel Çözüm Stratejileri

TSP için geliştirilmiş olan sezgisel çözüm stratejileri üç grup içinde ele alınmaktadırlar. Bunlar, tur-inşa algoritmaları, tur iyileştirme algoritmaları ve karma algoritmalarıdır.

Aşağıda bu üç grup içinde yer alan en ünlü ve en önemli sezgiseller ayrı ayrı açıklanmaktadır. Çalışmamızın bundan sonraki kısmında basitlik amacı ile, aksi belirtilmedikçe maliyetlerin her bir  $(i, j)$  çifti için belirli simetrik ( $c_{ij} = c_{ji}$ ) ve üçgen eşitsizliğine uygun ( $c_{ki} \leq c_{kj} + c_{ji}$ ) oldukları varsayılmaktadır.

#### 3.1.2.1. Tur-Inşa Algoritmaları

Tur-inşa algoritmaları, uzaklık veya zaman olarak tanımlanmış olan bir maliyet matrisinden yaklaşık olarak optimal olan bir tur oluşturmaktadırlar. Aşağıda bu yöntemler kısaca açıklanmaktadır.

##### - En yakın komşu algoritması

Rosenkrantz, Stearns ve Lewis (1977) tarafından geliştirilmiş olan yöntem, her bir adımda bir sonraki ziyaret yeri olarak en yakın komşunun tahsisine dayanmaktadır. Yöntemin uygulama adımları aşağıdaki gibidir.<sup>10</sup>

Adım 1. Herhangi bir düğümü bir yolun başlangıcı olarak al.

Adım 2. Yola ilave edilen son düğüme en yakın olan düğümü bul ve bu düğümü yola ilave et.

Adım 3. Yol bütün düğümleri kapsamış oluncaya kadar Adım 2'yi tekrarla.

---

<sup>10</sup> D. ROSENKRANTZ, R. STERNS, P. LEWIS, "An analysis of several heuristics for the travelling salesman problem", SIAM J. Computer 6, 1977, s. 563-581.

Adım 4. Birinci ve sonuncu düğümleri birleştir.

Prosedürün en kötü durum davranışı:

$$\frac{\text{en yakın komşu turunun uzunluğu}}{\text{optimal turun uzunluğu}} \leq \frac{1}{2} \lceil \log(n) \rceil + \frac{1}{2} \quad (3.7)$$

Formül (3.7)'deki logaritma, 2 tabanındadır,  $\lceil x \rceil$  x den büyük olan en küçük tam sayıyı göstermektedir ve n şebekedeki düğümlerin (müşterilerin) sayısıdır.

En yakın komşu algoritmasının icra süresi  $n^2$  ile orantılı olarak artmaktadır. Bu yöntemin alternatif bir uygulama şekli, her düğümü bir kere başlangıç düğümü olarak seçerek çözüm sürecini n kere tekrarlamak ve sonra en düşük maliyetli çözümü seçmektir. Bu durumda icra süresi  $n^3$  ile orantılı olarak artmaktadır.

#### - Clarke-Wright tasarruf algoritması

Bu algoritma, her bir çift düğümü birbirine bağlamanın sağlayacağı tasarrufları seçim kriteri olarak kullanmakta ve aşağıdaki adımlar içerisinde yerine getirilmektedir.<sup>11</sup>

Adım 1. Her bir (i, j) düğüm çifti (i, j = 1,2,...,n) için, bu düğümleri birbirine bağlamanın yani arka arkaya ziyaret etmenin kazandıracağı tasarrufları ( $S_{ij}$ ) hesapla.

$$S_{ij} = C_{oi} + C_{oj} - C_{ij} \quad (3.8)$$

Formüldeki "o" alt indisi depoyu veya bir başka ifade ile satıcının yola çıktığı ve geri döndüğü ana şehri göstermektedir.

Adım 2. Tasarrufları büyükten küçüğe listele.

<sup>11</sup> G. CLARKE, J. WRIGHT, "Scheduling of vehicles from a central depot to a number of delivery points", Operations Research 12, 1964, s. 568-581.

Adım 3. Tasarruflar listesinin en üstünden başlayarak ve aşağıya doğru devam ederek uygun  $i$  ve  $j$  düğümlerini birbirlerine bağla.

Adım 4. Bir tur oluşuncaya kadar Adım 3'ü tekrarla.

Yukarıda özetlenen yöntemin bir versiyonu, her bir adımda en iyi tasarrufun tura son ilave edilmiş olan düğümden seçilmesine dayanan ardışık versiyondur. Bu durumda bütün tasarruflar matrisinin depolanması zorunluluğu ortadan kalktığı için ardışık yöntem hesapsal açıdan daha elverişlidir.

Clarke-Wright tasarruf algoritmasının her iki versiyonu için de en kötü durum performansı bilinmektedir. Golden (1977)<sup>12</sup> ardışık versiyon için, Ong (1981) ise paralel versiyon için en kötü durum oranının  $\log(n)$ 'nin bir lineer fonksiyonu ile sınırlandığını göstermişlerdir.<sup>13</sup>

Algoritmanın Adım 1'indeki tasarruflar matrisi  $S=[S_{ij}]$ 'nin hesaplanması, yaklaşık olarak  $c n^2$  adet işlem gerektirmektedir. Adım 2'de tasarruflar, Williams (1974) ve Floyd (1964) Heapsort metodları ile maximum  $c n^2 \log(n)$  adet karşılaştırma ve yerleştirme ile büyükten küçüğe doğru sıralanabilmektedirler.

Adım 3 ise en fazla  $n^2$  adet operasyon içermektedir. Bu durumda Clarke-Wright tasarruf prosedürü,  $n^2 \log(n)$  adet hesaplama gerektirmektedir.

Çözüm sürecinde her bir düğümün başlangıç düğümü olarak seçilmesi ve elde edilen en iyi çözümün problem çözümü olarak kabul

<sup>12</sup> B GOLDEN, "Evaluating a sequential vehicle routing algorithm", AIIE Transportation 9, 1977, S. 204-208.

<sup>13</sup> H. ONG, "Design and analysis of heuristics for some routing and packing problems", Ph.D. Thesis, University of Waterloo, 1981.

edilmesi durumunda ise gerekli hesaplamaların sayısı  $n^3 \log(n)$  olmaktadır.<sup>14</sup>

### **-Giriş algoritmaları**

Giriş algoritmaları, bir tur oluşturmak için bir seçim veya giriş kriteri kullanılmaktadırlar. Prosedür, k yıncı iterasyonda k adet düğüm üzerinde bir alt tur alarak, bu alt turda bulunmayan düğümler arasından o anda alt tura ilave edilmesi gereken düğümü belirlemeye çalışmaktadır. Bu, seçim adımıdır. İlave edilecek düğüm seçildikten sonra bu düğümün alt turda nereye yerleştirilmesi gerektiği belirlenmektedir. Bu da giriş (insertion) adımıdır. Aşağıda seçim ve giriş adımlarında kullanılacak kriterlere göre farklılık gösteren bazı giriş algoritmaları açıklanmaktadır.

#### **a) En yakın giriş algoritması**

En yakın giriş algoritması, her bir iterasyonda alt turdaki herhangi bir düğüme en yakın olan düğümü seçerek ve bu düğümü turda en az maliyetle girebileceği yere sokarak çalışmaktadır. Algoritmanın uygulama adımları aşağıdaki gibi özetlenebilir.

Adım 1. Sadece düğüm i den ibaret olan bir alt tur ile başla.

Adım 2.  $c_{ik}$  nın minimum olduğu düğümü (k) bul ve i-k-i alt turunu oluştur.

Adım 3. Seçim adımı. Alt turda bulunmayan ve alt turdaki herhangi bir düğüme en yakın olan düğümü (l) bul.

Adım 4. Giriş adımı. Alt turda  $c_{il} + c_{ij} - c_{ij}$  yi minimum yapan kanalı (i, j) bul ve düğüm l yi i ve j arasına sok.

Adım 5. Bir Hamilton dairesi elde edinceye kadar Adım 3 ve 4'ü tekrarla.

---

<sup>14</sup> Bodin ve diğerleri, 1983, Agm., S.88.

Bu prosedür ile elde edilen turun uzunluğu, en kötü durumda optimal turun 2 katı kadardır.<sup>15</sup>

Yani,

$$\frac{\text{en yakın giriş turunun uzunluğu}}{\text{optimal turun uzunluğu}} \leq 2 \quad (3.9)$$

Metod  $n^2$  adet hesaplama gerektirmektedir. Ancak eğer  $n$  adet düğümün her biri başlangıç düğümü olarak seçilerek,  $n$  adet çözüm elde edilir ve bunlar arasından en düşük maliyetli olan tur seçilirse, bu miktar  $n^3$  olmaktadır.<sup>16</sup>

#### b) En ucuz giriş algoritması

Bu yöntem, en yakın giriş algoritmasına çok benzemektedir. Ancak en yakın giriş algoritmasının seçim adımında alt turdaki herhangi bir düğüme en yakın olan düğüm seçilirken, burada alt turda bulunmayan bütün düğümler için en düşük maliyetli giriş yerleri belirlenerek, bunlar arasından minimum maliyetli giriş yerine sahip olan düğüm alt tura sokulmaktadır. Algoritmanın uygulama adımları şöyledir.

Adım 1. Sadece düğüm  $i$ 'den ibaret olan bir alt tur ile başla.

Adım 2.  $c_{ik}$  nın minimum olduğu düğümü ( $k$ ) bul ve  $i$ - $k$ -i alt turunu oluştur.

Adım 3.  $c_{il} + c_{lj} - c_{ij}$  yi minimum yapan ve alt turda bulunan ( $i, j$ ) ve alt turda bulunmayan  $l$  düğümünü bul ve  $l$  yi  $i$  ve  $j$  arasına yerleştir.

Adım 4. Bir Hamilton dairesi elde edinceye kadar Adım 3'ü tekrarla.

<sup>15</sup> Bodin ve diğerleri, 1983, Agm., S. 88.

<sup>16</sup> Bodin ve diğerleri, 1983, Agm., S.88.

En ucuz giriş algoritmasının en kötü durum performansı en yakın giriş algoritmasınıninkinin aynısıdır. Fakat bu algoritmanın icra süresi daha fazla olup,  $n^2 \log n$  olarak verilmektedir.<sup>17</sup>

### c) Keyfi giriş algoritması

Keyfi giriş algoritmasının düğüm seçme aşaması, daha önce seçilmemiş olan düğümler arasından rastgele bir düğümün seçimine dayanmaktadır. Yöntemin diğer adımları ise en yakın giriş algoritmasınıninkinin aynısıdır. Kolayca anlaşılabileceği gibi bu yöntemdeki düğüm seçiminin keyfiliği, yöntemin en kötü durum performansını azaltmaktadır. Keyfi giriş algoritmasının en kötü durum performansı

$$\frac{\text{keyfi giriş turunun uzunluğu}}{\text{optimal turun uzunluğu}} \leq \lceil \log n \rceil + 1 \quad (3.10)$$

ve icra süresi  $n^2$  olarak verilmektedir.<sup>18</sup>

### d) En uzak giriş algoritması

Düğüm seçiminin, alt turdaki herhangi bir düğüme en yakın olan yerine en uzak olan düğümü seçmeye dayanması dışında, bu yöntem de en yakın giriş algoritmasının aynısıdır. Yöntemin en kötü durum performansı da en yakın giriş algoritmasınıninkinin aynısı olmaktadır.

### e) Çabuk giriş veya en yakın ekleme algoritması

Bu algoritmanın uygulama adımları aşağıdaki gibi özetlenebilir.<sup>19</sup>

Adım 1. Bir tek düğüm ve sıfır kanallı bir başlangıç devresi (T) olarak herhangi bir düğümü seç.

<sup>17</sup> Bodin ve diğerleri, 1983, Agm., S. 88.

<sup>18</sup> Bodin ve diğerleri, 1983, Agm., S. 88.

<sup>19</sup> Bodin ve diğerleri, 1983, Agm., S. 89.

Adım 2.  $k$  adet düğüme sahip olan  $T_k$  devresi için,  $T_k$  üzerindeki bir düğüme ( $Y_k$ ) en yakın olan ve  $T_k$  üzerinde bulunmayan düğümü ( $Z_k$ ) bul.

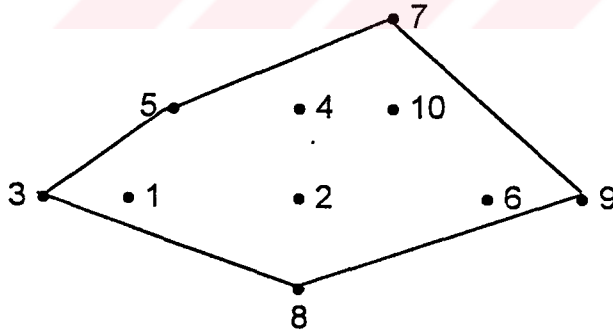
Adım 3.  $Z_k$  düğümünü  $T_k$  üzerindeki  $Y_k$  nın hemen önüne yerleştirerek  $k + 1$  adet düğüme sahip olan  $T_{k+1}$  devresini elde et.

Adım 4. Bütün düğümleri içeren bir Hamilton dairesi elde edinceye kadar Adım 2 ve 3'ü tekrarla.

Çabuk giriş algoritmasının en kötü durum performansı ve icra süresi en yakın giriş prosedürününkinin aynısıdır.

#### f) Dışbükey kabuk giriş algoritması

Bir düğümler setinin dışbükey kabuğu, setin bütün elemanlarını içeren en küçük dışbükey alt settir.<sup>20</sup> Şekil 3.3 de 10 düğümden oluşan bir düğümler setinin dışbükey kabuğu gösterilmektedir. Şekilde yerleşimleri verilmiş olan 10 düğüm için, 3, 5, 7, 8 ve 9 nolu düğümler dışbükey kabuğu oluşturmaktadırlar.



Şekil 3.3. Bir düğümler setinin dışbükey kabuğu

<sup>20</sup> S.V. BEGUR, "Capacitated Multi Period Vehicle Routing with Fived Period Length," Ph. D. Thesis, University of Alabama, 1994, S. 17.

Hardgrave ve Nemhauser (1962),  $c_{ij}$  maliyetinin öklid uzaklıklarını gösterdiği iki boyutlu bir düzlemde bir TSP için dışbükey kabuk (H) üzerindeki düğümlerin ortaya çıkış sıralarının aynı zamanda optimal turdaki sıralama olacağını göstermişlerdir.<sup>21</sup> Buradaki "öklid uzaklığı" terimi,  $(X_i, Y_i)$  koordinatları ile verilen iki nokta arasında aşağıdaki formüle göre hesaplanan en kısa uzaklığı ifade etmektedir.

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.11)$$

Giriş algoritması gibi bir tur inşa metodu için, dışbükey kabuğun bir başlangıç çözümü olarak kullanılması ümit verici sonuçlar ortaya koymuştur. Bu konuda Preparata ve Hong (1977), Wiorowski ve Mc Elvain (1975) ve Norback ve Love (1977) özellikle dikkate değer çalışmalardır. Bütün bu çalışmalardaki ortak düşünceden yola çıkılarak, dışbükey kabuk giriş algoritmasının uygulama adımları aşağıdaki gibi özetlenebilir.

Adım 1. Düğümler setinin dışbükey kabuğunu oluştur. Bu kabuk bir başlangıç turunu vermektedir.

Adım 2. Henüz alt turda bulunmayan her bir düğümün (k) alt turdaki hangi iki düğüm (i ve j) arasına sokulacağına karar vermek için  $c_{ik} + c_{kj} - c_{ij}$  nin minimum olması kriterini kullan.

Adım 3. Adım 2'de bulunan bütün (i, k, j) üçlülere arasından,  $(c_{ik} + c_{kj} - c_{ij})/c_{ij}$  değerini minimum yapan üçlüyü ( $i^*$ ,  $k^*$ ,  $j^*$ ) belirle.

Adım 4. Düğüm  $k^*$  yı alt turdaki düğüm  $i^*$  ve  $j^*$  arasına yerleştirir.

Adım 5. Bir Hamilton dairesi elde edinceye kadar Adım, 2, 3 ve 4'ü tekrarla.

---

<sup>21</sup> Begur, Age., S. 18.

Bu prosedürün en kötü durum performansı bilinmemektedir. İcra süresi ise  $n^3 \log(n)$  olarak verilmektedir.<sup>22</sup>

#### g) En büyük açılış algoritması

Başlangıç turu olarak düğümler setinin dışbükey kabuğunu alan bir diğer yöntem en büyük açılış algoritmasıdır ve uygulama aşamaları aşağıda özetlendiği gibidir.

Adım 1. Düğümler setinin dışbükey kabuğunu oluşturarak bir başlangıç turu elde et.

Adım 2. Alt turda bulunan kanallar  $(i, j)$  ve alt turda bulunmayan düğümler  $(k)$  için,  $(i^*, k^*)$  ve  $(k^*, j^*)$  şeklinde oluşturulan kanallar arasındaki en büyük açığı veren düğümü  $(k^*)$  ve kanalı  $(i^*, j^*)$  seç.

Adım 3.  $k^*$  düğümünü alt turdaki  $i^*$  ve  $j^*$  düğümleri arasına yerleştir.

Adım 4. Bir Hamilton dairesi elde edinceye kadar Adım 2 ve 3'ü tekrarla.

Adım 2'de açılar, üç köşesinin koordinatları bilinen bir üçgenin iç açılarının bulunması için kullanılan yöntemlerden herhangi biri ile bulunabilirler.

Yöntemin en kötü durum davranışı bilinmemekle birlikte, en büyük açılış turunun uzunluğunun optimal turun uzunluğuna oranını sınırlayan bir sabitin mevcut olmadığı şeklinde olumsuz bir yargı bulunmaktadır. Yöntemin icra süresi ise  $n^2 \log(n)$  olarak verilmektedir.<sup>23</sup>

<sup>22</sup> Bodin ve diğerleri, 1983, Agm., S. 89.

<sup>23</sup> Bodin ve diğerleri, 1983, Agm., S. 90.

### h) Fark x oran giriş algoritması

Or (1976) tarafından geliştirilmiş olan algoritmanın uygulama adımları aşağıdaki gibidir.<sup>24</sup>

Adım 1. Döğümler setinin dışbükey kabuğunu oluşturarak bir başlangıç alt turu elde et.

Adım 2. Alt turda bulunan kanallar (i, j) ve alt turda bulunmayan döğümler (k) arasından,  $\{C_{i-k} + C_{k-j} - C_{i-j}\} \times \{C_{i-k} + C_{k-j} / C_{i-j}\}$  çarpımını en küçük yapan  $k^*$  döğümünü ve  $(i^*, j^*)$  kanalını seç.

Adım 3.  $k^*$  döğümünü alt turdaki  $i^*$  ve  $j^*$  döğümleri arasına yerleştir.

Adım 4. Bir Hamilton dairesi elde edinceye kadar Adım 2 ve 3'ü tekrarla.

Fark x oran giriş algoritmasının en kötü durum performansı bilinmemekte fakat icra süresi  $n^2 \log$  olarak verilmektedir.

### - Minimal tarama ağacı yaklaşımı

Minimal tarama ağacı yaklaşımı ile TSP turunun belirlenmesi, öncelikle problemdeki döğümlerin minimal tarama ağacının bulunmasına dayanmaktadır.

Bu amaçla kullanılabilecek bir çok polynomial algoritma mevcut olmakla birlikte bizim çalışmamızda Prim (1957) tarafından geliştirilmiş olan algoritma kullanılmaktadır.

Bir tarama ağacı, bütün döğümlerin birbirlerine bağlı oldukları fakat dairelerin bulunmadığı bir alt grafiktir. Bir minimal tarama ağacı ise,

---

<sup>24</sup> I. OR, "Travelling salesman- type combinatorial problems and their relation to the logistics of blood banking", Ph. D. Thesis, Northwestern University, 1976.

düğümleeri birbirine bağlayan dalların (kanalların) maliyetleri toplamının minimum olduđu bir ağaçtır.

Şekil 3.4'de bir minimal tarama ağacı örneđi gösterilmektedir. Bu şekilden de anlaşılacağı gibi bir minimal tarama ağacı örneđi gösterilmektedir. Bu şekilden de anlaşılacağı gibi bir minimal arama ağacının maliyeti, bir optimal TSP turunun maliyetinden daha düşüktür. Çünkü tarama ağacı bir optimal TSP turu-bir kanala denktir. Minimal tarama ağacının belirlenmesi için Prim (1957) nin algoritması aşağıdaki gibi özetlenebilir.<sup>25</sup>

Adım 1. Bir kök düğüm seç (düğüm o)

Adım 2. Ağaca dahil olmayan her bir j düğümünü ( $i, c_{ij}$ ) ile etiketle. Burada "i" j düğümünün öncelini ve  $c_{ij}$  ise j düğümünü i düğümüne bağlamanın maliyetini göstermektedir. Bu durumda birinci aşamada etiketler ( $o, c_{oj}$ ) şeklinde olmaktadır.

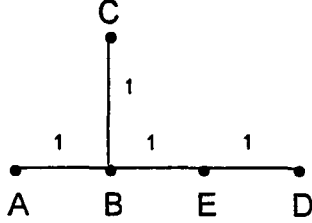
Adım 3. En küçük  $c_{ij}$  değerine sahip olan düğümü (k) seç ve bir kanal ile i düğümüne bağla.

Adım 4. Ağaç dışındaki her bir j düğümünün etiketini kontrol et. Eğer  $c_{kj} < c_{ij}$  ise etiketi ( $k, c_{kj}$ ) olarak değiştir. Aksi halde, olduğu gibi bırak.

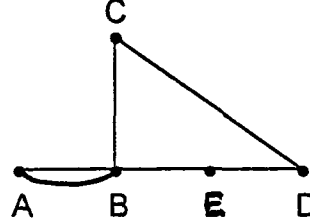
Adım 5. Daireler oluşmamasına dikkat ederek, bütün düğümler ağaca dahil oluncaya kadar Adım 3 ve 4 ü tekrarla.

Minimal tarama ağacı ile TSP turunun belirlenmesinde kullanılan bazı ifadelerin burada açıklanması gerekmektedir. Bunlardan birincisi, "tek" veya "çift" dereceli düğüm ifadesidir. Bir düğümün derecesi, o düğüme giren ve çıkan toplam kanal sayısına göre belirlenmektedir.

Şekil 3.4'de A, B, C ve D düğümleri tek dereceli, E düğümü ise çift derecelidir. Açıklanması gereken ikinci ifade "Euler grafiği" veya "Euler dairesi"dir. Bir Euler dairesi, bütün düğümlerin kapsandığı ve her bir düğümün çift dereceli olduğu bir kapalı şekildir. Şekil 3.5 de bir Euler dairesi örneği gösterilmektedir.



Şekil 3.4. Minimal tarama ağacı



Şekil 3.5. Euler dairesi

Yukarıdaki bilgilerin ışığında şimdi, Kim (1975) tarafından geliştirilmiş olan minimal tarama ağacı yaklaşımı aşağıdaki gibi özetlenebilir.<sup>26</sup>

- Adım 1. Bütün düğümlerin bir minimal tarama ağacını (T) oluştur.
- Adım 2. Minimal tarama ağacındaki kanalları, bir Euler dairesi elde edecek şekilde çiftleştir.
- Adım 3. 2'den büyük dereceli düğümler üzerindeki poligonları kaldır ve Euler dairesini bir Hamilton dairesine dönüştür.

Yukarıda özetlenen prosedürün en kötü durum performansı,

$$\frac{\text{Minimal tarama ağacı turunun uzunluğu}}{\text{optimal turun uzunluğu}} \leq 2 \quad (3.12)$$

ve icra süresi  $n^2$  dir.<sup>27</sup>

<sup>25</sup> R.C. PRIM, "Shortest connection networks and some generalisations", Bell system Technique Journal 36, 1957, S. 1389-1401.

<sup>26</sup> C. KIM, "A minimal spanning tree and approximate tours for a travelling salesman", Comp. Sci. Tech. Report, University of Maryland, 1975.

<sup>27</sup> Bodin ve diğerleri, 1983, Agm., S. 90.

### - Christofides sezgiseli

Christofides (1976), TSP'leri çözmek için oldukça etkin bir sezgisel teknik geliştirmiştir. Bu teknik, gene düğümlerin minimal tarama ağacının oluşturulmasına dayanmakla birlikte, Euler grafiğın oluşturulması için bir minimum ağırlıklı eşleştirme problemi, genel anlamı itibari ile belirli bir çift sayıda nesnenin (burada düğümlerin) ikişer ikişer eşleştirilmesi problemidir ve Lawler (1979) standart algoritması ile  $O(n^2)$  süre içinde çözülebilmektedir. Söz konusu standart algoritma ile bulunan optimal eşleştirme çözümünün ağırlığı ( $c_m$ ), minimum TSP turunun uzunluğunun yarısından daha büyük olmamaktadır.<sup>28</sup>

Christofides sezgiselinin uygulama adımları aşağıdaki gibidir.<sup>29</sup>

- Adım 1. Bütün düğümlerin bir minimal tarama ağacını (T) oluştur.
- Adım 2. T deki tek dereceli düğümleri belirle. Orijinal maliyet matrisini kullanarak tek dereceli düğümler üzerinde bir minimum maliyetli eşleştirme problemini çöz.
- Adım 3. Eşleştirme çözümünden elde edilen dalları hali hazırda T de bulunan dallara ilave ederek bir Euler dairesi elde et.
- Adım 4. 2'den büyük dereceli düğümler üzerindeki poligonları kaldırarak Euler dairesini bir Hamilton dairesine dönüştür.

Christofides sezgiselinin icra süresi  $n^3$ 'dür ve en kötü durum performansı,

$$\frac{\text{Christofides turunun uzunluğu}}{\text{optimal turun uzunluğu}} \leq 1,5 \quad (3.13)$$

<sup>28</sup> E. LAWLER, "Shortest path and network flow algorithms", Ann. Discrete Math. 4, 1979, S. 251-263.

<sup>29</sup> N. CHRISTOFIDES, "Worst-case analysis of a new heuristic for the travelling salesman problem", Report 388, Carnegie Mellon University, 1976.

olduğu için üçgen eşitsizliğinin geçerli olduğu TSP'ler için bilinen en iyi algoritmadır.<sup>30</sup>

#### - Karp (1977) dörtgen bölge parçalama metodu

Karp (1977) tarafından çok büyük TSP'lerin çözülebilmeleri için bir parçalama algoritması geliştirilmiştir.<sup>31</sup> Algoritmadaki ana düşünce, problemdeki müşteri ve depo yerleşimlerinin oluşturduğu büyük dörtgeni bir çok alt dörtgene bölmek ve her bir alt dörtgende bir TSP'yi çözmektir. Bu küçük TSP'lerin çözümleri sonucunda bir Euler dairesi ortaya çıkmakta ve bu da daha önce açıklanmış olan yöntemlerden birisi ile kolayca bir Hamilton dairesine yani bir bütün TSP turuna dönüştürülebilmektedir.

Dörtgen parçalama algoritmasında, N adet noktadan oluşturulan bir dörtgen ve bir  $t \geq 2$  tam sayısı için, orijinal dörtgen her biri en fazla t adet nokta içeren  $2^k$  adet alt dörtgene bölünmektedir. Bu bölme işlemi sırasında her bir alt dörtgenin her bir kenarının noktalardan birinden geçmesine veya orijinal dörtgenin bir kenarı ile çakışmasına dikkat edilmelidir.

$$k = \lceil \log_2 \{(N-1)/(t-1)\} \rceil \quad (3.14)$$

#### 3.1.2.2. Tur İyileştirme Metodları

İyileştirme metodları optimal olmayan uygun bir başlangıç çözümünün, bir araştırma mekanizması vasıtası ile ardışık olarak iyileştirilmesine dayanmaktadırlar. İyileştirme sezgisellerinin genel adımları aşağıdaki gibi özetlenebilirler.

Adım 1. Uygun bir başlangıç çözümü (T) oluştur.

Adım 2. T nin bir dönüşümü ile daha iyi bir uygun çözüm (T') bulmaya çalış.

---

<sup>30</sup> Bodin ve diğerleri, 1983, Agm., S. 90.

Adım 3. Eğer daha iyi bir çözüm bulursan, yani uygun bir kazanç fonksiyonuna ( $f$ ) göre  $f(T') < f(T)$  ise  $T$  yerine  $T'$  yü yerleştir ve Adım 2'ye git.

Adım 4. Eğer daha iyi bir çözüm bulamazsan,  $T$  yerel optimum çözümdür. Hesaplama süresi sona erinceye veya cevaplar tatmin edici oluncaya kadar bütün adımları tekrarla.

Yöntemin ikinci adımındaki sürekli olarak daha iyi bir çözüm arama şartı, başlangıç çözümlerini yerel optimuma yönlendirmekte ve çözümün kalitesi başlangıç çözümünün kalitesine bağlı olmaktadır. Başlangıç çözümünü oluşturan sezgisel iyileştikçe yerel optima seti küçülmekte ve global optimuma götüren rastgele başlangıç çözümlerinin oranı artmaktadır. Algoritmanın her bir adımında daha iyi bir çözüm arama yerine amaç fonksiyonunda geçici bir kötüleşmeye izin verilmesi durumunda, yerel optimuma yaklaşma ve başlangıç çözümünün yerel iyileştirme metodlarına etkisi bertaraf edilebilmektedir. Bölüm 3.2 de sunulan Yasak araştırması (YA) ve Faz dönüşüm (FD) yaklaşımları, böyle global optimum araştırma metodları geliştirmek için kullanılabilirler.

$S$ , bir TSP deki bütün düğümleri ikişer ikişer birbirlerine bağlayabilecek olan bütün olası kanalların setini göstermek üzere, Adım 2'de daha iyi bir uygun çözüm ( $T'$ ) elde etmek için  $T$  nin sabit  $k$  sayıda elemanı  $S/T$  nin  $k$  adet elemanı ile değiştirilmektedir. Böyle gruplar bulunabildiği sürece bu işlem tekrarlanmakta ve sonuçta  $T$  nin böyle dönüşümler ile iyileştirilmesi artık mümkün olmadığı zaman yerel optimum bulunmuş olmaktadır. Bulunan tur,  $k$ -optimal olarak adlandırılmaktadır. Bir  $k$ -optimal tur ayrıca  $k' < k$  için  $k'$ -optimaldır.<sup>32</sup>

<sup>31</sup> R. KARP, "Probabilistic analysis of partitioning algorithms for the travelling salesman problem in the plane", Math. Ops. Research 2, 1977, S. 209-224.

<sup>32</sup> M. DROR, L. LEVY, "A vehicle routing improvement algorithm comparison of a greedy and matching implementation for inventory routing", Computers and Operations Research 13, 1986, S. 33-45.

Doğal olarak, doğru kanal grupları değiştirilerek daima optimal çözümün bulunabileceği açıktır. Ancak  $k$  adet kanalın bütün olası kombinasyonlarının test edilmesi için gerekli operasyon sayısı  $n^k$  ile artmaktadır ( $n$ , düğüm sayısıdır). Bu komplekslikten dolayı genellikle  $k=2$  ile Croes (1958) ve  $k=3$  ile Lin (1965) tarafından geliştirilmiş olan stratejiler kullanılmaktadır.

Bununla birlikte Lin ve Kennighan (1973), her bir iterasyonda kaç tane kanalın değiştirilmesi gerektiğine karar veren bir algoritma geliştirmişlerdir. K-opt olarak adlandırılan bu algoritma aşağıda bu bölüm içerisinde ayrıntılı olarak açıklanmaktadır.

2-opt değişim algoritması, her bir adımda iki kanal yerine diğer iki kanalın ve 3-opt değişim algoritması ise üç kanal yerine diğer üç kanalın yerleştirilmesine dayanmaktadır. Çözümünden çıkartılacak ve bunların yerine çözüme sokulacak kanalların belirlenmesinde bir kabul ve seçim kriterine ihtiyaç vardır. 2-opt algoritmasında, çözümünden çıkartılacak kanallar  $i-j$  ve  $k-l$  ve çözüme sokulacak kanallar  $i-k$  ve  $j-l$  olmak üzere kabul ve seçim kriteri, aşağıdaki tasarruf ifadesi ile verilmektedir.

$$OS(i, j; k, l) = C_{ij} + C_{kl} - C_{ik} - C_{jl} \quad (3.15)$$

3-opt algoritmasında çözümünden çıkartılacak kanallar  $i-j$ ,  $k-l$  ve  $m-n$  ve çözüme girecek kanallar  $i-l$ ,  $j-m$  ve  $j-n$  olmak üzere kabul ve seçim kriteri aşağıdaki gibi alınabilir.

$$OS(i, j; k, l; m, n) = C_{ij} + C_{kl} + C_{mn} - C_{il} - C_{jm} - C_{jn} \quad (3.16)$$

Kabul açısından yukarıdaki tasarruf değerlerinin pozitif olmaları yeterlidir. Seçim aşaması ise farklı kombinasyonlara ait tasarruf değerlerinin en büyüğüne göre belirlenmektedir.

2-opt ve 3-opt algoritmalarında, değiştirilecek kanal kombinasyonlarının değerlendirilmelerindeki sınırlayıcı bir kanal kombinasyonlarının değerlendirmeye alınmamalarıdır

uygun olmayan alt turlara yol açmaktadır. Ayrıca bu prosedürlerin her aşamasında alt turların oluşmamasına ve orijinal kanalların tekrarlanmamasına dikkat edilmelidir.

### - K-opt algoritması

Optimal olmayan bir çözüm (T) de doğru yerde bulunmayan k adet  $X_1, X_2, \dots, X_k$  elemanının bulunduğu söylenebilir. T'yi optimal hale getirmek için  $X_1, X_2, \dots, X_k$  elemanlarının yerine S/T'den k adet  $Y_1, Y_2, \dots, Y_k$  elemanı yerleştirilmektedir. Bu durumda problem, basit olarak k, X ve Y lerin belirlenmesi problemidir. k'nın belirlenmesi ve sonra bu sabit k değeri için T nin bütün olası k-alt setlerinin ele alınması aşırı yorucu ve zaman alıcı bir iştir.

Lin ve Kernighan (1973), yapılabilecek en iyi şeyin k,  $X_1, X_2, \dots, X_k$  ve  $Y_1, Y_2, \dots, Y_k$  y, eleman eleman bulmaya çalışmak olduğunu açıklamaktadır.<sup>33</sup> Bu nedenle ilk olarak en yanlış yerdeki eleman olarak  $X_1$  ve  $Y_1$  belirlenmeye çalışılmakta ve sonra bu  $X_1$  ve  $Y_1$  dışında kalan setlerdeki en yanlış yerdeki  $X_2$  ve  $Y_2$  bulunmakta ve bu şekilde devam edilmektedir. Bu algoritmanın işleyişi daha formel olarak aşağıdaki gibi ifade edilebilir.

Adım 1. Rastgele bir başlangıç çözümü oluştur.

Adım 2.

(a)  $i=1$

(b)  $X_1, X_2, \dots, X_i$  yerine  $Y_1, Y_2, \dots, Y_i$  çözüme sokulduğunda elde edilen iyileşme maximum olacak şekilde  $T/\{X_1, X_2, \dots, X_{i-1}\}$  den i yinci adımda en yanlış yerdeki elemanı ( $X_i$ ) ve  $S/T/\{Y_1, Y_2, \dots, Y_{i-1}\}$  den bunun yerine çözüme sokulacak elemanı ( $Y_i$ ) seç.

---

<sup>33</sup> S. LIN, B. W. KERNIGHAN, "An effective heuristic algorithm for the travelling salesman problem", Operations Research 21, 1971, S. 498-516.

- (c) Eğer uygun bir durdurma kuralına göre daha fazla kazanç elde edilemeyeceği görülürse Adım 3'e git. Aksi halde  $i=i+1$  ve Adım 2 (b) ye geri dön.

Adım 3. Eğer  $i=k$  için en fazla iyileşme bulunursa  $X_1, X_2, \dots, X_k$  ile  $Y_1, Y_2, \dots, Y_k$  yı değiştirerek yeni bir çözüm (T) elde et. Adım 2'ye git. Eğer iyileşme bulunmazsa Adım 4'e git.

Adım 4. Eğer istersen, Adım 1'den itibaren tekrarla.

Yukarıdaki algoritmanın işleyişi sırasında aşağıdaki araçlara ihtiyaç duyulmaktadır.

- (1) Mevcut durumda en yanlış yerdeki  $X_i$  ve buna karşılık  $Y_i$  çiftinin hangisi olduğunu çabuk ve elverişli olarak söyleyecek bir seçim kuralı.
- (2) Olası değişiklikler setinden elde edilecek toplam kazancı temsil eden basit bir fonksiyon  $X_1, Y_1, \dots, X_{i-1}, Y_{i-1}$ 'in daha önce seçilmiş olmaları şartı ile  $g_i$ ,  $X_i$  ve  $Y_i$ 'nin birbirleri ile değiştirilmelerine ait "kar" veya "kazanç" olmak üzere,  $X_1, X_2, \dots, X_i$  ile  $Y_1, Y_2, \dots, Y_i$  nin değiştirilmesinden sağlanan kümülatif kazancın  $(g_1+g_2+\dots+g_i)$  kullanılması çok elverişli olmaktadır. Bu kümülatiflik, bir  $g_i$ 'nin negatif olması halinde seçim sürecinin durmasını engellemektedir. Bunun yerine  $\sum_{i=1}^k g_i \leq 0$  olduğu zaman seçim süreci durmakta ve bu da yerel optimayı savmaya yardımcı olmaktadır.
- (3) Seçim sürecinin herhangi bir  $k$  değeri ve  $X_1, X_2, \dots, X_k$  ile  $Y_1, Y_2, \dots, Y_k$  değişikliğini seçebilmesi için, bu değişikliğin uygun olduğunun peşinen bilinmesi, önemli zaman ve emek tasarrufuna yol açacaktır.
- (4) Değiştirilecek elemanları ararken, daha başka kazanç elde edilemeyeceğini veya kazançların azalma noktasına varmış olduğunu söyleyecek bir durdurma kuralı. Bu kural, verimsiz yolları izlemek için

çok fazla zaman ve emek harcayabilen "iyimserlik" ile çok iyi fırsatları kaçırabilen "kötümserlik" arasında hassas bir denge gerektirmektedir.

- (5) Son ihtiyacımız,  $X_1, X_2, \dots, X_k$  ve  $Y_1, Y_2, \dots, Y_k$  setlerinin birbirleri ile çakışmamasıdır. Bunun anlamı, bir elemanın bir iterasyonda çözümden çıkartıldıktan sonra, aynı iterasyonda tekrar çözüme sokulamamasıdır. Ancak bu eleman, daha sonraki bir iterasyonda tekrar çözüme sokulabilmektedir.

Şimdi temel sezgisel algoritma ayrıntılı olarak aşağıdaki gibi açıklanabilir.

Adım 1. Rastgele bir başlangıç turu (T) oluştur.

Adım 2.  $G^*$  o ana kadar gerçekleştirilmiş olan en fazla iyileşme olmak üzere,  $G^*=0$  yap. Herhangi bir düğüm ( $t_1$ ) ve T de bir uç noktası  $t_1$  olan bir kanal ( $X_1$ ) seç.  $i=1$ .

Adım 3.  $g_1 > 0$  olması şartı ile  $X_1$ 'in diğer uç noktası ( $t_2$ ) ile  $t_3$  arasındaki  $Y_1$ 'i seç.

Eğer bu şartı sağlayan bir  $Y_1$  mevcut değil ise Adım 6(d) ye git (Bu, kazanç kriterinin birinci uygulamasıdır).

Adım 4.  $i=i+1$   $t_{2i-1}$  ile  $t_{2i}$  arasındaki  $X_i$  ve  $Y_i$ yi aşağıdaki gibi seç.

(a)  $X_i$  yi  $t_{2i}$  ile  $t_1$  birleştirildiği zaman ortaya çıkan konfigürasyon bir tur olacak şekilde seç. Böylece verilen bir  $Y_{i-1}$  için  $X_i$  nin tek seçeneği söz konusudur. Bu, uygunluk kriterinin uygulanmasıdır ve herhangi bir  $i \geq 2$  için  $t_{2i}$  yi  $t_1$  ile birleştirerek daima bir turu kapatabilmemizi sağlamaktadır. Adım 4 (e) deki  $Y_{i-1}$ 'in seçimi, daima böyle bir  $X_i$  nin varlığını garanti etmektedir.

(b)  $Y_i$ , (c), (d) ve (e) ye göre  $t_{2i}$  uç noktasında  $X_i$  ile birleşen bir mevcut kanaldır. Eğer  $Y_i$  mevcut değil ise Adım 5'e git. Açıktır ki

i'inci adımda büyük bir maliyet indirimi elde etmek için,  $|Y_i|$  küçük olmalıdır.

Bu yüzden genellikle en yakın komşular seçilir.

(c) X ve Y lerin çakışmalarını garanti etmek için  $X_i$  daha önce birleştirilmiş olan (yani bir  $Y_j$ ,  $j < i$ ) ve  $Y_i$  de daha önce silinmiş olan bir kanal olamaz.

$$(d) G_i = \sum_{j=1}^i g_j > 0 \text{ (kazanç kriteri)} \quad (3.17)$$

(e)  $i+1$  de (a) daki uygunluk kriterine ters düşmemek için, seçilen  $Y_i$  bir  $X_{i+1}$ 'in silinmesine imkan vermelidir.

(f)  $Y_i$  oluşturulmadan önce,  $t_{2i}$  ile  $t_1$ 'i birleştirerek turu kapatmanın daha önce görülmüş olan en iyi kazanç değerinden daha iyisini verip vermeyeceği kontrol edilmelidir.  $i \geq 2$  için uygunluk kriteri gerçekleşmiş olduğu için bunun bir tur oluşturacağı bilinmektedir.  $Y_i^*$ ,  $t_{2i}$  ile  $t_1$ 'i birleştiren bir kanal ve  $g_i^* = |Y_i^*| - |X_i|$  olmak üzere, eğer  $G_{i-1} + g_i^* > G^*$  ise  $G^* = G_{i-1} + g_i^*$  ve  $k=i$  olur. ( $G^*$  daima o ana kadar T de kaydedilen en fazla iyileşmedir ve bu yüzden algoritmanın karşılaştırma kriteridir.  $G^* \geq 0$  dır ve monoton artandır. Index k,  $G^*$ 'yi elde etmek için değiştirilecek setleri ifade etmektedir.).

Adım 5.  $X_i$  ve  $Y_i$  kanalları 4(c)-(e) yi sağlamadığı veya  $G_i \leq G^*$  olduğu zaman Adım 2-4 arasındaki  $X_i$  ve  $Y_i$  nin oluşturulmasını sona erdir. Bu, durdurma kriteridir. Eğer  $G^* > 0$  ise  $f(T') = f(T) - G^*$  ile  $T'$  turunu al ve  $T'$  yi başlangıç turu olarak kullanarak Adım 2 den itibaren bütün süreci tekrarla.

Adım 6. Eğer  $G^*=0$  ise,

- (a)  $Y_2$  leri uzunluklarına göre küçükten büyüğe sıra ile seçerek  $g_1+g_2>0$  şeklindeki kazanç kriterine uygun oldukları sürece, Adım 4 ve 5'i tekrarla. Eğer herhangi bir seferde bir iyileşme bulunursa bu tabi ki Adım 2 ye geri dönmeye neden olmaktadır.
- (b) Eğer Adım 4(b) de bütün  $Y_2$  seçeneklerinin kazançsız oldukları görülürse Adım 4(a) ya geri dön ve  $X_2$  için alternatif seçeneği incele (Alternatif  $X_2$ 'nin silinmesi uygunluk kriterinin geçici olarak bozulmasına sebep olmaktadır).
- (c) Eğer bu da iyileşme sağlamaz ise, Adım 3 de  $Y_1$ 'leri uzunluklarına göre küçükten büyüğe sıra ile incele.
- (d) Eğer bütün  $Y_1$  seçenekleri de kazançsız iseler Adım 2 de alternatif  $X_1$  i dene.
- (e) Eğer bu da başarısız olur ise, yeni bir  $t_1$  seç ve Adım 2 den itibaren süreci tekrarla.

Adım 7.  $t_1$  in n adet değerinin hepsi kazançsız iseler algoritmayı bitir. Bu sefer, Adım 1 deki daha başka rastgele turları ele alabilirsin.

### 3.1.2.3. Karma Yöntemler

Karma yöntemler, genellikle bir tur inşa algoritması ile bir tur iyileştirme algoritmasının kombinasyonudurlar. Karma yöntemlerin ardındaki temel düşünce, hızlı olarak iyi bir başlangıç çözümü elde etmek ve sonra optimale yakın bir çözüm bulmak için 2-opt ve 3-opt algoritmalarını uygulamaktır. Hesaplama açısından nispeten hızlı olan ve oldukça iyi sonuçlar veren karma yöntem aşağıdaki gibi özetlenebilir.

Adım 1. Tur-inşa algoritmalarından birini kullanarak bir başlangıç turu elde et.

Adım 2. Adım 1 de bulunan tura bir 2-opt algoritması uygula.

Adım 3. Adım 2 de bulunan tura bir 3-opt algoritması uygula.

Yukarıda özetlenen karma yöntemin bazı olası varyantları bulunabilir. Bunlardan birisinde, başlangıç turuna sadece 3-opt algoritması uygulanırken, bir diğerinde ise sadece 2-opt algoritması uygulanarak karma yöntem çok hızlandırılmakta, ancak sonuçların hassaslığından biraz fedakarlık yapılmaktadır. Üçüncü bir varyant, Adım 1 de farklı tur-inşa algoritmaları kullanılarak karma yöntemin birkaç kere uygulanması şeklinde açıklanabilir. Bu, diğerlerinden daha fazla zaman gerektirmesine rağmen daha iyi sonuçlar vermektedir. Dördüncü bir varyant da ise, Adım 1 de tur-inşa algoritması az sayıda başlangıç düğümü için uygulanmakta ve sonra bu çözümlerin en iyisi için Adım 2 ve 3 uygulanmaktadır.

#### 3.1.2.4. Yönlü (asimetrik) TSP İçin Bir Yaklaşım

Bölüm 3 de şimdiye kadar açıklanan sezgisel algoritmaların bazıları asimetrik problemlere de uygulanabilmesine rağmen, tamamı simetrik TSP'ler için geliştirilmişlerdir. Akl (1978), Frieze (1982), Karp (1979), Van Der Cruyssen ve Rijckaert (1978), Kanellakis ve Papadimitriou (1980) vb. bir çok araştırmacı özellikle asimetrik TSP ler için algoritmalar geliştirmişlerdir. Ancak burada sadece Akl (1978) yaklaşımı açıklanmaktadır.<sup>34</sup>

Akl (1978) tarafından asimetrik TSP ler için geliştirilmiş olan çözüm metodu, Christofides sezgiseli ile yakın olarak ilişkilidir. Ancak Christofides sezgiselinin aksine, burada minimal yönlü tarama ağacı (MDSG) belirlenmektedir. Bir MDSG,  $W(i, j)$  ağırlıklarına sahip olan bir tamamen

---

<sup>34</sup> S. AKL, "An analysis of various aspects of the travelling salesman problem", Ph. D. Thesis, McGill University, 1978.

yönlü grafiğin  $n$  adet düğümünü tarayan minimum ağırlığa sahip olan ve yönler ihmal edildiklerinde bir ağaç oluşturan bir alt grafiktir. Bir başka ifade ile yönler ihmal edildikleri zaman grafik,  $l_{ij} = \min\{w_{ij}, w_{ji}\}$  kanal uzunluklarına sahip olan bir minimal tarama ağacıdır.

G grafiği üzerindeki bir yönlü TSP için, yaklaşık olarak optimal bir çözüm bulan algoritmanın adımları aşağıda özetlenmektedir.

Adım 1. G nin bir MDSG sini bul.

Adım 2. MDSG yi yönlü grafiğe dönüştürebilecek bir kanallar seti ilave et.

Adım 3. Bu yönlü grafikte bir Euler dairesi bul.

Adım 4. Euler dairesini bir Hamilton dairesine dönüştür.

Adım 1,  $l_{ij} = \min\{w_{ij}, w_{ji}\}$  kanal maliyetlerine sahip olan yönsüz G grafiği üzerinde bir minimal tarama ağacı (MST) çözümünü gerektirmektedir. Ortaya çıkan MST de bulunan bir  $(i, j)$  kanalı için  $l_{ij} = w_{ij}$  olması halinde bu kanala  $(i, j)$  yönü ve  $l_{ij} = w_{ji}$  olması halinde ise  $(j, i)$  yönü verilmektedir.

Adım 2 de, düğümler kanal girişleri ve çıkışları yönünden değerlendirilerek, girişlerin daha fazla olduğu düğümler "arz noktaları" ve çıkışların daha fazla olduğu düğümler ise "talep noktaları" gibi alınarak grafiği dengelemek için bir transport problemi çözülmektedir. Adım 3 ve 4 de daha önce Christofides sezgiselinde açıklanmış olan yöntemlerin aynıları uygulanmaktadır. Adım 4 de, Euler dairesinden elde edilebilecek olan bütün Hamilton dairelerinin oluşturulmaları için her bir düğümün bir kez başlangıç düğümü olarak alınması ve standart algoritmanın uygulanması yeterli olmaktadır.

### 3.1.2.5. Zaman Kısıtlı TSP İçin Sezgisel Yaklaşımlar

Zaman kısıtlı TSP, her bir sıralı  $(i, j)$  düğüm çiftinin bir net kazanca ve bir seyahat süresine sahip olduğu çok daha gerçekçi bir problem ifadesidir. Bu problemde amaç,  $\tau$  birim zamanı aşmaksızın kazancı maximize eden ve orijinden (düğüm 1) başlayıp aynı yerde sona eren bir alt tur bulmaktır. Cloonan (1966), Gensch (1978) ve Golden (1981) tarafından ele alınmış olan zaman kısıtlı TSP için Bodin (1983) tarafından geliştirilen sezgisel çözüm metodu aşağıda açıklanmaktadır.<sup>35</sup>

$P=[p(i,j)]$  net kazanç matrisi ve  $t=[t(i,j)]$  zaman matrisi olmak üzere, her bir iterasyon  $I$  de oluşturulan alt turun toplam kazancı ve toplam zamanı, sırası ile  $P$  ve  $T$  ile gösterilmektedir. Ayrıca  $\Delta P$  ve  $\Delta T$ , alt tura her bir düğüm girişine dair toplam kazanç ve toplam zaman değişimleridir. Söz konusu metoddaki temel düşünce, bir düğüm girişinin cazipliğinin bir ölçüsü olarak  $\Delta p/\Delta T$  oranını kullanmaktır. Ancak  $\Delta P$  ve  $\Delta T$  negatif olabilecekleri için burada alternatif bir yaklaşım kabul edilmektedir.

Yöntemin adımlarını özetlemeden önce birkaç noktanın açıklanmasında fayda vardır.  $P/T$  oranı, mevcut alt turda birim zamanın değeridir.  $R_i$ , iterasyon  $I$  de bir birim zamanın değerinin en iyi tahminidir. Yani  $R_i$ , daha önceki bütün  $P/T$  oranlarını dikkate alan fakat daha sonrakilere daha fazla ağırlık veren bir tahmindir.

$R_0$ , optimal tur için kazancın zamana oranının bir tahminidir ve muhtemelen bir ön analize dayanmaktadır.  $R_i$  ifadesi, çeşitli tahmin koşullarında kullanılan standart üstel düzgünleştirme modelidir. Son olarak, esnekliği artırmak ve bir çok sezgisel çözüm oluşturmak için  $\alpha$ , 0 ile  $I$  arasında değişebilmekte ve  $R_0$  için farklı başlangıç değerleri seçilebilmektedir. Bu bilgilerden sonra, zaman kısıtlı TSP için sezgisel çözüm algoritması aşağıdaki gibi özetlenebilmektedir.

---

<sup>35</sup> Bodin ve diğerleri, 1983, Agm., S. 95.

Adım 1.  $l$ ,  $p$  ve  $t$  yi sıfırla.  $R_l$  ye başlangıç değeri ver ve  $\alpha$ 'yı seç.

Adım 2. Her bir düğüm  $i$  için,

$$\Delta T = t(l, i) + t(i, l) \quad (3.18)$$

ve

$$\Delta P - R_o \Delta T = \{p(l, i) + p(i, l) - R_o \{t(1, i) + t(i, 1)\}\} \quad (3.19)$$

hesapla.  $\Delta T \leq \tau$  ve  $\Delta P - R_o \Delta T$  nin maximum olduğu düğümü ( $i^*$ ) bul. Eğer böyle bir düğüm mevcut değil ise, dur. Aksi halde, alt tur  $1-i^*-1$  i oluştur ve kaydet.  $P = P + \Delta P$  ve  $T = T + \Delta T$  yap.

Adım 3.  $l = l + 1$

$$R_l = \alpha(P/T) + (1-\alpha)R_{l-1} \quad (3.20)$$

Hesapla.

Adım 4. Alt turda bulunan komşu düğümler ( $i$  ve  $j$ ) ve alt turda bulunmayan her bir düğüm ( $k$ ) için,

$$\Delta T = t(i, k) + t(k, j) - t(i, j) \quad (3.21)$$

ve

$$\Delta P - R_l \Delta T = \{p(i, k) + p(k, j) - p(i, j)\} - R_l \{t(i, k) + t(k, j) - t(i, j)\} \quad (3.22)$$

hesapla.  $T + \Delta T \leq \tau$  ve  $\Delta P - R_l \Delta T$  nin maximum olduğu  $i, k, j$  üçlüsünü ( $i^*, k^*, j^*$ ) bul. Eğer böyle bir üçlü mevcut değil ise, Adım 6'ya git. Aksi halde,  $k^*$  ı alt turdaki  $i^*$  ile  $j^*$  arasına sok ve alt turu kaydet  $P = P + \Delta P$  ve  $T = T + \Delta T$  yap.

Adım 5. Adım 3'e git.

Adım 6. Kaydedilen bütün alt turlar arasından en büyük P değerine sahip olanını seç.

Düğümlerden geçen minimum uzunluktaki Hamilton turunun arandığı,  $l(i, j)$  kanal uzunluklarına sahip olan bir geleneksel TSP de,  $p(i, j)=T-l(i, j)$  ve  $t(i, j)=0$  (Bütün  $i$  ve  $j$  için) ve  $\tau \leq \max\{l(i, j)\}$  alınarak, yukarıdaki algoritma ile yaklaşık optimal bir çözüm elde edilebilmektedir. Bu durumda sezgisel, iyi bilinen en ucuz giriş algoritmasına dönüşmektedir. Elde edilen turda her bir düğümün yer almasını sağlamak için  $l(i, j)$  den  $p(i, j)$  ye bir dönüşüm gerekli olmaktadır.

### 3.1.2.6. TSP Tur Uzunluğunun Tahmin Edilmesi

Chien (1992) tarafından 5-30 müşteri içeren bir TSP turunun uzunluğu için hassas tahminler üretebilen iki tahminci geliştirilmiştir.<sup>36</sup> Söz konusu tahminciiler aşağıdaki gibidir.

$$EST1 = 2.10 \times \bar{r} + 0.67 \times (Txn)^{1/2} \quad (3.23)$$

$$EST2 = 2.05 \times \bar{l} + 0.60 \times n/\rho^{1/2} \quad (3.24)$$

$\bar{r}$  : müşteri yerleşimleri ile depo arasındaki uzaklıkların ortalaması

$T$  : Sadece müşterileri kapsayan en küçük dörtgenin alanı

$n$  : müşterilerin sayısı

$\bar{l}$  : müşteri yerleşimleri ile depo arasındaki en kısa ve en uzun mesafelerin ortalaması

$\rho$  : bütün hizmet bölgesinin alansal yoğunluğu ( $n+t/hizmet$  bölgesinin alanı).

<sup>36</sup> T.W. CHIEN, "Operational estimators for length of travelling salesman tour", Computer and Operations Research 19, 1992, S. 469-478.

EST1 ve EST2'nin fonksiyonel formları, Monte-Carlo denemeleri ile oluşturulan 4160 adet veri seti üzerinde, çoklu regresyon analizi kullanılarak doğrulanmıştır.

Kullanılan sabit katsayılar, 4160 adet veri seti ile tahmincilerin ortalama mutlak hata yüzdelerini minimize etmiş olan katsayılardır. Bu iki tahmincinin ortalama tahmin hatası %7 civarındadır.

### **3.2. Tek Depo/Çok Araç Rotalama Problemi**

Tek depo/çok araç rotalama problemi, adından da anlaşıldığı gibi, bir tek depoda üslenen bir araçlar filosu ile farklı yerleşimlere ve bilinen hizmet ihtiyaçlarına ya da talep miktarlarına sahip olan bir çok müşterinin ihtiyaçlarının minimum maliyet ile karşılanması problemidir. Filodaki her bir aracın sabit bir kapasiteye sahip olması, probleme uzaklık veya zaman kısıtının yanısıra ilave bir kapasite kısıtı yüklemektedir. Söz konusu kısıtlar nedeni ile bütün müşterilere bir tek araç ile hizmet verilmesi mümkün değildir. Bu durumda problem, her bir araç için kapasite ve uzaklık veya zaman kısıtlarına uygun olarak bütün müşteri taleplerinin karşılanması amacı ile, bütün araçlar için toplam seyahat mesafesini veya süresini minimize edecek şekilde merkez depoda başlayan ve aynı yerde sona eren rotaların ve bu rotadaki yolların belirlenmesi problemidir.

#### **3.2.1. Katlı TSP**

Katlı seyyar satıcı problemi, TSP nin birden fazla satıcının (araç) söz konusu olduğu ve gerçek yaşam problemlerine daha uygun olan bir genellemesidir.

Katlı TSP de  $M$  adet satıcı  $n$  adet düğümden oluşan belirli bir şebekenin düğümlerini, bütün satıcılar tarafından kapsanan toplam seyahat uzaklığı minimum olacak şekilde ziyaret etmek zorundadır. Her bir satıcı, düğümlerin ortak bir depo içeren bir alt seti boyunca seyahat etmelidir ve depo hariç her düğüm kesinlikle bir satıcı tarafından ve kesinlikle bir kere ziyaret edilmelidir. MTSP nin aşağıda sunulan

matematik programlama formülasyonu, TSP'nin tahsis-esaslı formülasyonunun doğal bir uzanımıdır.<sup>37</sup>

$$\text{Minimize} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (3.25)$$

$$\sum_{i=1}^n x_{ij} = b_j = \begin{cases} M, & \text{eğer } j = 1 \text{ ise} \\ 1, & \text{eğer } j = 2, 3, \dots, n \text{ ise} \end{cases} \quad (3.26)$$

$$\sum_{j=1}^n x_{ij} = a_i = \begin{cases} M, & \text{eğer } i = 1 \text{ ise} \\ 1, & \text{eğer } i = 2, 3, \dots, n \text{ ise} \end{cases} \quad (3.27)$$

$$x = (x_{ij}) \in S \quad (3.28)$$

$$x_{ij} = 0 \text{ veya } 1 \quad (i, j = 1, \dots, n) \quad (3.29)$$

(3.28) de görülen S, daha önce TSP formülasyonundaki S setine benzer şekilde belirlenmekte, ancak TSP de bütün alt turları elimine etmek amacını taşıırken burada sadece depoyu içermeyen alt turları elimine etmek için kullanılmaktadır. Ayrıca kısıt (3.26), M adet satıcının hepsinin kullanılmasını gerektirmektedir.

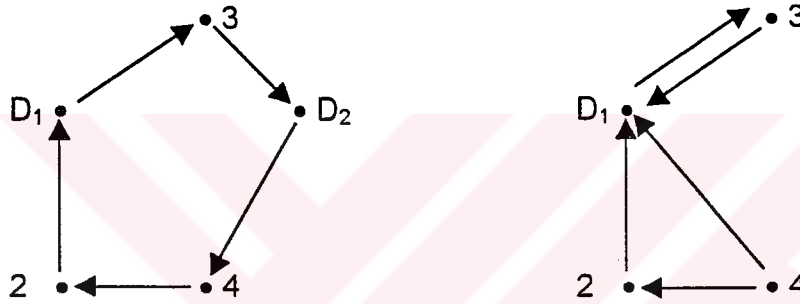
MTSP, orijinal deponun düğümlere olan uzaklıkları alınarak, her biri bu uzaklıklar ile diğer düğümlere bağlanan M adet kopya depo ( $D_1, D_2, \dots, D_M$ ) oluşturularak, denk bir TSP ye dönüştürülebilmektedir. M adet kopya depo birbirine bağlanmış olmayabilir veya her biri  $\sum_{i=1}^n \sum_{j=1}^n |c_{ij}|$  yi aşmayan bir uzunluğa sahip olan kanallar ile birbirlerine bağlı olabilir. Bu yol ile genişletilmiş şebekede ( $D_1, D_2, \dots, D_M, 2, 3, \dots, n$  düğümler seti üzerinde) bir optimal TSP turu hiçbir zaman iki kopya depoyu birleştiren bir kanalı kullanmayacaktır.<sup>38</sup> Bu aşamadan sonra kopya depolar tekrar bir tek düğüm halinde birleştirilerek tek satıcıdan ibaret olan TSP turu, MTSP nin gerektirdiği gibi M adet alt tura bölünmektedir.

<sup>37</sup> Bodin ve diğerleri, 1983, Agm., S.95.

<sup>38</sup> Bodin ve diğerleri, 1983, Agm, S.96.

Şekil 3.6'da iki satıcı ve dört düğümlü bir problem verilmektedir. Orijinal olarak, düğüm 1, 2, 3 ve 4 mevcuttur ve düğüm 1 depoyu göstermektedir. Genişletilmiş şebeke,  $D_1$ ,  $D_2$ , 2, 3 ve 4 düğümlerinden ibarettir ve  $D_1$  ve  $D_2$  düğümleri iki satıcıyı temsil etmektedir. Şekil 3.6'da genişletilmiş şebekedeki  $D_1$ -3- $D_2$ -4-2- $D_1$  turunun iki satıcı durumundaki 1-3-1 ve 1-4-2-1 alt turlarına denk olduğu görülmektedir.

Yukarıda özetlenen dönüşüm bu nedenle, daha önce açıklanan TSP algoritmalarından herhangi birini kullanarak MTSP leri çözme imkanı vermektedir. Bununla birlikte MTSP leri bir kısıt gevşetme yaklaşımı kullanarak direkt olarak çözmek daha elverişli ve etkindir.<sup>39</sup>



Şekil 3.6. İki-satıcı problemi

### 3.2.2. Tek Depo/Çok Araç Rotalama Problemi (VRP) Formülasyonları

Bölüm 3.2'nin girişinde tanımlanan standart araç rotalama problemi (VRP) için bir çok alternatif formülasyonlar geliştirilmiş olmakla birlikte, bizim çalışmamızda Golden, Magnanti, Nguyen (1977) tarafından geliştirilmiş ve Bodin, Golden, Assad, Ball (1983) tarafından da benimsenmiş olan formülasyon tercih edilmektedir.<sup>40</sup>

Aşağıda sunulan formülasyon rota süresi kısıtlarını da içeren genel bir yapıda olmasına karşın, bu kısıtların ihmal edilmeleri problemi kolayca

<sup>39</sup> Bodin ve diğerleri, 1983, Agm., S. 96.

<sup>40</sup> B. GOLDEN, T. MAGNANTI, H. NGUYEN, "Implementing vehicle routing algorithms", Networks 7, 1977, S. 113-148.

standart araç rotalama problemine (VRP) dönüştürmektedir. Problem saf dağıtım problemidir. Burada "dağıtım" terimi, merkez depodan alınan belirli miktardaki mal veya insanın belirli noktalara teslim edilmeleri anlamını ifade etmektedir. Çalışmamızda "toplama" olarak kullanılan bir diğer terim ise belirli noktalardan belirli miktarda mal veya insanın teslim alınması anlamına gelmektedir. Araç rotalama probleminin sadece toplama problemi olması durumunda, dağıtım problemine denk bir problem ortaya çıkmakta fakat hem dağıtım hem de toplamanın söz konusu olduğu problemler çok daha kompleks bir yapı sergilemektedirler.

$$\text{Minimize } \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^{NK} c_{ij} x_{ij}^k \quad (3.30)$$

$$\sum_{i=1}^N \sum_{k=1}^{NK} x_{ij}^k = 1 \quad (j=2,3,\dots,N) \quad (3.31)$$

$$\sum_{j=1}^N \sum_{k=1}^{NK} x_{ij}^k = 1 \quad (i=2,3,\dots,N) \quad (3.32)$$

$$\sum_{i=1}^N x_{ip}^k - \sum_{j=1}^N x_{pj}^k = 0 \quad (k=1,2,\dots,NK; p=1,2,\dots,N) \quad (3.33)$$

$$\sum_{i=1}^N d_i \left( \sum_{j=1}^N x_{ij}^k \right) \leq Q_k \quad (k=1,2,\dots,NK) \quad (3.34)$$

$$\sum_{i=1}^N t_i^k \sum_{j=1}^N x_{ij}^k + \sum_{i=1}^N t_j^k x_{ij}^k \leq T_k \quad (k=1,2,\dots,NK) \quad (3.35)$$

$$\sum_{j=2}^N x_{1j}^k \leq 1 \quad (k=1,2,\dots,NK) \quad (3.36)$$

$$\sum_{i=1}^N x_{i1}^k \leq 1 \quad (k=1,2,\dots,NK) \quad (3.37)$$

$$X \in S \quad (3.38)$$

$$x_{ij}^k = 0 \text{ veya } 1 \quad (\text{bütün } i, j, k\text{'lar için}) \quad (3.39)$$

Yukarıdaki formülasyonda;  $N$  : düğümlerin sayısı,  $NK$  : araçların sayısı,  $Q_k$ :  $k$  aracının kapasitesi,  $T_k$  :  $k$  aracının bir rota için izin verilen maximum seyahat süresi,  $d_i$  :  $i$  düğümündeki talep ( $d_1=0$ ),  $t_i^k$  :  $k$  aracının  $i$  düğümünde dağıtım veya toplama yapması için gerekli olan süre ( $t_1^k = 0$ ),  $t_{ij}^k$  :  $k$  aracının  $i$  düğümünden  $j$  düğümüne seyahat süresi,  $c_{ij}$  :  $i$  düğümünden  $j$  düğümüne seyahatin maliyeti,  $x$  : araç tipini dikkate almaksızın düğümler arasındaki bağlantıları ifade eden ve  $x_{ij} = \sum_{k=1}^{NK} x_{ij}^k$  bileşenlerinden oluşan matris ve

$$x_{ij}^k = \begin{cases} 1, & \text{eğer kanal } i, j, k \text{ aracı ile katedilirse} \\ 0, & \text{aksi halde} \end{cases}$$

olarak tanımlanmaktadır.

Eşitlik (3.30), toplam seyahat mesafesinin minimize edilmesini ifade etmektedir. Alternatif olarak  $c_{ij}$  yerine, araç tipine bağlı olan bir maliyet katsayısı ( $c_{ij}^k$ ) yerleştirilerek maliyetler minimize edilebilir. Eşitlik (3.31) ve (3.32), her bir talep düğümüne kesinlikle bir araç tarafından hizmet verilmesini temin etmektedirler. Eşitlik (3.33), rota sürekliliğini yani bir talep düğümüne giren bir aracın bu düğümden çıkması zorunluluğunu ifade etmektedir. Eşitlik (3.34), araç kapasitesi kısıtı ve benzer olarak (3.35) toplam rota süresi kısıtıdır. Örneğin bir gazete dağıtım kamyonu, bir turda 1 saatten daha fazla zaman harcamamak üzere kısıtlanabilir. Eşitlik (3.36) ve (3.37), mevcut araç sayısının aşılmamasını temin etmektedirler. Son olarak (3.38), alt tur eliminasyon kısıtıdır ve biraz değişiklik ile Bölüm 3.1.1. de belirlenen eşitliklerden herhangi birisi olabilir. (3.31) ve (3.33) zaten (3.32) yi ve (3.33) ve (3.36) da (3.37) yi kapsadıkları için (3.32) ve (3.37)'nin silinmeleri, modeli etkilemeyecektir. Modelde  $\max_{1 \leq i \leq N} d_i < \min_{1 \leq k \leq NK} Q_k$  olduğu yani her bir düğümdeki talebin herhangi bir aracın kapasitesini aşmadığı ve bir talep düğümüne hizmet verilirken düğümün ihtiyacının

tam olarak karşılandığı yani her bir düğüm talebinin bir seferde ve bir tek araç ile karşılandığı varsayılmaktadır.

Formülasyonda araç kapasitesi kısıtı (3.34) ve rota süresi kısıtı (3.35) ihmal edildikleri zaman, kısıt (3.33) elimine edilerek ve amaç fonksiyonuna  $x_{ij} = \sum_{k=1}^{NK} x_{ij}^k$  yerleştirilerek model bir MTSP ye dönüştürülebilmektedir. Ayrıca model incelendiğinde de görülebileceği gibi araç rotalama problemi (VRP), NK adet farklı TSP'nin çözümünü gerektirmektedir. Bu nedenle, TSP NP-hard olduğu için VRP de NP-hard dır ve bunun sonucu olarak VRP'leri çözmek konusundaki araştırmaların büyük bir kısmı, sezgisel metodlarla ilgilidir.

### 3.2.3. VRP İçin Seçilen Sezgisel Çözüm Teknikleri

Bölüm 2.4'de belirtilmiş olduğu gibi araç rotalama problemini dal ve sınır teknikleri ile optimal olarak çözen teknikler mevcuttur. Ancak bu teknikler sadece çok küçük problemleri çözebildikleri için bizim çalışmamızda sezgisel algoritmalar ele alınmaktadır. Aşağıda 1000 müşterili problemleri etkin olarak çözmüş olan bir çok sezgisel metod açıklanmaya çalışılmaktadır.

Burada incelenen metodların hepsinde araç filosunun homojen olduğu kabul edilmektedir.

#### 3.2.3.1. Clarke-Wright Tasarruf Algoritması

Clarke ve Wright (1964) tarafından geliştirilmiş olan tasarruf algoritması, her bir adımda bir turlar setinin daha iyi bir set elde etmek üzere değiştirilmesine dayanmaktadır.<sup>41</sup> Bu nedenle Clarke-Wright tasarruf algoritması bir değişim yöntemidir.

---

<sup>41</sup> Clarke, Wright, Agm, S. 568-581.

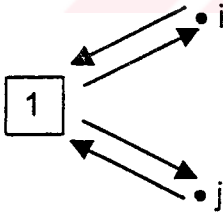
Yöntemde başlangıç olarak her bir araç için ayrı bir rota oluşturulmakta yani her bir talep noktasına ayrı bir araç ile hizmet verilmektedir. Sonra, oluşturulabilecek en büyük tasarrufa ve uygunluk şartlarına göre iki rota birleştirilmektedir. Burada ifade edilen tasarruflar,  $i$  ve  $j$  düğümlerine iki ayrı araç yerine bir tek araç ile hizmet verilmesi durumunda elde edilecek maliyet azalmasıdır.

Şekil 3.7 (a) da görüldüğü gibi,  $i$  ve  $j$  düğümlerine iki ayrı araç ile hizmet verilmesi durumunda, toplam seyahat mesafesi  $2c_{1i} + 2c_{1j}$  olarak hesaplanmaktadır. İki rota bir tek rota içersinde birleştirildiği yani  $i$  ve  $j$  düğümlerine bir tek araç ile hizmet verildiği zaman, Şekil 3.7 (b) de görüldüğü gibi toplam seyahat mesafesi  $c_{1i} + c_{ij} + c_{j1}$  olmakta ve bu dönüşüm ile sağlanan tasarruf aşağıdaki gibi bulunmaktadır.

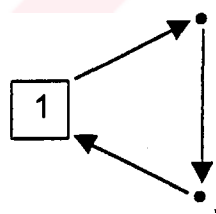
$$S_{ij} = 2c_{1i} + 2c_{1j} - (c_{1i} + c_{1j} + c_{ij}) = c_{1i} + c_{1j} - c_{ij} \quad (3.40)$$

Yukarıdaki tasarruf ifadesinde maliyetler simetriktir ( $c_{ij} = c_{ji}$ ). Maliyetler asimetric olduklarında tasarruf formülü biraz değişmektedir.

$$S_{ij} = c_{1i} + c_{1j} - c_{ij} \quad (3.41)$$



(a) Başlangıç oluşumu



(b) Düğüm  $i$  ve  $j$  birleştirilmiştir.

Şekil 3.7. Tur Birleştirme

Clarke-Wright tasarruf algoritmasının paralel ve ardışık olmak üzere iki versiyonu bulunmaktadır. Orijinali paralel versiyon olan algoritmanın uygulama adımları aşağıdaki gibi özetlenebilir.

Adım 1.  $(1, i, 1)$  ( $i=2,3,...,n$ ) şeklinde  $n-1$  adet araç rotası oluştur.  
Düğüm 1 depoyu göstermektedir.

Adım 2.  $S_{ij} = c_{i1} + c_{1j} - c_{ij}$  ( $i, j = 2, 3, ..., n$  ve  $i \neq j$ ) tasarruflarını hesapla.

Adım 3. Tasarrufları büyükten küçüğe doğru sıralayarak tasarruflar listesini oluştur.

Adım 4. Tasarruflar listesinin en üstünden başlayarak ve problemin kısıtlarına uygun olarak rotaları birbirlerine ekleyip genişlet. Bir rota artık genişletilemediği zaman, yeni bir rotaya başla. Bütün düğümler rotalara yerleşmiş olduğu zaman, dur.

Algoritmanın ardışık versiyonunda, bütün tasarruflar matrisini hesaplamak yerine, başlangıç rotası olarak keyfi bir düğüm seçilmekte ve her bir adımda rotaya sokulmuş olan en son düğüme dair en yüksek tasarrufa karşılık gelen bağ ilave edilmektedir. Ardışık versiyon, hesaplanacak tasarruf değerlerinin sayısını azalttığı için hesaplama açısından daha elverişlidir.

### 3.2.3.2. Paralel Tasarruf Esaslı Sezgisel Algoritma

Altinkemer ve Gavish (1991) tarafından Clarke-Wright tasarruf sezgiseline dayanılarak geliştirilmiş olan paralel tasarruf sezgiselinin orijinal tasarruf yaklaşımından ana farklılığı şöyle açıklanabilir.<sup>42</sup> Clarke-wright tasarruf yaklaşımında bir tek tur ele alınarak genişletilmeye çalışılmakta ve turlar art arda oluşturulmaktadır. Bunun aksine burada, her bir adımda bir çok kısmi tur aynı anda genişletilmeye çalışılmaktadır.

---

<sup>42</sup> K. ALTINKEMER, B. GAVISH, "Parallel savings based heuristics for the delivery problem", Operations Research 39 (3), 1991, S. 456-469.

Paralel tasarruf sezgiselleri eşleştirme (matching) esaslı sezgisellerdir ve her bir iterasyonda bir çok düğüm kümesi birbirleri ile eşleştirilmektedir. Her bir iterasyonda eşleştirilecek düğümler, o iterasyonda elde edilen tasarrufları maximize etmeyi amaçlayan bir eşleştirme problemi çözülerek belirlenmektedir.

Buradaki "eşleştirme" terimi, düğümlerin ikişer ikişer eşleştirilerek birbirlerine bağlanmalarını ifade etmektedir ve bu problemin çözülmesi için alternatif algoritmalar mevcuttur. Altinkemer ve Gavish (1991), eşleştirme probleminin hangi algoritma ile çözüleceği konusunda herhangi bir açıklamada bulunmamışlardır. Bu nedenle herhangi bir algoritma kullanılabilmektedir.

Altinkemer ve Gavish (1991)'deki paralel tasarruf sezgiselinde kullanılan maliyet matrisi simetriktir, üçgen eşitsizliği geçerlidir ve kullanılan rotasyon aşağıdaki gibidir.

$d_i =$  i düğümünde talep edilen yük miktarı

$R_p =$  p aracına tahsis edilen düğümlerin seti

$C(R_p) =$  Düğüm 0 (depo) ve  $R_p$  üzerindeki optimum TSP turunun maliyeti

$S_{pq} =$  p ve q kümelerini birleştirmekten kaynaklanan tasarruf

$p_r =$  p kümesindeki turun ilk uç noktası

$p_l =$  p kümesindeki turun son uç noktası

Algoritmanın birinci iterasyonunda her bir küme, orijinal grafiğin bir tek düğümünden ibarettir. Bu kümeler birleştirildikleri zaman yeni kümedeki bütün düğümlere bir tek araç ile hizmet verilmektedir. İki küme kapasite kısıtına uygun olarak birleştirildiği zaman elde edilecek tasarrufları hesaplamak için her bir iterasyonda bir TSP çözmek yerine tasarrufların aşağıda verilen tahminleri kullanılmaktadır.

$$S_{pq} = \max \begin{cases} c_{1p_r} + c_{1q_r} - c_{p_r q_r} \\ c_{1p_r} + c_{1q_l} - c_{p_r q_l} \\ c_{1p_l} + c_{1q_l} - c_{p_l q_l} \\ c_{1p_l} + c_{1q_r} - c_{p_l q_r} \end{cases} \quad (3.42)$$

Yukarıdaki formül, p ve q kümelerini birleştirerek elde edilen tasarrufları gerçek değerlerinin altında tahmin etmektedir. Formül (3.42), birleştirilmiş kümedeki turun, p ve q kümelerindeki turların uç noktalarından birleştirilmiş bir kombinasyonu olacağını varsaymaktadır. Bu varsayıma göre, birleştirilmiş turdaki düğümlerin ziyaret sıralaması değişmemektedir. TSP çözmek yerine bu tahminlerin kullanılmalarının tek nedeni, bir poynomyal algoritma elde edebilmektir. Bu nedenle problemin ve dolayısıyla oluşturulan kısmi turların küçük olmaları halinde, bu turların TSP lerin çözülerek belirlenmelerinin daha doğru olacağı söylenebilir. Bu durumda tasarruf formülü aşağıdaki gibi ifade edilmektedir.

$$S_{pq} = c(R_p) + c(R_q) - c(R_q \cup R_p) \quad (3.43)$$

Tasarrufların hesaplanmalarından sonraki aşamada, bu tasarruf değerleri kullanılarak bir maximum ağırlıklı eşleştirme problemi çözülmekte ve problemin kısıtlarına uygun olarak kümeler eşleştirilerek yeni kümeler oluşturulmaktadır. Aşağıda bu algoritmanın temel bir ifadesi verilmektedir.

Adım 1:  $R_p = \{i\}$ , ( $p=i=1, \dots, n$ ) şeklinde n adet küme oluştur.

Adım 2:  $S_{pq} = c(R_p) + c(R_q) - c(R_p \cup R_q)$ , ( $p=2, 3, \dots, n-1$ ,  $q=3, 4, \dots, n$ ;  $p \neq q$ ) tasarruflarını hesapla.

Adım 3: Ağırlıklar olarak  $S_{pq}$ 'ları alarak, bir maximum ağırlıklı eşleştirme problemini çöz.

Adım 4: Eşleştirme problemi p-q eşleşmesini vermiş ise ve bu iki kümedeki toplam yük miktarı araç kapasitesini aşmıyorsa, p ve q kümelerini birleştir.  $P < q$  olmak üzere,  $R_p = p \cup q$  yap. Bütün böyle eşleşmeleri yaptıktan sonra, Adım 2 ye git.

Eğer hiçbir uygun eşleşme mümkün değil ise, dur ve elde etmiş olduğun rotaları kaydet.

### 3.2.3.3. Süpürme Yaklaşımı

Gillett ve Miller (1974) tarafından geliştirilmiş olan ve 250 düğümlüye kadar ki problemleri elverişli olarak çözebilen süpürme algoritması, çözümü iki ayrı aşamada kurmaktadır.<sup>43</sup> Birinci aşamada, yerleşimler çok sayıda rotaya bölünmekte yani müşteriler araçlara tahsis edilmektedir. İkinci aşamada ise her bir araç için ziyaret sıralarını belirlemek amacı ile bir TSP algoritması uygulanmaktadır. Gillett ve Miller (1974)'de ikinci aşama için bir değişim sezgiseli önerilmektedir.

Süpürme yaklaşımında problem, daha küçük alt problemlere bölündüğü için hesaplama süresi, yerleşim (düğüm) sayısı ile üstel olarak değil lineer olarak artmakta ve bunun bir sonucu olarak çok sayıda yerleşime sahip olan problemler optimale yakın olarak çözülebilmektedir.

Tek depolu homojen araç rotalama problemleri için geliştirilmiş olan süpürme algoritmasında kullanılan rotasyon aşağıdaki gibidir.

$N$  : depoyu da içeren yerleşim sayısı (depo, yerleşim 1 ile gösterilmektedir.)

$d_i$  :  $i$  talep noktasındaki talep miktarı ( $i=2,3,...,N$ )

$(x_i, y_i)$  :  $i$ 'yinci yerleşimin dik koordinatları ( $i=2,3,...,N$ )

$Q$  : her bir aracın kapasitesi (araçlar homojendirler)

$T$  : herbir araç için kabul edilen maximum seyahat süresi

$c(i, j)$  :  $i$  ve  $j$  yerleşimleri arasındaki uzaklık

---

<sup>43</sup> B. GILLETT, L. MILLER; "A heuristic algorithm for the vehicle dispatch problem", Operations Research 22, 1974, S. 340-349.

$An_i$  : i.yinci yerleşimin kutupsal koordinat açısı ( $i=2,3,...,N$ )

$$An_i = \arctg [(y_i - y_1)/(x_i - x_1)] \quad (3.44)$$

Eğer  $y_i - y_1 \leq 0$  ise  $-\pi < An_i < 0$  ve eğer  $y_i - y_1 \geq 0$  ise  $0 \leq An_i \leq \pi$  dir.

$R_i$  : depodan i yerleşimine uzaklık (yarıçap,  $i=2,3,...,N$ )

Problemin kısıtları ise aşağıdaki gibidir.

$$d_i \leq Q, \quad \text{bütün } i\text{'ler için} \quad (3.45)$$

$$c(i,j) > 0, \quad \text{bütün } i \neq j\text{'ler için} \quad (3.46)$$

$$c(i,i) = 0 \quad (3.47)$$

$$c(i,1) + c(1,i) \leq T, \quad \text{bütün } i\text{'ler için} \quad (3.48)$$

Algoritmanın birinci aşamasında önce yerleşimlerin kutupsal koordinat açıları belirlenerek, küçüktен büyüğe numaralandırılmaktadır. Yani  $An_i < An_{i+1}$ 'dir (bütün i'ler için). Aynı kutupsal koordinat açısına sahip olan iki veya daha fazla yerleşim bulunması durumunda, yarıçapı (depodan uzaklığı) küçük olan yerleşim daha önce alınmaktadır. Yani eğer  $An_i = An_j$  ve  $c(1,i) < c(1,j)$  ise  $i < j$  dir.

Süpürme algoritması, ileriye doğru süpürme ve geriye doğru süpürme olmak üzere iki kısımdan oluşmaktadır. Algoritmanın ileriye doğru kısmında, yerleşimler en küçük açılı olandan yani yerleşim 2 den başlanarak ve yerleşim numaraları izlenerek rotalara bölünmektedir. Yerleşimler, rotalara bölünürlerken tabi ki kapasite ve uzaklık (veya zaman) kısıtlarına uygunluk gözetilmektedir. Bu durumda birinci rota 2,3,...,j olmaktadır. Bu rotadaki j yerleşimi, kapasite ve/veya uzaklık kısıtlarını aşmadan rotaya eklenebilecek son yerleşimdir. Aynı şekilde ikinci rota, j+1, j+2,...,L olmakta ve bütün rotalar bu yol ile oluşturulmaktadır. Bu durumda toplam seyahat uzunluğu basit olarak her bir rota için seyahat uzunluklarının toplamına eşittir.

Her bir yerleşim bir rotaya sokulduktan sonra  $k$  rotasındaki bir yerleşim yerine  $k+1$  rotasındaki bir veya daha fazla yerleşimi  $k$  rotasına sokarak toplam rota uzunluğunda bir azalma yapıp yapılamayacağı araştırılmaktadır ( $k=1,2,...,NK-1$ ). Bu yerleşim (düğüm) değiştirme metodunun işleyişi aşağıdaki gibidir.

$k$  rotasından silinecek yerleşim,  $A_{i_1}$  ve  $R_{i_1}$  nin bir fonksiyonu minimize edilerek belirlenmektedir. Burada çok iyi işleyen bir fonksiyon,  $R_{i_1}+A_{i_1}AVR$  dir. Fonksiyondaki AVR, bütün yerleşimlerin depoya uzaklıklarının ortalamasıdır ve bu fonksiyon hem depoya hem de bir sonraki rotaya yakın olan bir yerleşimi seçmektedir.

$k$  rotasına dahil edilmek üzere ele alınan birinci yerleşim ( $p$ ),  $k+1$  rotasında bulunan ve  $k$  rotasına sokulmuş en son yerleşime en yakın olan yerleşimdir.  $k$  rotasına sokulmak üzere ele alınan ikinci yerleşim  $k+1$  rotasında  $p$  ye en yakın yerleşimdir. Eğer bu plan ile  $k$  rotasına bir veya daha fazla yerleşim sokulursa,  $k+1$  rotasındaki bir sonraki yerleşimin de  $k$  rotasına sokulup sokulamayacağı kontrol edilmektedir.  $k+1$  rotasında  $k$  rotasına sokulabilecek en son yerleşim  $j$  yerleşimi oluncaya kadar bu prosese devam edilmekte ve sonra  $j+2$  yerleşiminin  $k$  rotasına sokulup sokulamayacağı kontrol edilmektedir. Yerleşimlerin bu şekilde seçilmeleri, en iyi çözümü vermese bile çok hızlı bir seçim planıdır ve iyi sonuçlar üretmektedir.

Bir veya daha fazla yerleşimi  $k$  ( $k=1,2,...,NK$ ) rotasına sokma ve bir diğer yerleşimi silme sürecine iyileşme bulunamayınca kadar devam edilmektedir. Sonra birinci yerleşim son ve ikinci yerleşim ilk yerleşim olacak şekilde  $X$  ve  $Y$  eksenleri ters çevrilmekte ve rotalar arasındaki yerleşim değiştirme süreci tekrarlanmaktadır. Bu sürecin sonunda gene bir minimum toplam uzunluk hesaplanmakta, bütün olasılıklar tükeninceye kadar eksenlerin ters çevrilmeleri sürecine devam edilmekte ve hesaplanan minimum toplam uzunlukların en küçüğü araç rotalama problemi için iyi bir sezgisel çözüm vermektedir.

Geriye doğru süpürme olarak adlandırılan ikinci bir algoritma, rotaları ters yönde oluşturması hariç, ileriye doğru süpürme algoritmasının tamamen aynısıdır. Bu algortmada başlangıç olarak rota 1 yerleşim  $N$ ,  $N-1$ ,  $N-2, \dots, L$ 'yi rota 2 yerleşim  $L-1$ ,  $L-2, \dots, M$  yi v.b. içermektedir. Çoğu durumlarda bu iki algoritma, farklı rotalar ve bunun sonucu olarak farklı minimum toplam uzunluklar üretmektedir. Tabi ki bu iki minimum arasından küçük olanı, en iyi yaklaşık çözümdür. Aşağıda ileriye doğru süpürme algoritması adım adım açıklanmaktadır. Algoritmanın akışı sırasında eğer  $j$  yerleşimi belirli bir rota içerisinde ise ve  $j+1$  yerleşimi bu rotaya sokulamıyorsa  $j+2$  yerleşimi kontrol edilmemektedir. Yerleşimleri tekrar numaralandırma yerine en küçük açılı yerleşim  $K(2)$  ile, ikinci en küçük açılı yerleşim  $K(3)$  ile v.b. gösterilmektedir.

Adım 1. Her bir yerleşim için,  $(0, 0)$  daki depo ile kutupsal koordinatları belirle.  $i$  yerleşimi için  $A_{n_i}$  açığı ve  $R_i$  yarıçapı göstermektedir ( $i=2,3,\dots,N$ )

Adım 2. Her bir yerleşim için  $A_n(K(i)) \leq A_n(K(i+1))$  olarak  $K(i)$  leri belirle ( $i=2,3,\dots,n$ )

Adım 3.  $j=2$  ve  $TOP=q_{K(2)}$  ile birinci rotaya başla.

Adım 4.  $j=j+1$  ile açığı büyüt.

Adım 5. Eğer  $TOP + d_{K(j)} > Q$  ise, Adım 7 ye git.

Adım 6.  $TOP = TOP + d_{K(j)}$  yaparak rotayı  $K(j)$  yerleşimi ile büyüt. Eğer  $j=N$  ise, Adım 16 ya git. Eğer  $j \neq N$  ise, Adım 4 e git.

Adım 7. Bir TSP algoritması vasıtası ile rota için minimum uzunluğu ( $T_1$ ) hesapla. Uzaklık kısıtını kontrol et. Eğer uzaklık kısıtı aşılrısa  $K(j-1)$  i rotadan çıkart.  $TOP = TOP - d_{K(j-1)}$  ve  $j=j-1$  yap. Uzaklık kısıtını tekrar kontrol et. Uzaklık kısıtına uygunluk sağlanıncaya kadar bu adıma devam et.

Adım 8.  $K(j-1)$ 'e en yakın olan ve bir rotada bulunmayan yerleşimin sıra numarasını  $(jjx)$  belirle,  $K(jjx)$ 'e en yakın olan ve bir rotada bulunmayan yerleşimin sıra numarasını  $(jii)$  bul. Rotadaki bütün yerleşimler için,  $R_{K(i)} + A n_{K(i)} AVR$  fonksiyonunu minimize eden  $i$ 'yi belirle. Kii bu  $i$ 'yi göstere. Rotadan  $K(Kii)$ 'yi silip  $K(jjx)$ i dahil ederek minimum uzunluğu ( $T_2$ ) belirle.

Adım 9. Eğer  $T_2 \leq T$  ve yük kısıtına uygunluk sağlanmış ise, Adım 11'e git. Aksi halde Adım 10'a git.

Adım 10. Rotayı kaydet ve  $ToP = q_{K(i)}$  ile yeni bir rotaya başla. Adım 4'e git.

Adım 11. Depodan başlayıp  $K(j)$ ,  $K(j+1)$ , ...,  $K(j+4)$  yerleşimlerini ziyaret ederek  $K(j+5)$  de sona eren seyahat için minimum uzunluğu ( $T_3$ ) belirle.  $K(jjx)$  i çıkartıp  $K(Kii)$  yi dahil etme dışında aynı yerleşimlerden geçen seyahatin uzunluğunu ( $D_4$ ) belirle. Eğer  $K(jjx)$   $K(j)$ ,  $K(j+1)$ , ... veya  $K(j+4)$  değil ise, Adım 10'a git. Eğer  $T_1 + T_3 < T_2 + T_4$  ise Adım 13'e git. Aksi halde Adım 12'ye git.

Adım 12.  $K(jjx)$  i rotaya sok ve  $K(Kii)$  yi rotadan çıkart. Adım 4'e git.

Adım 13. Rotaya  $K(ii)$  yerine  $K(jjx)$  ve  $K(jii)$  yi sokma durumunda minimum rota uzunluğunu ( $T_5$ ) belirle. Eğer  $K(jjx)$  ve  $K(jii)$   $K(j)$ ,  $K(j+1)$ , ... veya  $K(j+4)$  değil ise, Adım 10'a git. Eğer  $T_5 < T$  ise ve yük kısıtı tatmin edilmiş ise, Adım 14'e git. Aksi halde Adım 10'a git.

Adım 14. Depodan başlayıp  $K(j)$ ,  $K(j+1)$ , ...,  $K(j+4)$  yerleşimlerinden geçerek  $K(j+5)$  de sona eren ve  $K(jjx)$  ve  $K(jii)$  yi çıkartıp  $K(Kii)$  yi sokarak gerçekleştirilen seyahat için minimum uzunluğu ( $D_6$ ) belirle. Eğer  $T_1 + T_3 < T_5 + T_6$  ise Adım 10'a git.

Adım 15. Rotadan  $K(K_{ii})$  yi çıkartıp  $K(jj_x)$  ve  $K(jj_i)$  yi ilave et. Adım 4 e git.

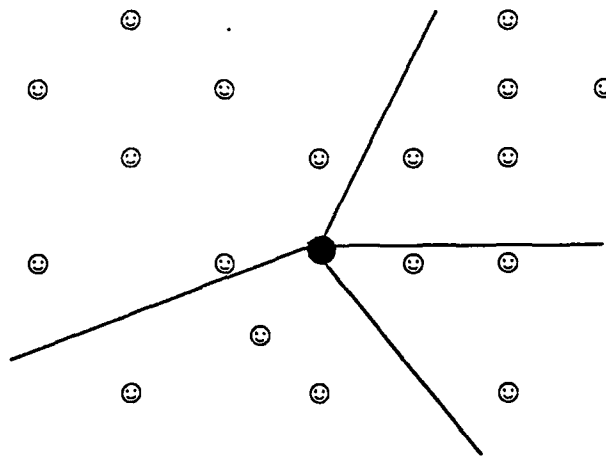
Adım 16. Rota için minimum uzunluğu belirle ve uzaklık kısıtını kontrol et. Eğer uzaklık kısıtı tatmin edilemiyorsa, Adım 17 ye git. Aksi halde bu rotalar seti tamamlanmıştır. Başka rotalar setine ihtiyaç olup olmadığını kontrol et. Eğer ihtiyaç yoksa Adım 19 a git. Aksi halde Adım 18 e git.

Adım 17. Bir yerleşimi rotadan sil ( $j=j-1$ ). Adım 10 a git.

Adım 18. Bir yerleşim ile açığı büyüt (yani ikinci rotalar setine  $K(3)$  ile başla). Adım 2 ye git.

Adım 19. DUR.

Gillett ve Miller (1974) süpürme yaklaşımı ile yerleşimlerin kümeler (rotalara) bölünmelerinde uygulanan yöntemin geometrik yorumu şöyledir. Rastgele olarak seçilen bir çekirdek düğüm ile depoyu birleştiren çizgi saat yönünde veya ters yönde yerleşimleri (düğümleri) süpürmektedir. Problemin kısıtlarına uyulduğu sürece düğümler bir rotaya ilave edilerek kısıtların bozulduğu noktada yeni bir rotaya başlanmaktadır. Şekil 3.8, bu yolla oluşturulmuş olan kümelerin bir örneğini göstermektedir.



Şekil 3.8. Süpürme prosedürü ile oluşturulan kümeler

### 3.2.3.4. Lagrangean Gevşetme Yaklaşımı

Stewart ve Golden (1984) tarafından geliştirilen lagrangean gevşetme yaklaşımı, kapasite kısıtlarını amaç fonksiyonuna taşımak ve bir rota üzerindeki talep kapasiteyi aştığı zaman bir ceza yüklemek amacı ile bir çarpan uygulama fikrine dayanmaktadır.<sup>44</sup> Bu şekilde ortaya çıkan problem, bir MTSP'yi andırmaktadır.

Yaklaşımın geliştirilmesinde araç filosunun homojen olduğu ve zaman kısıtlarının bulunmadığı varsayılmaktadır. Bu durumda en basit şekli ile klasik VRP formülasyonu aşağıdaki gibi olmaktadır.

$$\text{Minimize } \sum_k \sum_{i,j} c_{ij} x_{ij}^k \quad (3.49)$$

$$\sum_{i,j} d_i x_{ij}^k \leq Q \quad (k=1, \dots, NK) \quad (3.50)$$

$$X = [x_{ij}^k] \in S \quad (3.51)$$

Yukarıdaki formülasyonda kullanılan notasyon, bölüm 3.2.2'deki notasyonun aynısıdır. Lagrangean relaxation algoritmasında bu formülasyona aşağıdaki kısıt seti ilave edilmektedir.

$$\sum_{i,j} d_i x_{ij}^k \geq \sum_{i,j} d_i x_{ij}^l \quad (l=k+1 \text{ ve } k=1, 2, \dots, NK) \quad (3.52)$$

Kısıt (3.52)'nin anlamı, en büyük talepli rotaya  $k=1$ , ikinci büyük talepli rotaya  $k=2$  vb. tahsis etmek yani rotaları toplam talep miktarlarına göre büyükten küçüğe doğru sıralamaktır. Şimdi kısıt (3.50) amaç fonksiyonuna taşınarak VRP, gevşetilebilir. Bu durumda ortaya çıkan problem aşağıdaki gibi olmaktadır.

### LVRP

---

<sup>44</sup> W.R. STEWART, B.L. GOLDEN, "A Lagrangean relaxation heuristic for vehicle routing", European Journal of Operational Research 15, 1984, S. 84-88.

$$\text{Minimize } \sum_k \sum_{i,j} c_{ij} x_{ij}^k + \sum_k \lambda_k (\sum_{i,j} d_i x_{ij}^k - Q) \quad (3.53)$$

$$\sum_{i,j} d_i x_{ij}^k - \sum_{i,j} d_i x_{ij}^l \geq 0, \quad l=k+1 \text{ ve } k=1, 2, \dots, NK-1 \quad (3.54)$$

$$X \in S \quad (3.55)$$

LVRP formülasyonunda görülen  $\lambda_k \geq 0$  ( $k=1,2,\dots,NK$ ) dır ve  $k$  rotası üzerindeki araç kapasitesini aşan her birim talep için bir ceza olarak düşünülebilmektedir. Eğer  $\lambda_k$ ,  $\lambda_1 \geq \lambda_2 \dots \geq \lambda_{NK}$  şeklinde seçilirse, yüksek talepli rotalar daha yüksek cezalar ile karşılaşacaklardır. Bütün diğer  $\lambda$  lar sıfır ( $\lambda_k=0$ ) olmak üzere  $\lambda_1$  için büyük bir pozitif değer belirlenmesi durumunda, en büyük talepli rotaya yüksek bir ceza yüklenmiş olacaktır. Bu durumda mevcut talep miktarını azaltacak bir değişiklik, toplam maliyeti azaltırken uygun olmayan bir çözümü de uygun olmaya yöneltecektir. Burada sunulan algoritmanın temel dayanağı budur.

Lagrangean gevşetme sezgiselinde kullanılan ve LR3OPT olarak adlandırılan algoritmada, önce bir  $\lambda_1$  değeri ile başlanarak problem (3.53)-(3.55) çözülmektedir. Sonra her aşamada VRP nin optimal çözümüne yaklaşmak için  $\lambda_1$  değiştirilerek her bir  $\lambda_1$  değeri için problem tekrar çözülmektedir. (3.53)-(3.55) yaklaşık olarak çözüldüğü için bu algoritma sezgiseldir. Ayrıca (3.53)-(3.55) kesin olarak çözülsede bile bu çözümün orijinal VRP nin optimal çözümü olması gerekmemektedir. Algoritma bu yönü ile de sezgiseldir. Aşağıda LR3OPT algoritmasının adımları açıklanmaktadır.

Adım 1.  $\lambda_1=0$ . Ortaya çıkan MTSP yi çöz. Eğer çözüm, kapasite kısıtlarına uygun ise, algoritmaya son ver. Aksi halde devam et.

Adım 2.  $\lambda_1 > 0$  ve  $\Delta > 0$  seç.

Adım 3. Problem (3.53)-(3.55) i 3-opt algoritması ile çöz. Bu, en büyük talepli rotanın  $\lambda_1$  cezasına sahip olduğu bir MTSPdir.

Eğer çözüm, VRP için uygun ise, Adım 5 e git. Aksi halde devam et.

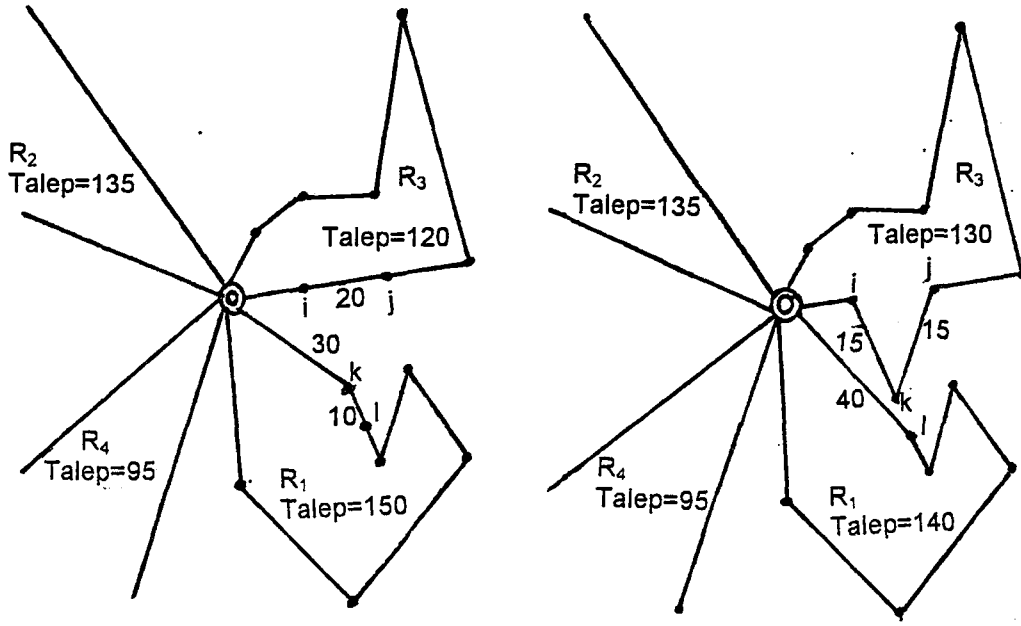
Adım 4. Rota 1 in talebi ( $Y_1$ ) aracın kapasitesini aşmaktadır.

$\lambda_1 = \lambda_1 + \Delta$  yap ve Adım 3 e git.

Adım 5.  $\lambda_1 - \Delta$  ile  $\lambda_1$  arasında bulunan ve VRP için uygun bir çözüm üretebilecek olan en küçük  $\lambda_1$  değerini bul. Bu değeri  $\lambda_1^*$  ile göster ve problem (3.53)-(3.55) i 3-opt algoritması ile çözerek algoritmayı tamamla.

LR3OPT algoritması VRP için uygun bir çözüm aramaktadır. Eğer tek amaç uygunluk olsa idi  $\lambda_1$  için çok büyük bir başlangıç değeri seçilerek, bu başarılabilirdi. Ancak bu durumda ortaya çıkan rotalar optimalden uzak olurlar. Buradaki gerçek amaç, "iyi" bir uygun çözüm bulmak olduğu için,  $\lambda_1$  için uygun bir çözüm üretebilecek olan düşük bir değer aranmaktadır. Bu nedenle algoritma çok muhtemelen başlangıç olarak uygun olmayan bir çözüm üretecek olan bir  $\lambda_1$  değeri seçmekte ve sonra uygun bir çözüm elde edilinceye kadar  $\lambda_1$  i yavaş yavaş artırmaktadır.

Algoritmanın diğer adımlarında 3-opt algoritması kullanılırken, Adım 1 de farklı seçenekler mevcuttur. Adım 1 in uygulanması için bir olasılık, tasarruf algoritmasını kullanarak VRP yi çözmek ve sonra küçük talepli rotaları NK adet rota oluşacak şekilde birleştirmektir. Bu rotaların hepsinin uygun olmaları gerekmemektedir. Adım 1 için bir diğer seçenek, MTSP yi bir TSP ye dönüştürmek ve sonra bir TSP sezgiseli uygulamaktır. Adım 3 de en büyük talepli rotanın cezalandırıldığı MTSP yi çözmek için bir 3-opt algoritması kullanılmaktadır. Ancak burada orijinal 3-opt algoritmasından farklı olarak her bir adımda her bir rotadaki toplam talep yeniden hesaplanmakta ve amaç fonksiyonu kontrol edilmektedir. Şekil 3.9 Adım 3 ü açıklamaktadır.



Şekil 3.9. Rotaların karşılaştırılması, 3-opt algoritması  
( $Q=140, \lambda_1=2$ )

Şekil 3.9 (a) daki i-j, i-k ve k-l kanalları yerine Şekil 3.9 (b) de i-k, k-j ve o-l kanalları yerleşmektedir. Maliyet normal tasarruf formülüne göre artmış gibi görünmesine karşın amaç fonksiyonu (3.53) e göre azalmaktadır. Çünkü Şekil 3.9 (a) da rota 1 uygun değilken Şekil 3.9 (b) de bütün rotalar uygundur.

$$\text{Maliyet azalması} = c_{ij} + c_{ok} + c_{kl} + \lambda_1 (150-Q)=80$$

$$\text{Maliyet artışı} : c_{ik} + c_{kj} + c_{ol} + \lambda_1(140-Q)=70$$

$$\text{Toplam tasarruf} = 80-70=10$$

### 3.2.3.5. Matematik Programlama Esaslı Yaklaşık Algoritmalar

VRP nin gevşetilmiş bir formülasyonunu optimal olarak çözmeye dayanan matematik programlama esaslı algoritmik çözüm algoritmalarının en yaygın olarak kullanılan iki türü, Fisher ve Jaikumar (1981) in genel tahsis sezgiseli ve Christofides, Mingozzi ve Toth (1979) un iki aşamalı algoritmasıdır. Aşağıda bu iki algoritma sıra ile ele alınmaktadır.

### - Genel Tahsis Sezgiseli

Fisher ve Jaikumar (1981) tarafından geliştirilmiş olan sezgisel, VRP yi karşılıklı iki bileşenden ibaret olarak ele almaktadır.<sup>45</sup> Bu bileşenlerden birisi, bir TSP (rotalama problemi) ve diğeri bir genel tahsis (paketleme) problemidir. Aşağıdaki formülasyonlar VRP'nin bu iki bileşenini ortaya koymaktadırlar.

$$\text{Minimize } \sum_{i,j,k} c_{ij} X_{ij}^k \quad (3.56)$$

$$\sum_i d_i Y_{ik} \leq Q_k, \quad k=1,2,...,NK \quad (3.57)$$

$$\sum_k Y_{ik} \begin{cases} NK, & i=1 \text{ için} \\ 1, & i=2,3,...,N \text{ için} \end{cases} \quad (3.58)$$

$$Y_{ik} = 0 \text{ veya } 1, \quad i=1,2,...,N \quad (3.59)$$

$$k=1,2,...,NK$$

$$\sum_i X_{ij}^k = Y_{ik}, \quad j=1,2,...,N, \quad k=1,2,...,NK \quad (3.60)$$

$$\sum_j X_{ij}^k = y_{ik}, \quad i=1,2,...,N, \quad k=1,2,...,NK \quad (3.61)$$

$$\sum_{i,j} X_{ij}^k \leq |S| - 1, \quad S \subseteq \{2,3,...,N\}, \quad k=1,2,...,NK \quad 2 \leq |S| \leq n-2 \quad (3.62)$$

$$X_{ij}^k = 0 \text{ veya } 1, \quad i=1,2,...,N, \quad k=1,2,...,NK \quad j=1,2,...,N \quad (3.63)$$

Yukarıdaki formülasyonda,

---

<sup>45</sup> M. FISHER, R. JAIKUMAR, "A generalised assignment heuristic for vehicle routing", Networks 11(2), 1981, S.109-124.

$$Y_{ik} = \begin{cases} 1, & \text{eğer } i \text{ müşterisine } k \text{ aracı ile hizmet verilirse} \\ 0, & \text{aksi halde} \end{cases}$$

şeklinde tanımlanmaktadır. Kısıt (3.57)-(3.59) bir genel tahsis problemini ifade etmekte ve her bir müşteriye bir aracın tahsis edilmesini, araç yükünün ( $\sum_i d_i Y_{ik}$ ) aracın kapasitesini aşmamasını ve her bir araç rotasının depoyu (düğüm 1) içermesini temin etmektedirler.  $Y_{ik}$  değişkenleri, (3.57)-(3.59) a uygun olacak şekilde belirlendikleri zaman, her bir araç için kısıt (3.60)-(3.63) bu araca tahsis edilen müşteriler üzerindeki bir TSP yi ifade etmektedirler.

TSP ve genel tahsis problemi, literatürde yaygın olarak incelenmiş oldukları için, yukarıdaki VRP formülasyonu, iki bileşenli problemler için geliştirilmiş olan teori ve algoritmalarından faydalanabilmektedir. Ayrıca aşağıda açıklanan ve bu formülasyona dayanan sezgisel, eğer mevcut ise, daima uygun bir çözüm bulmaktadır.

Aşağıda VRP nin doğrusal olmayan bir genel tahsis problemi olarak yeniden formülasyonu verilmektedir.

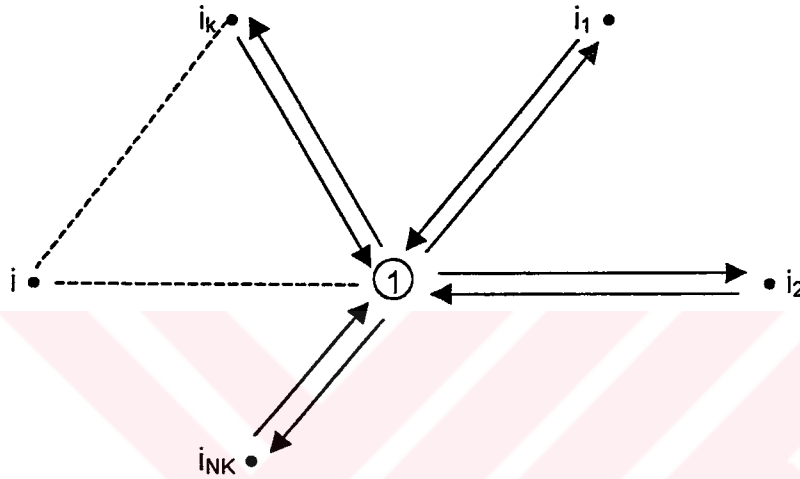
$$\text{Minimize } \sum_k f(Y_k) \quad (3.64)$$

$$\sum_i d_i Y_{ik} \leq Q_k, \quad k=1,2,\dots,NK \quad (3.65)$$

$$\sum_k Y_{ik} = \begin{cases} NK, & i=1 \\ 1, & i=2,3,\dots,N \end{cases} \quad (3.66)$$

$$Y_{ik}=0 \text{ veya } 1, \quad i=1,2,\dots,N, \quad k=1,2,\dots,NK \quad (3.67)$$

Formülasyonda görülen  $f(Y_k)$ ,  $N(Y_k)=\{i | Y_{ik}=1\}$  deki bütün müşteriler üzerindeki TSP turunun maliyetidir.  $f(Y_k)=\text{Min}\{\sum_{i,j} c_{ij} X_{ij}^k | k \text{ aracı kısıt (3.60)-(3.63) e uygundur.}\}$  şeklinde tanımlanmış olduğu için,  $f(Y_k)$  aslında karmaşık bir fonksiyondur. Bu nedenle genel tahsis sezgiseli, (3.64) deki  $f(Y_k)$  yerine bir doğrusal tahmini olan  $\sum_i \tilde{c}_{ik} Y_{ik}$  yı yerleştirmekte ve sonra (3.64)-(3.67) yi çözmektedir.



Şekil 3.10.  $\tilde{c}_{ik}$ 'nin hesaplanması

Ortaya çıkan doğrusal genel tahsis probleminin çözümü, müşterilerin araçlara uygun bir tahsisini ifade etmektedir. Bundan sonra her bir araç için bir TSP'nin çözülmesi gerekmektedir.  $F(Y_k)$  nın doğrusal tahmininin bulunmasında birçok yöntem bulunmakla birlikte, burada bu yöntemlerden sadece bir tanesi açıklanmaktadır.  $\tilde{c}_{ik}$ 'nin bulunmasında, önce sıra ile 1,2,...,NK araçlarına tahsis edilmek üzere çekirdek müşterilerden  $(i_1, i_2, \dots, i_{NK})$  oluşan bir set seçilmektedir. Şekil 3.10 da gösterildiği gibi  $\tilde{c}_{ik}$ , i müşterisini 1- $i_k$ -1 rotasına sokmanın maliyetine eşitlenebilir.

$$\tilde{c}_{ik} = \min \{c_{1,i} + c_{i,i_k} + c_{i_k,1}, c_{1,i_k} + c_{i_k,i} + c_{i,1}\} - \{c_{1,i_k} + c_{i_k,i}\} \quad (3.68)$$

Formül (3.68) e bakıldığı zaman simetri varsayımının bulunmadığı görülebilir. Çekirdek müşteri değiştirilerek bu süreç tekrarlanabilir. Ayrıca çekirdeklerin seçimi otomatikleştirilebilir. Yukarıdaki doğrusal yaklaşım basit olmasına karşın hesaplama sonuçları son derece çarpıcıdır.

### - İki-Aşamalı algoritma

Christofides, Mingozzi ve Toth (1979) tarafından geliştirilmiş olan iki aşamalı algoritma, ardışık rota-inşa aşaması ve paralel rota-inşa aşaması olmak üzere iki kısma ayrılmaktadır.<sup>46</sup> Birinci aşamada, problem çözücü tarafından verilen  $\lambda \geq 1$  parametresi ile bir minimum maliyetli giriş prosedürüne göre  $k$  adet rota oluşturulmaktadır. Algoritmanın ikinci aşamasında ise  $\mu \geq 1$  parametresi ile müşterilerin alternatif rotalara sokulmalarının maliyetleri karşılaştırılarak, müşteriler bu rotalara tahsis edilmektedirler. Algoritmanın uygulama adımları, aşağıdaki gibidir.

#### Aşama 1. Ardışık rota-inşa aşaması

Adım 1.  $\lambda$  yı seç.  $k=1$  yap.

Adım 2.  $k$  rotasına başlamak için herhangi bir rotalanmamış müşteriyi ( $i_k$ ) seç. Her rotalanmamış müşteri ( $i$ ) için,  $\delta_i = c_{1i} + \lambda c_{iik}$  yi hesapla.

Adım 3.  $S_k$ ,  $k$  rotasına uygun olarak sokulabilecek olan rotalanmamış müşteriler setidir.  $S_i^* = \min_{i \in S_k} \{\delta_i\}$  olsun  $i^*$  müşterisini  $k$  rotasına sok. Bir r-opt algoritması kullanarak  $k$  rotasını optimize et.  $k$  rotasına başka müşteri tahsis edilemeyinceye kadar, Adım 3 ü tekrarla.

Adım 4. Eğer bütün müşteriler rotalara sokulmuş iseler, dur. Aksi halde  $k=k+1$  yap ve Adım 2ye git.

---

<sup>46</sup> N. CHRISTOFIDES, A. MINGOZZI, P. TOTH, "The vehicle routing problem, Combinatorial Optimization", Chap 11, Wiley, Newyork (1979)

### Aşama 2. Paralel rota-inşa aşaması

Adım 5.  $\mu$  yü seç. Aşama 1'in sonunda oluşmuş olan  $k$  adet rota  $R_t = (1, i_t, 1)$ , ( $t=1,2,...,k$ ) ile başla.  $K = \{R_1, R_2, ..., R_k\}$  olsun.

Adım 6. Her bir  $R_t \in K$  ve henüz bir rota ile ilişkisi olmayan her bir müşteri ( $i$ ) için,  $\epsilon_{ti} = C_{1i} + \mu C_{1kt} - C_{1it}$  ve  $\epsilon_{ti} = \min_{t \in K} \{\epsilon_{ti}\}$  yi hesapla.  $i$  müşterisini  $R_{t^*}$  rotasına bağla ve bütün müşteriler bir rotaya bağlanmış oluncaya kadar Adım 6 yı tekrarla.

Adım 7. Herhangi bir rotayı ( $R_t \in K$ ) ele al. Ve  $K = K \setminus \{R_t\}$  yap.  $R_t$  rotası ile ilişkili olan her müşteri ( $i$ ) için  $\epsilon_{ti} = \min_{R_t \in K} \{\epsilon_{ti}\}$  ve  $\tau_i = \epsilon_{ti} - \epsilon_{ti^*}$  yi hesapla

Adım 8.  $\tau_i^* = \max_{i \in S_t} \{\tau_i\}$  olan müşteriyi ( $i^*$ )  $R_t$  rotasına sok (Burada  $S_t$ ,  $R_t$  rotasına uygun olarak sokulabilecek olan ve  $R_t$  rotası ile ilişkili olan rotalanmamış müşteriler setidir). Bir r-opt algoritması kullanarak  $R_t$  rotasını optimize et.  $R_t$  rotasına başka müşteri sokulamayıncaya kadar Adım 8'i tekrarla.

Adım 9. Eğer  $K \neq \emptyset$  ise, Adım 6 ya git. Aksi halde eğer bütün müşteriler rotalanmış iseler, dur. Eğer rotalanmamış müşteri varsa, Adım 1'den başlayarak yeni rotalar oluştur.

### 3.2.3.6. Faz Dönüşüm Yaklaşımı

Bir optimizasyon problemi için faz dönüşümünün (FD) kullanımı ile bir fiziksel faz dönüşüm prosesi arasında çok basit bir benzerlik vardır.<sup>47</sup> Fiziksel faz dönüşümünde düzenli yapılarda atomlara sahip denge durumunda olan bir katı malzeme elde etmek amaçlanmaktadır. Fiziksel

<sup>47</sup> A.V.BREADAM, "Improvement heuristics for the vehicle routing based on simulated annealing", European journal of Operational Research 86(3), 1995, S. 480-490.

faz dönüşüm prosesinin birinci adımı, malzemeyi yüksek bir efektif sıcaklıkta eriterek daha sonra malzeme donuncaya kadar sıcaklığı aşama aşama azaltmayı (soğutmayı) içermektedir. Denge durumu elde etmek için bu soğutmanın çok hızlı olmaması gerekmektedir. Malzemenin denge durumu, malzemenin atomlarının minimum enerji konfigürasyonunu ifade etmekte ve bir optimizasyon probleminin amaç fonksiyonunun minimum değerine karşılık gelmektedir.

Bir optimizasyon probleminin her bir uygun çözümü, atomların herhangi bir konfigürasyonuna karşılık gelmektedir. Fiziksel faz dönüşümünde belirli bir sıcaklıkta ısı dengeye ulaşmak için, sıcaklığın yeterince uzun bir süre sabit kalması gerekmektedir. Bu faz dönüşüm yaklaşımı terimleri ile ifade edildiğinde, belirli bir sıcaklıkta oluşturulan uygun komşu çözümlerin sayısının yeterince yüksek olması gerektiği anlamına gelmektedir. Bir komşu çözüm, mevcut çözümden biraz farklı olan yeni oluşmuş bir uygun çözümdür ve yeni bir konfigürasyon elde etmek için malzemedeki bir atomun küçük bir yer değişikliğine karşılık gelmektedir. Amaç fonksiyonunun her bir uygun çözümünün değeri, malzemenin yeni bir konfigürasyonunun enerjisine uygun şekilde hesaplanmaktadır. Mevcut çözümün amaç değeri ile yeni oluşan uygun çözümün amaç değeri arasındaki farklılık, bir fiziksel faz dönüşüm sürecinde ortaya çıkan enerji değişimi ile aynı anlama gelmekte ve  $\delta = c(R_{\text{yeni}}) - c(R)$  ile gösterilmektedir. Eğer  $\delta < 0$  ise, komşu çözüm 1 olasılığı ile,  $\delta > 0$  olduğunda ise  $\exp\{-[c(R_{\text{yeni}}) - c(R)]/T\}$  olasılığı ile kabul edilmektedir. Burada  $R$  mevcut çözümü,  $R_{\text{yeni}}$  bulunan yeni çözümü ve  $T$  mevcut sıcaklığı göstermektedir.

FD, geleneksel iyileştirme algoritmalarının aksine amaç fonksiyonunun değerini artıran uygun çözümleri reddetmeyerek, bunları belirli bir olasılıkla kabul etmektedir. Bu kabul olasılığı, sürecin başlangıcındaki yüksek bir sıcaklıkla 1'e ve sürecin sonuna doğru ise sıfıra yakın olmakta ve bu FD algoritmasını bir yerel minimum tuzağına düşmekten korumaktadır.

FD nin uygulanmasında önemli bir konu, soğutma programının belirlenmesidir. Soğutma programı, sabit bir başlangıç sıcaklığından, soğutma oranlı bir sıcaklık fonksiyonundan, her bir sıcaklıkta oluşturulan uygun çözümlerin sayısından ve bir durdurma kuralından oluşmaktadır. Başlangıç sıcaklığının büyüklüğü ( $T_{baş}$ ), bütün uygun çözümlerin elde edilebilirliği ile sürecin uzunluğu arasında bir denge oluşturmalıdır. Bunun için bir kural, başlangıç sıcaklığını problem için en iyi ve en kötü çözümler arasındaki maximal farkın bir katına eşit almaktır.

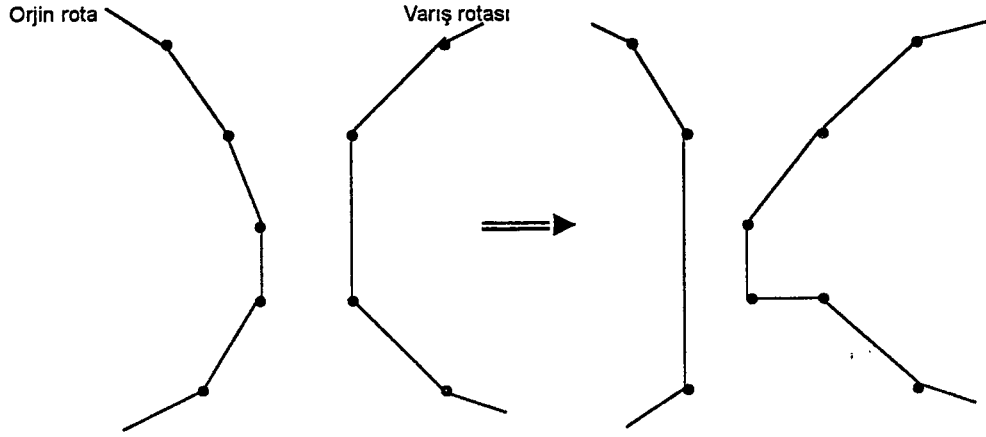
Soğutma oranı, optimal çözümün bulunmasını engelleyecek kadar hızlı sıcaklık azalmalarına yol açmamalıdır. Her bir sıcaklıktaki uygun çözümlerin sayısı bir dengeye ulaşmaya yetecek kadar fazla olmalıdır. Fakat eğer her bir sıcaklık için yeni oluşan uygun çözümlerin sayısı çok fazla olur ise sürecin uzunluğu çok hızlı olarak artacaktır. Bu nedenle burada da bir denge gözetilmelidir. Bir durdurma kuralı, bir durdurma sıcaklığından, sıcaklık azalmalarının sabit bir sayısından, bir minimal kabul oranından veya bir maximum toplam iterasyon sayısından ibaret olabilmektedir. Kabul oranı, kabul edilen uygun çözümlerin sayısının toplam uygun çözüm sayısına bölümü olarak ifade edilmektedir. Breadam (1995), bir örnek olarak Christofides, Mingozi ve Toth (1979) da verilen ondört test probleminin her biri için, soğutma oranını ( $R_c$ ) 0.9 a, toplam iterasyon sayısını ( $N_{top}$ ) 1.000.000 a, bir sıcaklıktaki toplam iterasyon sayısını ( $N_i$ ) 1000'e, kabul oranını 0.01'e ve son sıcaklığı ( $T_{son}$ ) 1 e eşit almaktadır.

Aşağıda FD için Breadam (1995) de kullanılan üç iyileştirme sezgiseli açıklanmaktadır.

(i) Tekrar katar yerleştirme metodu:

Bu metod, bir düğüm veya düğümler katarını bir rotadan çıkartıp bir başka rotaya yerleştirmeyi denemektedir. Metod, ( $m,o$ ) ile sembolize edilmektedir. Buradaki  $m$ , orjin rotasından çıkartılıp varış rotasına

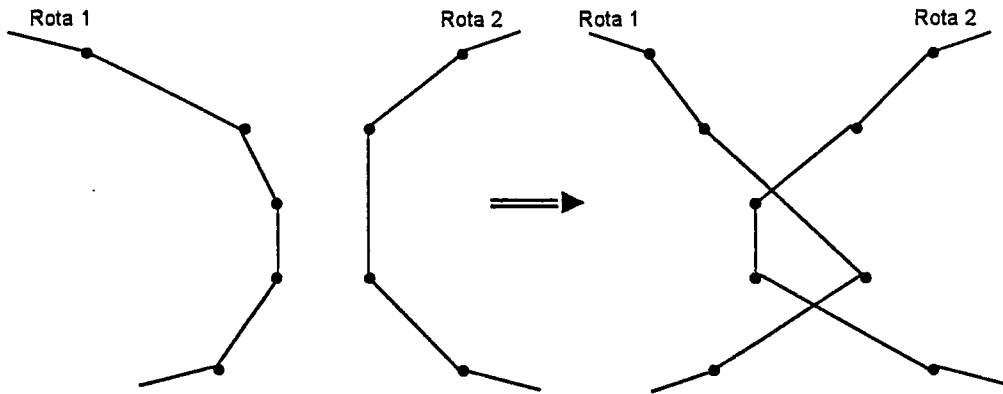
yerleştirilecek katarın uzunluğunu gösteren bir pozitif tamsayıdır. Şekil 3.11, bir (2, 0) tekrar katar yerleştirme örneğini göstermektedir.



Şekil 3.11. İki rota arasında tekrar katar yerleştirme örneği (2, 0)

(ii) Karşılıklı katar değiştirme metodu:

Bu ikinci metotta, her iki rota arasında düğümler veya düğüm katarları birbirleri ile değiştirilerek çözüm iyileştirilmeye çalışılmaktadır. Metod sembolik olarak  $(m, n)$  ile gösterilmektedir. Burada  $m$  birinci rotadaki ve  $n$  ikinci rotadaki değiştirilecek katarların uzunluklarıdır. Şekil 3.12. bir (2, 1) değişimini göstermektedir.



Şekil 3.12. İki rota arasında karşılıklı katar değiştirme örneği (2, 1)

## (iii) Katar karışım metodu:

Üçüncü metod, önceki iki metodun bir karışımıdır. Katar karışım metodu, düğümleri veya düğüm katarlarını yeniden yerleştirmeye veya birbirleri ile değiştirmeye çalışmaktadır. Metod bunu yaparken iki hareketten daha büyük tasarruf sağlayanını seçmektedir. Katar karışım metodunda maximum 3 düğümlü katar hareketleri ele alınmaktadır. Yani  $m \leq 3$  ve /veya  $n \leq 3$  tür.

Şimdi bu üç metod bir FD sezgiseli içersinde birleştirilerek, faz dönüşüm algoritması açıklanabilir. FD sezgiselinde gerekli parametreler, başlangıç sıcaklığı ( $T_b$ ), soğutma oranı ( $R_c$ ), toplam iterasyon sayısı ( $N_{top}$ ) ve bir sıcaklıktaki maximum toplam iterasyon sayısı ( $N_t$ )'dir. Bunlara ilaveten kabul oranı ve son sıcaklık ( $T_s$ ) tercihsel parametrelerdir ve ikisi de durdurma kuralı parametreleri olarak ifade edilmektedir. Bütün bu teknik parametrelere verilmesi gereken değerler deneme-yanılma yolu ile belirlenmektedirler. Çok fazla sayıda kötü kaliteli uygun çözüm oluşmasından kaçınmak için, bu parametrelere ilaveten uygun komşuluk hareketlerini araştırma uzayını bir miktar sınırlayacak bir uzaklık limiti (DL) ve yeniden yerleştirilecek ve/veya birbirleri ile değiştirilecek katarlardaki düğümlerin sayısı için bir üst limit olarak katar uzunluk parametresi (SL) kullanılmaktadır. R, başlangıç uygun rotalar setini ve M kullanılacak iyileştirme metodunu göstermektedir. Sezgiselin uygulama adımları, aşağıdaki gibi özetlenebilir.

Adım 1. Bir başlangıç uygun rotalar seti (R) oluşturun.  $T_b$ ,  $T_s$ ,  $R_c$ ,  $N_t$ ,  $N_{top}$ , Kritik oran, M, DL, SL değerlerini seç.  $T = T_{baş}$  ve  $n_{top} = 0$ .

Adım 2. Eğer  $T \leq T_s$  veya Kabul oranı < Kritik oran veya  $n_{top} = N_{top}$  ise, dur. Aksi halde  $n_t = 0$  ve Adım 3 e git.

Adım 3. Eğer  $n_t = N_t$  ise veya  $n_{top} = N_{top}$  ise  $T = R_c \times T$  ile sıcaklığı azalt ve Adım 2 ye git. Aksi halde Adım 4 e git.

Adım 4.  $N_{top} = n_{top} + 1$  ve  $R_{yeni} = \text{Komşu}(R, M, DL, SL)$ 'yi bul. Eğer  $R_{yeni}$  uygun ise veya  $n_{top} = N_{top}$  ise, Adım 5 e git. Aksi halde Adım 4 ü tekrarla.

Adım 5. Eğer  $n_{top} < N_{top}$  ise,  $\delta = c(R_{yeni}) - c(R)$  yi hesapla. (0,1) aralığında rastgele bir sayı seç. Eğer  $\delta < 0$  ise  $R = R_{yeni}$  ve  $R' = R$ . Aksi halde Adım 6 ya git.

Adım 6. Eğer  $r < \exp(-\delta/T)$  ise  $R = R_{yeni}$  ve  $n_t = n_t + 1$  ve Adım 3 e git.

### 3.2.3.7. Yasak Araştırması Sezgiseli

Yasak araştırması, ilk olarak Glover (1989, 1993) tarafından geliştirilmiş olan ve kompleks araştırma uzaylarını incelemek için yerel optimuma tırmanma tekniklerinin aksine mevcut çözümde amaç değerini yükselten (minimizasyon problemi için) ayarlamalar yapılmasına izin vererek yerel minimumdan kaçınabilen bir tekniktir.<sup>48</sup> Yasak araştırması algoritmasında uygun olmayan ara çözümler kabul edilmekte fakat bu çözümlerin uygunluktan ayrılma dereceleri, amaç fonksiyonundaki bir ceza terimi ile ölçülmektedir. Algoritma her bir adımda mevcut çözümün komşuluğunu belirlemekte ve bu komşuluk içerisindeki en iyi çözümü yeni çözüm olarak seçmektedir. Bununla birlikte bu prosedür saf yerel araştırma tekniklerinin aksine, herhangi bir iyileşme gerçekleşmediği zaman durmayarak yerel optimadan kaçınmakta ve bunun sonucu olarak araştırma uzayının daha büyük bir kısmını inceleme imkanı vermektedir.

Bizim çalışmamızda ele alınan yasak araştırması algoritması, Gendreau, Hertz ve Laporte (1984) tarafından geliştirilmiş olan TABUROUTE'dir. Algoritmada kullanılan notasyon bizim çalışmamızın bütünlüğüne uygun olarak değiştirilmektedir. Bu notasyona göre, NK depoda üslenmiş olan homojen araçların sayısıdır ve ya sabit olarak

---

<sup>48</sup> M. GENDREAU, A. HERTZ, G. LAPORTE, "A tabu Search heuristics for the vehicle routing problem", Management Science 40(10), 1994, S. 1276-1290.

belirlenmiştir ya da  $\overline{NK}$  ile üstten sınırlandırılmıştır.  $G=(N,A)$  yönlü bir grafiktir,  $N=\{N_0,N_1,...,N_n\}$  bir düğümler (şehirler) setidir ve  $A=\{(N_i, N_j), i \neq j\}$  ise bir kanallar setidir ( $N_0$  depoyu göstermektedir). Ele alınan VRP de, her zamanki gibi her bir rota depodan başlayıp aynı noktada sona ermektedir, her şehir  $N/\{N_0\}$  kesin olarak bir araç ile ve kesin olarak bir kere ziyaret edilmelidir. Taburoute sezgiselinin geliştirmesi sırasında kabul edilmiş olan yan kısıtlar aşağıdaki gibidir.

(i) Her şehir negatif olmayan bir talebe ( $q_i$ ) sahiptir ( $q_0=0$ ). Herhangi bir araç rotasının toplam talebi, aracın kapasitesini ( $Q$ ) aşamaz (kapasite kısıtı).

(ii) Her şehir ( $N_i$ ), bir hizmet süresine ( $\delta_i$ ) ihtiyaç duymaktadır. Herhangi bir rotanın toplam uzunluğu, seyahat süresi ile hizmet süresinin toplamından oluşmaktadır ve bu süre bir  $L$  sınırını aşamaz (rota uzunluk kısıtı).

Taburoute sezgiselinde bir çözüm  $NK$  adet rotadan ( $R_1, R_2,...,R_{NK}$ ) oluşan bir  $S$  setidir. Burada,  $NK \in [1, \overline{NK}]$ ,  $R_k=(N_0,N_{r1}, N_{r2},...N_0)$ 'dir ve her bir düğüm ( $N_i, i \geq 1$ ) bir rotaya aittir. Bu rotalar, kapasite ve uzunluk kısıtlarına göre uygun olabilir veya olmayabilirler. Eğer  $N_i, R_r$ 'nin bir bileşeni ise  $N_i \in R_r$  ve eğer  $N_i$  ve  $N_j, R_r$ 'nin iki ardışık düğümü ise  $(N_i, N_j) \in R_r$  yazılmaktadır. Prosedürde kullanılan amaç fonksiyonu herhangi bir uygun çözüm ( $S$ ) ile,

$$F_1(S) = \sum_r \sum_{(N_i, N_j) \in R_r} c_{ij} \quad (3.69)$$

ve herhangi bir çözüm  $S$  ile (uygun veya uygun olmayan)

$$F_2(S) = F_1(S) + \alpha \sum_r [(\sum_{N_i \in R_r} q_i) - Q]^+ + \beta \sum_r [(\sum_{(N_i, N_j) \in R_r} c_{ij} + \sum_{N_i \in R_r} \delta_i - L)]^+ \quad (3.70)$$

olarak verilmektedir. Formül (3.70) de görülen  $\alpha$  ve  $\beta$  iki pozitif ceza parametresidir ve  $[x]^+ = \max(0, x)$ 'dir. Eğer çözüm uygun ise  $F_2(S) = F_1(S)$  olmakta, aksi halde  $F_2(S)$  araç kapasitesini aşma ve rota süresini aşma

durumları için iki ceza terimi içermektedir.  $F_1^*$  ve  $F_2^*$ , algoritmanın herhangi bir adımında, o ana kadar karşılaşılan en küçük  $F_1(S)$  ve  $F_2(S)$  değerlerini göstermektedirler. Ayrıca  $S^*$ , bilinen en iyi uygun çözüm ve  $\bar{S}^*$  bilinen en iyi çözümdür.

Yukarıdaki açıklamalardan sonra şimdi TABUROUTE sezgiselinin merkezinde yer alan SEARCH(P) algoritması açıklanabilir. SEARCH (P), yasak araştırması kullanarak belirli bir çözümü (S) iyileştirmeye çalışmakta ve bunu yaparken, Gendreau, Hertz ve Laporte (1992) tarafından TSP için geliştirilmiş olan iki sezgiseli (GENI ve US) kullanmaktadır.<sup>49</sup> GENI sezgiseli, genel bir giriş (düğümleri rotalara sokma) rutindir. US ise, GENI'yi kullanarak her düğümü ardışık olarak bir rotadan çıkartıp bir diğer rotaya sokan bir post-optimizasyon algoritmasıdır. GENI ve US'nin kombinasyonu, TSP için güçlü bir iki-aşama sezgiseli vermektedir. SEARCH (P) de kullanılan parametreler şunlardır:

W: Mevcut rotadan çıkmasına izin verilen düğümleri içeren,  $N \setminus \{N_0\}$ 'in boş olmayan bir alt seti.

q : W'nin başka bir rotaya girmeye aday olan düğümlerinin sayısı

$p_1$  : N düğümünün sokulacağı rota, N'nin  $p_1$  adet en yakın komşusundan en az birini içermelidir.

$p_2$ : GENI de kullanılan komşuluk hacmi

g : Bir yapay amaç fonksiyonu değeri belirlemek için kullanılan bir ölçek faktörü

$Q_{min}, Q_{max}$  : bir hareketin yasak olarak açıklanması için gerekli iterasyon sayısı üzerindeki sınırlar

h :  $\alpha$  ve  $\beta$ 'nin değiştirilme frekansı

---

<sup>49</sup> M. GENDREAU, A. HERTZ, G. LAPORTE, "New insertion and post-optimisation procedures for the TSP", Operations Research 40, 1992, S. 1086-1094.

$n_{\max}$  : amaç fonksiyonunda herhangi bir iyileşme olmadan algoritmanın son adımının çalışmasına izin verilen maximum iterasyon sayısı.

### SEARCH (P) Algoritması=

Adım 0 (Başlatma). İterasyon sayısını 1 e eşitle ( $t:=1$ ), hareket tabu değildir.

Adım 1. (Düğüm seçimi). Çözüm S yi ele al ve W den rastgele olarak q adet düğüm seç.

Adım 2. (Aday hareketlerin değerlendirilmesi). Seçilen bütün düğümler (N) için aşağıdaki süreci tekrarla. N nin mevcut rotasından ( $R_r$  den) hiç şehir içermeyen (eğer  $NK < \overline{NK}$  ise) veya N nin  $p_1$  adet en yakın komşusundan en az birini içeren başka bir rotaya ( $R_s$  ye) doğru bütün potansiyel hareketleri ele al. Bütün aday hareketler için aşağıdaki işlemleri tekrarla.

(a) N yi  $R_r$  den çıkart ve  $p_2$  parametresi ile GENI algoritmasını kullanarak  $R_s$  ye sokmanın maliyetini hesapla ve buna uyan S' yi belirle.

(b) Eğer hareket yasak ise, eğer S' uygun ise ve  $F_1(S') < F_1^*$  ise veya eğer S' uygun değil ise ve  $F_2(S') < F_2^*$  ise bu hareketi ihmal et.

(c) Aksi halde eğer  $F_2(S') < F_2(S)$  ise  $F(S') = F_2(S')$ , aksi halde  $F(S') = F_2(S') + \Delta_{\max} \sqrt{NK \cdot g \cdot f_N}$  dir. Buradaki  $\Delta_{\max}$ , iki ardışık iterasyonda elde edilen  $F_2(S)$  değerleri arasındaki en büyük mutlak farktır ve  $f_N$ , N düğümünün hareket sayısının t'ye oranıdır.

Adım 3. (En iyi hareketin belirlenmesi). En küçük F değerini veren aday hareketi ve  $\overline{S}$  çözümünü belirle.

Adım 4. (Bir sonraki çözüm). Adım 3'de belirlenen hareketin yerine getirilmesi zorunlu değildir. Aslında  $S$  nin her bir rotasına US prosedürü uygulayarak  $S$  yi iyileştirmeye çalışmak avantajlı olabilir. Aşağıdaki üç koşul tatmin edilemediği zaman çözüm  $S$  yi  $\bar{S}$  ye eşitle.

(a)  $F_2(\bar{S}) > F_2(S)$  ise, (b)  $S$  uygun ise, (c)  $t-1$  iterasyonunda US kullanılmamış ise. Böyle bir durumda US sürecini uygulayarak  $S$  yi elde et.

Adım 5. (Güncelleştirme). Eğer Adım 4 de US süreci kullanılmamış ve  $N$  düğümü  $R_r$  rotasından  $R_s$  rotasına ( $S \neq r$ ) aktarılmamış ise,  $N$  yi  $R_r$  ye sokmayı  $t+Q$  iterasyonuna kadar yasak olarak açıkla. Burada  $Q$ ,  $[Q_{\min}, Q_{\max}]$  aralığında rastgele seçilen bir tamsayıdır.  $t=t+1$  yap.  $F_1^*$ ,  $F_2^*$ ,  $S^*$ ,  $\bar{S}^*$ ,  $\Delta_{\max}$ ,  $NK$ ,  $f_N$  yi uygun şekilde güncelleştir.

Adım 6. (Ceza ayarlama). Eğer  $t$ ,  $h$  nin bir katı ise  $\alpha$  ve  $\beta$  yi aşağıdaki gibi ayarla. Daha önceki  $h$  adet çözümün hepsinin araç kapasitesine göre uygun olup olmadıklarını kontrol et. Eğer hepsi uygun iseler  $\alpha=\alpha/2$  yap. Eğer hiçbiri uygun değil ise  $\alpha=2\alpha$  yap. Benzer şekilde eğer daha önceki  $h$  adet çözümün hepsi rota uzunluğuna göre uygun iseler  $\beta=\beta/2$  ve hiçbiri uygun değil ise  $\beta=2\beta$  yap.

Adım 7. (Bitiş kontrolü). Eğer son  $n_{\max}$  adet iterasyon için  $F_1^*$  ve  $F_2^*$  azalmamış iseler, dur. Aksi halde adım 1'e git.

Şimdi ana algoritma açıklanabilir. Taburoute algoritmasında ilk olarak bir çok başlangıç çözümü oluşturulmakta, bunların her birine sınırlı iterasyon için SEARCH süreci uygulanmakta ve Taburoute için bir başlangıç noktası olarak en ümit verici olan çözüm seçilmektedir. Sonra farklı parametre değerleri ( $P_1$  ve  $P_2$ ) ile SEARCH süreci iki kere uygulanmaktadır. Birinci uygulama, genel olarak başlangıç çözümünde en

önemli iyileşmeyi yapmaktadır. İkinci uygulama ise, eğer var ise, bilinen en iyi uygun çözümün veya aksi halde bilinen en iyi uygun olmayan çözümün şehirlerinin özel bir alt seti üzerinde odaklanarak, araştırmayı yerel olarak güçlendirmektedir.

### TABURROUTE Algoritması

Adım 0 (Başlatma).  $\alpha=\beta=1$  ve  $F_1^* = \infty$ . Eğer düğümler iki-boyutlu koordinatlar ile tanımlı iseler onları depo ve yatay bir hat ile oluşturdukları açılara göre yeniden etiketle.

Adım 1 (İlk çözüm). Aşağıdaki operasyonları  $\lambda$  kere tekrarla ( $\lambda$  bir girdi parametresidir).

(a) Rastgele bir düğüm seç.

(b)  $(N_0, N_i, N_{i+1}, \dots, N_n, N_1, \dots, N_{i-1})$  düğüm serisini kullanarak, TSP için GENIUS sezgiseli ile bütün düğümler üzerinde bir tur oluştur.

(c)  $N_0$  ile başlayarak ve turu izleyerek en fazla  $\overline{N\overline{K}}$  adet araç rotası oluştur. Birinci araç, turdaki birinci düğümden başlayarak fakat rotaya dahil edilmesi kapasite ve uzunluk kısıtının ihlaline neden olacak olan ilk düğümü ( $N_i$ ) hariç tutarak bütün düğümleri içermektedir. Sonra  $N_i$ 'den başlayarak ve bütün düğümler rotalara girmiş oluncaya (bu durumda çözüm uygundur) veya  $\overline{N\overline{K}} - 1$  adet araç kullanılmış oluncaya kadar (bu durumda kalan bütün düğümler  $\overline{N\overline{K}}$  aracına tahsis edilirler ve çözüm uygun olmayabilir), bu süreci tekrarla.  $S$ , bu süreç ile elde edilen çözüm (uygun veya uygun olmayan) olsun.  $F_1^*$ ,  $F_2^*$ ,  $S^*$  ve  $\overline{S}^*$  yi uygun şekilde değiştir.

(d) SEARC ( $P_1$ ) i çağır.

(e) Eğer  $F_1^* < \infty$  ise  $S=S^*$ , aksi halde  $S=\overline{S}^*$  yap.

Adım 2. (Çözüm iyileştirme) SEARCH ( $P_2$ ) yi çağır. Eğer  $F_1^* < \infty$  ise  $S=S^*$ , aksi halde  $S=\bar{S}^*$  yap.

Adım 3. (Güçlendirme). SEARCH ( $P_3$ ) ü çağır. Eğer  $F_1^* < \infty$  ise  $S^*$  bilinen en iyi uygun çözümdür. Aksi halde uygun çözüm bulunamamıştır.

DUR.

Şimdi SEARCH ve TABURROUTE de kullanılan parametrelerin seçimi ve birkaç algoritmik konu üzerinde yorumlar yapılabilir. Ele alınması gereken birinci konu, SEARCH'ın Adım 2c'sindeki güçlendirme stratejisidir. Glover (1989) da, aday çözümün amaç fonksiyonuna o anda taşınan  $N$  düğümünün mutlak hareket frekansına orantılı bir terim eklenerek, sık sık taşınan düğümlere ceza yüklenmektedir.<sup>50</sup>

Bir kural olarak  $g$  için çok yüksek bir değer kullanmak iyi bir çözüm elde etme olasılığını azaltırken çok düşük bir değer kullanmak ise istenilen güçlendirme etkisini sağlamamaktadır. Çünkü bu durumda algoritma, mevcut çözümden uzaklaşmamaktadır. Gendreau, Hertz ve Laporte (1994) de algoritmanın  $[0,005; 0.02]$  aralığındaki  $g$  değerlerine karşı oldukça hassas olduğu ifade edilmektedir.<sup>51</sup>

SEARCH'ın Adım 5 inde kullanılan yasak listesinin uzunluğu ( $Q$ ), gerekli kayıt miktarını sınırlamakta ve bunun sonucu olarak algoritmanın hızını artırmaktadır. Glover ve Laguna (1993),  $Q_{\min}=5$  ve  $Q_{\max}=10$  değerlerinin uygun olduğunu ifade etmektedirler.<sup>52</sup>

SEARCH ın Adım 6 sında kullanılan  $\alpha$  ve  $\beta$  nın algoritmanın akışı esnasında değiştirilmesi düşüncesi, amaç fonksiyonuna ceza terimlerinin ilave edildiği başka alanlara da uygulanabilir. Genellikle uygun bir katsayı

<sup>50</sup> F. GLOVER, "Tabu search, Part I", ORSA J. Computing 1, 1989, S. 190-206.

<sup>51</sup> Gendreau ve diğerleri, 1994, Agm., S. 1279.

<sup>52</sup> F. GLOVER, M. LAGUNA, "Tabu search, Modern heuristic techniques for combinatorial problems", C. Reeves (Ed.), Blackwell Scientific publications, Oxford, 1993, S. 70-150.

değerinin seçilmesi zordur ve yanlış bir seçim, algoritmanın performansı üzerinde olumsuz bir etki yaratabilir. Burada daha önceki  $h=10$  adet çözümün uygun olmaması halinde ceza katsayısı iki katına çıkartılmakta, uygun olmaları durumunda ise yarıya indirilmektedir. Bu kural ile  $\alpha$  ve  $\beta$  nin uygun ve uygun olmayan çözümler karışımını üreten değerlerine çabuk olarak varılmaktadır. Hertz (1992) ye göre algoritma  $h$  nin değeri karşısında fazla hassas değildir ve  $h=5$  veya  $h=20$  ile oluşturulan çözümler,  $h=10$  ile oluşturulanlardan en fazla %1 daha kötüdür. Ayrıca bu tip algoritmada uygun olmayan çözümlerin elde edilmeleri önemlidir. Çünkü bu, yerel optimumdan uzaklaşmaya yardımcı olmaktadır.<sup>53</sup>

Başlangıç çözümlerinin sayısını gösteren  $\lambda$  için kullanılan değer  $\lfloor \sqrt{n}/2 \rfloor$  ye eşit alınmaktadır.<sup>54</sup> Aşağıda SEARCH (P) nin çeşitli çağrılarında  $w$ ,  $q$ ,  $p_1$ ,  $p_2$  ve  $n_{\max}$  parametrelerinin seçimleri üzerinde durulmaktadır.

$W$  parametresi, SEARCH sürecinde farklı rotalara taşınabilen düğümlerin alt setini ifade etmektedir. TABUROUTE nin güçlendirme adımı (Adım 3) hariç, bu parametre daima  $N \setminus \{N_0\}$  a eşittir. Adım 3'de  $W$  parametresi, en büyük  $f_N$  ye sahip olan  $\lfloor N/2 \rfloor$  adet düğüm ( $N$ ) olarak tanımlanmaktadır. Bu düğümler sık sık hareket etmiş oldukları için muhtemelen bir çözüm iyileşmesi sağlamaktadırlar. Adım 3 de  $q$  nun değeri  $|W|$  ye eşittir. Yani hareketine izin verilen bütün düğümler yeniden girmeye adaydırlar. Bununla birlikte  $P_1$  ve  $P_2$  de bu kadar çok sayıda düğüm girişinin ele alınması çok engelleyici olacaktır. Burada  $q$ , her bir rotadan en az bir düğümün seçilmesi ihtimalinin yeterince yüksek olmasını sağlayacak şekilde seçilmektedir. Her bir rotadaki düğümlerin sayılarının yeterince yüksek oldukları varsayımı ile ve  $S(q, NK)$  ikinci türden bir Stirling sayısını göstermek üzere, bu ihtimal

$$P(q, NK) = S(q, NK) NK! / NK^q \quad (3.71)$$

<sup>53</sup> A. HERTZ, "Finding a feasible course schedule using tabu search", Discrete Applied Math 35, 1992, S. 255-270.

<sup>54</sup> Gendreau ve diğerleri, 1994, Agm., S. 1280.

olarak verilmektedir.  $q$  nun en uygun değeri,  $NK$  ya bağlıdır ve  $NK \leq 30$  iken  $q=5NK$  alınması  $P(q, NK) \geq 0.9$  olmasını sağlamaktadır.<sup>55</sup>  $p_2$  parametresi GENI deki komşuluk hacmini göstermektedir.  $p_2=5$  seçilmesi nispeten hızlı olarak optimale yakın TSP çözümünün bulunmasını sağlamakta,  $p_2 \leq 4$  düşük kaliteli çözümler üretmekte ve  $p_2 \geq 6$  ise algoritmanın icra süresini aşırı miktarda artırmaktadır.<sup>56</sup> Görüldüğü gibi algoritma  $p_2$  değeri karşısında oldukça hassastır ve bu nedenle  $P_1$ ,  $P_2$  ve  $P_3$ 'ün hepsinde  $p_2=5$  değeri kullanılmaktadır.

$p_1$  parametresi  $\max(k, p_2)$  ye eşittir.<sup>57</sup> Buradaki  $k$  o anda hareket eden düğümü içeren rotadaki düğümlerin sayısıdır ve  $p_1$  in bu değeri en az bir potansiyel hareketin sözkonusu düğümü farklı bir rotaya tahsis etmesini sağlamaktadır. Son olarak  $n_{\max}$  in değeri,  $P_1$  ve  $P_3$  de  $n$  ye ve algoritmanın en önemli kısmı Adım 2 de icra edildiği için  $P_2$  de  $50n$  ye eşittir.<sup>58</sup> Algoritmanın icra süresi bu parametrenin  $P_2$  deki değeri ile lineer olarak ilişkilidir. Eğer  $n_{\max}$  çok düşük ise bazı iyi çözümler kaçırılacak, aksine  $n_{\max}$  çok yüksek olduğunda ise algoritmanın uzun bir süre iyileşme olmadan çalışması riski ortaya çıkacaktır.

### 3.2.3.8. Çözüm İyileştirme İçin Doyumsuzluk ve Denklik Değişimi Sezgiselleri

Dror ve Levy (1986) tarafından geliştirilmiş olan ve temel girdisi uygun araç rotaları seti olan iki çözüm iyileştirme sezgiselinden birincisi doyumsuzluk değişimi sezgiseli, ikincisi ise denklik değişimi sezgiselidir.<sup>59</sup> Her iki sezgisel de başlangıç çözümünü iyileştirmek için düğüm değiştirme algoritmalarını ve bir tek rota iyileştirme sezgiseli olan SRIH i kullanmaktadırlar.

<sup>55</sup> J. RIORDAN, "An introduction to combinatorial analysis", Wiley, New York, 1958.

<sup>56</sup> Gendreau ve diğerleri, 1994, Agm., S. 1280.

<sup>57</sup> Gendreau ve diğerleri, 1994, Agm., S. 1280.

<sup>58</sup> Gendreau ve diğerleri, 1994, Agm., S. 1280.

<sup>59</sup> M. DROR, L. LEVY, "A vehicle routing improvement algorithm comparison of a greedy and matching implementation for inventory routing", Computers and Operations Research 13(1), 1986, S. 33-45.

Doyumsuzluk deęiřimi algoritması, her bir düęüm deęiřtirme iřleminde, en yüksek tasarrufu saęlayan düęüm deęiřimi seęmekte ve uygun olan ve pozitif tasarruf saęlayan düęüm deęiřimi kalmadıęı zaman, SRIH algoritmasını kullanarak her bir rotayı yeniden optimize etmeye çalışmaktadır. Her bir rota optimize edildikten sonra, artık pozitif tasarruflu uygun düęüm deęiřimi kalmadıęı zaman algoritma sona ermektedir. Ařaęıda önce her iki sezgiselin de aynı noktada çağırıldıkları SRIH algoritması ve ardından doyumsuzluk ve denklik deęiřim algoritmaları sunulmaktadır.

#### SRIH:

Adım 0. (Bařlatma): Bir tek rota gir.

Adım 1. Bir 2-opt kanal deęiřtirme sezgiseli uygula ve Adım 2 ye git.

Adım 2. Bir 3-opt kanal deęiřtirme sezgiseli uygula. Eęer tur ayrıca 2-opt ise, dur. Aksi halde Adım 1'e git.

#### **Doyumsuzluk deęiřimi algoritması**

Toplam rota sayısı  $NK$  ile ve  $I$ .yinci rota  $TNUM(I)$ , ( $I = 1, 2, \dots, NK$ ) ile gösterilmektedir.

Adım 1.  $I = 1, 2, \dots, NK$  için SRIH [ $TNUM(I)$ ] çağır.

Adım 2. 1-düęüm ve 2-düęüm deęiřtirme dönüşümleri için bütün tasarrufları hesapla. Her bir tasarrufu hesaplarken pozitif olup olmadığını ve bu tasarrufu saęlayan düęüm deęiřiklięinin uygun olup olmadığını da kontrol et. Pozitif ve uygun olmayan deęiřimleri elimine et ve dięerlerini kaydet. Eęer pozitif ve uygun tasarruf yok ise, dur.

Adım 3. Tasarrufları büyükten küçüęe doęru sırala.

Adım 4. Eğer tasarruflar listesi boş ise Adım 1 e git. Aksi halde listedeki birinci düğüm değişimini gerçekleştir.

Adım 5. Listedeki diğer tasarrufları yeniden hesapla ve pozitif ve uygun olmayanları elimine et, sonra Adım 3 e git.

Doyumsuzluk değişimi algoritmasının başlangıcındaki düğüm değişimleri, algoritmanın sonunda zayıf kaliteli (yüksek maliyetli) rotalara yol açabilmektedirler. Aşağıda sunulan denklik değişimi algoritması, bu dezavantajı yok etmeye çalışmaktadır.

#### Denklik değişimi algoritması

Bu algoritmanın birinci ve ikinci adımları doyumsuzluk değişimi algoritmasındakinin aynısıdır.

Adım 3. Her bir rota için, rotaya yeni bir düğümün girmesine neden olan bir değişim için bütün tasarrufların bir listesini oluştur.

Adım 4. Her bir rotaya ait tasarruf listesindeki en büyük elemanı bul.

Adım 5. Eğer bütün listeler boş iseler, Adım 8'e git. Aksi halde Adım 4 de bulunan en büyük elemanların maksimumunu seç ve bu elemanı  $\overline{OS}$  ile göster.

Adım 6.  $\overline{OS}$  ye ait düğüm değişimi en fazla dört düğümü etkilemektedir. 2-düğüm değişimi durumunda bundan etkilenen müşteriler, değiştirilen düğümlerden önceki ve sonraki iki çift müşteridir. 1-düğüm değiştirme durumunda ise, aralarına bir düğüm sokulan iki düğüm ve aralarından bir düğüm silinen iki düğüm etkilenmektedir. Etkilenen rotalardan birisinin yapay rota olması durumu, bu adımda ayrıca ele alınmalıdır. Etkilenen düğümlere ait olan bütün tasarrufları rota tasarruf listelerinden sil.

Adım 7. Aşağıdaki gibi bir G grafiği oluştur. Eğer  $\overline{OS}$  2-düğüm değişimine ait ise her bir düğüm G deki bir düğümdür ve bu iki düğümü birbirine bağlayan kanala bu  $\overline{OS}$  değerini tahsis et. Eğer  $\overline{OS}$ , 1-düğüm değişimine ait ise, G ye bir yapay düğüm ilave et. Bu durumda diğer düğüm rotası değişen düğümü göstermektedir. İki düğümü birleştiren kanala gene  $\overline{OS}$  değerini ver,  $\overline{OS}$  yi rota listelerinden sil ve Adım 5'e git.

Adım 8. G grafiği üzerindeki maksimum ağırlıklı eşleştirme problemini çöz. Her bir rota için, eşleştirme probleminin çözümünde rota ile ortak bir düğüme sahip olan değişimlerin bir listesini oluştur.

Adım 9. Eşleştirme probleminin çözümünden elde edilen değişimleri gerçekleştirerek ortaya çıkan rotaların uygunluklarını incele. Bazı durumlarda ortaya çıkan rotalar kapasite kısıtına uygun olmayabilirler. Eğer bütün rotalar uygun iseler, Adım 1 e git. Aksi halde Adım 10'a git.

Adım 10. Talebin kapasite kısıtını en fazla aştığı rotayı bul. Bu rota için, eşleştirme probleminin çözümündeki düğüm değişimleri listesini incele ve rotaya talep ilave etmeyen değişimleri bu listeden sil.

Adım 11. MIR maximum ölçüde uygun olmayan rotaya talep ilave eden düğümlerin listesini,  $\bar{d}_i$   $i \in \text{MIR}$  değişimi için ilave talebi,  $\bar{q}$  bu rotadaki talep fazlasını ve  $OS_i$  i değişimi için tasarruf değerini göstermek üzere aşağıdaki problemi çöz.

$$\text{Minimize } \sum_{i \in \text{MIR}} OS_i Y_i \quad (3.72)$$

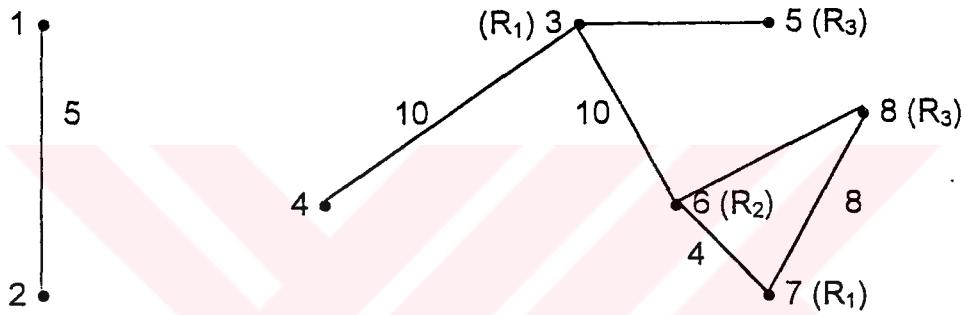
$$\sum_{i \in \text{MIR}} \bar{d}_i Y_i \geq \bar{q} \quad (3.73)$$

$Y_i=0$  veya 1, (bütün  $i$  ler için)

(3.74)

Problemin çözümünde  $Y_i=1$  olan bütün düğüm değişimlerini rotadan sil. Adım 9 a git.

Şekil 3.13 de, Adım 7'de oluşturulabilecek bir G grafiğinin basit bir gösterimi sunulmaktadır. Bu gösterimde sadece üç rota ( $R_1$ ,  $R_2$ ,  $R_3$ ) mevcuttur ve 2, 3, 5, 6, 7, ve 8 müşteri düğümlerini, 1 ve 4 ise yapay düğümleri göstermektedir. Düğümlerin yanlarındaki  $R_1$ ,  $R_2$ ,  $R_3$ , müşterilerin rota numaralarını ve kanalların yanlarındaki rakamlar, düğüm değişimine ait tasarruf değerlerini göstermektedir.



Şekil 3.13. Denklik değişimi algoritması, Adım 3'de oluşturulan bir grafik örneği

Doyumsuzluk değişimi algoritmasında, amaç fonksiyonunun değerinde en büyük iyileşmeye yol açan düğüm değişimi bulunarak uygulanmakta ve sonra uygun olan ve amaç fonksiyonunun değerini azaltan bütün düğüm değişimleri yeniden hesaplanmaktadır. Bu algorithmada bir seferde bir tek düğüm değişimi yapılmaktadır. Denklik değişimi algoritmasında, aynı anda bir çok düğüm değişimi gerçekleştirilmeye çalışılmaktadır. Ortaya çıkan eşleştirme probleminin çözümündeki düğüm değişimlerinin her birinin uygulanmasında bu algoritmanın geçerli bir rota iyileşmesi sunması için, bu düğüm değişimleri birbirlerinden bağımsız olmak zorundadırlar. Yani bir düğüm değişimi, herhangi bir diğer düğüm değişimini etkileyemez. Bu bağımsızlık, denklik

değişimi algoritmasında temin edilmekte ve prosedürün altıncı adımında açıklanmaktadır.

### 3.3. Çok depo/araç Rotalama Problemi (MVRP)

Çok depo/ araç rotalama problemi (multiple vehicle routing problem, MVRP) belirli yerleşimlere sahip olan  $M$  adet depodan, belirli talep ve yerleşimlere sahip olan  $N$  adet müşterinin ihtiyaçlarını karşılayabilmek için, toplam rota uzunluğu minimum olacak tarzda müşterilerin depolara tahsis edilmeleri ve her bir depoya ilişkin rotaların belirlenmesi problemi olarak tanımlanabilir. Bu problemde de daha önce incelenmiş olan tek depo/araç rotalama problemindeki (VRP) gibi zaman (uzunluk) ve kapasite kısıtlamalarına ilaveten her bir müşteriye bir tek depodan ayrılan bir tek araç (rota) ile hizmet verilmesi kısıtı vardır.

MVRP, VRP nin aksine, literatürde çok az incelenmiş ve bu konuda sadece birkaç sezgisel çözüm tekniği geliştirilmiştir.

#### 3.3.1. MVRP Formülasyonları

VRP için (3.30)-(3.39) da verilen tamsayılı programlama formülasyonunda yapılan küçük ayarlamalar, MVRP ler için model oluşturulması imkanı vermektedirler. 1, 2,...,M depoları göstermek üzere kısıt (3.31) ve (3.32) deki indexler sırası ile  $j=M+1, M+2, \dots, N$  ve  $i=M+1, M+2, \dots, N$  şeklinde değiştirilerek ve kısıt (3.36) ve (3.37) uygun şekilde düzeltilerek MVRP için aşağıdaki tamsayılı programlama formülasyonu elde edilmektedir.

$$\text{Minimize } \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^{NK} c_{ij} X_{ij}^k \quad (3.75)$$

$$\sum_{i=1}^N \sum_{k=1}^{NK} X_{ij}^k = 1 \quad (j=M+1, M+2, \dots, N) \quad (3.76)$$

$$\sum_{j=1}^N \sum_{k=1}^{NK} X_{ij}^k = 1 \quad (i=M+1, M+2, \dots, N) \quad (3.77)$$

$$\sum_{i=1}^N X_{ip}^k - \sum_{j=1}^N X_{pj}^k = 0 \quad (k=1,2,\dots,NK; p=1,2,\dots,N) \quad (3.78)$$

$$\sum_{i=1}^N d_i \left( \sum_{j=1}^N X_{ij}^k \right) \leq Q_k \quad (k=1,2,\dots,NK) \quad (3.79)$$

$$\sum_{i=1}^N t_i^k \sum_{j=1}^N X_{ij}^k + \sum_{i=1}^N \sum_{j=1}^N t_{ij}^k X_{ij}^k \leq T_k \quad (k=1,2,\dots,NK) \quad (3.80)$$

$$\sum_{i=1}^M \sum_{j=M+1}^N X_{ij}^k \leq 1 \quad (k=1,2,\dots,NK) \quad (3.81)$$

$$\sum_{j=1}^M \sum_{i=M+1}^N X_{ij}^k \leq 1 \quad (k=1, 2,\dots,NK) \quad (3.82)$$

$$X \in S \quad (3.83)$$

$$X_{ij}^k = 0 \text{ veya } 1 \quad (\text{bütün } i, j, k \text{ lar için}) \quad (3.84)$$

Kısıt (3.83) de görülen S, alt tur-eliminasyon setidir ve aşağıdaki üç seçenekten biri olabilir.

(1)  $S = \{(X_{ij}): \sum_{i \in W} \sum_{j \in W} X_{ij} \geq 1, \quad 1,2,\dots,M \text{ düğümlerini içeren } \{1,2,\dots,N\} \text{nin}$   
her  
uygun w alt seti için.

(2)  $S = \{(X_{ij}): \sum_{i \in R} \sum_{j \in R} X_{ij} \leq |R| - 1, \quad \{M+1, M+2,\dots,N\} \text{nin boş olmayan her R}$   
alt seti için

(3)  $S = \{(X_{ij}): y_j - y_i + N X_{ij} \leq N - 1, \text{ bazı reel } y_i \text{ sayıları için, } M+1 \leq i \neq j \leq N \text{ için}$

### 3.3.2. Tahsis-Etme-Süpürme Sezgiseli

MVRP için Gillett ve Johnson (1976) tarafından geliştirilmiş olan sezgisel, süpürme sezgiselinin çok depo problemini iki aşamada çözen bir uzanımdır. Birinci aşamada müşteriler depolara tahsis edilmektedir. Bu aşamadan ayrı olarak ele alınan ikinci aşamada ise süpürme sezgiseli

kullanılarak bir çok VRP çözülmektedir.<sup>60</sup> Algoritmanın uygulama adımları aşağıdaki gibidir.

Adım 1.  $t'(i)$ ,  $i$  müşterisine en yakın ve  $t'(i)$   $i$  müşterisine ikinci en yakın depoyu göstermek üzere bütün müşteriler için  $r(i) = c_{i,t'(i)} / c_{i,t''(i)}$  oranlarını hesapla.

Adım 2. Bütün müşterileri  $r(i)$  değerlerine göre küçükten büyüğe doğru sırala. Bu sıra, müşterilerin bir depoya tahsis edilme sıralarını gösterir.

Adım 3. Listenin tepesinden başlayarak  $i$  müşterisini  $t'(i)$  deposuna tahsis et.  $r(i)$  oranı 1 e yakın olan müşterileri ayrıca ele alarak şöyle tahsis et. Eğer  $j$  ve  $k$  bir depoya tahsis edilmiş olan iki müşteri iseler,  $i$  müşterisinin  $t$  deposuna ait olan bir rotadaki  $j$  ve  $k$  arasına sokulması,  $c_{ji} + c_{ik} - c_{jk}$  ya eşit bir ilave maliyet yaratmaktadır.  $i$  müşterisini bu ilave maliyeti minimum yapacak şekilde halihazırda  $t$  deposuna tahsis edilmiş olan iki müşteri arasına yerleştir. Bütün müşterileri depolara tahsis ettikten sonra Adım 4 e git.

Adım 4. Her bir depoya tahsis edilmiş olan müşteri kümesi için ayrı ayrı süpürme algoritmasını kullanarak rotalar oluştur.

Adım 5. Çözüm iyileştirme sezgiselleri kullanarak, rotaları iyileştir.

### 3.3.3. Tillman ve Cain (1972) Tasarruf Yaklaşımı

Bu algoritma, her bir müşteriye en yakın depodan ve sadece o müşteriye ait olan bir rota ile hizmet verilen bir ilk çözüm ile başlamaktadır.<sup>61</sup>  $c_{ij}$ ,  $i$  ve  $j$  müşterileri arasındaki uzaklığı ve  $c_i^d$ ,  $i$  müşterisi

<sup>60</sup> B. GILLET, J. JOHNSON, "Multi-terminal vehicle-dispatch algorithm", Omega 4, 1976, S. 711-718.

<sup>61</sup> F. TILLMAN, T. CAIN, "An upper bounding algorithm for the single and multiple terminal delivery problem", Management Science 18, 1972, S. 664-682.

ile  $d$  deposu arasındaki uzaklığı göstermek üzere, başlangıç çözümünün maliyeti,  $D = \sum_{i=1}^N 2\min\{c_i^d\}$  olmaktadır. Algoritma, toplam maliyeti azaltacak şekilde müşteri çiftlerini ard arda birbirlerine bağlamaktadır. Burada kabul edilen temel bir kural şöyle açıklanabilir. Müşterilerin kendilerine en yakın depolara başlangıçtaki tahsisleri geçicidir. Fakat iki veya daha fazla müşteri bir kez bir depoya ait olan ortak bir rotaya tahsis edildikleri zaman, bu tahsis sürekli olmaktadır.

Her bir adımda  $d$  deposuna ait bir rotadaki  $i$  ve  $j$  müşterilerinin birbirlerine bağlanmalarına dair seçim, bu bağlantı gerçekleştirildiği zaman elde edilecek tasarrufa göre ve kısıtlara uygun olarak yapılmaktadır.

Tasarrufların hesaplanmaları, tek-depo durumundakinden farklıdır.  $S_{ij}^d$   $i$  ve  $j$  müşterileri ve  $d$  deposunun her kombinasyonu için hesaplanmakta ve  $d$  deposunda  $i$  ve  $j$  müşterilerinin birbirlerine bağlanmaları durumunda toplam maliyetteki azalmayı göstermektedir.

$$S_{ij}^d = \tilde{c}_i^d + \tilde{c}_j^d - c_{ij} \quad (3.85)$$

$$\tilde{c}_i^d = \begin{cases} 2 \min\{c_i^t\} - c_i^d, & \text{eğer } i \text{ henüz sürekli olarak tahsis edilmemiş ise} \\ c_i^d, & \text{aksi halde} \end{cases} \quad (3.86)$$

Birinci durumda,  $i$  müşterisi kendisine en yakın olan depodan ( $t$ ) bir başka depoya yeniden tahsis edilmekte ve  $2 \min\{c_i^t\}$  kadar maliyet tasarrufu elde edilmektedir. İkinci durumda  $j$  müşterisi  $d$  deposu ile  $i$  müşterisi arasına sokulmakta ve  $i$  müşterisi ile  $d$  deposu arasındaki kanal çözümden çıkartılmaktadır. Prosedürün her bir adımında  $S_{ij}^d$  ( $i, j=1, 2, \dots, N$ ,  $i \neq j$ ,  $k=1, 2, \dots, M$ ) tasarrufları hesaplanmakta ve problemin kapasite ve diğer

kısıtlarına uygun olarak  $S_{ij}^d$  yi maximize eden k deposundaki i-j kanalı seçilerek, bu bağlantılar gerçekleştirilmektedir.

### 3.3.4. Golden, Magnanti ve Nguyen (1977) Tasarruf Yaklaşımı

MVRP, birincisi tahsis ve ikincisi rota oluşturma olmak üzere iki aşamada çözülebilmekle birlikte aslında ideal olanı, Tillman ve Cain (1972) tasarruf algoritmasındaki gibi, iki adımı aynı anda ele almaktır. Bununla birlikte 1000 düğümlü gibi büyük problemler için bu metod hesaplama açısından elverişli değildir. Bu durumda uygun bir yaklaşım, problemi depo sayısı kadar alt probleme bölmek ve her bir alt problemi ayrı ayrı çözmek olabilir. Bu bölümde sunulan tasarruf yaklaşımı, tahsis etme-süpürme algoritmasındaki  $r(i)$  oranları ile birlikte bu stratejiyi kullanmaktadır.<sup>62</sup>

Golden, Magnanti ve Nguyen (1977) tasarruf algoritmasına göre, eğer bir  $i$  düğümü herhangi bir depoya bir başka depodan çok daha yakın ise, bu düğüme en yakın depodan hizmet verilmektedir.  $i$  düğümü bir çok depoya eşit uzaklıkta olduğu zaman, bu düğümün bir depoya tahsis edilmesi daha zor olmaktadır. Prosedürde her bir  $i$  düğümü için, en yakın depo ( $d_1$ ) ve ikinci yakın depo ( $d_2$ ) belirlenmektedir. Eğer  $r(i) = c_i^{d_1} / c_i^{d_2}$  oranı belirli bir seçilmiş  $\delta$  ( $0 \leq \delta \leq 1$ ) parametresinden daha küçük ise,  $i$  düğümü  $d_1$  deposuna tahsis edilmekte,  $r(i) \geq \delta$  olması durumunda ise  $i$  düğümü "sınır düğümü" olarak adlandırılmaktadır. Açıkça görüldüğü gibi,  $\delta=0$  olduğunda bütün düğümler sınır düğümleridir.  $\delta=1$  olduğunda ise bütün düğümler kendilerine en yakın olan depolara tahsis edilmektedirler. Belirli bir problem için  $\delta$  değeri değiştirilerek sınır düğümlerinin sayısı istenildiği gibi ayarlanabilmektedir.

---

<sup>62</sup> B. GOLDEN, T. MAGNANTI, H. NGUYEN, "Implementing vehicle routing algorithms", Networks 7, 1977, S. 113-148.

Algoritmanın birinci adımında, sınır düğümleri dışındaki düğümler ihmal edilmekte ve sınır düğümlerine Tillmen-Cain (1972) tasarruf algoritması uygulanmaktadır. Algoritma, sınır düğümlerini depolara tahsis etmekte ve aynı anda bu düğümleri birbirlerine bağlayarak, kısmi rotalar oluşturmaktadır. Birinci adımın sonunda, problemdeki bütün düğümler depolara tahsis edilmiş ve bütün sınır düğümleri rotalar üzerinde birbirlerine bağlanmış olmaktadır. Bu aşamadan sonra tek depo-VRP teknikleri kullanılarak, her bir depo için ayrı ayrı VRP çözümleri elde edilmekte ve sınır düğümler üzerinde inşa edilen rota parçaları, kalan düğümlere genişletilmektedir.

### 3.4. Filo Hacmi ve Karışımı ile Araç Rotalama Problemi (FSMVRP)

Filo hacmi ve karışımı problemi (fleet size and mix vehicle routing problem, FSMVRP) basit olarak, bir müşteriler şebekesinin toplam talebini minimum maliyetle karşılamak için, hangi büyüklüklerde ve kaç adet araca ihtiyaç olduğu sorusuna cevap arayan bir problemdir. Bu problemde amaç, hem filo hacmini hem de araçların karışımını optimize etmektir.

Toplam dağıtım maliyeti, araçlara ait rotalama maliyetleri ve seçilen araç tipine bağlı olan bir sabit satın alma ve/veya kiralama maliyeti ile yakıt, bakım-onarım ve işgücü maliyetlerinin toplamı olarak ifade edilebileceği için, dağıtım maliyetlerinin minimizasyonunda, rotalama ve filo hacmi ve karışımı bileşenlerinin birlikte ele alınması gerekmektedir. Kullanılan araç tipine bağlı maliyetler, bir başka anlamda sabit ve değişken maliyetler olarak iki ana kısma ayrılmakta ve bunlardan değişken maliyet bileşeni, seyahat edilen mesafeye bağlı olarak değişmektedir.

Çalışmamızın bu aşamasına kadar araç filosunun homojen olduğu kabul edilmesine rağmen gerçek yaşamda böyle bir zorunluluğun bulunmadığı açıktır. İyi bir araç filosu karışımı, muhtemelen en iyi sonuçları verecek olan karışımdır. Aşağıda bir dağıtım (teslimat) problemi için, filo hacmi ve karışımı ile araç rotalama probleminin (FSMVRP) matematik formülasyonları verilmektedir.

### 3.4.1. FSMVRP İçin Matematiksel Formülasyonlar

Burada sunulan matematiksel formülasyonlar bir dağıtım problemi için geliştirilmiş olmasına karşın bir toplama problemine de kolayca uyarlanabilecekleri açıktır. Kullanılan rotasyonda,  $N$  : müşterilerin sayısı,  $K$ : araç tiplerinin sayısı,  $q_i$  :  $i$ .yinci müşterinin talebi,  $c_{ij}$  :  $i$ - $j$  hattında seyahat etmenin maliyeti,  $t_{ij}$  :  $i$ - $j$  hattında seyahatin süresi,  $F_k$  :  $k$  tipli araç için sabit araç maliyeti,  $Q_k$  :  $k$  tipli araç için kapasite kısıtı ve  $T_k$  :  $k$  tipli araç için maximum seyahat süresi kısıtlıdır. Ayrıca  $Y_{ij}$ ,  $i$  ve  $j$  müşterileri arasında taşınan yük miktarını,  $R_{ij}$  ise bir aracın  $i$ - $j$  hattını katettikten sonra kalan yedek mesafesini (zamanını) gösteren birer sürekli değişkendir.

$$X_{ij}^k = \begin{cases} 1, & \text{eğer } k \text{ tipli araç } i\text{-}j \text{ hattından geçerse} \\ 0, & \text{aksi halde} \end{cases}$$

$$\text{Minimize } \sum_{k=1}^K F_k \sum_{j=1}^N X_{oj}^k + \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N c_{ij} X_{ij}^k \quad (3.87)$$

$$\sum_{k=0}^K \sum_{i=0}^N X_{ij}^k = 1, \quad j=1, 2, \dots, N \quad (3.88)$$

$$\sum_{i=0}^N X_{ij}^k - \sum_{i=0}^N X_{ji}^k = 0, \quad j=0, 1, \dots, N; k=1, 2, \dots, K \quad (3.89)$$

$$\sum_{i=0}^N Y_{ij} - \sum_{i=0}^N Y_{ji} = q_j, \quad j=1, 2, \dots, N \quad (3.90)$$

$$\sum_{i=1}^N Y_{io} = 0 \quad (3.91)$$

$$\sum_{j=1}^N Y_{oj} = \sum_{i=1}^N q_i \quad (3.92)$$

$$Y_{ij} \leq \sum_{k=1}^K Y_{ij}^k \cdot Q_k, \quad i \neq j=0, 1, \dots, N \quad (3.93)$$

$$R_{ij} \leq \sum_{k=1}^K T_k X_{ij}^k, \quad i, j=0, 1, \dots, N \quad (3.94)$$

$$R_{oj} = \sum_{k=1}^K X_{oj}^k T_k - \sum_{k=1}^K X_{oj}^k \cdot t_{oj}, \quad j=1, 2, \dots, N \quad (3.95)$$

$$\sum_{i=0}^N R_{ip} - \sum_{j=0}^N R_{pj} = \sum_{k=1}^K \sum_{j=0}^N X_{pj}^k \cdot t_{pj}, \quad p=1, 2, \dots, N \quad (3.96)$$

$$X_{ij}^k \in \{0, 1\}, \quad Y_{ij} \geq 0, \quad i=0, 1, \dots, N; \quad j=0, 1, \dots, N; \quad k=1, 2, \dots, K \quad (3.97)$$

$$Y_{ii}=0, \quad R_{ij} \geq 0, \quad i=0, 1, \dots, N; \quad j=0, 1, \dots, N \quad (3.98)$$

Formülasyondaki amaç fonksiyonu (3.87), toplam sabit maliyet ile toplam ortalama maliyetinin toplamından ibarettir. Kısıt (3.88), her bir müşterinin sadece bir kere ziyaret edileceğini, kısıt (3.89) her bir aracın girdiği bir müşteri yerleşiminden çıkacağını göstermektedir. Kısıt (3.90), bir müşteri yerleşimine varan ve oradan ayrılan bir araçtaki toplam yük miktarı farkının kesin olarak bu müşterinin talebine eşit olduğunu, kısıt (3.91) depoya hiçbir yükün geri dönmediğini yani araçların depoya boş olarak döndüklerini ve kısıt (3.92) depodan ayrılan toplam yük miktarının toplam müşteri talebine eşit olduğunu ifade etmektedir. Kısıt (3.93), i-j hattı üzerindeki toplam yük miktarının bu hatta tahsis edilen aracın kapasitesini aşmaması gerektiğini ve ayrıca sadece i müşterisinden j müşterisine giden bir araç mevcut olduğu zaman i den j ye yük taşınabileceğini göstermektedir.

Bir başka ifade ile kısıt (3.90)-(3.93), kapasite ve alt-tur eliminasyon kısıtlarıdır. Kısıt 3.94 i-j hattını katetmek için kullanılacak bir yedek zamanın (mesafenin) mevcut olması zorunluluğunu ve kısıt (3.95) depodan ayrılan kanallardan geçildikten sonra kalan yedek zamanın maximum zaman-bu ilk müşteriye ulaşmak için gerekli olan zamanı aşmaması zorunluluğunu ifade etmektedir. Son olarak kısıt (3.96), iki müşteri arasında seyahat eden bir aracın sahip olduğu yedek zamanın bu hattın uzunluğu ile azalmakta olduğunu göstermektedir.

Yukarıdaki formülasyonda, herhangi bir problem için toplam değişken sayısı,  $K(N+1)^2$  adet 0-1 değişken ve  $2(N+1)$  adet sürekli değişken olmak üzere toplam  $K(N+1)^2 + 2(N+1)$  dir. Toplam kısıt sayısı ise  $2(N+1)^2 + (K+4)N + K + 2$  dir. Açıkça görüldüğü gibi, değişken ve kısıt sayılarının çok fazla oluşları kesin algoritmaların kullanılmalarını nerede ise imkansız hale getirmektedir. Bu nedenle FSMVRP yi çözmek için bazı sezgisel algoritmalar geliştirilmiştir.

### 3.4.2. Tasarruf Sezgiselleri

FSMVRP için tasarruf sezgisellerinin geliştirilmesinde ilk düşünülen yaklaşım Clarke-Wright tasarruf tekniğinin benimsenmesidir. Ancak bu teknik, sabit araç maliyetlerini dikkate almadığı için yetersiz kalmaktadır. Problemden sonsuz sayıda aracın mevcut olduğu kabul edilmekte ve bu nedenle uygunluk kesinlikle bir sonun oluşturulmamaktadır. Bunun sonucu olarak, Clarke-Wright tasarruf metodu (CW), en büyük aracın kapasitesine ulaşıncaya kadar alt-turları birleştirme eğiliminde olmaktadır. Fakat hemen hemen tamamen en büyük araçlardan oluşan bir filonun en az maliyetli filo olması gerekmemektedir. Bu nedenle CW yaklaşımının eksikliklerini gideren daha gerçekçi tasarruf sezgiselleri geliştirilmiştir. Aşağıda bunlardan bazıları sunulmaktadır.

#### 3.4.2.1. Kombine Tasarruf Sezgiseli (CS)

Kombine tasarruf sezgiseli (combine savings, CS) tasarrufların sabit araç maliyetlerini de içeren bir formda ifa edilmeleri ile CW yaklaşımından ayrılmaktadır. CS sezgiselinde her alt-tur, ortalama maliyetlerinin ve bu alt-turdaki müşterilere hizmet veren aracın maliyetinin toplamına eşit olan bir toplam maliyete sahiptir. İki alt-tur birleştirildiği zaman oluşan büyük alt-turun maliyeti de buna benzer şekilde hesaplanmaktadır. Algoritmanın

herhangi bir noktasında hangi iki alt-turun birleştirileceğini belirlemek için, aşağıda ifade edilen tasarruf formülü kullanılmaktadır.<sup>63</sup>

$$\bar{S}_{ij} = S_{ij} + F(Z_i) + F(Z_j) - F(Z_i + Z_j) \quad (3.99)$$

(3.99) tasarruf ifadesinde  $F(Z)$ , bir alt-tur için  $Z$  büyüklüğündeki bir talebe hizmet verebilecek en küçük aracın sabit maliyetini ifade eden ve  $[0, \infty]$  aralığında değişebilen bir kesikli fonksiyondur ( $F(0)=0$ ). Uç noktalarından birisi  $i$  düğümü olan bir  $I$  alt-turu ile uç noktalarından birisi  $j$  düğümü olan bir  $J$  alt-turu birleştirildiği zaman elde edilebilecek kombine tasarruf,  $i$  ve  $j$  düğümlerini birleştirmenin sağlayacağı tasarruf ( $S_{ij}$ ) ile iki ayrı araç yerine bir tek araç kullanmanın sağlayacağı tasarrufun toplamına eşittir.  $S_{ij}$ , orijinal CW sezgiselindeki gibi hesaplanmakta ve bütün uygun  $i, j$  çiftleri (uç düğümler) için,  $\bar{S}_{ij}$  negatif olduğu zaman CS algoritması sona ermektedir.

CS yaklaşımı, CW yaklaşımının filo hacmi ve karışımı problemine mantıklı bir uzanımıdır. Bununla birlikte,  $\bar{S}_{ij}$  sadece  $i$  ve  $j$  düğümlerini birleştirerek kazanılan ilk tasarrufu içerdiği için algoritmada bir belirsizlik elemanı bulunmaktadır. Bu konuyu açıklayabilmek için, her biri toplam 100 birim talebe sahip olan ve 100 birim kapasiteli birer araç ile hizmet verilen  $I, J$  ve  $K$  alt turları ele alınabilir. Bu örnekte bir sonraki en büyük araç 300 birim kapasiteye sahiptir ve 100 birim kapasiteli araçtan önemli ölçüde daha pahalıdır.  $I, J$  ve  $K$  alt turlarının kombinasyonuna bir adet 300 birim kapasiteli araç ile hizmet vermek, ayrı ayrı 100 birim kapasiteli üç araç ile hizmet vermekten daha ucuz olabilir. Fakat CS yaklaşımı bu kombine rotayı oluşturmaz. Çünkü CS yaklaşımı, bir seferde sadece iki alt turu birleştirebilmekte ve bu nedenle herhangi bir  $i, j$  müşteri çifti ( $I$  ve  $J$  alt-turlarının uç noktaları) için kombine tasarruf, 300 birim kapasiteli aracın sabit maliyetinin büyük olmasından dolayı negatif olabilmektedir. Bu

<sup>63</sup> B. GOLDEN, A. ASSAD, L. LEVY, F. GHEYSENS, "The fleet size and mix vehicle routing problem", Computers and Operations Research 11(1), 1982, S. 52.

durumda CS yaklaşımı, daha sonra ilave alt-turları kapsamak üzere kullanılabilecek olan 100 birimlik bir kapasiteyi göz ardı ederek, bu kapasitenin atıl kalmasına yol açmaktadır. CS sezgiselinin bu dezavantajı, fırsat tasarrufu algoritmaları olarak adlandırılan çeşitli varyantlarının geliştirilmesini teşvik etmiştir.

#### 3.4.2.2. İyimser Fırsat Tasarrufu Sezgiseli (OOS)

Fırsat tasarrufu algoritmaları, temel olarak CS algoritmasındaki gibi belirlenen rotalama maliyeti tasarrufları ve sabit araç maliyeti tasarruflarının yanı sıra fırsat tasarruflarını da dikkate almaktadırlar. Burada sözü edilen fırsat tasarrufları, birleştirilen alt turlara hizmet veren aracın kullanılmayan (atıl) kapasitesinin bir fonksiyonudur.

Fırsat tasarrufu algoritmalarının birincisi, iyimser fırsat tasarrufu algoritması (optimistic options savings, OOS) olarak adlandırılmaktadır. OOS algoritmasında fırsat tasarrufu, yeni aracın bütün atıl kapasitesine eşit olan bir talebe hizmet verebilen en küçük aracın maliyeti olarak ifade edilmektedir. Örneğin, mevcut araç kapasitelerinin 50, 100 ve 200 birim olmaları durumunda, 40 ve 80 birim toplam talepli iki alt turu 120 birim talepli bir rota şeklinde birleştirilerek elde edilen tasarruf,  $\bar{S}_{ij}$  ve fırsat tasarrufunun toplamına eşit olmaktadır. Buradaki fırsat tasarrufu, 80 birim kullanılmayan kapasiteye eşit bir talebe hizmet verebilecek olan en küçük aracın yani 100 birim kapasiteli aracın maliyetidir. OOS sezgiseli, iyimser olarak R alt turu üzerindeki kullanılmayan kapasiteye eşit bir toplam talebe sahip olan bir S alt turunun daima mevcut olduğunu ve daha sonra S alt turunun R alt turu ile birleştirilerek R'ye hizmet veren araç tarafından kapsanacağını varsaymaktadır. Fırsat tasarrufu, bu kombinasyonun gerçekleştirilmesi halinde ortaya çıkacak olan sabit araç maliyetini göstermektedir. Gelecekteki kombinasyon sonucunda rotalama maliyetinde ortaya çıkabilecek olan tasarruf kolayca tahmin edilememekte ve bu yüzden OOS algoritması, bu tasarrufu içermemektedir. Yukarıda

sunulan bilgilere uygun olarak OOS sezgiselindeki tasarruf formülü aşağıdaki gibidir.<sup>64</sup>

$$S^*_{ij} = S_{ij} + F(Z_i) + F(Z_j) - F(Z_i + Z_j) + F(P(Z_i + Z_j) - Z_i - Z_j) \quad (3.100)$$

Formül (3.100)'deki  $P(Z)$ ,  $Z$  birim toplam talepli bir alt tura hizmet verebilecek en küçük aracın kapasitesidir. Bu durumda yukarıda en son verilen örnek için,  $P(40)=50$ ,  $P(80)=100$  ve  $P(120)=200$  olmaktadır.

#### 3.4.2.3. Gerçekçi Fırsat Tasarrufu Sezgiseli (ROS)

Gerçekçi fırsat tasarrufu sezgiseli (realistic options savings, ROS), iki temel varsayıma dayanmaktadır. Birinci varsayıma göre, fırsat tasarrufunun tek amacı, daha büyük araçların kullanılmaları kârlı görüldüğü zaman bunu yerine getirmek olmaktadır. Bu yüzden iki alt turun birleştirilmesi, daha büyük bir aracın kullanılmasını gerektirmedikçe, tasarruf formülünde fırsat tasarrufu bulunmamalıdır. Eğer alt turların kombinasyonuna hizmet verebilecek araç, şu anda kullanılmakta olan araçların herhangi birinden daha büyük ise, bir araç eşiği ile karşılaşıldığı söylenmektedir. Buna göre sadece bir araç eşiği ile karşılaşılmaması durumunda tasarruf formülüne fırsat tasarrufu dahil edilmelidir.

ROS metodundaki ikinci varsayım, bir alt turun toplam talebinin bir aracın kapasitesine yakın olması eğilimidir. Bu yüzden, gelecekteki bir kombinasyonda kullanılabilir olan aracın, kullanılmayan kapasiteye hizmet verebilecek en küçük araç olduğu varsayımı yerine, kullanılmayan kapasite içine sıkıştırılabilecek en büyük aracı kullanmak muhtemelen daha uygundur. Bu araç, OOS algoritmasının kullandığından bir küçük olan araçtır. Ancak kullanılmayan (atıl) kapasitenin bir aracın kapasitesine eşit olması durumunda, her iki algoritma da ayrı aracı kullanmaktadır.

ROS algoritmasında kullanılan tasarruf formülü aşağıdaki gibidir.<sup>65</sup>

<sup>64</sup> Golden ve diğerleri, 1982, Agm., S.53.

<sup>65</sup> Golden ve diğerleri, 1982, Agm., S. 55.

$$S'_{ij} = \bar{S}_{ij} + F'(P(Z_i + Z_j) - Z_i - Z_j) \quad (3.101)$$

Burada görülen  $F'(Z)$ , kapasitesi  $Z$  ye eşit veya daha küçük olan en büyük aracın sabit maliyetini gösteren bir süreksiz fonksiyondur.  $Z$ 'nin en küçük aracın kapasitesinden daha küçük olması durumunda  $F'(Z)=0$  dir. Birer uç noktaları  $i$  ve  $j$  olan ve  $Z_i$  ve  $Z_j$  ( $Z_i \geq Z_j$ ) toplam taleplerine sahip olan  $I$  ve  $J$  alt turları için; eğer  $F(Z_i + Z_j) = F(Z_i)$  ise  $\bar{S}_{ij}$ , CS yaklaşımındaki gibi  $i$  ve  $j$  müşterilerini birleştirmekten kaynaklanan tasarruftur. Bununla birlikte eğer  $F(Z_i + Z_j) > F(Z_i)$  ise, bir araç eşiği ile karşılaşmış olmakta ve fırsat tasarrufunun dikkate alınması gerekmektedir. Toplam tasarruf hesaplandıktan sonra ROS algoritması CS ile aynı tarzda devam etmektedir.

#### 3.4.2.4. Gerçekçi Fırsat Tasarrufu Sezgiseli (ROS-v)

ROS-v algoritması, her bir problem için bir çok farklı çözüm üreterek, bunlar arasından en az maliyetli olanını final çözümü olarak seçmeyi amaçlamaktadır. Algoritma bunu başarmak için,  $v$  ile gösterilen bir rota şekil parametresi kullanmakta ve  $i$  ve  $j$  düğümlerini birleştirmekten kaynaklanan rota tasarrufunu aşağıdaki şekilde değiştirmektedir.

$$S_{ij} = C_{oi} + C_{oj} - vC_{ij} \quad (3.102)$$

Bu durumda ROS-v algoritmasındaki toplam tasarruf ifadesi aşağıdaki gibidir.<sup>66</sup>

$$S_{ij}'' = S_{ij}' + (1-v)C_{ij} \quad (3.103)$$

Herhangi bir FSMVRP de,  $v$  nin değeri sıfırdan başlanarak her bir deneme için çok küçük aralıklarla (örneğin 0,1 aralığı ile) değiştirilmektedir.<sup>67</sup>

<sup>66</sup> Bodin ve diğerleri, 1983, Agm., S. 108.

<sup>67</sup> Bodin ve diğerleri, 1983, Agm., S. 109.

ROS-v yaklaşımının ardındaki mantık oldukça nettir. Bu mantığa göre, bir düğümün herhangi bir tura bir kez tahsis edilmiş olması durumunda, bu düğümün daha sonraki bir aşamada başka bir tura tahsis edilmesi söz konusu olamaz. Algoritmanın ilk aşamalarında iki bağlantının hemen hemen denk oldukları görülebilir ve bunlar arasında yapılacak seçim, final çözümü üzerinde çok büyük etkiye sahip olabilir. v yı değiştirmek, algoritmanın iyi bir çözüm elde etmesi için zorunlu olan çeşitliliği de kapsamayı vermektedir.

### 3.4.3. Dev Tur (GT) Algoritmaları

Dev tur (gient tour, GT) algoritmaları, FSMVRP leri çözmek için tasarruf metodlarından önemli ölçüde farklı bir yaklaşım sergilemektedirler.

GT algoritmaları önce rota-sonra küme sezgiselleri sınıfına giren iki-adım algoritmalarıdır. Birinci adımda bütün müşterileri ziyaret eden bir tur oluşturulmaktadır. İkinci aşamada ise bu dev tur, her biri problem kısıtlarına uygun olan ve her iki uçlarından depoya bağlı olan bir çok alt tura bölünmektedir.

GT algoritmasının ikinci adımında çözülmesi gereken tur parçalama problemi, çok sayıda en kısa yol (shortest path) problemine dönüştürülebilmektedir. Bu nedenle çeşitli dev tur algoritmaları, esas olarak başlangıç turunun hangi yolla oluşturulduğuna göre farklılık göstermektedirler. Başlangıç turunun oluşturulması için kullanılabilecek pek çok yol vardır ve bu yollardan bir çoğu Bölüm 3.1 de açıklanmıştır. TSP lerin çözümlerinde esas amaç rotalama maliyetini minimize etmek olduğu için komşu müşterilerin birbirlerine yakın olmaları garanti edilmekte fakat müşteri taleplerini kümelemenin sabit araç maliyetleri üzerindeki etkisi hiçbir şekilde dikkate alınmamaktadır. Algoritmanın tur parçalama bileşeni, müşteri taleplerini kümeleme faktörünü analize dahil etmekte ve çok sayıda uygun çözümü etkili bir tarzda mukayese etme imkanı vermektedir.

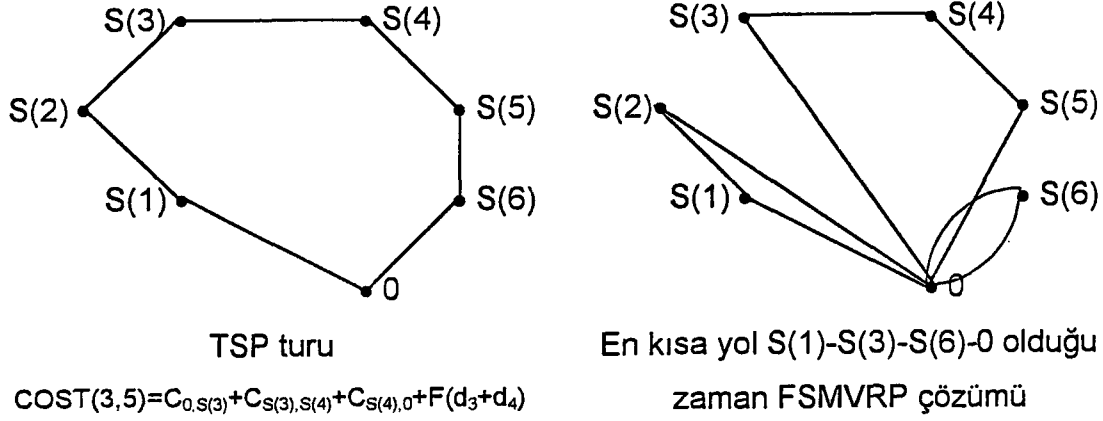
### 3.4.3.1. Tek Bölümlü GT Algoritması (TGT)

TGT algoritması, merkez depodan başlayan, her bir müşteriyi kesinlikle bir kere ziyaret eden ve depoya geri dönen bir tur ile başlamaktadır. 0-S(1)-S(2)...S(n)-0 şeklinde de yazılabilen bu başlangıç TSP turunun alt-turlara bölünmesi için bir en kısa yol problemi çözülmektedir. En kısa yol probleminde, S(k) ve S(m) (k<m) müşterileri arasındaki uzaklık COST (k, m) ile gösterilmekte ve müşterilere S(k), S(K+1),...,S(m-1) sırası ile hizmet vermenin toplam maliyetini ifade etmektedir. (COST(k,n+1), S(k) müşterisi ile depo arasındaki uzaklıktır). Tabi ki kullanılan araç böyle bir alt-tur için yeterli olan en küçük araçtır. Bu durumda COST(k, m) formülü aşağıdaki gibi ifade edilmektedir.<sup>68</sup>

$$\text{COST}(k,m)=C_{0,S(k)}+\sum_{r=k}^{m-2} C_{S(r), S(r+1)}+C_{S(m-1),0}+F\left(\sum_{r=k}^{m-1} d_{S(r)}\right) \quad (3.104)$$

(3.104) deki ilk üç terimin toplamı, toplam rotalama maliyetini vermektedir. Dördüncü terim, sabit araç maliyetidir. Eğer S(k), S(k+1),...,S(m-1) müşterilerinin toplam talebi  $\left(\sum_{r=k}^{m-1} d_{S(r)}\right)$  en büyük aracın kapasitesinden daha büyük ise COST (k,m)= $\infty$  alınmaktadır. En kısa yol problemi için uzaklıklar belirlendikten sonra, S(1) müşterisinden (turdaki ilk müşteriden) depoya en kısa yol aranmaktadır. Bütün kanalların yönleri ileriye doğru oldukları için, müşteriler turda ortaya çıkış sıralarına göre etiketlenmektedirler. Eğer en kısa yol probleminin çözümü p adet kanal gerektirirse TGT çözümü p adet alt-tura sahip olmaktadır (Şekil 3.14).

<sup>68</sup> Golden ve diğerleri, 1982, Agm., S. 56.



Şekil 3.14. TGT algoritması

TGT algoritması, tasarruf algoritmaları gibi dev turda müşterilerin ortaya çıkış sıralamasına bağlı olarak tatmin edici olmayan sonuçlar üretebilmektedir. Bu nedenle bir çok iyi TSP turları ile başlayarak, her bir tura en kısa yol algoritması uygulamak çok daha iyi sonuçlar verebilmektedir.

#### 3.4.3.2. Çok Bölümlü GT Algoritması (ÇGT)

ÇGT algoritmasında başlangıç turu, depo hariç olmak üzere bütün müşterileri ziyaret etmektedir. TGT algoritmasında depo, başlangıç turunda sadece bir kere, final çözümünde ise mevcut araç sayısı kere ziyaret edildiği için bu değişiklik bize fazla bir şey kaybettirmemektedir. Esasen depoyu hariç tutmak, hepsi aynı TSP turu üzerinde bulunan çok daha fazla sayıda bölümü ele alma imkanı sağlamaktadır.

$S(1)-S(2), \dots, S(n)$  TSP turu için,  $S(1)-S(2) \dots S(n)-S(n+1) \dots S(n+M)$  katarında,  $S(n+i)$  ( $i=1,2,\dots,M$ ) düğüm  $S(i)$  nin bir kopyasıdır ve  $M$  nin değeri aşağıdaki gibi belirlenmektedir.

$$\sum_{r=1}^{M-1} d_{S(r)} < \text{en büyük aracın kapasitesi} \leq \sum_{r=1}^M d_{S(r)} \quad (3.105)$$

$S(k)$  ve  $S(m)$  müşterileri arasındaki uzaklık TGT algoritmasındaki gibi belirlenmektedir.  $S(i)$ , ( $i=1,2,\dots,M$ )  $S(i)$  ve  $S(n+i)$  arasındaki en kısa yolun

uzunluğu olarak ifade edilmektedir. Bu durumda  $\min_{i=1,2,\dots,M} S(i)$  ye karşılık gelen yol, başlangıç TSP turunun optimal parçalamasını vermektedir. Burada her bir yolun  $\{S(1), S(2), \dots, S(M)\}$  setindeki en az bir düğümden geçmesi gerektiği için sadece  $i=1,2,\dots,M$  üzerinden minimizasyon yeterli olmaktadır.<sup>69</sup> TGT algoritması gibi ÇGT algoritması da bir çok farklı TSP turları için tekrarlanarak daha iyi çözümler üretilebilmektedir.

#### 3.4.4. İyileştirme Algoritması

ÇGT algoritması oldukça iyi sonuçlar vermesine rağmen iyileştirme algoritmaları daima ele alınmaya değerler. Esasen TSP turu, komşu düğümlerin birbirlerine önemli ölçüde yakın olmalarını garanti etmekte fakat her bir aracın seyahatine depodan başlayıp depoda sona erdirmeye zorunluluğunu dikkate almamaktadır. Ayrıca parçalama süreci, çok fazla sayıda uygun çözümü birbirleri ile mukayese etmesine rağmen bu sayı hala bütün uygun çözümlerin çok küçük bir kısmını oluşturmaktadır. Bunun bir sonucu olarak bir parçalama algoritması ile elde edilen bir çözüm bir iyileştirme algoritması ile daha da iyileştirilebilir.

Golden, Assad, Levy ve Gheysens (1982), bu amaçla TSP iyileştirme sezgisellerinin adaptasyonları olan iki iyileştirme algoritması önermektedirler. Söz konusu algoritmalarda önce, ÇGT algoritması ile bir başlangıç çözümü oluşturulmakta ve bu çözümdeki araç sayısına eşit sayıda depo kopyası dahil edilerek denk bir TSP turu inşa edilmektedir. Şekil 3.14 deki üç alt-turlu çözüm başlangıç çözümü olarak alınır ve deponun kopyaları c1, c2 ve c3 ile gösterilirse denk bir TSP turu, c1-S(1)-S(2)-c2-S(3)-S(4)-S(5)-c3-S(6) olmaktadır. İyileştirme algoritmasının halen kullanılmakta olandan daha fazla sayıda araç kullanabilmesine imkan vermek için tura iki ilave depo düğümü sokulmaktadır. Bu durumda TSP turu; c1-S(1)-S(2)-c2-S(3)-S(4)-S(5)-c3-c4-c5-S(6) olmaktadır. Böyle bir oluşumda iki depo kopyası arasındaki uzaklık sıfıra eşit alınmaktadır.

<sup>69</sup> Golden ve diğerleri, 1982, Agm., S.57.

Kullanılan iyileştirme algoritmaları, TSP iyileştirme sezgisellerinin ortalama maliyetlerinin yanı sıra sabit maliyetleri de dikkate alan ve her seferinde uygunluk kontrolü yapan adaptasyonlarıdır. Birinci iyileştirme algoritması (MGT+2-opt), 2-opt algoritmasının bir adaptasyonudur.

Bu algortmada mevcut çözümü iyileştiren bir uygun değişim kalmayıncaya kadar bütün kanal çiftlerinin birbirleri ile değişimleri test edilmektedir.

İkinci iyileştirme algoritması, OROPT algoritmasının bir adaptasyonu olan MGT+OROPT algoritmasıdır. Bu yöntem, bütün 2-değişimleri ve bir-iki veya üç komşu düğümlük bir katarın başka iki düğüm arasına sokulması anlamındaki 3-değişimleri ele almaktadır. Bu yüzden OROPT, ele alınması gereken değişimlerin sayısını sınırlayarak 3-opt algoritmasından çok daha hızlı çalışmaktadır.

#### 3.4.5. Salhi ve Rand (1993) Sezgisel Çözüm Tekniği

Salhi ve Rand (1993) tarafından FSMVRP için geliştirilmiş olan sezgisel algoritma, yedi aşamalı bir yöntemi benimsemektedir.<sup>70</sup> Yöntemin her bir aşamasında yeni ve daha az toplam maliyetli bir filo belirlenmektedir.

Bazı aşamalarda ortalama maliyeti artmasına rağmen, bu maliyet artışı araç maliyetlerindeki azalma ile dengelenmekte ve toplum maliyetin minimize edilmesi amacı gerçekleştirilmeye çalışılmaktadır. Söz konusu algoritmanın adımları aşağıda özetlenmektedir.

Adım 1. Belirli bir araç kapasitesini kullanarak, VRP'yi çöz ve çözümü (S) kaydet.

---

<sup>70</sup> S. SALHI, G.K.RAND, "Incorporating vehicle routing in the vehicle fleet composition problem", European Journal of Operational Research 66(3), 1993, S. 313-330.

Adım 2. Her bir rotaya hizmet vermek için yeterli olan en küçük aracı belirlemek için, eşleştirme algoritmasını kullan ve çözümü (P) kaydet.

Adım 3. Belirli bir rotayı sistemden çıkartıp bu rotanın müşterilerini diğer rotalara sokmanın daha ekonomik olup olmadığını görmek için indirgeme sürecini uygula. Eğer pozitif kazanç elde edilirse SAYAÇ=1 yap.

Adım 4. Sıra ile aşağıdaki süreçleri uygula.

- Yeniden tahsis süreci : mevcut bir rotaya daha küçük bir araç ile hizmet verilmesine imkan sağlamak için müşterileri bu rotadan çıkart ve diğer rotalara sok.

-Birleştirme süreci : Herhangi iki rotanın daha büyük bir araç gerektirecek daha uzun bir rota içinde birleştirilmesi kazançlı ise, bunu yerine getir. Daha başka iyileşme yapılabileceğini test etmek için yeniden tahsis sürecini kullan.

-Bölüştürme süreci : Herhangi bir rotayı daha küçük birkaç rotaya bölmek kazançlı ise bunu uygula. IND=1 yap ve bu yeni oluşan rotaların bölüştürme sürecinden etkilenmemiş olan diğer rotalar ile olası kombinasyonlarını test et. Bu test, BÖL/BRLŞ ile gösterilmektedir.

-Değiş-tokuş süreci : bazı rotalara daha küçük araçlar ile hizmet verilebilmesini sağlamak için, bu rotaların bazı müşterilerini değiş tokuş et.

-Gevşetme süreci : Bu süreç hem birleştirme hem de bölüştürme süreçlerinin bir uzanımıdır. Mesela, belirli iki rotayı birleştirmek, önemli bir atıl kapasiteye sahip olabilecek daha büyük ve dolayısıyla daha fazla sabit maliyetli bir araç

gerektirdiğinden dolayı ekonomik olmayabilir. Bu durumda bazı müşterilere yeniden tahsis süreci uygulayarak bir küçük aracı kullanmak cazip olabilir (bu proses GVŞ/BRLŞ ile gösterilmektedir). Benzer olarak bölüştürme süreci, kapasite açısından kötü kullanılabilecek bir ilave araç gerektirebilir ve bu durum daha sonra reddedilecek birçok çözüme yol açabilir. Bu durumda bazı müşterileri diğer rotalara yerleştirmek ve başlangıçta bölüştürme ile elde edilenden daha küçük bir araç kullanmak kazançlı olabilir (bu süreç GVŞ/BÖL ile gösterilmektedir).

Adım 5. Eğer  $IND=0$  ise ve GVŞ/BÖL süreci pozitif bir kazanç vermemiş ise, Adım 6 ya git. Aksi halde indirgeme sürecini kullan (yeni bazı küçük rotaların sistemden çıkartılmaları toplam maliyeti azaltabilir) ve Adım 6'ya git.

Adım 6. Eğer  $SAYAÇ=1$  ise, Adım 2'deki sonuçları (P) kullan,  $SAYAÇ=0$  ve  $IND=0$  yap ve Adım 4 e git. Aksi halde, yeni en düşük maliyetli çözümü kaydet ve Adım 7 ye git.

Adım 7. Eğer başka araç tipleri kombinasyonu kullanarak problemi incelemek istersen, bir kombinasyon seç, Adım 1 deki sonuçları (S) kullan ve Adım 2 ye git. Aksi halde Adım 8 e git.

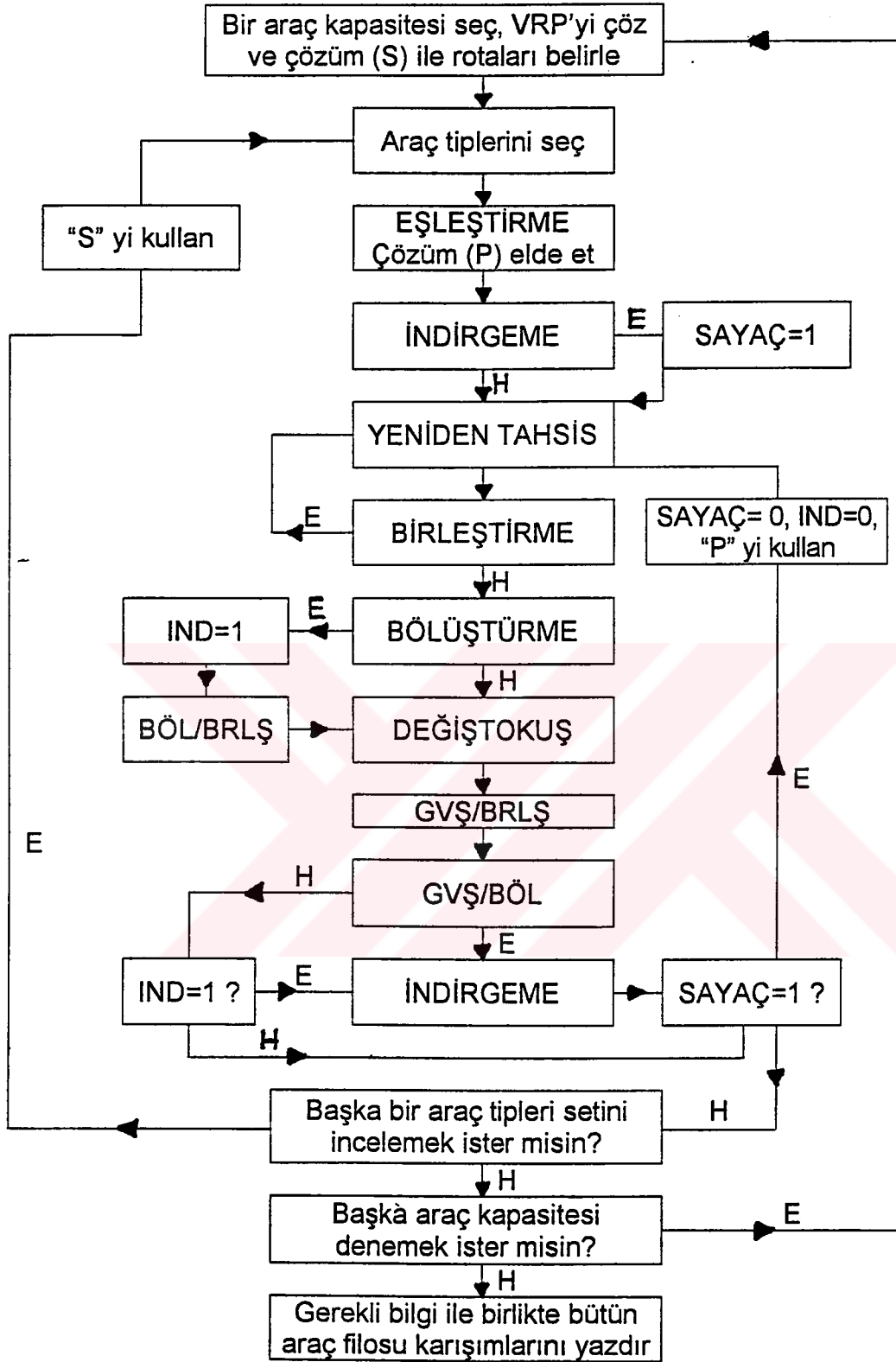
Adım 8. Eğer başka bir araç kapasitesi kullanarak bütün süreci tekrarlamak istersen, bir kapasite değeri seç ve Adım 1 e git. Aksi halde, DUR. En düşük toplam maliyetli çözümü, FSMVRP nin çözümü olarak seç. Her bir süreçte bulunan bütün olası filo karışımlarını da toplam maliyetleri ile birlikte yazdır.

Şekil 3.15 de akış diyagramı verilen sezgisel algoritma esnasında uygunluk terimi, zaman (mesafe) kısıtlarına uygunluğu ifade etmektedir. Çünkü kapasite kısıtlarına uygunluk otomatikman sağlanmaktadır.

Algoritmanın birinci adımında herhangi bir uygun rotalama sezgiseli kullanılabilir. Çünkü farklı rotalama sezgiselleri farklı çözümler üretebilmelerine rağmen, daha sonra her bir çözüm bir tasfiye süreçleri seti ile iyileştirileceği için rotalama sezgiselinin seçimi final çözüm üzerinde fazla bir etkiye sahip olmamaktadır. Dolayısıyla bu aşamada yapılacak bir seçim sadece hesaplama süresini ve çabasını dikkate almaktadır.

Adım 1 deki rotalama sezgiselinde, klasik VRP çözülürken araç kapasitesinin toplam rota talebini ne kadar aştığına yani aracın atıl kapasitesine dikkat edilmemektedir. Buna karşın Adım 2 de, birbirlerinden farklı kapasitelere sahip olan bir çok tipte araç vardır ve her bir rotaya toplam talebini karşılayabilecek en küçük araç ile hizmet verilmesi, toplam araç maliyetlerini önemli ölçüde azaltabilmektedir. Daha sonra, tek başlarına birer rota oluşturan büyük müşteriler başka rotalara sokularak çözüm iyileştirilmeye çalışılmaktadır.

Adım 3'deki indirgeme sürecinde, belirli bir rotanın müşterilerinin yeni rotalara tahsis edilmeleri için, önce her bir müşterinin bir rotaya sokulmasının uygun olup olmadığı kontrol edilmektedir. Uygunluğun bozulmaması durumunda müşterinin rotaya sokulabileceği en iyi yer minimum extra maliyete göre belirlenmektedir. Bu işlemler bütün rotalar için tekrarlandıktan sonra söz konusu müşterinin tahsis edilebileceği en iyi rota ve o rotadaki en iyi yer belirlenmiş olmaktadır. Müşterileri başka rotalara tahsis edilmek istenen rotadaki bütün müşteriler için bu süreç tekrarlanarak, rota tamamen elimine edilmektedir.



Şekil 3.15. FSMVRP için Salhi, Rand (1993) Sezgiselinin akış diyagramı

İndirgeme sürecinde müşterilerin değerlendirilme sıraları, sürecin başarısını etkileyen bir faktördür. Büyük müşterilerin herhangi bir rotaya geç aşamalarda sokulmaları, bu rotanın kapasite kısıtı açısından uygunluğunun bozulma olasılığını yükselmektedir. Küçük müşterilerin diğer rotalara ileri aşamalarda sokulma şansları, büyük müşterilerinkinden daha fazladır. Çünkü araçların kullanılmamış kapasiteleri giderek azalmaktadır. Bu nedenle eliminasyon konusu olan rotanın müşterilerinin taleplerine göre büyükten küçüğe doğru sıralanarak indirgeme sürecine en büyük müşteriden başlamak uygun bir yoldur. En büyük müşteri ile başlamanın diğer bir avantajı, bunun hesaplama süresini azaltabilmesidir. Çünkü eğer bir müşteri başka hiçbir rotaya sokulamazsa, indirgeme sürecine devam etmenin bir anlamı kalmayacaktır.

İndirgeme süreci bütün rotalar için uygulanarak en büyük toplam kazancı veren rota seçilmekte ve elimine edilmektedir. Bu aşamada yeni ilave müşterilerin sokuldukları bütün rotalara kendi içlerinde 2-opt ve 3-opt algoritmaları uygulanarak rotalar iyileştirilebilmektedir.

Adım 4 deki yeniden tahsis sürecinde, rotalama maliyetleri artmakta fakat bir yandan da toplam maliyet azalmaktadır. Bu süreçte bir rotadan çıkartılan müşteriler, en düşük maliyetli giriş kuralına uygun olarak tekrar başka rotalara yerleştirilmektedirler. Daha ileri testler için, sadece uygun kalan rotalar ele alınmaktadırlar. Burada amaç, bir rotaya daha küçük bir araç ile hizmet verilebilmesini sağlamak için, bu rotanın bazı müşterilerini yeterli yedek kapasitesi bulunan araçlara tahsis ederek maliyeti düşürmektir.

Birleştirme sürecinde, en büyük araç ile hizmet verilenler dışında kalan bütün rotaların birbirleri ile kombinasyonları incelenmektedir. Rotaların birleştirilmelerindeki amaç, iki küçük araç ile hizmet verilen rotaları daha büyük bir tek araç ile hizmet verilen bir rotaya dönüştürerek maliyet tasarrufu sağlamaktır. Sürecin sonunda bütün olası kombinasyonlar arasından en büyük tasarrufu sağlayan kombinasyon seçilmekte ve seçilen bu iki rota birer uçlarından birleştirilmektedir.

Birleştirmeden sonra oluşan rotaya 2-opt ve 3-opt algoritmaları uygulanarak iyileştirme yapılmakta ve bu rotaya daha küçük bir araç ile hizmet verilebilmesini sağlamak için yeniden tahsis prosesi uygulanmaktadır.

Bölüştürme sürecinde, sadece en küçükten daha büyük araç ile hizmet verilen ve ikiden fazla müşteriye sahip olan rotaların iki veya daha fazla rotaya bölünmeleri ele alınmaktadır. Amaç, gene toplam maliyetin azalmasını sağlamaktır. Her bir araç tipi için, olası daha küçük rotaların sayısı (alt sınır), büyük rotanın toplam talebinin söz konusu aracın kapasitesine bölünmesi ile belirlenmektedir. Küçük rotalar, rotadaki birinci müşteriden başlanarak ardışık olarak oluşturulmaktadırlar. Bu süreçte, herhangi bir müşterinin herhangi bir rotaya sokulması bu rotanın kapasite açısından uygunluğunu bozuyor ise ve bu rotanın hala başka müşterileri kabul etmeye yetecek atıl kapasitesi mevcut ise, bir sonraki müşterinin bu rotaya girişi kontrol edilmektedir. Bu, araçları tam kapasiteye yakın olarak kullanma imkanı vermektedir. Bu süreç sırasında oluşturulan küçük rotalara genellikle aynı tip araçlar ile ve kalan düşük talepli müşterilere de daha küçük bir araç ile hizmet verilmektedir. Küçük rotaların toplam maliyetlerinin toplamının orijinal büyük rotanın toplam maliyetinden daha küçük olması durumunda rotalama maliyetlerini biraz daha azaltmak için, bu rotalara 2-opt ve 3-opt algoritmaları uygulanabilmektedir.

Yukarıda özetlenen süreç bütün araç tipleri için tekrarlandıktan sonra, söz konusu rota için en fazla kazancı sağlayan çözüm seçilmektedir. Bu aşamada, elde edilen çözümdeki küçük rotalardan herhangi biri bölüştürme sürecinden etkilenmemiş olan bir başka rota ile birleştirilebilmektedir.

Değiş tokuş sürecinde, farklı rotalardaki müşteriler birbirleri ile değiştirilmektedir. Bu durumda her bir rotadaki müşteri sayısının sabit kalacağı açıktır. Yeni rota maliyeti, öncekinden çok daha yüksek olabileceği için bu yöntem son çare olarak başvurulmaktadır.

GVŞ/BRLŞ süreci, birleştirme sürecinden farklıdır. Birleştirme sürecinde sadece pozitif bir iyileşme bulunması halinde yeniden tahsis prosesi uygulanırken GVŞ/BRLŞ sürecinde elverişli şekilde birleştirme uygulayabilmek için bazı müşteriler diğer rotalara tahsis edilmektedirler.

GVŞ/BÖL sürecinde, dikkate alınan küçük rotaların sayısı, bölüştürme sürecindekinden bir eksiktir. Bu daha az sayıda araç kullanılmasını ve tahsis edilmemiş müşterilerin diğer rotalara sokulabilmelerini sağlamaktadır. Bu süreçte ortaya çıkabilecek bir extra rota maliyeti, daha ucuz bir araç kullanarak elde edilen finansal kazanç ile dengelenebilmektedir. Bu süreç, sadece başka rotalara sokulduklarında hiçbir kısıtı bozmayan ve en düşük extra rotalama maliyetini oluşturan müşterilerin yeniden tahsislerini ele almaktadır. Küçük müşterilerin diğer rotalara sokulabilme olasılıkları daha fazla olduğu için burada amaç, küçük müşterileri tahsis edilmemiş halde bırakmaktır. Bunu yapmak için, en büyük müşteriler çekirdek noktalar olarak seçilmekte ve söz konusu rotanın diğer müşterileri yakınlık sıralarına göre bu çekirdek noktalara tahsis edilmektedirler. Pozitif bir kazanç elde edilmesi durumunda, bütün rotalara 2-opt ve 3-opt algoritmaları uygulanarak rotalar iyileştirilmektedir. GVŞ/BÖL süreci bütün araç tipleri için tekrarlandıktan sonra en iyi olan çözüm seçilmektedir.

### 3.5. Stokastik Araç Rotalama Problemi (SVRP)

Çalışmamızın şimdiye kadar incelenen bütün bölümlerinde, bütün müşterilerin deterministik taleplere sahip oldukları kabul edilmişti. Bu bölümde, aşağıda sıralanan farklılıklar ile stokastik araç rotalama problemi (SVRP) ele alınmaktadır.

- (1) Müşteri talebi, bilinen bir ihtimal dağılımına sahip olan bir rastgele değişkendir.
- (2) Rotalar, gerçek talepler bilinmeden önce oluşturulmalıdırlar.

- (3) Amaç, beklenen seyahat mesafesini minimize etmektir. Fakat bazı müşterilere özel bir tur üzerinde hizmet verilememesi durumunda başka maliyetler ile karşılaşılabilmektedir.

SVRP en az VRP kadar zor bir problemdir. Esasen probabilistik taleplerden dolayı kısıtlardaki ve/veya amaç fonksiyonundaki lineer olmayan ilişkilerin varlığı, bu problemi çok daha zor bir hale getirebilmektedir. Bu nedenle bu problem için de sadece sezgisel çözüm metodlarından söz edilmektedir.

SVRP lerin çözümleri konusunda oldukça az sayıda araştırma yapılmıştır. Bizim bu çalışmada ele aldığımız çözüm yöntemi, SVRP yi VRP ye dönüştürme ve sonra bu dönüştürülmüş probleme mevcut VRP çözüm algoritmalarından birini uygulamaya dayanmaktadır. Söz konusu çözüm yönteminde kullanılan "yapay kapasite" terimi, bir VRP çözüm algoritmasındaki gerçek kapasite yerine kullanılacak bir kapasiteyi ifade etmektedir. Bu yapay kapasite gerçek kapasiteden daha küçüktür. Rotalama, yapay talepler olarak ortalama müşteri taleplerini alarak ve araçları yapay kapasitelerine kadar yükleyerek gerçekleştirilmektedir.

SVRP nin aşağıda sunulan bir matematiksel formülasyonunda, her bir müşteri yerleşimindeki talep bir rastgele değişkendir ve bu durumda problem bir stokastik araç rotalama problemi olmaktadır. Formülasyondaki  $d_i$  talepleri, bilinen bir olasılık dağılımından gelen rastgele değişkenlerdir.  $\alpha$  parametresi ise maximum kabul edilebilir rota başarısızlık parametresidir.<sup>71</sup>

$$\text{Minimize } \sum_k \sum_{ij} c_{ij} X_{ij}^k \quad (3.106)$$

$$\text{Prob} \left\{ \sum_{ij} d_i X_{ij}^k \leq Q \right\} \geq 1 - \alpha \quad (k=1,2,\dots,NK) \quad (3.107)$$

---

<sup>71</sup> Bodin ve diğerleri, 1983, Agm., S. 110.

$$X=[X_{ij}^k] \in S \quad (3.108)$$

Formülasyon (3.106)-(3.108) de;  $c_{ij}$  :  $i$  den  $j$  ye seyahatin maliyeti,  $Q$  : araç kapasitesi,  $NK$  : araçların sayısı,  $S$  : bütün  $m$ -TSP çözümlerinin seti ve

$$X_{ij}^k = \begin{cases} 1, & \text{eğer } i \text{ ve } j \text{ rota } k \text{ üzerinde birleşiyorlar ise} \\ 0, & \text{aksi halde} \end{cases}$$

şeklinde tanımlanmaktadır ve kısıt (3.107) bir şans-kısıtı olarak adlandırılabilir.

Kapasite kısıtına uygunsuzluk durumu için bir cezanın açıkça belirlenemediği durumda bu şans-kısıtlı model kullanışlı olmaktadır. Bu şekilde, rota başarısızlık olasılığına göre müşteri hizmetinin sübjektif bir tahmini elde edilebilmektedir.

Bu bölümde söz edilen çözüm yöntemleri, mevcut VRP çözüm algoritmalarının küçük bir değişiklik ile SVRP lere adaptasyonlarına dayanmaktadır. Aşağıda bir şans-kısıtının denk bir deterministik kısıta dönüştürülmesi açıklanmaktadır.<sup>72</sup> Birçok durumda, bu denk kısıttan yapay kapasite için kapalı-formlu bir ifade türetilebilmektedir. Ele alınan kısıt formu aşağıdaki gibidir.

$$\text{Prob} \left\{ \sum_i a_i Y_i \leq b \right\} \geq 1 - \alpha \quad (3.109)$$

Eğer  $a_i$ , birbirlerinden bağımsız olan ve  $\bar{a}_i$  ortalama ve  $S_i$  standart sapması ile dağılan rastgele değişkenler ise ve  $Y_i$  karar değişkenleri ise,  $\sum_i a_i Y_i$  nin ortalaması  $M = \sum_i \bar{a}_i Y_i$  ve standart sapması  $S = (\sum_i S_i^2 Y_i^2)^{1/2}$  dir. Bu durumda,

---

<sup>72</sup> Bodin ve diğerleri, 1983, Agm., S. 110.

$$\text{Prob}\{(\sum_i a_i Y_i - M) / S \leq T\} = 1 - \alpha \quad (3.110)$$

şartını sağlayan bir T sabiti aranmaktadır. Eğer  $b \geq M + T.S$  ise, eşitsizlik (3.109) un sol taraf toplamı da  $1 - \alpha$  dan daha küçük değildir. Bundan dolayı kısıt (3.109) yerine aşağıdaki deterministik kısıt yerleştirilebilmektedir.

$$M + T.S \leq b \quad (3.111)$$

Bu kısıt aşağıdaki gibi  $y_i$  ye göre yeniden düzenlendiği zaman genellikle doğrusal olmayan bir yapı oluşturmaktadır.

$$\sum_i \bar{a}_i Y_i + T(\sum_i S_i^2 Y_i^2)^{1/2} \leq b \quad (3.112)$$

Bununla birlikte özel şartlar altında bu kısıt, doğrusal bir form alabilmektedir. Açık ki SVRP lerde  $a_i$  ler probabilistik talepleri ve  $Y_i$  ler 0-1 değişkenlerini ve  $b$  araç kapasitesini göstermektedir. Olasılık dağılımlarının gama, binomial, poisson ve negatif binomial dağılımlarındaki gibi  $S^2 = \lambda M$  (bir  $\lambda$  sabiti için) şartına uygun olmaları durumunda, (3.112) aşağıdaki gibi yazılabilmektedir.

$$\sum_i \bar{a}_i Y_i + T(\lambda \sum_i \bar{a}_i Y_i)^{1/2} \leq b \quad (3.113)$$

$$M + T\sqrt{\lambda.M} \leq b \quad (3.114)$$

Bu eşitsizlik uygun şekilde işlendiği zaman,

$$M \leq \bar{b} \quad (3.115)$$

Formundaki basit bir ifade ortaya çıkmaktadır.  $W = T\sqrt{\lambda}$  tanımlayarak ve  $b - M \geq W M^{1/2}$  olduğuna dikkat ederek aşağıdaki eşitsizlik elde edilmektedir.

$$M \leq \frac{2b + W^2 - \sqrt{(W^4 + 4bW^2)}}{2} = \bar{b} \quad (3.116)$$

$Z$  bir birim normal değişken olmak üzere eğer  $T$ ,  $\text{Prob}(Z \leq T) = 1 - \alpha$  olarak belirlenirse bu sonuç Golden ve Yee (1979) tarafından türetilen genel ifadenin aynısıdır.<sup>73</sup> Bununla birlikte bir çok olasılık dağılımı için,  $\bar{b}$  nin (3.116) dakine benzer bir kapalı formlu ifadesi mevcut değildir. Bu nedenle sık sık daha genel bir algoritmaya ihtiyaç duyulmaktadır. Çeşitli araştırmacılar tarafından, SVRP için bir ceza fonksiyonu içeren böyle algoritmalar geliştirilmiştir. Ancak SVRP ler başlı başına ayrı bir araştırma konusu oldukları için burada daha fazla incelenmeyeceklerdir.

### 3.6. Zaman Pencere Arac Rotalama ve Programlama Problemi (VRSPTW)

Bir müşteriler setinde bulunan her bir müşteri hizmetinin o müşteriye ait bir zaman aralığı içinde yerine getirilmesi zorunluluğu söz konusu olduğunda, zaman pencere araç rotalama ve programlama problemi (vehicle routing and scheduling problem with time windows, VRSPTW) ortaya çıkmaktadır. Buradaki "zaman penceresi" terimi herhangi bir müşterinin talep ettiği hizmetin kendisine sunulmasını kabul edebileceği en erken ( $e_i$ ) ve en geç ( $l_i$ ) zamanlar ile belirlenen ve  $[e_i, l_i]$  şeklinde gösterilen zaman aralığını ifade etmektedir. Örneğin bir teslimat probleminde, herhangi bir müşteriye ait teslimatın saat 10:00-11:30 arasındaki herhangi bir zamanda yapılması şartı söz konusu olabilir. Bu durumda bu müşteriye ait rota, söz konusu teslimatın bu zaman sınırları içerisinde yapılabilmesini sağlayacak şekilde oluşturulmalıdır. Depodaki zaman penceresi ise, planlama hattını tayin etmektedir. Yani her bir rota, bu planlama hattı içerisinde başlayıp sona ermek zorundadır.

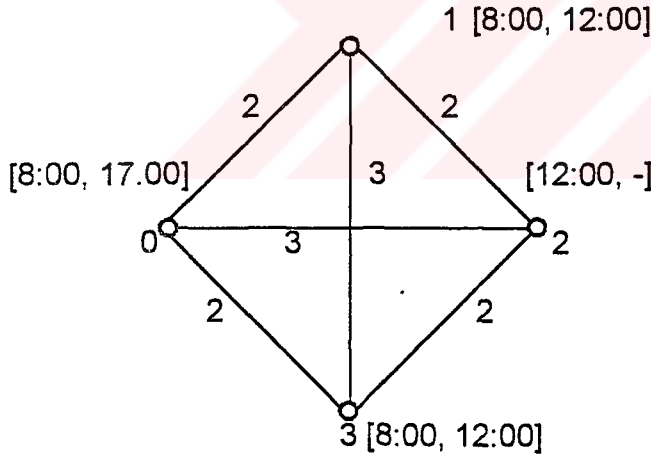
Gerçek yaşamda sabit zamanlı programlar ile çalışan işletme organizasyonlarının yüz yüze oldukları problemlerde zaman pencereleri ortaya çıkmaktadır. Bu tür problemlere örnek olarak, posta teslimatları,

<sup>73</sup> B. GOLDEN, J. YEE, "A framework of probabilistic vehicle routing", AIEE Trans. 11(2), 1979, S. 109-112.

endüstriyel ve evsel atık toplama, işyeri ve okul ulaşım hizmetleri vb. verilebilir.

Zaman pencerelerinin varlığı, araç rotalama problemine mesafe boyutunun yanı sıra zaman boyutunu da ilave etmekte ve bu nedenle problem, rotalama bileşenine ilaveten programlama bileşenini de içermektedir. Bu iki bileşen birbiri ile karşılıklı ve çok yakın etkileşim halindedir ve bu yüzden problem sadece "rotalama problemi" olarak değil "rotalama ve programlama problemi" olarak adlandırılmaktadır.

VRSPWTW, ilave zaman penceresi kısıtlarının varlığından dolayı, VRP den çok daha karmaşıktır. Şekil 3.16 daki örnek, bu karmaşıklığın nedenini açıkça ortaya koymaktadır. Söz konusu örnekte, düğüm 0 (depo) her biri 1/3 birim mal talep eden üç noktaya (1, 2 ve 3) hizmet vermektedir. Araç saat 8:00'e kadar depodan ayrılamamaktadır ve en geç 17:00 de depoya geri dönmek zorundadır. Düğümler arasındaki seyahat süreleri kanallar üzerindeki rakamlar ile verilmektedir.



Şekil 3.16. Zaman pencereli bir rotalama problemi

0-1-2-3-0 optimal bir turdur ve 8 birim uzunluğa sahiptir. Eğer düğüm 1 ve 3 e saat 8:00-12:00 arasında ve düğüm 2 ye saat 12:00 den sonra hizmet verilmesi zorunluluğu varsa bu durumda yukarıdaki optimal tur artık uygun olmayacak ve yeni optimal tur 7 birim uzunluğa sahip 0-1-

2-0 ve 4 birim uzunluğa sahip 0-3-0 şeklindeki iki ayrı turdan oluşacaktır. Bu durumda iki ayrı araca ihtiyaç duyulacağı açıktır.

### 3.6.1. VRSPW İçin Matematiksel Formülasyonlar

Bu bölümde sunulan VRSPW formülasyonları, Soloman ve Desrosiers (1988) tarafından en genel formda geliştirilmiş olan tek depo, homojen filo, dağıtım ve toplama problemi formülasyonlarına dayanmaktadır.<sup>74</sup> Burada kullanılan notasyon standart VRP formülasyonlarında kullanılanlardan oldukça farklıdır. Bu nedenle notasyonun baştan sona yeniden tanımlanması uygun görülmektedir.

$i$  ile indekslenmiş olan  $n$  adet müşteri vardır. Her bir müşteri belirli bir miktardaki malın belirli bir noktadan gene belirli bir başka noktaya taşınmasını istemektedir.  $i$  müşterisinden malın teslim alınacağı nokta  $i$  düğümü ile ve malın teslim edileceği nokta da  $n+i$  düğümü ile gösterilmektedir. 0 ve  $2n+1$  düğümleri sırası ile herhangi bir aracın seyahatinin başlangıç ve bitiş noktaları olarak depoyu göstermektedirler. Bu gösterim şekli, müşterilerin yerleşimleri ile şebekenin düğümleri arasında net bir farklılık oluşturmaktadır. Bununla birlikte farklı düğümler, aynı fiziksel yerleşime karşılık gelebilirler. Bu nedenle,  $N=\{0,1,2,\dots,n, n+1, n+2,\dots,2n, 2n+1\}$  şebekedeki düğümlerin setidir.  $P=P^+ \cup P^-$ , ( $P^+=\{1,2,\dots,n\}$ ,  $P^-=\{n+1, n+2,\dots,2n\}$  ise depo düğümleri dışındaki düğümlerin setidir.  $i$  müşterisi  $d_i$  birim malın  $i$  düğümünden  $n+i$  düğümüne taşınmasını talep etmektedir.  $[e_i, l_i]$ ,  $i$  müşterisine ait "teslim alma zaman penceresi" ve  $[e_{n+i}, l_{n+i}]$  de aynı müşteriye ait "teslim etme zaman penceresi" olarak tanımlanmaktadır.

$K=\{1,2,\dots,|K|\}$ , rotalanacak ve programlanacak olan araçların setidir. Her bir aracın kapasitesi birbirine eşittir ve  $Q$  ile gösterilmektedir.

---

<sup>74</sup> M.M. SOLOMON, J. DESROSIERS, "Time windows constrained routing and scheduling problems", Transportation Science 22(1), 1988, S. 3.

Ayrıca  $[e_0, l_0]$  araçların depodan ayrılma zaman penceresi ve  $[e_{2n+1}, l_{2n+1}]$  de araçların depoya geri dönme zaman penceresidir.

N deki her bir ayırık  $u, w$  düğüm çifti için  $t_{uw}$  ve  $c_{uw}$  sırası ile  $u$  dan  $w$  ye seyahatin süresini ve maliyetini göstermektedir.

Formülasyonlarda üç tip değişken kullanılmaktadır. Bunlardan birincisi 0-1 akış değişkenleri ( $X_{uw}^k, u, w \in N, k \in K, u \neq w$ ), ikincisi zaman değişkenleri ( $T_u, T_o^k, T_{2n+1}^k, u \in P, k \in K$ ) ve üçüncüsü de yük değişkenleri ( $Y_u, u \in P$ ) dir. 0-1 karar değişkeni  $X_{uw}^k$ ,  $k$  aracının  $u$  düğümünden  $w$  düğümüne seyahat etmesi halinde "1" e, aksi halde "0" a eşittir ( $k \in K$  ve  $u, w \in N, u \neq w$ ).  $T_u$   $u$  müşterisinde hizmete başlama zamanını ( $u \in P$ ),  $T_o^k$   $k$  aracının depodan ayrılma zamanını ve  $T_{2n+1}^k$   $k$  aracının depoya geri dönüş zamanını ( $k \in K$ ) göstermektedir. Araçların depodan boş olarak ayrıldıkları yani  $Y_0 = 0$  olduğu kabul edilmektedir. Son olarak  $f_u(T_u)$ ,  $u$  düğümündeki hizmetin  $T_u$  zamanında başlamasına dair cezayı ve  $g_k(T_{2n+1}^k - T_o^k)$  da  $k \in K$  aracının rota süresine dair cezayı göstermektedir. Bu cezalar aynı birim ile ( $C_{uw}$  maliyetleri ile) ifade edilmelidirler. Şimdi VRSPWTW'nin matematiksel formülasyonları aşağıdaki gibi ifade edilebilir.

$$\text{Min } \sum_{k=1}^K \sum_{u=0}^{2n-1} \sum_{w=0}^{2n-1} c_{uw} X_{uw}^k + \sum_{u=1}^{2n} f_u(T_u) + \sum_{k=1}^K g^k(T_{2n+1}^k - T_o^k) \quad (3.117)$$

$$\sum_{k=1}^K \sum_{w=0}^{2n-1} X_{uw}^k = 1, \quad u=\{1,2,\dots,n\} \quad (3.118)$$

$$\sum_{w=0}^{2n-1} X_{uw}^k - \sum_{w=0}^{2n-1} X_{wu}^k = 0, \quad u=\{1,2,\dots,n, n+1,\dots,2n\}, k=\{1,2,\dots,K\} \quad (3.119)$$

$$\sum_{w=1}^n X_{ow}^k = 1, \quad k=\{1,2,\dots,K\} \quad (3.120)$$

$$\sum_{u=n+1}^{2n} X_{u,2n+1}^k = 1, \quad k=\{1,2,\dots,K\} \quad (3.121)$$

$$\sum_{w=0}^{2n+1} X_{uw}^k - \sum_{w=0}^{2n+1} X_{w,n+u}^k = 0, \quad u=\{1,2,\dots,n\}, k=\{1,2,\dots,K\} \quad (3.122)$$

$$T_u + S_u + t_{u,n+u} \leq T_{n+u}, \quad u=\{1,2,\dots,n\} \quad (3.123)$$

$$X_{uw}^k = 1 \Rightarrow T_u + S_u + t_{uw} \leq T_w, \quad u,w=\{1,2,\dots,n,n+1,\dots,2n\}, k=\{1,2,\dots,K\} \quad (3.124)$$

$$X_{ow}^k = 1 \Rightarrow T_o^k + t_{ow} \leq T_w, \quad w=\{1,2,\dots,n\}, k=\{1,2,\dots,K\} \quad (3.125)$$

$$X_{u,2n-1}^k = 1 \Rightarrow T_u + S_u + t_{u,2n+1} \leq T_{2n+1}^k, \quad u=\{n+1,n+2,\dots,2n\}, k=\{1,2,\dots,K\} \quad (3.126)$$

$$e_u \leq T_u \leq l_u \quad u=\{1,2,\dots,n, n+1,\dots,2n\} \quad (3.127)$$

$$e_o \leq T_o^k \leq l_o \quad k=\{1,2,\dots,K\} \quad (3.128)$$

$$e_{2n+1} \leq T_{2n+1}^k \leq l_{2n+1} \quad k=\{1,2,\dots,K\} \quad (3.129)$$

$$X_{uw}^k = 1 \Rightarrow Y_u + d_w = Y_w, \quad u=\{1,2,\dots,n,n+1,\dots,2n\}, w=\{1,2,\dots,n\}, k=\{1,2,\dots,K\} \quad (3.130)$$

$$X_{uw}^k = 1 \Rightarrow Y_u - d_w - d_{w-n} = Y_w, \quad u=\{1,2,\dots,n,n+1,\dots,2n\}, w=\{n+1, n+2,\dots,2n\} \\ k=\{1,2,\dots,K\} \quad (3.131)$$

$$X_{ow}^k = 1 \Rightarrow Y_o + d_w = Y_w \quad w=\{1,2,\dots,n\}, k=\{1,2,\dots,K\}, u=\{1,2,\dots,n\} \quad (3.132)$$

$$Y_o = 0, \quad 0 \leq Y_u \leq Q \quad u=\{1,2,\dots,n\} \quad (3.133)$$

$$X_{uw}^k = 0 \text{ veya } 1 \quad u,w=\{0,1,2,\dots,n,n+1,\dots,2n+1\}, k=\{1,2,\dots,K\} \quad (3.134)$$

Amaç fonksiyonu (3.117) de, toplam seyahat maliyeti müşterilere çok erken veya çok geç hizmet sunmaya dair ceza ve maximum rota uzunluğu kısıtını aşmaya dair ceza toplamalarının minimize edilmesi amaçlanmaktadır. Kısıt (3.118)-(3.121) ve (3.134), minimum maliyetli akış problemini oluşturmaktadırlar. Kısıt (3.122), aynı k aracının hem u yu hem de n+u yu ziyaret etmesini sağlamaktadır. Kısıt (3.123), u düğümünün n+u düğümünden önce ziyaret edilmesini sağlayan öncelik kısıtıdır. Kısıt

(3.124)-(3.126), rotalar ile programların birbirlerine uygun olmaları şartını gösterirken, kısıt (3.127)-(3.129) ise zaman penceresi kısıtlarıdır. Kısıt (3.130)- (3.132) rotalar ile araç yükleri arasındaki uygunluk şartını göstermektedirler ve kısıt (3.133) de kapasite kısıtıdır. Formülasyon ayrıca bir rota süresi (uzunluğu) kısıtı  $(l_{2n+1}-e_0)$  içermektedir. Formülasyonda alt-tur eliminasyon kısıtları bulunmamaktadır. Çünkü kısıt (3.124), alt-tur eliminasyonunu sağlamaktadır.

Yukarıda en genel formu ile verilen formülasyonda, tek araç varyantını elde etmek için  $|K|=1$  almak ve kısıt (3.122) yi elimine etmek yeterli olmaktadır. Ayrıca filo hacminin minimizasyonunu da dikkate almak için, amaç fonksiyonuna  $\sum_{k=1}^K \sum_{w=1}^n FX_{ow}^k$  terimini ilave etmek ve kısıt (3.120) ve (3.121) deki "=" yerine " $\leq$ " yerleştirmek gerekmektedir. Formülasyonda sadece dağıtım veya sadece toplama problemi elde etmek için,  $P=P^+=P^-=\{1,2,...,n\}$  ve  $N=PU\{0,n+1\}$  tanımlayarak, kısıt (3.122), (3.123) ve (3.131) silmek ve toplama problemi için  $d_w \geq 0$ , dağıtım problemi için ise  $d_w \leq 0$  yaparak kısıt (3.133) deki  $Y_u \leq D$  yi silmek ve  $Y_0 = D$  almak yeterli olmaktadır.

VRP NP-hard olduğu için, ilave kısıtlar içeren VRSPWTW de NP-hard dır. Bu yüzden bu problemin çözülmesi için de sezgisel algoritmalar üzerinde yoğunlaşmıştır. VRSPWTW için sezgisel çözüm algoritmalarının büyük bir kısmı VRP algoritmalarının uzanımlarıdır ve çözüm süreci içerisine mesafe boyutunun yanı sıra zaman boyutunu da dahi etmeye dayanmaktadırlar.

### 3.6.2. VRSPWTW İçin Sezgisel Çözüm Algoritmaları

Bu bölümde sunulan sezgisellerde, notasyon kolaylığı amacı ile homojen bir araç filosu varsayımı yapılmaktadır. Herhangi bir  $i$  müşterisine ( $i=1,2,...,n$ ) hizmet, en erken  $e_i$  ve en geç  $l_i$  ile tanımlanan bir zaman penceresi içinde ve müşterinin kabul edebileceği bir  $T_i$  zamanında başlayabilir. Bu hizmetin tamamlanması  $S_i$  kadar bir süre almaktadır.

Bundan dolayı eğer bir araç direkt olarak  $i$  müşterisinden  $j$  müşterisine gider ve  $j$  müşterisine çok erken varırsa beklemektedir. Yani,

$$T_j = \max \{e_j, T_i + S_i + t_{ij}\} \quad (3.135)$$

olarak belirlenmektedir. Burada hizmetlerin başlama zamanları  $T_i$ , ( $i=1,2,\dots,n$ ) karar değişkenleridir.

Ele alınan bütün sezgisellerde  $i$  müşterisinden  $j$  müşterisine direkt seyahatin maliyeti aşağıdaki formül ile ifade edilmektedir.

$$c_{ij} = p_1 d_{ij} + p_2 (T_j - T_i) \quad (3.136)$$

(3.136) daki  $d_{ij}$ ,  $i$  ve  $j$  müşterileri arasındaki direkt mesafedir ve  $p_1 \geq 0$ ,  $p_2 \geq 0$  dir.<sup>75</sup>Buna göre  $p_1=0$  ve  $p_2=1$  olması durumunda amaç, toplam program süresinin minimize edilmesi olmaktadır.  $p_1=1$  ve  $p_2=0$  olması ise, zaman penceresiz araç rotalama problemini ortaya koymaktadır.

Sezgisel algoritmaların sunulmalarında kabul edilen bir diğer varsayım, araç sayısının serbest olması yani filo hacminin de en iyi rotalar ve programlar seti ile birlikte belirlenmesidir. Araçlar depodan en erken  $e_0$  zamanında ayrılmakta ve en geç  $l_0$  zamanında depoya geri dönmektedirler. Araçların depodan ayrılma zamanları da karar değişkenleridir.

Bu bölümde VRSPWTW ler için rota-inşa sezgiselleri üzerinde durulmaktadır. Bu nedenle öncelikle, kısmi olarak inşa edilmiş olan ve hizmete başlama zamanları  $(T_{ir}, 0 \leq r \leq m)$  bilinen bir uygun rota  $(i_0, i_1, i_2, \dots, i_m)$ , ( $i_0=i_m=0$ ) üzerindeki  $i_{p-1}$  ve  $i_p$  ( $i \leq p \leq m$ ) müşterileri arasına bir  $u$  müşterisinin sokulması durumunda zaman açısından uygunluğun korunması için gerek ve yeter şartların açıklanması gerekmektedir. Bu konuyu açıklamak için bazı varsayımlar yapılmakta ve ilave notasyon

---

<sup>75</sup> M.M.SOLOMON, "Algorithms for the vehicle routing and scheduling problems with time window constraints", Operations Research 35(2), 1987, S. 255.

kullanılmaktadır. İlk olarak her bir aracın depodan en erken olası zamanda ( $e_0$ ) ayrıldığı varsayılmaktadır. Bütün araç programları oluşturulduktan sonra herhangi bir gereksiz bekleme süresini elimine etmek için, her bir aracın depodan ayrılma zamanı ayrı ayrı ayarlanabilmektedir.

$T_{ip}^{Yeni}$ , u müşterisinin  $i_{p-1}$  ile  $i_p$  arasına sokulması durumunda  $i_p$  müşterisine hizmetin yeni başlama zamanı göstermektedir. Ayrıca  $w_{ir}$  ( $p \leq r \leq m$ ),  $i_r$  deki bekleme süresidir. Eğer hem seyahat mesafeleri hem de seyahat süreleri için üçgen eşitsizliğinin geçerli olduğu kabul edilirse, u müşterisinin  $i_{p-1}$  ile  $i_p$  arasına sokulması, programda u müşterisinin ileriye doğru itilmesine neden olmaktadır. bu itilme miktarı, aşağıdaki bağıntı ile hesaplanmaktadır.

$$PF_{ip} = T_{ip}^{Yeni} - T_{ip} \geq 0 \quad (3.137)$$

Ayrıca rotada  $i_p$  müşterisinden sonraki her bir müşteri için itilme miktarı,

$$PF_{ir+1} = \max\{0, PF_{ir} - w_{ir+1}\}, \quad p \leq r \leq m-1 \quad (3.138)$$

ile verilmektedir. Benzer şekilde programda  $i_r$  müşterisinin ( $r=0,1,\dots,p-1$ ) geriye doğru itilme miktarı da belirlenebilir. Fakat sadece programda  $i_{p-1}$  e kadar bekleme süresi bulunmaması ve aracın depodan  $e_0$  dan daha sonra ayrılması durumunda program geriye doğru itilebileceği için, bu konu pek önemli değildir. Yukarıda, her bir aracın depodan  $e_0$  zamanında ayrıldığı varsayımı yapılmıştı. Bunun nedeni şimdi açıktır. Bu varsayım bize programı mümkün olduğunca maximum ölçüde ileriye doğru itme imkanı vermektedir.

Eğer  $PF_{ip} > 0$  ise, bazı müşteriler ( $i_r$ ,  $p \leq r \leq m$ ) uygunsuz olabilmektedirler.  $PF_{ir}=0$  olan veya zaman açısından uygun olmayan bir  $i_r$  ( $r < m$ ) müşterisi ile karşılaşılıncaya kadar, bu müşterilerin zaman açısından uygunluklarının ard arda incelenmesi gerekmektedir.

Kısmi olarak oluşturulmuş uygun bir  $(i_0, i_1, i_2, \dots, i_m)$ ,  $(i_0=i_m=0)$  rotası üzerindeki  $i_{p-1}$  ve  $i_p$  ( $i \leq p \leq m$ ) müşterileri arasında bir  $u$  müşterisinin sokulması durumunda zaman penceresi kısıtlarına uygunluk için gerek ve yeter şartlar aşağıdaki gibidir.

$$T_u \leq l_u \text{ ve } T_{ir} + PF_{ir} \leq l_{ir} \quad (p \leq r \leq m) \quad (3.139)$$

$i_m=0$  olduğu için (3.139) daki koşullar, aracın programlanan süre içinde depoya geri dönmesini engelleyen müşterilerin kısmi rotaya sokulmamalarını sağlamaktadır.

### 3.6.2.1. VRSPTW için Tasarruf Sezgiselleri

VRSPTW leri çözmek için bu bölümde sunulan tasarruf sezgiseli, VRP için orijinal Clarke-Wright (1964) tasarruf sezgiselinin bir uzanımıdır. Burada zaman pencerelerinin varlığı, tasarrufların hesaplanmasında rota yönlendirmesini de dikkate almayı gerektirmektedir. Eğer  $i$  müşterisi bir kısmi rota üzerindeki ilk (son) müşteri ve  $j$  müşterisi bir başka kısmi rota üzerindeki son (ilk) müşteri ise, "bu iki kısmi rota uyumlu yönlere sahiptirler" denilmektedir. Yani iki kısmi rotanın birleştirilmesinde kabul edilebilir kanallar, bir rotadaki son müşteriden  $(l)$  diğer rotadaki ilk müşteriye  $(f)$  doğrudur.

Ayrıca sezgisel algoritmanın her adımında araç kapasitesi kısıtlarına ilaveten zaman penceresi kısıtlarına uygunluk da kontrol edilmelidir. (3.139) daki gerek ve yeter şartlar kolayca  $(l, f)$  formundaki kanallar durumuna adapte edilebilirler. (3.139) da basit olarak, zaman uygunluğunu etkin şekilde test etmek için  $f$  de oluşan ileri itme kullanılmaktadır.

Tasarruf sezgiseli orijinal hali ile, birbirlerine mesafe olarak çok yakın fakat zaman açısından oldukça uzak olan iki müşteriyi bir rotada birleştirmeyi kazançlı bulabilmektedir. Böyle kanallar, bekleme zamanı periyotlarını genişletmekte ve araç bir müşteriye hizmet için beklemek yerine bu süre içinde başka müşterilere hizmet verebileceği için bu,

yüksek bir fırsat maliyeti oluşturabilmektedir. Solomon (1987), müşterilerin hem alansal hem de zamansal yakınlıklarını dikkate almak için, iki müşterinin birleştirilmesi durumunda ortaya çıkan bekleme süresini sınırlamayı önermektedir.<sup>76</sup> Bu amaçla  $w_j^{yeni}$ , l ve f birleştirildiği zaman f de ortaya çıkan bekleme süresi ve w bir parametre olmak üzere,  $w_j^{yeni} > w$  olduğu zaman (l,f) kanalı kullanılmamaktadır.

### 3.6.2.2. Zaman-Esaslı En Yakın Komşu Sezgiseli

Ardışık tur-inşa algoritmaları sınıfına ait olan zaman-esaslı en yakın komşu sezgiseli aşağıda açıklanmakta olan bir ölçüye göre, her rotayı depoya en yakın olan rotalanmamış müşteri ile başlatmaktadır. Sezgisel, her ardışık iterasyonda son müşteriye en yakın olan müşterinin rotaya dahil edilmesini incelemektedir. Bu inceleme, zaman pencereleri, aracın depoya varış zamanı ve kapasite kısıtlarına uygun olarak söz konusu rotanın sonuna ilave edilebilecek olan bütün müşteriler arasında yapılmaktadır. Halen programlanacak müşteri mevcut olduğu halde bu inceleme ile mevcut rotaya artık hiçbir müşterinin ilave edilememesi durumunda, yeni bir rotaya başlanmaktadır.

Zaman-esaslı en yakın komşu sezgiselinde kullanılan yakınlık ölçüsü, müşterilerin hem coğrafik hem de zamansal yakınlıklarını dikkate almaktadır. i müşterisi mevcut kısmi rotadaki son müşteriye ve j müşterisi bir sonra ziyaret edilebilecek herhangi bir rotalanmamış müşteriye göstermek üzere, kullanılan ölçü ( $c_{ij}$ ), iki müşteri arasındaki direkt mesafeyi ( $d_{ij}$ ), i müşterisinde hizmetin tamamlanması ile j müşterisinde hizmetin başlaması arasındaki zaman farkını ( $T_{ij}$ ) ve j müşterisine hizmetin acilliğini ( $V_{ij}$ ) ölçmektedir. Formel olarak;

$$T_{ij} = T_j - (T_i + S_i) \quad (3.140)$$

$$V_{ij} = I_j - (T_i + S_i + t_{ij}) \quad (3.141)$$

---

<sup>76</sup> Solomon, Agm., S. 256.

$$C_{ij} = \delta_1 d_{ij} + \delta_2 T_{ij} + \delta_3 V_{ij} \quad (3.142)$$

$$\delta_1 \geq 0, \delta_2 \geq 0, \delta_3 \geq 0 \quad \delta_1 + \delta_2 + \delta_3 = 1$$

### 3.6.2.3. Ardışık Giriş Sezgiseli

Ardışık tur-inşa algoritmaları sınıfına giren ardışık giriş sezgiseli, her rotayı bir çok kriterden birisini kullanarak başlatmaktadır.

Her bir rotayı başlatmak için gerekli olan çekirdek müşterilerin seçiminde kullanılabilecek bazı kriterler, depodan en uzak rotdanmamış müşterinin seçilmesi, hizmet verilmesi en acil olan rotdanmamış müşterinin seçilmesi veya direkt rota süresinin ve mesafesinin eşit ağırlıklı bir kombinasyonunun minimum değerine sahip olan müşterinin seçilmesi olabilmektedir.<sup>77</sup>

Sezgisel, bir rotayı başlattıktan sonra bu kısmi rota üzerindeki iki komşu  $i$  ve  $j$  müşterileri arasına yeni bir  $u$  müşterisini sokmak için, her iterasyonda  $c_1(i, u, j)$  ve  $c_2(i, u, j)$  olmak üzere iki kriter kullanmaktadır.  $i_0, i_1, i_2, \dots, i_m$ , ( $i_0 = i_m = \text{depo}$ ), şeklindeki bir kısmi rota ele alındığında, her bir rotdanmamış müşteri için önce bu müşterinin rotadaki en iyi uygun giriş yeri,

$$c_1(i(u), u, j(u)) = \min_{p=1, \dots, m} [c_1(i_{p-1}, u, i_p)] \quad (3.143)$$

olarak hesaplanmaktadır. Sonra rotaya sokulacak en iyi müşteri,

$$c_2(i(u^*), u^*, j(u^*)) = \text{optimum} [c_2(i(u), u, j(u))] \quad (3.144)$$

olarak seçilmektedir. Burada seçilen  $u^*$  müşterisi, rotdanmamış ve uygun bir müşteri olmalıdır. Seçim aşamasının ardından  $u^*$  müşterisi rotadaki  $i(u^*)$  ile  $j(u^*)$  arasına sokulmaktadır. Eğer bütün müşteriler rotdanmamış değil

---

<sup>77</sup> Solomon, Agm., S. 259.

iseler ve uygun olarak rotaya sokulabilecek başka müşteri kalmamış ise sezgisel yeni bir rotaya başlamaktadır.

Yukarıda sunulan genel giriş kriterinin yanısıra daha spesifik yaklaşımlar da kullanılabilir. Aşağıda bu yaklaşımlardan en önemli üç tanesi ele alınmaktadır.<sup>78</sup>

$$(i) \ c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \ (\alpha_1, \alpha_2 \geq 0, \alpha_1 + \alpha_2 = 1) \quad (3.145)$$

$$c_2(i, u, j) = \lambda d_{ou} - c_1(i, u, j), \ \lambda \geq 0 \quad (3.146)$$

Burada görülen  $c_{11}(i, u, j)$  ve  $c_{12}(i, u, j)$ , aşağıdaki gibi ifade edilmektedir.

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij}, \ \mu \geq 0 \quad (3.147)$$

$$c_{12}(i, u, j) = T_{ju} - T_j \quad (3.148)$$

( $T_{ju}$ , u müşterisi rotaya sokulduğu zaman j müşterisine hizmetin yeni başlama zamanıdır).

(i) tipi giriş sezgiseli, bir müşteriye direkt bir rota yerine kısmi bir rota üzerinden hizmet vermenin sağlayacağı faydayı maximize etmeye çalışmaktadır. Örneğin  $\mu = \alpha_1 = \lambda = 1$  ve  $\alpha_2 = 0$  olduğu zaman  $c_2(i, u, j)$ , u müşterisine bireysel direkt hizmetin aksine i ve j müşterileri ile aynı rota üzerinden hizmet vermekten kaynaklanan uzaklık tasarrufudur. Rotalanmamış bir müşteri için en iyi uygun giriş yeri, bu müşteriye ziyaret etmek için gerekli olan extra mesafe ve extra sürenin ağırlıklı bir kombinasyonunu minimize eden yerdir.

(ii)  $c_1(i, u, j)$ , (3.145) dekinin aynısıdır.

$$c_2(i, u, j) = \beta_1 R_d(u) + \beta_2 R_t(u) \quad (\beta_1, \beta_2 \geq 0, \beta_1 + \beta_2 = 1) \quad (3.149)$$

---

<sup>78</sup> Solomon, Agm., S. 257.

$R_d(u)$  ve  $R_t(u)$ ,  $u$  nun sokulması halinde kısmi rotanın toplam mesafe ve toplam süresidir. Bu tip giriş sezgiseli, toplam rota mesafesi ve toplam rota süresinin ağırlıklı bir kombinasyonunu minimize eden müşterileri seçmektedir.

$$(iii) \ c_1(i,u,j) = \alpha_1 c_{11}(i,u,j) + \alpha_2 c_{12}(i,u,j) + \alpha_3 c_{13}(i,u,j)$$

$$(\alpha_1, \alpha_2, \alpha_3 \geq 0, \alpha_1 + \alpha_2 + \alpha_3 = 1) \quad (3.150)$$

Burada  $c_{11}(i,u,j)$  ve  $c_{12}(i,u,j)$  (3.147) dekinin aynısıdır.

$$c_{13}(i,u,j) = l_u - T_u \quad (3.151)$$

$c_2(i,u,j)$ ,  $c_1(i,u,j)$  ye eşittir.

(iii) tipi yaklaşımda giriş için kullanılan kriterin zamansal yönü, bir müşteriye hizmetin acilliğini de hesaba katmaktadır.

Yukarıda sunulan bütün yaklaşımlarda, rotdalanmamış müşterilerin rotalara sokulmaları hem coğrafik hem de zamansal kriterler ile yönlendirilmekte ve bunun sonucu olarak bu sezgisel ile üretilen programlardaki bekleme süresi, sadece mesafesel kriterler ile üretilenden önemli ölçüde daha az olabilmektedir.

#### 3.6.2.4. Paralel Giriş Sezgiseli

Paralel giriş sezgiseli adından da kolayca anlaşılabileceği gibi, aynı anda birden fazla rotanın oluşturulmasını amaçlamaktadır. Bu bölümde Potvin ve Rousseau (1993) tarafından oluşturulan ve  $n_r$  adet rotayı aynı anda başlattıktan sonra kalan müşterileri bu rotalardan herhangi birine sokmak için bütün rotdalanmamış müşteriler üzerinden bir genel pişmanlık ölçütü kullanan paralel giriş sezgiseli ele alınmaktadır.

Paralel yaklaşımdaki bir zorluk, başlangıç rota sayısının ( $n_r$ ) ne olması gerektiğinin belirlenmesidir. Bunun için elverişli bir yol, rota başlatma için depodan en uzak müşteriye kullanarak ve rotalara girecek

müşterilerin seçimi için  $(\beta, \alpha_1, \alpha_2) = (1, 1, 0)$  ile Solomon (1987) deki ardışık algoritmayı sadece bir kere uygulayarak  $n_r$  yi tahmin etmektir.<sup>79</sup>

Aşağıda da açıklanacağı gibi Solomon (1987) algoritması ile oluşturulan rotalar seti, uygun çekirdek müşterileri seçmek için de kullanılmaktadır. Şimdi paralel algoritma aşağıdaki gibi açıklanabilir.

1. Solomon (1987) ardışık algoritması (bir tek başlatma kriteri ve bir tek parametre seti) ile oluşturulan rotaların her birindeki depodan en uzak müşteri seçilerek başlanmaktadır. Bu rotalar kullanılarak coğrafik alan üzerinde  $n_r$  adet çekirdek müşterinin iyi bir başlangıç dağılım elde edilmekte ve kalan müşterilerin rotalara sokulmaları sırasında son rotalanmamış müşterilerin birbirlerinden çok fazla ayırık kalmaları olasılığı daha az olmaktadır.

2. Her bir rotalanmamış  $u$  müşterisi için önce  $n_r$  adet başlangıç rotasının herbirindeki en iyi uygun giriş yeri belirlenmektedir.

$$c_{1r}(i_r(u), u, j_r(u)) = \min_{p=1, \dots, m} [c_{1r}(i_{rp-1}, u, i_{rp})], \quad (r=1, \dots, n_r)$$

$$c_{1r}(i_r, u, j_r) = \alpha_1 c_{11r}(i_r, u, j_r) + \alpha_2 c_{12r}(i_r, u, j_r), \quad (\alpha_1, \alpha_2 \geq 0, \alpha_1 + \alpha_2 = 1) \quad (3.152)$$

Burada;

$$c_{11r}(i_r, u, j_r) = d_{i_r, u} + d_{u, j_r} - d_{i_r, j_r}$$

$$c_{12r}(i_r, u, j_r) = T_{u, j_r} - T_{j_r}$$

$T_k = k$  daki hizmete başlama zamanı

$T_{k,l} = k$  rotaya sokulduğu zaman  $l$  deki yeni hizmete başlama zamanıdır.

<sup>79</sup> C. FOISY, J.Y.POTVIN, "Implementing an insertion heuristic for vehicle routing on parallel hardware", Computers and Operations Research 20, 1993, S. 738.

Sonra  $u^*$  aşağıdaki gibi seçilmektedir.

$$c_2(u^*) = \max_u [c_2(u)]$$

$$c_2(u) = \sum_{r=r'} [c_{1r}^*(i_r(u), u, j_r(u)) - c_{1r}^*(i_r(u), u, j_r(u))] \quad (3.153)$$

Burada;

$$c_{1r}^*(i_r(u), u, j_r(u)) = \min_{r=1, \dots, n_r} [c_{1r}^*(i_r(u), u, j_r(u))] \quad (3.154)$$

$U^*$  seçildikten sonra  $i_r(u^*)$  ile  $j_r(u^*)$  arasına sokulmaktadır. Bu işlem bütün müşteriler rotalanmış oluncaya veya kalan rotalanmamış müşteriler için uygun giriş yerleri bulunamayınca kadar tekrarlanmaktadır.

Yukarıda görüldüğü gibi, giriş maliyeti formülü ifadesi Solomon (1987) dekinin aynısıdır. Fakat rotaya sokulacak bir sonraki müşterinin seçimi, bütün rotalar üzerinden genel bir pişmanlık ölçütüne dayandığı için,  $c_2(u)$  oldukça farklıdır.

Temel olarak pişmanlık, belirli bir müşterinin derhal en iyi rota içine sokulmaması durumunda ileride karşılaşılabilecek zararı gösteren bir tür ileri görüşlülük olmaktadır. Burada büyük bir pişmanlık ölçütü, bir müşteri için en iyi giriş yeri ile bu müşterinin diğer rotalardaki en iyi giriş yerleri arasında büyük bir farklılık bulunduğunu ifade etmektedir. Bundan dolayı, büyük pişmanlık ölçütlerine sahip olan müşterilerin sokulabilecekleri alternatif rotaların sayısı az olduğu için, ilk olarak bu müşterilerin rotalara sokulmaları ele alınmalıdır.

$U$  müşterisi için  $r$  rotasında uygun giriş noktası mevcut olmadığı zaman  $c_{1r}^*(i_r(u), u, j_r(u))$  keyfi ölçüde büyük bir değere eşitlendiği için genel pişmanlık ölçütü, mevcut çözümün uygunluk meselesinden kaynaklanabilecek güçlükleri de öngörebilmektedir. Bir müşteri için bir rota uygun olmadığı zaman bu müşterinin pişmanlık ölçütü çok büyük bir değer almaktadır (bu müşteri için hiçbir rota uygun olmadığı zaman pişmanlık

ölçütü sıfıra eşittir). Bu nedenle genel pişmanlık ölçütü, rotalanmamış müşteriler setini bu müşterilerin her biri için uygun kalan rota sayısına göre parçalamaktadır. Örneğin eğer keyfi büyük değerin  $L$  olduğu varsayılırsa  $n$  adet alternatif uygun rotaya ( $i \leq n \leq n_r$ ) sahip olan bütün rotalanmamış müşterilerin pişmanlık ölçütü  $L(n_r - n)$  komşuluğu içerisinde. Algoritma en büyük pişmanlık ölçütüne sahip olan müşteriyi aradığı için, otomatik olarak en az sayıda alternatif uygun rotaya sahip olan müşteriyi seçmektedir. Bu arzu edilen bir davranıştır. İki müşteri için uygun rotaların sayısı aynı olduğu zaman genel pişmanlık ölçütü, bu müşterilerin uygun rotaları içindeki en iyi giriş yerlerine bakarak seçim yapmaktadır. Şimdi paralel giriş algoritmasının uygulama adımları aşağıdaki gibi özetlenebilir.

Adım 0.  $(\beta, \alpha_1, \alpha_2) = (1, 1, 0)$  parametreleri ile Solomon (1987) ardışık algoritmasını uygulayarak başlangıç rota sayısını ( $n_r$ ) ve başlangıç çekirdek müşteriler setini ( $S_{\text{çek}, n_r}$ ) belirle. Çekirdek müşteriler her bir rotada depodan en uzak olan müşterilerdir.

Adım 1.  $S_{(\alpha_1, \alpha_2), n_r}$  parametreler setine  $\{(0.5, 0.5), (0.75, 0.25), (1.0, 0.0)\}$  başlangıç değerlerini ver.

Adım 2.  $S_{(\alpha_1, \alpha_2), n_r}$  den bir parametre grubu seç ve onu bu setten çıkart.

Adım 3. Rotalar-kur ( $n_r, \alpha_1, \alpha_2, S_{\text{çek}, n_r}$ ) yi çağır.

Adım 4. Eğer Rotalar-kur bir çözüm verirse, bunu bilinen en iyi çözüm olarak kaydet.  $n_r = n_r - 1$  ve  $S_{\text{çek}, n_r} = S_{\text{çek}, n_r - 1} - S^*$  yap.

Burada

$$S^* = \arg \left[ \min_{S \in S_{\text{çek}, n_r + 1}} \left\{ \min_{S' \in S_{\text{çek}, n_r + 1}, S = S'} d_{S, S'} \right\} \right] \text{ dir.}$$

Adım 1 e git.

Aksi halde,

eğer  $S_{(\alpha_1, \alpha_2), n_r}$  de parametre grubu yoksa,

$n_{r_{\min}} = n_r + 1$  yap ve Adım 5 e git.

Aksi halde, Adım 2'ye git.

Adım 5. Eğer  $S_{(\alpha_1, \alpha_2), n_{r_{\min}}}$  de parametre grubu yoksa, bilinen en iyi çözüm ile süreci sona erdir.

Adım 6.  $S_{(\alpha_1, \alpha_2), n_{r_{\min}}}$  den bir parametre grubu  $(\alpha_1, \alpha_2)$  seç ve onu bu setten çıkart.

Adım 7. Rotalar-kur  $(n_{r_{\min}}, \alpha_1, \alpha_2, S_{\text{çek}}, n_{r_{\min}})$  i çağır. Eğer Rotalar-kur bir çözüm verirse, bu çözümü bilinen en iyi çözüm ile karşılaştır. Eğer şu andaki çözüm daha iyi ise onu bilinen en iyi çözüm olarak kaydet.

Adım 8. Adım 5 e git.

#### - Rotalar-kur $(n_r, \alpha_1, \alpha_2, S_{\text{çek}}, n_r)$ Algoritması

Adım 0.  $S_{\text{çek}}, n_r$  deki  $n_r$  adet müşteri ile  $n_r$  adet rotayı başlat.

Adım 1. Her bir rotalanmamış  $u$  müşterisi için,  $r$  rotasındaki bütün komşu  $i_r$  ve  $j_r$  müşterileri için uygun giriş maliyetlerini  $(c_{1r}, (i_r, u, j_r))$  belirle ve  $c_{1r}^*(i_r(u), u, j_r(u))$  yu bul ( $r=1, \dots, n_r$ ). Eğer uygun giriş yeri yoksa  $c_{1r}(i_r(u), u, j_r(u))$  çok büyük bir değere eşit olur.

Adım 2. Bir sonra rotaya sokmak üzere  $c_2(u)$  nun maximum olduğu müşteriyi ( $u^*$ ) seç.

Adım 3.  $u^*$  müşterisinin  $r$  rotasındaki  $i_r(u^*)$  ile  $j_r(u^*)$  arasına girişini ele al. Burada,

$$c_{1r}^*(i_r(u^*), u^*, j_r(u^*)) = \min_{r=1, \dots, n_r} c_{1r}^*(i_r(u^*), u^*, j_r(u^*)) \quad \text{dur.}$$

Eğer bu giriş uygun ise,  $u^*$  yu rotaya sok.

Eğer rotalanmamış müşteri varsa, Adım 1'e git.

Aksi halde, Ana programa dön (çözüm ile).

Aksi halde, Ana programa dön (çözumsuzlük ile).

### 3.6.2.5. Zaman Esaslı Süpürme Sezgiseli

Zaman esaslı süpürme sezgiseli, VRSPTW yi bir kümeleme aşaması ve bir programlama aşamasına bölen sezgisel metodlar sınıfının bir üyesidir. Birinci aşamada orijinal süpürme sezgiselindeki gibi, müşteriler araçlara tahsis edilmektedirler. İkinci aşamada ise bir tur-inşa sezgiseli kullanılarak, o ana kadar tahsis edilmiş olan bütün müşteriler için bir tek-aracı programı oluşturulmaktadır. Zaman penceresi kısıtlarından dolayı bu kümedeki bazı müşteriler programlanmamış halde kalabilmektedirler. Söz konusu programlanmamış müşteriler elimine edildikten sonra kümeleme-programlama süreci tekrarlanmaktadır.

Bu algoritmada coğrafik yakınlığı gözönünde bulundurmak amacı ile bir sonraki küme için farklı çekirdek seçim kriteri dikkate alınabilir. Bunun için elverişli bir yol, en son oluşturulmuş olan bölgeyi bir çizgi ile iki eşit parçaya ayırmak ve saat yönünün aksi yönde kalan bölgede saat yönüne ters bir süpürme uygulayarak, programlanmamış müşterileri depoya birleştiren çizgiler ve ayırma çizgisi arasında kalan en küçük açığa sahip olan müşteriyi yeni bir kümenin çekirdeği olarak seçmektir.

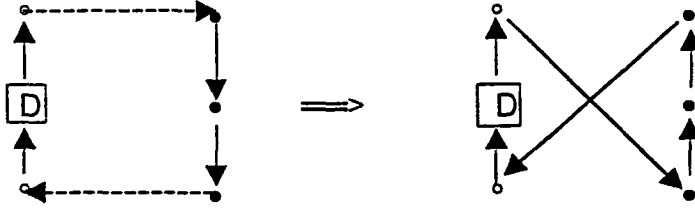
Programlanmamış müşterileri iki alt bölgeye bölmedeki mantık, saat yönüne daha yakın yarı bölgedeki müşterilerin yeni kümeden nispeten daha uzak olmalarıdır. Bu müşterilerin rotalara daha ileri aşamalarda sokulmaları daha iyi bir program oluşturabilmektedir. Zaman esaslı

süpürme sezgiseli bütün müşteriler programlanmış oluncaya kadar tekrarlanmaktadır.

### 3.6.2.6. Yasak Araştırması Sezgiseli

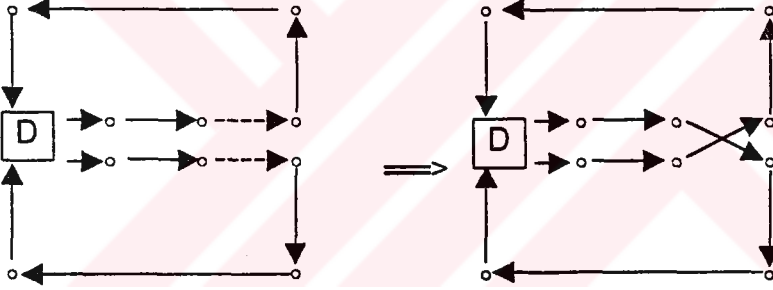
Daha önce Bölüm (3.2.3.7) de açıklanmış olduğu gibi, Yasak araştırması sezgiseli, bir iyileştirme sezgiselidir ve her bir adımda mevcut çözümün komşuluğunu araştırarak bu komşuluk içindeki en iyi çözümü yeni çözüm olarak kabul etmeye dayanmaktadır. Bu sezgisel saf yerel araştırma tekniklerinin aksine, iyileşme gerçekleşmediği zaman durmamakta ve komşuluk içindeki en iyi çözüm mevcut çözümden daha kötü olsa bile yeni çözüm olarak seçilmektedir. Bu strateji, sezgisel metodun yerel optimadan kaçınılabilmesini ve bunun sonucu olarak daha büyük bir araştırma uzayının incelenebilmesini sağlamaktadır.

Yerel araştırma sezgiselleri, uygun bir başlangıç çözümünden başlayarak yeni bir iyileşmiş çözüm bulmak için mevcut çözüm üzerindeki çeşitli değişiklikleri ele almaktadırlar. Daha önce de söz edilmiş olduğu gibi, bu sınıftaki bilinen en iyi sezgiseller k-opt değişimleridir. k-opt değişimlerinde yeni bir tur elde etmek için mevcut turdan k adet kanal çıkartılmakta ve bunların yerine k adet yeni kanal sokulmaktadır. Özellikle 2-opt değişimleri, rotaları iyileştirmek için sık sık kullanılmakla birlikte rotaların yönlerini dikkate almadıkları için zaman pencereli problemleri pek uygun değildirler. Örneğin, şekil 3.17 depodan (D) başlayan ve aynı yerde sona eren bir tek rota üzerindeki bir 2-opt u göstermektedir. Burada yeni bir tur oluşturmak için kesikli çizgi ile gösterilen iki kanal yerine iki yeni kanal yerleştirilmektedir. Bununla birlikte yeni çözümde koyu renkli düğümler, ilk çözümdekine ters bir sıra içinde bulunmaktadırlar. Müşteriler tipik olarak zaman pencerelerinin üst sınırlarına göre sıralandıkları için (en düşük üst sınırlı müşteriye ilk önce hizmet verilmelidir), başlangıç rotasının bir kısmının ters çevrilmesi muhtemelen uygun olmayan bir çözüm oluşturmaktadır.



Şekil 3.17. Bir 2-opt değişimi

Çok rotalı problemler için özel bir 2-opt\* değişimi ele alınabilir.<sup>80</sup> Burada değişim, iki farklı rotadaki iki kanalı içermekte ve birinci rotadaki ilk(son) müşteriyi ikinci rotadaki son (ilk) müşteri ile birleştirmektedir (Şekil 3.18). Bir 2-opt\*, rotaların yönlerini dikkate almakta ve geç zaman pencereli müşterileri erken zaman pencereli müşterilerin arkasına ilave etmektedir. Bundan dolayı yeni çözüm muhtemelen uygun olmaktadır.

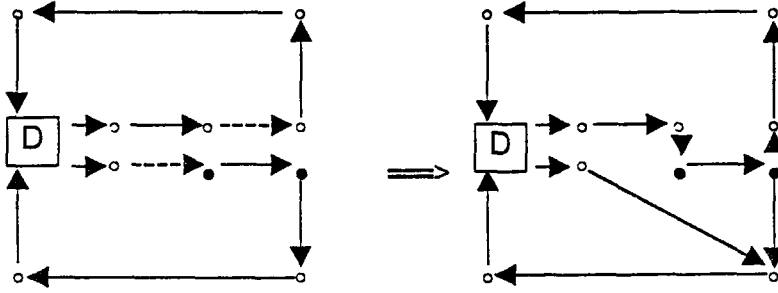


Şekil 3.18. Bir 2-opt\* değişimi

Bu bölümde sunulmakta olan ve Garcia, Potvin, Rousseau (1994) tarafından geliştirilen yasak araştırması sezgiselinde 2-opt\* değişimleri Or-opt değişimleri ile birleştirilmektedir. Or-opt değişimleri, mevcut çözümde bir, iki ve üç komşu müşteriden oluşan serileri ele alarak bu serileri aynı rotadaki veya diğer rotalardaki bütün diğer olası yerlere sokmayı denemektedir (Or-opt konusunda daha ayrıntılı bilgi, I. Or (1976)

<sup>80</sup> B.L.GARCIA, J.Y. POTVIN, J.M.ROUSSEAU", A parallel implementation of the tabu search heuristic for vehicle routing problems with time windows constraints", Computers and Operations Research 21(9), 1994, S. 1026.

da bulunabilir). Şekil 3.19 da bir Or-opt değişimi kullanılarak iki komşu müşteriden (siyah düğümler) oluşan bir seri bir rotadan bir diğerine taşınmaktadır. Bu değişimler, 3-opt ile sınırlıdır ve zaman pencereli problemler için çok iyi işlemektedirler.



Şekil 3.19. İki müşterili bir seri üzerinde Or-opt

Yasak araştırması sezgiselinin iki temel bileşeni (2-opt\* ve Or-opt) açıklandıktan sonra şimdi yasak araştırması algoritmasının açıklanmasına devam edilmektedir.

Algoritmanın işleyişi sırasında tekrarlamaları önlemek için, en son seçilen çözümler bir "Yasak" listesine yerleştirilerek saklanmaktadır. Genel olarak çözümler Yasak listesinde doğrudan saklanmayıp bunun yerine yeni bir çözüme yol açan her bir değişikliğin aksi yönündeki değişiklik "Yasak" olarak etiketlenmekte ve Yasak listesine yerleştirilmektedir.

Burada ele alınan yasak araştırması sezgiseli, bölüm (3.2.3.7) de VRP için açıklanmış olan sezgisele çok benzemektedir. Ancak oradakini aksine bu bölümdeki yasak araştırması sezgiselinde çözümün uygunluğunun daima korunmasına dikkat edilmektedir. Çünkü uygun olmayan bir hareketten sonra uygun bir çözüme geri dönmek, zaman penceresi kısıtlarının varlığından dolayı çok zor olmaktadır.

Yasak araştırması, hesaplama süresini kısaltmak ve en ümit verici değişimler üzerinde odaklanmak amacı ile bütün 2-opt\* ve Or-opt değişimlerini araştırmamakta ve sadece mesafe açısından birbirlerine

yakın olan müşteriler arasındaki değişimleri ele almaktadır. Söz konusu mesafe kısıtlamaları Or-opt ve 2-opt\* değişimleri için şöyledir.

- (a) Or-opt: eğer  $i$  müşterisi mesafe olarak  $j$  müşterisinin en yakın  $h$  adet komşusundan birisi ise,  $i$  müşterisi (muhtemelen  $i+1$  ve  $i+2$  müşterileri ile birlikte)  $j$  ve  $j+1$  müşterileri arasına yerleştirilir.
- (b) 2-opt\*: eğer  $j+1$  mesafe olarak  $i$  müşterisinin en yakın  $h$  adet komşusundan birisi ise,  $(i, i+1)$  ve  $(j, j+1)$  kanalları yerine  $(i, j+1)$  ve  $(j, i+1)$  kanalları yerleştirilir.

Mesafe açısından en yakın komşuların sayısı ( $h$ ), algoritmanın bir parametresidir. Ayrıca  $j$  müşterisini  $i$  müşterisinin bir komşusu olarak niteleyebilmek için,  $i$  den  $j$  ye direkt olarak gitmek ve her iki müşterinin de zaman penceresi kısıtlarını karşılamak mümkün olmalıdır. Şimdi, yasak araştırması sezgiselinin uygulama adımları aşağıdaki gibidir.<sup>81</sup>

#### Adım 1. Başlatma adımı.

Her bir müşterinin mesafe olarak en yakın komşularını belirle. Komşuların sayısı,  $h_A$  ve  $h_B$  nin maximumu ile verilmektedir. Sonra, mevcut ve bilinen en iyi çözümü, Solomon (1987) sezgiseli ile elde edilen çözüme eşitle. Bilinen en iyi çözümde herhangi bir iyileşme olmaksızın  $I_{tot}$  adet iterasyon icra edilmiş oluncaya kadar Adım 2'yi tekrarla.

#### Adım 2. Yasak araştırması adımı.

Bilinen en iyi çözümde herhangi bir iyileşme olmaksızın  $I_{2-opt^*}$  adet ardışık iterasyon icra edilmiş oluncaya kadar Adım 2.1'i tekrarla.

---

<sup>81</sup> Garcia ve diğerleri, Agm., S. 1028.

### Adım 2.1.

#### Aşama A. Rota tasarruf aşaması

Üç veya daha az müşteriye sahip olan rotalardaki bütün müşterileri seç. Sonra, seçilen her bir  $i$  müşterisi için, bu müşteriyi (muhtemelen  $i+1$  ve  $i+2$  müşterileri ile birlikte) mevcut rotasından çıkartan Or-opt değişimlerini ele al. Değişimler uygun olmalı ve mesafe kısıtlarını tatmin etmelidirler ( $h=h_A$  için). Eğer bu komşuluk içindeki en iyi çözüm mevcut çözümden daha iyi ise bu çözümü yeni çözüm olarak seç ve Aşama A yı tekrarla.

#### Aşama B. 2-opt\* ile yasak aşaması.

Her bir  $i$  müşterisi için,  $(i, i+1)$  kanalını içeren 2-opt\* değişimlerini ele al. Değişimler uygunluğu korumalı ve mesafe kısıtlarını tatmin etmelidir ( $h=h_B$  için). Bu komşuluk içindeki en iyi çözüm mevcut çözümden daha kötü olsa bile onu yeni çözüm olarak seç. Bunun tersi hareketi beş iterasyon için yasak olarak belirle.

Adım 2.1'in sonu.

Bilinen en iyi çözümde herhangi bir iyileşme olmaksızın  $I_{or}$  adet iterasyon icra edilmiş oluncaya kadar Adım 2.2. yi tekrarla.

### Adım 2.2.

#### Aşama A. Rota tasarruf aşaması.

Adım 2.1 deki Aşama A nın aynısıdır.

#### Aşama B. Or-opt ile yasak aşaması.

Her bir  $i$  müşterisine (iki veya üç uzunluklu seriler için  $i+1$  ve  $i+2$  müşterileri ile birlikte) Or-opt değişimlerin uygulanması hariç, Adım 2.1 deki Aşama B nin aynısıdır.

Adım 2.2 nin sonu.

Adım 2 nin sonu.

Adım 3. Bilinen en iyi çözüm, yasak araştırması ün final çözümüdür.



## DÖRDÜNCÜ BÖLÜM

### KANAL ROTALAMA PROBLEMİ (ARP)

Kanal rotalama problemi (arc routing problem, ARP), bir  $G=(N, A)$  şebekesindeki her bir kanaldan veya önceden belirlenmiş bazı kanallardan geçerek başlangıç noktasına geri dönerken toplam seyahat mesafesini (maliyetini) minimize etme problemidir. Bu problemde talepler düğümlerde değil kanallar üzerinde ortaya çıktıkları için problem, "kanal rotalama problemi (ARP)" olarak adlandırılmaktadır.

ARP uygulamaları, sokak süpürücülerinin, çöp toplama araçlarının, kara yollarında buzlanmayı önleyici tuz püskürtme araçlarının, elektrik ve su sayacı kontrolörlerinin, okul ve iş yeri servis araçlarının vb. rotalanmalarını içermektedir. Bu tür uygulamalarda kesin talep noktaları bilinmedikleri veya birbirlerine çok yakın oldukları için, üzerlerinde taleplerin yer aldığı kanalların bir uç düğümden diğerine boydan boya katedilmeleri zorunluluğu ortaya çıkmaktadır.

Bu bölümde kanal rotalama problemi, şebekedeki her bir kanaldan geçilmesi gereken (CPP) ve sadece bazı kanallardan geçilmesi gereken (RPP) olmak üzere iki ve kapasite kısıtının söz konusu olduğu (CCPP) üçüncü bir alt bölüme ayrılarak incelenmektedir. Bu bölümdeki problemlerin bazıları NP-hard iken diğerleri için optimal çözüm teknikleri mevcuttur. Ancak çalışmamızın ana amacı sezgisel çözüm tekniklerinin kullanılmaları olduğu için NP-hard olmayan problem türleri sadece kısaca tanımlanmakta, NP-hard olanlar için ise sezgisel çözüm teknikleri sunulmaktadır.

#### 4.1. Çinli Postacı Problemi (CPP)

Kanal rotalama problemlerinin en yoğun olarak incelenmiş versiyonu olan Çinli postacı problemi (Chinese Postman Problem, CPP) Çin kültür devrimi sırasında bir süre postacı olarak çalışmış olan Çinli matematikçi Kwar Mei-Ko (1962) tarafından sunulmuştur.<sup>82</sup> CPP bu nedenle "Çinli postacı problemi" olarak adlandırılmaktadır.

CPP, toplam seyahat mesafesi minimum olacak şekilde, şebekedeki bütün kanallardan kesinlikle bir kere geçen ve başlangıç noktasına geri dönen kapalı bir yol bulma problemi olarak tanımlanmaktadır. Bir  $G=(N, A)$  grafiği üzerinde yukarıda tanımlanan gibi bir kapalı yolun mevcut olması durumunda bu grafik, "Euler grafiği" olarak adlandırılmaktadır. Bu durumda CPP için çözüm algoritmalarının birinci aşaması,  $G=(N,A)$  grafiğinin minimum maliyetle genişletilmesini yani grafiği Euler yapacak olan minimum maliyetli kanallar setinin belirlenmesini içermektedir. İkinci aşamada ise bu genişletilmiş grafiğin bütün kanallarından geçen bir tur daima polinomik süre içinde belirlenebilmektedir.<sup>83</sup>

CPP, yönsüz (simetrik), yönlü (asimetrik) ve karışık olmak üzere üç ayrı türe ayrılmaktadır. Yönsüz CPP durumunda problemde mevcut olan bütün kanalların maliyetleri seyahat yönünden bağımsızdırlar. Yönlü CPP, bunun aksine hepsi seyahat yönüne bağlı maliyetlere sahip olan kanallardan oluşmaktadır. Karışık CPP ise hem yönlü hem de yönsüz kanallar içermektedir. Yukarıda söz edildiği gibi CPP için çözüm algoritmalarının ikinci aşaması, her üç versiyonda da polinomik olarak çözülebilmektedir. Buna ilaveten yönsüz ve yönlü CPP için birinci aşama da polinomik olarak çözülebilmektedir. Yönsüz CPP durumunda Euler grafiği, bir eşleştirme problemi çözülerek, yönlü CPP durumunda ise

<sup>82</sup> H.A.EISELT, M. GENDREAU, G. LAPORTE, "Arc routing problems, Part I: The Chinese postman problem", Operations Research 43(2), 1995, S. 231.

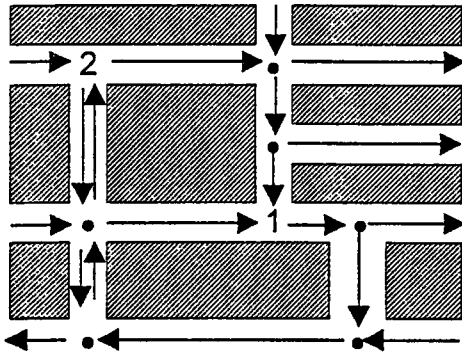
<sup>83</sup> Eiselt ve diğerleri, Part I, Agm., S. 232.

bir minimum maliyetli akış problemi çözülerek polynomik olarak elde edilebilmektedir. Yönlü ve yönsüz CPP için etkin optimal çözüm algoritmaları, Orloff (1974), Beltrami ve Bodin (1974), Kaufmann (1967) ve Fleischner (1991) de bulunabilir.

Karışık CPP lerin çözümleri yönlü ve yönsüz CPP ler kadar kolay değildir. G bir karışık grafik olduğunda, Euler grafiği belirleme problemi NP-harddır.<sup>84</sup> Bu nedenle karışık CPP'nin çözülmesi için sezgisel algoritmalara gerek duyulmaktadır.

#### 4.1.1. Karışık Çinli Postacı Problemi (MCP)

Karışık CPP de bazı kanalların maliyetleri seyahat yönünden bağımsız iken, diğerlerinin maliyetleri seyahatin yönüne göre değişmektedir. Bir başka deyişle bazı kanallar için  $c_{ij}=c_{ji}$  iken diğerleri için  $c_{ij} \neq c_{ji}$  dir. Karışık CPP(MCP), caddelerdeki tek yönlü gidiş, sağa veya sola dönüş kısıtlamaları veya trafiğin yoğun veya sakin olma durumlarında ortaya çıkmaktadır. Şekil (4.1), bir MCP ye ait sokak şebekesindeki tek veya çift yönlü gidiş ve sağa veya sola dönüş kısıtlamalarını göstermektedir.



Şekil 4.1. İzin verilen seyahat ve dönüş yönleri

<sup>84</sup> Eiselt ve diğerleri, Part I, Agm., S. 240.

Şekil 4.1 de görüldüğü gibi (1) noktasından (2) noktasına gidişte en kısa yol (1)-(2) arasındaki direkt hat olmayıp, trafik kısıtlamalarına göre (1) den (2) ye giderken katedilen mesafe (2) den (1) e giderken olandan farklıdır.

MCPP yi çözmek için mevcut bazı etkin sezgiselleri açıklamadan önce notasyon uygunluğu için MCPP nin bir  $G=(N,AUE)$  üzerinde ifade edildiği varsayılmaktadır. Burada N daha önceki gibi düğümler setini, A yönlü kanallar setini ve E de yönsüz kanallar setini göstermektedir. Ayrıca AUE ye ait bir maliyet matrisi  $C=(c_{ij})$  dir.

MCPP nin çözülmesi esas olarak denge özelliğinden yararlanmaktadır. Birbirlerine bağlı düğümlerden oluşan bir G grafiği üzerinde, her bir kanalı kesinlikle bir kere ve her bir düğümü en az bir kere içeren bir kapalı yolun mevcut olması halinde bu grafiğin "dengeli" olduğu söylenmektedir. Yönlü ve yönsüz kanallardan oluşan bir G grafiğinin "dengeli" olması için gerek ve yeter şartlar, her düğüme toplam çift sayıda yönlü ve yönsüz kanalın bağlı olması ve her  $S \subseteq N$  için S den  $N \setminus S$  ye yönlü kanalların sayısı ile  $N \setminus S$  den S ye yönlü kanalların sayısı arasındaki farkın S ile  $N \setminus S$  yi birbirlerine bağlayan yönsüz kanalların sayısına eşit veya daha küçük olmasıdır.<sup>85</sup> Şimdi MCPP nin çözülmesi, denge için gerek ve yeter şartlara uygun olarak bazı yönlü ve yönsüz kanalların kopyalarını ilave ederek G yi minimum maliyet ile genişletme ve sonra bu grafik üzerinde bir Euler turu belirlemeye denktir.

Eğer bir grafiğin her bir düğümüne bağlı yönlü ve yönsüz toplam kanal sayısı çift ise, bu grafik "çift" olarak adlandırılmaktadır. Eğer bir düğüm için giren kanalların sayısı çıkan kanalların sayısına eşit ise grafik simetriktir. Denge koşulunun sağlanması halinde grafik dengelidir ve bir karışık grafik yalnız ve yalnız çift ve dengeli olması durumunda Euler

---

<sup>85</sup> L.R. FORD, D.R. FULKERSON, "Flows in networks", Princeton University Press, Princeton, N.J., 1962, S. 70-71.

grafiğidir. Fakat denge için simetri bir gereklilik şartı değildir (bkz. Şekil 4.2).



Şekil 4.2. Simetrik olmayan fakat Euler olan bir grafik örneği

Bir  $G$  grafiğinin Euler olması durumunda problem,  $G$  üzerinde bir Euler dairesinin belirlenmesinden ibarettir. Bu, üç aşamada çözülebilen bir problemdir. 1)  $G$  yi simetrik hale dönüştürmek için bazı yönsüz kanallara yönler verme, 2) kalan yönsüz kanallara yönler verme ve 3)  $G$  üzerinde bir tur belirleme.

MCCP için Edmonds ve Johnson (1973) tarafından sezgiseller önerilmiş ve Frederickson (1979) ve Christofides, Benavent, Campos, Corberan ve Mota (1984) tarafından bu sezgiseller daha da iyileştirilmiştir. Söz konusu sezgiseller, denge için gerek ve yeter şartları sağlayan iyi çözümler aramaktadırlar. Frederickson (1979), herbiri  $O(\max |N|^3, |A|(\max\{|A|, |E|\})^2)$  süre içinde icra edilebilen, en kötü durum oranları 2 olan ve orijinal olarak MIXED1 ve MIXED2 olarak adlandırılan iki sezgisel önermektedir. Bu iki algoritmanın ardarda uygulanması halinde bu karışık sezgiselin en kötü durum performansı  $5/3$  e inmektedir. Aşağıda MIXED1 ve MIXED2 sezgisellerinin uygulama adımları verilmektedir.

-MIXED1 Sezgiseli:

Adım 1. Bir  $G=(N,AUE)$  grafiğindeki tek dereceli düğümleri belirle.

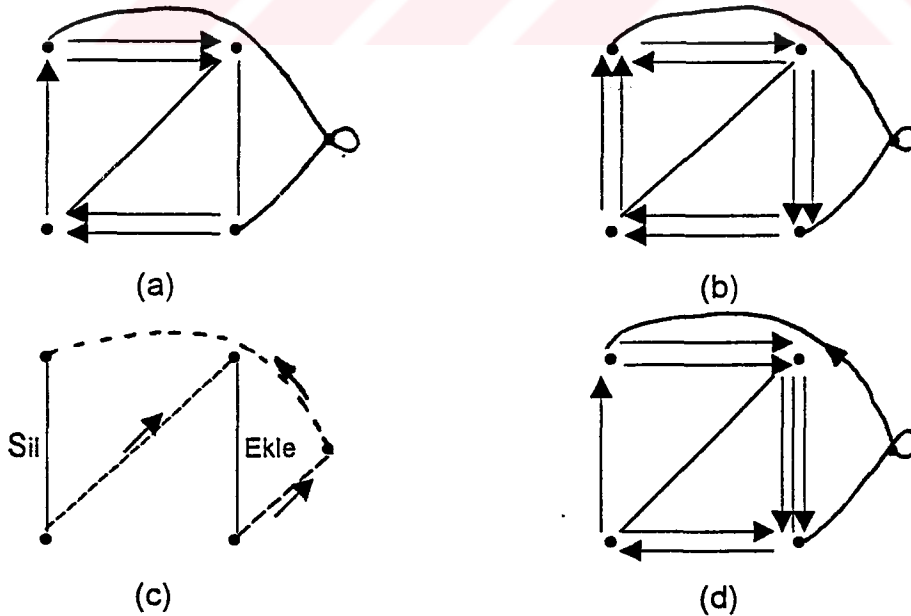
Kanal yönlerini ihmal ederek tek dereceli düğümler arasındaki bütün en kısa yolları bul ve tek dereceli düğümlerin minimum maliyetli eşleşmesini belirlemek için bu en kısa yolları kullan. Eşleştirme çözümü için kullanılan

bütün yönlü ve yönsüz kanalları dahil ederek orijinal grafiği genişlet.

Adım 2. Bir minimum maliyetli akış algoritması kullanarak grafiği simetrik yap. Ortaya çıkan grafik  $G'=(N,A'UE')$  olsun.

Adım 3.  $G'$  deki tek dereceli düğümler setini belirle. Her bir ucundan bir tek dereceli düğüme bağlı olan her yol yerine kanal yönlerini dikkate almayarak  $A'VA$  ve  $E'$  deki yolları yerleştirerek oluşan daireleri belirle. Bir daire oluşturulduğunda ortaya çıkan grafik simetrik kalacak ve çift olacak şekilde, bu dairenin yönlü kanallarını ya sil ya da grafiğe ilave et ve yönsüz kanalları yönlendir.

Adım 2 de bir çift grafiğin korunamayabilmesi olasılığından dolayı, Adım 3 gerekli olmaktadır. bu konuyu açıklamak için şekil 4.3 faydalı olabilir. Şekil 4.3 de (a) Adım 1 in sonunda elde edilen bir çift fakat simetrik olmayan grafiği göstermektedir, (b) Adım 2 nin sonundaki simetrik fakat çift olmayan grafik, (c) Adım 2 nin sonunda grafiğe uygulanan Adım 3 ü göstermektedir ve (d) final çift, yönlü ve simetrik grafik.



Şekil 4.3. MIXED1 sezgiseli ile elde edilen çözüm

-MIXED2 sezgiseli:

Adım 1. Mixed1 sezgiselinin Adım 2 sine bak. Ortaya çıkan grafik  $G'=(N,A'UE')$  olsun.

Adım 2.  $G''=(N,E')$  deki tek dereceli düğümleri belirle.  $\bar{G}=(N,E)$  grafiği üzerinde bu tek dereceli düğümler arasındaki en kısa yolları bul. En kısa yol mesafelerini kullanarak  $\bar{G}$  nin tek dereceli düğümlerinin bir minimum maliyetli eşleşmesini uygula. Eşleştirmede kullanılan yönsüz kanalları  $E'$  ye sok.  $A'$  daki kanallar ve sonradan yönlendirilmiş kanallar ve  $E'$ deki kanallar üzerinde bir tur bul.

Christofides, Benavent, Campos, Corberan ve Mota (1984) de, herhangi bir uygun çözüme uyabilen bir iyileştirme algoritması önerilmektedir. Bu algoritmaya göre eğer bir yönlü kanalın maliyeti ( $c_{ij}$ ),  $N_j$  den  $N_i$  ye bir en kısa yolun uzunluğunu aşıyorsa değişkenlerin sayısını azaltmak için bu yol yerine  $(N_i, N_j)$  yerleştirmek avantajlı olmaktadır. Bu fikir ayrıca Gendreau, Laporte ve Zhao (1990) da da kullanılmaktadır.<sup>86</sup>

#### 4.2. Kırsal Postacı Problemi (RPP)

Kanal rotalama problemlerinin ikinci bir türü olan kırsal postacı probleminde (rural postman problem, RPP) bütün kanallar değil sadece belirli bazı kanallar ziyaret edilmektedir. Bir başka ifade ile bir  $G=(N,A)$  grafiğinde kanalların sadece bir  $R \subseteq A$  alt setinden geçilmesi gerektiği zaman problem bir RPP ye dönüşmektedir. Pratikte bir şebekenin bütün kanallarına hizmet verilmesi zorunluluğu olan az sayıda problem söz konusudur. Gerçek yaşamda kanal rotalama uygulamalarının çoğu RPP olarak ortaya çıkmaktadır.

<sup>86</sup> Eiselt ve diğerleri, Part I, Agm., S. 239.

CPP nin hem yönlü hem de yönsüz versiyonu polynomial zaman içinde çözülebilirken, RPP nin yönsüz ve yönlü versiyonlarının her ikisi de NP-harddır.<sup>87</sup> R bir yönlü kanallar seti ve A\R bir yönsüz kanallar seti olmak üzere, RPP nin karışık versiyonu "stacker crane problemi (SCP)" olarak bilinmektedir ve bu problem de NP-harddır.<sup>88</sup>

MCPP ler için olduğu gibi RPP ler içinde standart bir algoritmik strateji, grafiği dengeli olacak şekilde genişletmek ve sonra genişletilmiş grafik üzerinde polynomial zaman içinde bir Euler dairesi elde etmektir. RPP ler NP-hard olduklarından dolayı birinci aşama için sezgiseller kullanılabilir. Bu sezgiseller genellikle, bir genişletilmiş grafik oluşturmak için eşleştirme algoritmalarını veya minimal tarama ağacı algoritmalarını kullanmaktadırlar. İkinci aşamada Euler uru elde etmek için ise kesin algoritmalar kullanılmaktadır.

#### 4.2.1. Yönsüz Kırsal Postacı Problemi

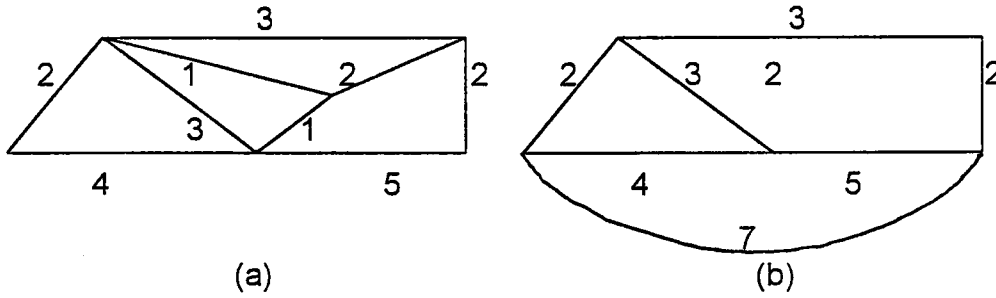
Yönsüz RPP de bütün kanallar yönsüzdürler. Birleştirilmiş  $\bar{G}=(N,R)$  grafiği üzerinde ifade edilen problem, tek dereceli düğümler arasında (G deki) en kısa zincirleri hesaplayarak ve sonra yönsüz CPP deki gibi ilerleyerek çözülebilmektedir.<sup>89</sup> Genel olarak RPP, değiştirilmiş bir  $G'=(N',A')$  grafiği üzerinde çözülmektedir. Dönüştürülmüş grafik  $G'$  de  $N'=\{N_i \in N: (N_i, N_j) \in R, \text{ bir } N_j \in N \text{ için}\}$  şeklinde ifade edilmektedir.  $A'$  ise önce  $A$  ya her bir  $N_i, N_j \in N'$  için maliyeti ( $c_{ij}$ ) olan bir yönsüz kanal ilave ederek ve sonra aynı maliyete sahip olan iki paralel yönsüz kanaldan birisinin yanı sıra bir  $k$  için  $c_{ij}=c_{ik}+c_{kj}$  olan bütün  $(i, j) \in A \setminus R$  yönsüz kanallarını silerek elde edilmektedir.

<sup>87</sup> J.K. LENSTRA, A. H.G. RINNOOY KAN, "On general routing problems", Networks 6, 1976, S. 273-280.

<sup>88</sup> G. N. FREDERICKSON, M. S. HECHT, C. E. KIM, "Approximation algorithms for some postman problems", SIAM J. Computer 7, 1978, S. 178-193.

<sup>89</sup> H.A. EISELT, M. GENDREAU, G. LAPORTE, "Arc routing problems, Part II: The rural postman problem", Operations Research 43(3), 1995, S. 402.

Bu konuyu açıklamak için Şekil 4.4 de gösterilen iki grafik yardımcı olabilir. Şekil 4.4 de R nin yönsüz kanalları üzerindeki rakamlar simetrik kanal maliyetlerini göstermektedirler. Dönüştürülmüş G' grafiğinde R seti, N' nin bir kısmını oluşturan  $N_1, N_2, \dots, N_p$  düğüm setlerine karşılık gelen ve birbirleri ile bağlantılı olan p adet  $G_1, G_2, \dots, G_p$  bileşenini ortaya koymaktadır.



Şekil 4.4. Yönsüz RPP için grafik dönüşümü (a) orijinal grafik, (b) dönüştürülmüş grafik (G')

-Yönsüz RP için sezgisel algoritma:<sup>90</sup>

Adım 1. (Minimal tarama ağacı).  $G_1, G_2, \dots, G_p$  yi birbirlerine bağlayan minimal tarama ağacını (T) oluştur.  $I(T)$  onun uzunluğu olsun. R nin bütün yönsüz kanallarının uzunlukları toplamını  $I(R)$  ile ve bir optimal RPP çözümünün değerini  $Z^*$  ile göster. Bu durumda,  $I(T) + I(R) \leq Z^*$  dır.

Adım 2. (Minimum maliyetli eşleştirme). RUT ile gösterilen grafiğin bütün tek dereceli düğümlerinin bir minimum maliyetli eşleştirmesini bul. M, optimal eşleştirmedeki yönsüz kanallar seti ve  $I(M)$  bunların toplam uzunluğu olsun.

Adım 3. (Euler dairesi). Bir RPP çözümü, RUTUM ile gösterilen grafik üzerindeki bir Euler dairesidir.  $I(M) \leq Z^*/2$  dir ve bu

<sup>90</sup> Eiselt ve diğerleri Part II, Agm., S.402.

nedenle Euler daresinin uzunluđu  $I(T)+I(R)+L(M)$ ,  $3Z/2$  yi aşmamaktadır.

#### 4.2.2. Yönlü Kırsal Postacı Problemi

Yönlü RPP de bütün kanallar yönlere sahiptirler. Birbirlerine bağılı düğüm ve kanallardan oluşan bir  $\bar{G}=(V,R)$  grafiğı üzerindeki problem, yönsüz durumdakine çok benzer bir yol ile oluşturulan bir değıştirilmiş  $G'=(N',A')$  grafiğı üzerinde çözülmektedir. Yönsüz durumdakinden farklı olarak burada  $G'$  grafiğı oluşturulurken dahil edilen  $(N_i,N_j)$  kanalları yönlüdürler ve  $N_i$  den  $N_j$  ye en kısa yolun uzunluđuna eşit bir  $c_{ij}$  uzunluđuna sahiptirler. Düğüm setlerine karşılık gelen birbirlerine bağılı  $G_1, G_2, \dots, G_p$  bileşenleri yönsüz RPP deki gibi belirlenmektedir. Christofides, Campos, Corberan ve Mota (1986) da yönlü RPP için aşağıdaki sezgisel önerilmektedir.

-Yönlü RPP için sezgisel algoritma:

Adım 1. (En kısa tarama ağacı). Keyfi bir düğümü kök olarak alan ve  $G_1, G_2, \dots, G_p$  yi birbirlerine bağlayan en kısa tarama ağacını oluştur (bkz. Emonds(1967)). Ortaya çıkan grafik  $\tilde{G}$  olsun.

Adım 2. (Dönüşüm problemi). Yönlü CPP için olduđu gibi, her bir düğümüne giren kanalların sayısı çıkan kanalların sayısına eşit olacak şekilde en düşük maliyetli kanalları ilave ederek  $\tilde{G}$  den bir Euler grafiğı türet.

Adım 3. (Euler dairesi). Genişletilmiş grafik üzerinde bir Euler dairesi belirle.

#### 4.2.3. Karışık RPP (Stacker Crane Problemi)

Stacker crane problemi (SCP) olarak da adlandırılan karışık kırsal postacı problemi, bir karışık  $G=(N,AUE)$  grafiğı üzerinde tanımlanmaktadır.

Burada problem, A'nın her bir kanalından (yönlü kanallardan) en az bir kere geçen bir en kısa devir belirleme problemidir. A'nın kanalları, bir vinç tarafından her biri kesinlikle bir kere belirli bir yönde gerçekleştirilecek hareketler gibi görülebilir. Kanallar ayrıca bir araç tarafından yapılacak teslimatlar olarak izah edilebilir. Bu durumda SCP, teslimatlar arası maliyetlerin toplamını minimize etmeye denk olmaktadır. Eğer A'nın her  $(N_i, N_j)$  kanalının maliyeti  $(c_{ij})$  sıfır ise SCP bir TSP'ye denktir ve bu nedenle SCP NP-hard'dır.

Frederickson, Hecht ve Kim (1978), bu problemin çözümü için aşağıdaki iki koşulu taşıması gereken iki sezgisel önermişlerdir.

- (1) G'deki her bir düğüme en az bir yönlü kanal bağlı olmalıdır.
- (2) G'deki yönsüz kanalların maliyetleri  $(c_{ij})$  üçgen eşitsizliğini sağlamalıdır.

G grafiği yukarıdaki iki koşulu sağlamaması durumunda bu koşulları sağlayan denk bir  $G'$  grafiğine dönüştürülebilmekte ve sonra problem  $G'$  üzerinde çözülerek orijinal grafiğe göre izah edilebilmektedir. Şimdi aşağıda G'nin her iki koşulu da sağladığı varsayımı ile iki sezgisel (BÜYÜK KANALLAR ve KÜÇÜK KANALLAR) açıklanmaktadır. BÜYÜK KANALLAR sezgiseli, toplam kana maliyetinin optimum tur maliyetine oranla çok büyük olması durumunda ve KÜÇÜK KANALLAR sezgiseli de bunun aksi durumda daha iyi işlemektedir.

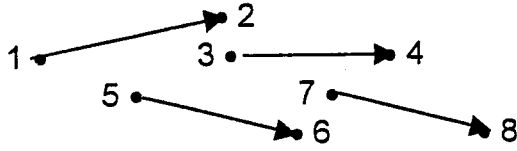
- BÜYÜK KANALLAR sezgiseli:

Adım 1. (İki parçalı grafik). G'den yönlü kanallar setini (A) al (Şekil 4.5a). Birinci düğümler seti A'nın kanallarının kuyrukları ve ikinci düğümler seti bu kanalların başlıkları olmak üzere iki parçalı bir  $G'$  grafiği oluştur. Aynı kanala ait olan kuyruk ve başlıklar hariç, bütün başlıklar ve kuyruklar arasında yönsüz kanallar oluştur (Şekil 4.5b). bu kanalların maliyetlerini en kısa yol uzunluklarına eşit al.

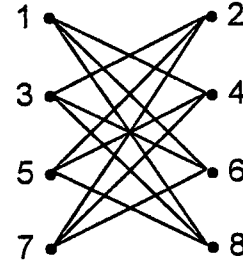
Adım 2. (Eşleştirme).  $G'$  üzerinde bir minimum maliyetli iki parçalı eşleştirme problemini çöz.  $A$  dan ibaret olan bir kanal setine ve başlıktan kuyruğa doğru yönlendirilen eşleştirme çözümünün yönsüz kanallarına sahip olan bir yönlü grafik ( $G''$ ) oluştur. Örneğin, şekil 4.5c de eşleştirme çözümünün  $\{(1,4), (2,3), (5, 8), (6,7)\}$  olduğu varsayılmaktadır.

Adım 3. (En kısa tarama ağacı).  $G''$  grafiği, birbirleri ile çakışmayan bir çok bağlantılı bileşenden ibarettir. Orijinal yönsüz kanal maliyetlerini kullanarak, bu bileşenleri birbirlerine bağlayan en kısa ağacı belirle (iki bileşen arasındaki bir yönsüz kanalın maliyeti, bu bileşenler arasındaki en ucuz bağlantının maliyetine eşittir) (Şekil 4.5d).

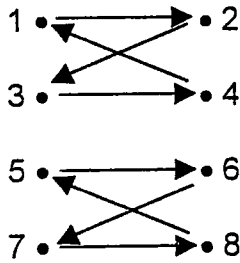
Adım 4. (Euler turu).  $G''$  nin bütün kanallarından oluşan bir kanal setine ve her bir yönde bir adet olmak üzere en kısa tarama ağacının her bir yönsüz kanalı için iki adet yönlü kanala sahip olan bir yönlü grafik ( $\bar{G}$ ) oluştur (Şekil 4.5e). Bu grafik Euler grafiğidir ve gerekli SCP çözümü,  $\bar{G}$  üzerinde bir yönlü CPP çözülerek belirlenmektedir.



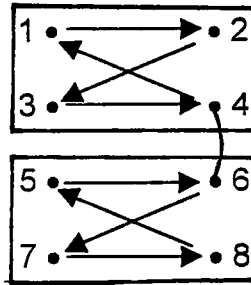
(a)



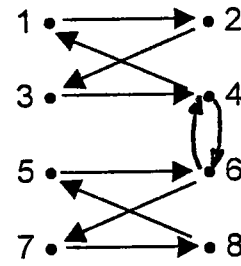
(b)



(c)



(d)



(e)

Şekil 4.5. BÜYÜK KANALLAR algoritmasının gösterimi. (a) orijinal yönlü kanallar seti (A). (b) iki parçalı grafik ( $G'$ ). (c)  $G''$  grafiği. (d) İki bağlantılı bileşenli en kısa tarama ağacı. (e) yönlü grafik ( $\bar{G}$ )

- KÜÇÜK KANALLAR sezgiseli:

Adım 1. (Daraltılmış grafik).  $G$  den aşağıdaki gibi bir daraltılmış yönsüz grafik ( $G^*$ ) oluştur.  $A$  nın her yönlü kanalı için bir düğüm belirle ve bu düğümler arasına yönsüz kanallar ilave et.  $(N_i, N_j)$  ile  $(N_k, N_l)$  kanallarına karşılık gelen düğümler arasındaki bir yönsüz kanalın maliyeti,  $\min\{c_{ik}, c_{il}, c_{jk}, c_{jl}\}$  ye eşittir (Şekil 4.6b).  $G^*$  nin düğüm çiftleri arasındaki bütün en kısa zincirleri belirle ve yönsüz kanal yerine uygun en kısa zinciri yerleştirerek  $G^{**}$  grafiğini elde et (Şekil 4.6c).

Adım 2. (En kısa tarama ağacı).  $G^{**}$  üzerinde en kısa tarama ağacını belirle (Şekil 4.6d).

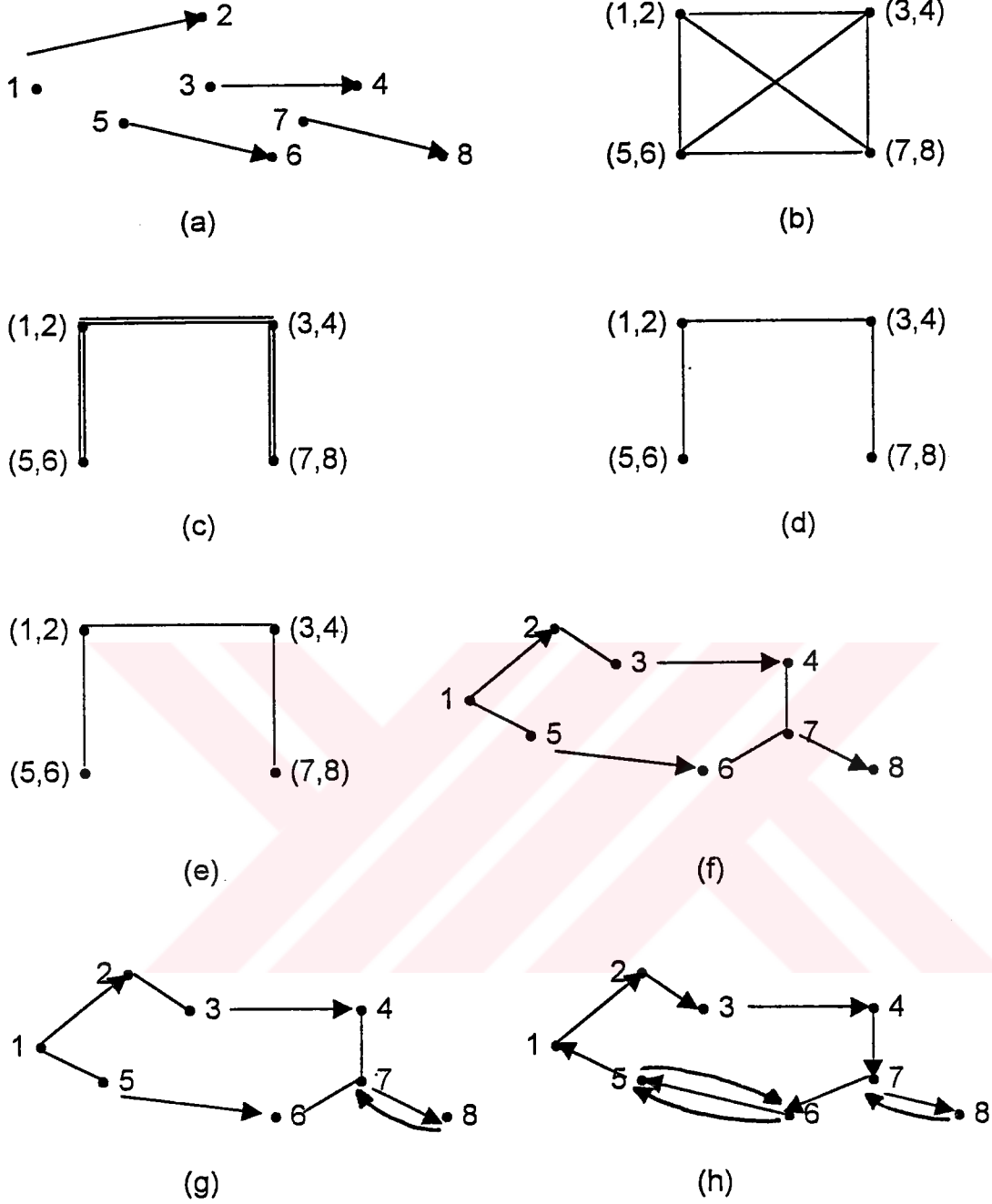
Adım 3. (Minimum maliyetli eşleştirme). Tarama ağacındaki tek dereceli düğümlerin bir minimum maliyetli eşleştirmesini uygula (Şekil 4.6e).

Adım 4. (Tekrar genişletilmiş grafik). Bütün düğümler yerine  $A$  nın uygun kanallarını yerleştir ve her bir tarama ve eşleştirme yönsüz kanalı yerine Adım 1 deki uygun en kısa zinciri yerleştir ve bir karışık grafik ( $\hat{G}$ ) elde et (Şekil 4.6f). Eğer  $(N_i, N_j)$   $G$  nin bir kanalı ise  $N_i$  ve  $N_j$  nin her ikisi de tek dereceye sahip iseler  $(N_i, N_j)$  yönsüz kanalını ilave et ve bu kanalı geçici olarak  $N_j$  den  $N_i$  ye doğru yönlendir. Ortaya çıkan grafik  $G^\#$  olsun (Şekil 4.6g).

Adım 5. (Birinci Euler turu). Çift kanal yönlerini ihmal ederek  $\bar{G}$  de bir Euler turu bul. Eğer geriye doğru katedilen çift kanalların uzunluğu,  $l(\bar{A})/2$  yi aşarsa, turun yönünü ters çevir.

Adım 6. (Kanal ilavesi). Her bir yönsüz kanalı turun yönüne uydur. Eğer bir çift kanal  $(N_i, N_j)$  yanlış yönde katediliyorsa, iki adet  $(N_j, N_i)$  kanalı ilave et. Ortaya çıkan yönlü grafik  $\tilde{G}$  olsun (Şekil 4.6h).

Adım 7. (Final Euler turu).  $\tilde{G}$  üzerinde bir Euler turu belirle.



Şekil 4.6. KÜÇÜK KANALLAR algoritmasının gösterimi. (a) orijinal kanal seti  $A$ . (b) yönsüz grafik  $(G^*)$ . (c) yönlü grafik  $(G^{**})$ . (d)  $G^{**}$  üzerinde en kısa tarama ağacı. (e) tek dereceli düğümleri eşleştirme. (f) Karışık grafik  $(\hat{G})$ . (g) karışık grafik  $(G^\#)$ . (h) yönlü grafik  $(\tilde{G})$ .

### 4.3. Kapasite Kısıtlı Kanal Rotalama Problemi (CARP)

Kapasite kısıtlı kanal rotalama problemi (capacitated arc routing problem, CARP) genel kanal rotalama problemlerinin gerçek yaşam şartlarını daha iyi yansıtan bir versiyonudur. Bu tür problemlerde her bir kanal, bir pozitif  $q_{ij}$  talebine sahiptir ve bu kanallara hizmet vermek üzere her biri  $Q$  kapasiteli olan bir homojen araçlar filosu mevcuttur.

Yönsüz bir  $G=(N,E)$  şebekesi üzerinde tanımlanan CARP, bütün kanal taleplerini tatmin etmek üzere her biri merkez depodan (düğüm 1) başlayıp aynı noktada sona eren ve üzerindeki toplam talep araç kapasitesini ( $Q$ ) aşmayan minimal maliyetli turlar setini oluşturmaktan ibarettir. Burada bir araç bir kanaldan hizmet vermeden geçebilmektedir. Fakat her bir kanalın talebi kesinlikle bir tek araç tarafından karşılanmalıdır.

CARP kapasite kısıtları içerdiği için, problemin bazı yaklaşık kısıtlanmış versiyonlarını bile optimal olarak çözmek çok zordur. Golden ve Wong (1981) de, basit şebekeler üzerinde optimal CARP çözümünün maliyetinin 1.5 katından daha düşük maliyetli bir uygun CARP çözümü bulma probleminin (0.5-yaklaşık CARP) NP-hard olduğu gösterilmektedir.

#### 4.3.1. CARP İçin Sezgisel Algoritmalar

NP-hard bir problem olan CARP için geliştirilmiş olan sezgiseller genel olarak üç kategoriye sınıflanabilmektedirler. (1) Basit inşa metodları, (2) İki-aşamalı inşa metodları ve (3) İyileştirme metodları.

İkinci kategoriye giren iki-aşamalı inşa metodları, önce küme-sonra rota ve önce rota-sonra küme sezgiselleri olmak üzere iki alt kategoriye ayrılmaktadırlar. Bunlardan önce küme-sonra rota yaklaşımında ilk olarak kanallar, her biri  $Q$  yu aşmayan bir toplam ağırlığa sahip olan kümelere

bölünmekte ve sonra bir CPP algoritması ile her bir küme için bir araç rotası belirlenmektedir.<sup>91</sup>

Önce rota-sonra küme yaklaşımında, önce pozitif talepli olan bütün kanallar üzerinde büyük bir Euler turu oluşturulmakta ve sonra bu tur uygun kümelerle bölünerek, her bir küme için bir araç turu inşa edilmektedir. Ulusoy (1985), ilk düğüm dışındaki düğümleri Euler turundaki kanallara ve kanalları da G üzerindeki uygun araç turlarına karşılık gelen bir yardımcı grafik ( $G^*$ ) oluşturmakta ve sonra  $G^*$  üzerinde bir en kısa yol problemini çözerek en düşük maliyetli araç turları setini elde etmektedir.

Herhangi bir tur inşa sezgiseli ile elde edilen çözüme iyileştirme algoritmaları uygulanabilmektedir (bkz. Ulusoy (1995) ve Win (1987)). Tipik olarak böyle iyileştirme metodları, TSP için kanal değiştirme sezgisellerinden kaynaklanmaktadır. Bu amaçla faz dönüşüm ve yasak araştırması metodları da kullanılabilmektedir. Şimdi aşağıda bazı basit tur inşa sezgiselleri açıklanmaktadır.

#### 4.3.1.1. Kur-Kır Algoritması

Christofides (1973) tarafından geliştirilmiş olan kur-kır algoritması üç adımdan oluşmaktadır. Birinci adımda, aşağıdaki iki özelliğe ((i) ve (ii)) uygun olarak, mevcut grafikten turlar oluşturulmaktadır.

(i) Tur üzerinde hizmet veren talep kanallarının turdan çıkartılmaları mevcut grafiği birbirinden ayrı bileşenlere bölmemelidir.

(ii) Tur üzerinde hizmet verilen kanal taleplerinin toplamı, araç kapasitesini aşmamalıdır.

Genelde (i) özelliğini taşıyan turlar grafiğin özel bir alanındaki kanallardan oluşmaktadır. (ii) özelliği ise üretilen çözümlerin uygunluklarını temin etmek için gerekli olmaktadır.

---

<sup>91</sup> Eiselt ve diğerleri, Part II, Agm., S. 410.

Algoritmanın ikinci adımında Adım 1 de oluşturulan turlar üzerinde hizmet verilen kanallar kaydedilmekte ve bunlar mevcut grafikten silinmektedir. Buna ilaveten Adım 3 de grafiğe ilave edilmiş olan yapay kanallar da algoritmanın bir sonraki döngüsünde ikinci adımda silinmektedirler. Bu durumdaki grafik, sadece düğümleri ve hizmet verilmemiş kanalları içermektedir.

Algoritmanın üçüncü ve son adımı, Adım 2 deki grafiği ele almakta ve deponun (düğüm 1) pozitif ve çift dereceli olmasını ve mevcut grafikteki bütün diğer düğümlerin çift dereceli olmalarını sağlamak için grafiğe bir minimum maliyetli yapay kanallar seti ilave etmektedir. Algoritma bunu mevcut grafiğin tek dereceli düğümleri üzerinde bir minimum maliyetli eşleştirme bularak yapmaktadır. Yapay kanallar ve bunların maliyetleri, tek dereceli düğümler arasındaki minimum maliyetli yollara dayanılarak ilave edilmektedirler. Depo sıfır dereceli olduğu zaman, bu depo yerine çok yüksek maliyetli bir kanal ile (eşleştirmede kullanılmasını önlemek için) birbirine bağlanan iki yapay düğüm yerleştirilmektedir. Daha sonra, eşleştirme çözümü ile belirlenen yapay kanalları içeren mevcut grafik Adım 1 e gönderilmekte ve hizmet verilmemiş kanal kalmayıncaya kadar Adım 1-3 tekrarlanmaktadır. Bu algoritmanın kompleksliği  $O(A N^3)$  dür.<sup>92</sup>

Pearn (1989) da, kur-kır çözümlerini iyileştirmek için bazı değişiklikler önerilmektedir. Bu değiştirilmiş algoritmada, daha başka tur bulunamayıncaya kadar ard arda uygun turlar oluşturulmakta ve silinmektedir. Kalan grafiği tekrar birbirine bağlamak için minimal tarama ağacı algoritması ve bir Euler turu oluşturabilmek için eşleştirme algoritması uygulanmaktadır. Eğer ortaya çıkan grafikte uygun turlar bulunabilirse, algoritma uygun tur kur-kır aşamasına geri dönmektedir. Aksi halde araç kapasitesi tükeninceye kadar keyfi kanallar seçilmektedir. Bütün grafik kapsanıncaya kadar kur-kır, minimal tarama ağacı ve

---

<sup>92</sup> W. L. PEARN, "Augment-Insert algorithms for the capacitated arc routing problem", Computers and Operations Research 18(2), 1991, S. 190.

eşleştirme algoritmaları tekrarlanmaktadır. Algoritmanın kompleksliği,  $O(A.N^4)$  dür.<sup>93</sup>

#### 4.3.1.2. Yol Tarama Algoritması

Yol tarama algoritması, belirli bir miyopik optimizasyon kriteri kullanarak bir seferde bir tur inşa etmeye dayanmaktadır. Her bir tur oluşturulurken, araç kapasitesi tükeninceye kadar en ümit verici görünen kanal eklenerek bir yol büyütülmekte ve sonra bir komple tur oluşturmak için depoya en kısa dönüş yolu izlenmektedir. Golden, De Armon ve Baker (1983) tarafından yapılan bir çalışmada kanal seçim sürecinde beş yol tarama optimizasyon kriteri kullanılmaktadır. Bunlar;

- (1) Kalan talep için birim başına  $c(i, j)$  maliyetinin minimize edilmesi
- (2) Kalan talep için birim başına  $c(i, j)$  maliyetinin maximize edilmesi
- (3)  $j$  düğümünden depoya geri dönüş maliyetinin minimize edilmesi.
- (4)  $j$  düğümünden depoya geri dönüş maliyetinin maximize edilmesi
- (5) Eğer kalan araç kapasitesi yarıdan az ise,  $j$  düğümünden depoya geri dönüş maliyetinin maximize, aksi halde minimize edilmesi.

Bu kriterlerin her biri, komple bir çözüm oluşturmak için kullanılmakta ve en iyi olanı final çözüm olarak seçilmektedir. Bu algoritmanın kompleksliği  $O(N^3)$  dür.<sup>94</sup>

Yol tarama algoritmasının Pearn (1989) tarafından geliştirilen bir versiyonunda, komple bir çözüm oluşturmak için her bir kriterin tek tek kullanılması yerine aşağıdaki kriter karışımları kullanılmaktadır.

<sup>93</sup> W. L. PEARN, "Approximate solutions for the capacitated arc routing problem", *Computeris and Operations Research* 16(6), 1989, S. 593.

<sup>94</sup> Pearn, 1989, Agm., S. 590.

- (1) Her bir adımda kullanmak için beş kriterden birisi rastgele olarak seçilir ve komple bir çözüm oluşturulur.  $k$  adet çözümün en iyi olanı bu yeni yaklaşımın final çözümü olarak seçilir.
- (2) Her bir adımda kullanmak için değişken olasılıklar ile beş kriterden birisi seçilir (Örneğin, kriter 1 0.6 olasılığını ve diğer dört kriter 0.1 olasılıklarını alabilir) ve bir komple çözüm oluşturulur.  $k$  adet çözümün en iyi olanı bu yaklaşım için final çözüm olarak seçilir.

#### 4.3.1.3. Genişlet-Birleştir Algoritması

Golden, De Armon ve Baker (1983) tarafından geliştirilmiş olan algoritmanın uygulama adımları aşağıdaki gibidir.

Adım 1. Bütün talep kanallarına ayrı birer tur ile hizmet ver.

Adım 2. Mevcut en uzun tur ile başlayarak, küçük bir tur üzerindeki bir kanala daha büyük bir tur üzerinde hizmet verilip verilemeyeceğini araştır.

Adım 3. Kapasite kısıtlarına göre, herhangi iki turu birleştirmeyi değerlendir. En büyük pozitif tasarrufu veren iki turu birleştir.

Adım 4. Bitinceye kadar Adım 3 ü tekrarla.

Görüldüğü gibi Adım 3 ve 4, Clarke-Wright tasarruf sezgiseli ile yakın olarak ilişkilidirler.

#### 4.3.1.4. Paralel Giriş Algoritması

Chepleau, Ferland, Lapalme ve Rousseau (1984) tarafından bir okul servisi rotalama uygulaması için geliştirilmiş olan paralel giriş algoritması, yol tarama algoritmasına benzemektedir. Ancak burada bir çok paralel rota oluşturulmaktadır. Paralel giriş algoritmasında iki tamamlayıcı strateji söz konusudur. Bunlar;

- (1) Belirli bir kanal için, turda boş geçen zamanı minimize etmek için bu kanalın sokulması gereken mevcut rotanın belirlenmesi,
- (2) Belirli bir rota için, kalan ve daha sonra sokulması gereken kanalın belirlenmesidir.

Algoritma, rotalar oluşuncaya kadar bu iki strateji arasında gidip gelmektedir. Aşağıda böyle bir teknik özetlenmektedir.

Algoritmada kullanılan değişkenler listesi:<sup>95</sup>

BC : maximum araç kapasitesi

$L(i)$  : "i" rotasının toplam uzunluğu

LB : araç sayısının alt sınırı

NA : toplanacak öğrencilerin bulunduğu kanalların sayısı

NR : rotaların sayısı

NRS : kaydedilen çözümdeki rota sayısı

PH : aşama numarası

$R(i)$  : "i" rotasındaki öğrenci sayısı

REL : kısıtlar gevşetilmiş ise "1", aksi halde "0"

$S(a)$  : "a" kanalı üzerindeki öğrenci sayısı

SI : bu adımda sokulan öğrenciler

TL : bütün rotaların toplam uzunluğu

TLS : kaydedilen çözümdeki toplam rota uzunluğu

---

<sup>95</sup> L. CHAPLEAU, J. A. FERLAND, G. LAPALME, J. M. ROUSSEAU", A parallel insert method for the capacitated arc routing problem", Operations Research Letters 3 (2), 1984, S. 98.

UB : bir rota uzunluğunun üst sınırı.

Adım 0. Varış yerinden en uzak olan kanalı (a) bul ve onu birinci rotayı oluşturmak için kullan.  $NR=1$ ,  $TL=bu\ rotanın\ uzunluğu$ ,  $NA=NA-1$ ,  $R(1)=S(a)$ ,  $PH=0$ ,  $REL=0$ ,  $NRS=\infty, TLS=\infty$  olsun.

Adım 1. (Strateji I).  $SI=0$

1.1. "a"= kalan en uzak kanal olsun. Eğer  $S(a)+R(i)>BC$  (bütün i,  $1 \leq i \leq NR$  için) ise, Adım 1.2 ye git. Aksi halde, "i"=a sokulduğunda en düşük atıl zamanı oluşturan ve a sokulduktan sonraki toplam uzunluğu UB yi aşmayan rota olsun. Eğer böyle bir "i" yoksa, Adım 1.2 ye git. a yı "i" rotasına sok,  $R(i)=R(i)+S(a)$  yap, "i" nin uzunluğunu ve TL yi düzelt,  $SI=SI+S(a)$ ,  $NA=NA-1$  yap. Eğer  $NA>0$  ise Adım 1.1 e git, aksi halde Adım 7 ye git.

1.2. Eğer  $NR<LB$  ise  $NR=NR+1$  yap. a yı NR rotasına sok,  $R(NR)=S(a)$  yap, NR rotasının uzunluğu bu rotanın uzunluğudur.  $NA=NA-1$  yap, TL yi düzelt, Adım 1.1 e git. Aksi halde, eğer  $PH=1$  ve  $SI>0$  ve  $REL=0$  ise Adım 3 e git.

Eğer  $PH=1$  ve  $SI>0$  ve  $REL=1$  ise Adım 5 e git.

Adım 2. (Strateji II).  $SI=0$  yap ve bütün rotaları "açık" olarak etiketlendir.

2.1. i= en boş olan "açık" rota olsun. Eğer  $R(i)+S(a)>BC$  (kalan bütün kanal a için) ise Adım 2.2 ye git. Aksi halde,  $R(i)+S(a) \leq BC$  olan ve "i" de en küçük atıl süreyi oluşturan kalan kanal olsun. Eğer a nın sokulması "i" nin uzunluğunu UB nin üzerine çıkartıyorsa Adım 2.2 ye git. Aksi halde a yı "i" rotasına sok,  $R(i)=R(i)+S(a)$  yap, i nin uzunluğunu ve TL

yi düzelt,  $SI=SI+S(a)$  ve  $NA=NA-1$  yap. Eğer  $NA>0$  ise Adım 2.1 e git. Aksi halde Adım 7 ye git.

2.2. "i" rotasını "kapalı" olarak etiketlendir. Eğer en az bir rota açık ise, Adım 2.1 e git. Aksi halde, eğer  $PH=1$  ve  $SI=0$  ve  $REL=0$  ise, Adım 4 e git. Eğer  $PH=2$  ve  $REL=1$  ise, Adım 5 e git.

Adım 3. Rotaların uzunluklarını iyileştirmek için bir değişim prosedürü kullan.  $PH=1$  yap ve Adım 1 e git.

Adım 4. (Kısıt gevşetme). BC ve UB ve öğrenci toplama mesafesini bir yüzde ile gevşet.  $REL=1$  yap. Adım 3 e git.

Adım 5. BC, UB ve öğrenci toplama mesafesini orijinal değerlerine ayarla.  $R=0$  yap.  $SI=0$  ise, Adım 3 e git.

Adım 6.  $NR=NR+1$  yap.  $a$ = kalan en uzak kanal olsun.  $a$  ile bir rota oluştur.  $R(NR)=S(a)$  yap,  $NR$  nin uzunluğu bu rotanın uzunluğudur.  $NA=NA-1$  yap, TL yi düzelt. Eğer  $NA>0$  ise, Adım 3 e git.

Adım 7. Eğer  $NR<NRS$  ve  $TL<TLS$  ise, Adım 8 e git. Aksi halde, dur.

Adım 8. Bu çözümü kaydet.  $NRS=NR$ ,  $TLS=TL$  yap. "i"= en boş rotanın indexi olsun. "i" rotasını elimine et.  $NA$ = bu rotadaki öğrencilere ait kanalların sayısı olsun.  $NR=NR-1$  yap, Adım 3 e git.

Yukarıda özetlenen algoritmada Adım 1 ve 2 de, "a" kanalının "i" rotasına sokulması, toplama mesafesi açısından test edilmektedir. Adım 5 de BC, toplama mesafesi ve UB orijinal değerlerine eşitlenmekte fakat bazı rotalar bu kısıtları ihlal edebilmektedir. Böyle rotalar korunmakta ve söz konusu ihlaller azaltılmaya çalışılmakta, fakat kısıtlara uygun olan

diğer rotalar oluşturulabilmektedir. Adım 8 de en boş olan yerine en kısa olan rotanın elimine edilmesi de seçilebilir.

#### 4.3.1.5. Genişlet-Yerleştir Algoritması

Pearn (1991) tarafından geliştirilmiş olan genişlet-yerleştir algoritması, genişlet-birleştir ve paralel-giriş algoritmalarında kullanılan standart düşüncelerin bir bileşimine dayanmaktadır. Bu yöntemde, depodan çok uzak olan talep kanallarına hizmet veren turlar daha kazançlı bir yol ile planlandıkları için seyrek şebekelere ve büyük taleplere sahip problemler daha kolayca çözülebilmektedirler.

Algoritmanın birinci aşamasında, mümkün olduğu sürece bütün kanallar aşamalı olarak depoya bağlı uygun turlar içine sokulmaktadır. Bu turların oluşturulmaları için, birisi maliyete ve diğeri talebe dayanan iki farklı kriter önerilmektedir. İkinci aşamada ise bir tasarruf kriteri kullanılarak, kalan kanallar mevcut turlara dahil edilmektedir.

Algoritmada kullanılan orijinal notasyon ve uygulama adımları aşağıdaki gibidir.<sup>96</sup>

$d_{ij}$ =i ve j düğümleri arasındaki en az maliyetli yolun maliyeti

$D_{ij}=d_{1i}+d_{1j}$

$DCYCLE_{ij} = (i,j)$  kanalından geçen en az maliyetli tur

$INCOST_{uv}=(u,v)$  kanalını mevcut tura dahil etmekten kaynaklanan ilave maliyet

$S = G$  deki hizmet verilmemiş talep kanalları seti.

Aşama 1 (Genişletme)

---

<sup>96</sup> Pearn, 1991, Agm., S. 191.

Adım 1. Başlangıç olarak  $S=G$  dir.  $D_{ij} = \max_{(u,v) \in S} \{D_{uv}\}$  olan  $(i,j) \in S$  talep

kanalı ile başla.  $DCYCLE_{ij}$  turunu bul, Adım 2 ye git. Burada, bir  $(i,j) \in S$  talep kanalı için  $DCYCLE_{ij}$  bulununcaya kadar  $DCYCLE_{ij}$  yi araştırma prosedürü,  $D_{uv}$  ye göre büyükten küçüğe doğru her  $(u, v) \in S$  için tekrarlanmalıdır. Eğer  $DCYCLE_{ij}$  mevcut değil ise, Aşama 2 ye git.

Adım 2. Araç kapasitesi tükeninceye kadar  $D_{uv}$  ye göre büyükten küçüğe doğru  $(u,v) \in DCYCLE_{ij}$  talep kanallarını ilave ederek,  $(i,j)$  kanalından ibaret olan başlangıç turunu genişlet. Komple bir tur oluşturmak için, genişletilmiş turun uç noktalarından depoya (düğüm 1) en az maliyetli dönüş yolları izlenmektedir. Bu komple turu  $G$  grafiğinden çıkart ve  $S$  yi düzelt.

Adım 3. Daha başka  $DCYCLE_{ij}$  (her  $(i,j) \in S$  için) bulunamayınca kadar Aşama 1 in Adım 1 ve 2 sini tekrarla. Aşama 2 ye devam et.

#### Aşama 2 (Kanal girişi)

Adım 1.  $D_{ij} = \max_{(u,v) \in S} \{D_{uv}\}$  olan  $(i,j) \in S$  talep kanalı ile başla.  $C$  kanal

sokma maliyeti üzerindeki üst sınırı göstermek üzere,  $INCOST_{uv} \leq C$  olacak şekilde  $(u,v) \in S$  talep kanalı  $(i,j)$  kanalından ibaret olan başlangıç turu içine yerleştir.

Adım 2. Araç kapasitesi tükeninceye veya depo düğümüne ulaşıncaya kadar  $D_{uv}$  ye göre büyükten küçüğe doğru her  $(u,v) \in S$  talep kanalı için bu giriş sürecini tekrarla. Komple bir daire oluşturmak için genişletilmiş daire yolunun iki uç noktasından depo düğümüne en düşük maliyetli dönüş

yolları izlenir. Bu komple turu G grafiğinden sil ve S yi uygun şekilde düzelt.

Adım 3. Bütün grafik kapsanıncaya kadar Aşama 2 nin Adım 1 ve 2 sini tekrarla.

Aşama 2 nin Adım 1 inde kullanılan kanal giriş maliyeti ( $INCOST_{uv}$ ) için üst sınırın (C) değeri çok büyük veya çok küçük alınmamalıdır. C nin çok büyük seçilmesi durumunda bu aşama, mevcut turdan oldukça uzak olan talep kanallarını tura sokmaya çalışacak ve bu yüzden uzun turlar oluşacaktır. Bunun aksine C değerinin çok küçük seçilmesi durumunda algoritma, sadece mevcut tura çok yakın olan talep kanallarını sokacak ve bunun sonucu olarak tur, araç kapasitesi tükenmeden önce depo düğümüne ulaşacaktır. Bu nedenle, bütün grafiği kapsamak için bir çok ilave turlar oluşturulmalıdır. Pearn (1991) de C için uygun bir aralığın  $[0, \max_{i \in N} \{d_{1i}\}]$  olduğu ifade edilmektedir.<sup>97</sup>

---

<sup>97</sup> Pearn, 1991, Agm., S. 192.

## BEŞİNCİ BÖLÜM

### İŞYERİ ULAŞIM SERVİS ARAÇLARINI ROTALAMA PROBLEMİ ÖRNEĞİ

Bu bölümde iş yeri ulaşım servis araçlarının rotalanmasına ilişkin bir gerçek yaşam problemi sunularak, bir sezgisel çözüm metodu geliştirilmekte ve söz konusu problem için optimale yakın bir sonuç üretilmektedir.

#### 5.1. Problemin Özellikleri

İstanbul'un Güneşli semtinde faaliyet göstermekte olan T. Garanti Bankası A.Ş. Genel Müdürlüğü, bünyesinde çalışmakta olan elemanlarına iş yerine geliş ve gidişlerinde ulaşım hizmeti sağlamak amacı ile merkezi gene Güneşli'de bulunan Dedetur Turizm, İthalat, İhracat, Ticaret ve Sanayi A.Ş. ile 1996 yılında anlaşmış bulunmaktadır.

Bankanın Güneşli'de bulunan Genel Müdürlük Binalarında şu anda toplam 512 personel görev yapmakta ve bunlardan 400 kişi işe gidiş ve gelişlerinde servis araçlarını kullanmaktadır. Servis araçlarından yararlanan elemanlar İstanbul'un Avrupa ve Anadolu yakasında oturanlar olmak üzere iki ana gruba ayrılmaktadırlar. İki yaka arasında deniz yolu alternatifinin varlığı ve toplam durak sayısının çok fazla oluşu, problemin çözümünü aşırı ölçüde karmaşıklştıracağı için, bu çalışmada Avrupa yakasında oturan toplam 294 elemanın ulaşım problemi ele alınmaktadır.

Dedetur A.Ş., banka personelinin Avrupa yakasında ulaşımını sağlamak için herbiri 18 kişi alabilen toplam 21 adet minibüs kullanmaktadır.

Banka personeli sabahları işe giderken kendilerine en yakın belediye otobüsü durakları civarından, servis araçlarına binmektedirler. Biniş durakları sabittir ve servis araç sürücüleri tarafından bilinmektedir.

Probleme ilişkin durakların Bankaya uzaklıklarının ortalaması 16,474 km, minimumu 1,771 km ve maksimumu 38,577 km'dir. Söz konusu duraklardaki talepler minimum 1 kişiden maksimum 9 kişiye değişmektedir.

Bir servis aracının rotasının uzunluğu, bu rota boyunca katedilen toplam mesafeye, rota üzerindeki trafik yoğunluğuna ve rotaya tahsis edilen durakların sayısına bağlı olarak değişmektedir. Ancak buradaki problemde duraklar arası mesafelerin genellikle oldukça uzun olması, elemanların araçlara biniş ve inişleri esnasında harcanan zamanların gözardı edilmelerini haklı kılmaktadır. Trafik yoğunlukları ise araç sürücülerinin geleneksel ve sezgisel olarak oldukça doğru tercihlerine bağlı olarak problemin başlangıcında, duraklar arasındaki en kısa mesafelerin belirlenmesi sırasında doğal olarak dikkate alınmış olmaktadır.

Problemdeki duraklar arasındaki mesafelerin büyük bir kısmı tamamen simetriktir. Duraklar arasında simetrik olmayan mesafelerin gerek sayıca az olması ve gerekse bunlara ilişkin her iki yöndeki mesafeler arasındaki farkların çok küçük olmaları nedeni ile duraklar arası mesafeler matrisi tamamen simetrik alınmaktadır. Ayrıca duraklar arası mesafeler hemen hemen hiçbir zaman iki durak arasındaki direkt uzaklık olmayıp, karayolu ulaşım ağının elverdiği şekilde belirlenmektedir. Ek 3'deki İstanbul toplu taşıma rehber haritası, bütün durakların konumlarını ve aralarındaki karayolu ulaşım ağını ortaya koymaktadır. Probleme ilişkin şu anda yürürlükte olan araç rotaları Ek 5'te ve bu rotalar üzerindeki duraklar, rota uzunlukları ve toplam rota yükleri Ek 4'te gösterilmektedir.

Açıkça görülebileceği gibi, ele alınan rotalama problemi tipik bir tek depo (Güneşli) -Homojen filo- Simetrik Araç Rotalama problemidir. Burada amaç, öncelikle rota ve dolayısı ile gerekli araç sayısını minimize ederken bir yandan da araçlar tarafından katedilen toplam günlük yol uzunluğunu

minimize etmektir. Araçlardaki koltuk kullanım yüzdesinin maksimize edilmesi veya bir başka ifade ile atıl kapasite oranının minimize edilmesi ise, araç sayısının minimize edilmesinin doğal bir sonucu olarak gerçekleşmektedir. Sezgisel metodların yapısı gereği buradaki maksimizasyon ve minimizasyon kavramlarının maksimum ve minimuma uygun ölçüde yakın olmayı ifade ettikleri açıktır.

## 5.2. Probleme İlişkin Verilerin Elde Edilmesi

T. Garanti Bankası Genel Müdürlüğü Güneşli personeline sunulmakta olan ulaşım hizmetlerine ilişkin problemin ele alınması sırasında gerekli olan veriler, öncelikle servis araç sürücülerine dağıtılan üç soruluk bir anket formu ile elde edilmiştir. Anket formunda yer alan sorular aşağıdaki gibi sıralanmaktadır. 1) Aracın alabileceği yolcu sayısı nedir? 2) Aracın izlediği güzergah nedir? 3) Araç hangi duraktan kaç elemanı almaktadır? 21 araç sürücüsü tarafından cevaplanan ve personel tarafından onaylanan anket formlarının değerlendirilmesi Ek 1'de alfabetik sıra ile sergilenen duraklar ve durak talepleri ile Ek 4'te ayrıntıları verilen ve Ek 5'te sergilenen güzergahlar elde edilmiştir.

Rotalama problemlerinin ana değişkeni olan duraklar arası mesafeler, İstanbul Büyükşehir Belediyesi İ.E.T.T. İşletmelerinden elde edilen ve Ek 3'te gösterilen İstanbul toplu taşıma rehber haritasından ölçek yardımı ile belirlenmiştir.

Problemin çözümünde esas alınan ve Ek 2'de sunulan duraklar arası en kısa mesafeler bir yandan coğrafik olarak Ek 3'deki haritadan ve öte yandan trafik yoğunluklarını dikkate almak amacı ile araç sürücüleri ve personelden edinilen bilgilerden yararlanılarak belirlenmiştir.

## 5.3. İnsan Faktörüne İlişkin Kısıtlamalar

Araç rotalama problemlerinde, insanların ulaşımı sözkonusu olduğunda, insani faktörler ilave kısıtlar oluşturmaktadırlar. Burada "insani faktörler" terimi ile ifade edilmek istenen şey, personelin araçlar ile işe

götürülmeleri sırasında yerine getirilmesi gereken bazı gerekliliklerdir. Personelin iş yerinden evlerine götürülmeleri sırasında ise aşağıda sunulmakta olan gerekliliklerin tamamen terslerinin dikkate alınması yeterli olmaktadır.

### 5.3.1. Toplam Mesafe Kısıtı

Yolculuklarına ait oldukları rotanın (aracın) ilk biniş noktasından başlayan elemanların iş yerine varıncaya kadar olan toplam seyahat mesafeleri (süreleri) üzerine bir üst sınır koymak için, bir araç tarafından katedilen yolun toplam uzunluğu, önceden belirlenmiş olan bir DK değerini aşmamalıdır. Söz konusu gereklilik aşağıdaki gibi formüle edilmektedir.

$$\sum D(I,J) \leq DK \quad \text{bütün K'lar için} \quad (5.1)$$

Burada  $D(I,J)$ , I ve J durakları arasındaki mesafeyi ve K da aracı göstermektedir.

Bizim problemimizde DK'nın değeri, Güneşli'ye en uzak olan Sarıyer durağı esas alınarak, 80.000 m. olarak belirlenmiştir. (Bu değer farklı senaryolar ve farklı DK değerleri ile deneme-yanılma yolu ile belirlenmekle birlikte, varış noktasına en uzak durak mesafesinin iki katının alınması da oldukça iyi sonuç vermektedir.)

Bu tip mesafe kısıtı, genellikle dayanıksız malların taşınmasında bozulmaları önlemek için bu tür mallara ilişkin zaman penceresi kısıtlı araç rotalama problemlerinde de elverişli olacaktır. Bununla birlikte, bizim çalışmamızda bu kısıt, insan faktöründen doğan bir özel kısıt olarak probleme dahil edilmektedir.

### 5.3.2. Yön Kısıtı

İnsan doğasından dolayı açıktır ki, işine gitmek üzere bir araca binen eleman daima iş yeri yönünde seyahat etmek isteyecek yani geri dönüşlerden rahatsız olacaktır. Bu gereklilik herhangi iki ardışık durak (I ve J) için aşağıdaki gibi ifade edilebilir.

$$D(I,1)-D(J,1)\geq 0 \quad \text{bütün } I, J \text{ ve } K \text{ lar için} \quad (5.2)$$

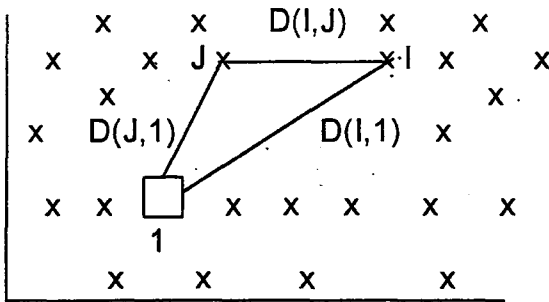
Burada "1" indisi varış noktasını (Güneşli) temsil etmektedir.

### 5.3.3. Maximum Sapma Kısıtı

Problemde daha yüksek koltuk kullanım oranı ve daha az sayıda araç amacını karşılamak için, işyerlerine en kısa yoldan sapmalara izin verilmesi durumunda doğal olarak söz konusu sapmalar üzerine bir sınır konulmalıdır. DK toplam rota uzunluğu üzerinde bir sınır oluştururken her bir biniş noktasından maksimum sapma üzerinde de bir sınır gerekmektedir. Bu amaçla I ve J duraklarının birbirlerine bağlanmaları sonucunda ortaya çıkan toplam uzaklık ile I durağından son varış noktasına en kısa uzaklık arasındaki farkın önceden belirlenmiş bir P değerini aşmaması durumunda, söz konusu bağlantının gerçekleştirilmesine izin verilmektedir.

$$D(I,J)+D(J,1)-D(I,1)\leq P, \quad \text{bütün } I, J \text{ ve } K' \text{lar için} \quad (5.3)$$

Maksimum sapma kısıtı Şekil 5.1'de gösterilmektedir. Şekil 5.1'de yalnızca  $D(I,J)+D(J,1)-D(I,1)\leq P$  olması durumunda I durağı J durağına bağlanabilir yani J'den I'ye gidilmesine izin verilmektedir.



Şekil 5.1. Maksimum sapma kısıtı (x= biniş noktası, □= merkez depo/iş yeri)

Yukarıda söz edilen (5.1), (5.2) ilave kısıtları, bu tip araç rotalama problemini literatürde sunulan klasik problemlerden ayırmaktadır. Açıkta ki

gerçek yaşamda ortaya çıkan VRP modeli çok sayıda kısıt ve çok fazla sayıda değişken içeren kompleks bir birleşik (combinatorial) problemidir. Bu çalışmada sözkonusu problemi çözmek için, ayrıntıları aşağıdaki bölümde açıklanan bir sezgisel metod önerilmektedir.

#### 5.4. Önerilen Sezgisel Metod

Yukarıda açıklanan araç rotalama probleminin özel yapısına uygun bir sezgisel algoritma geliştirilmiştir. Sözkonusu algoritma bir tur-inşa metodudur ve Bölüm 5.3'te açıklanan üç ilave kısıtı dikkate almaktadır.

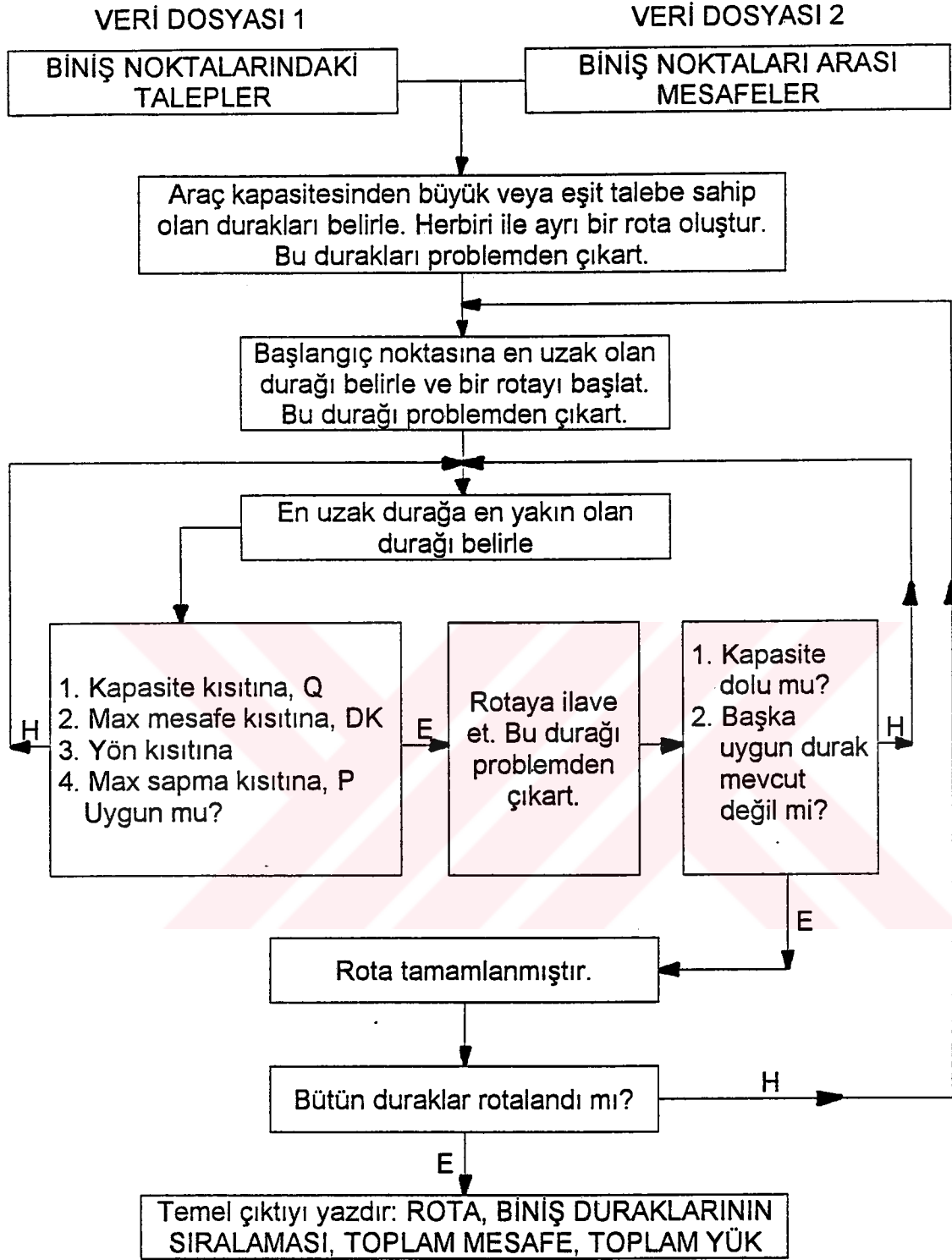
Bu rotalama algoritmasının temel girdisi, biri herbir duraktan araçlara binen elemanların sayısından (talepten) ve diğeri de herbir durak ile diğer duraklar ve iş yeri (durak 1) arasındaki mesafelerden oluşan iki veri dosyasından ibarettir. Ek 1 birinci veri dosyasının, Ek 2 ise ikinci veri dosyasının içeriğini sergilemektedir. Algoritma bütün duraklardaki taleplere hizmet verilinceye kadar ardarda icra edilen birçok rota-inşa prosedürünün bir toplamıdır. Herbir adımda, aracın kapasite kullanımına ilişkin kısıtın yanısıra daha önce açıklanan insan faktörüne ilişkin özel kısıtlara uygunluk da kesinlikle dikkate alınmaktadır.

Geliştirilen sezgisel metodun akış diyagramı Şekil 5.2'de sunulmaktadır.

Metodun bazı çarpıcı özellikleri aşağıdaki gibidir.

1. Sezgisel metodumuz içeriye doğru çalışmaktadır. Yani öncelikle son varış yerinden (iş yerinden) en uzak olan duraklar belirlenmekte ve rotalar bu duraklardan başlanarak geliştirilmektedir. Bunun nedeni, sözkonusu uzak durakların algoritmanın ileri aşamalarında rotalara yerleştirilmelerinin toplam mesafe kısıtının varlığından dolayı çok zor hatta çoğu zaman imkansız olabilmesidir.

2. Öncelikle talebin araç kapasitesine eşit veya daha fazla olduğu duraklar ele alınmaktadır. Böylece bu duraklar kendi başlarına birer rota oluşturarak problemden çıktıkları için problemi basitleştirmektedirler.
3. Metod, halihazırda bir araca tahsis edilmiş bulunan eleman sayısını sürekli kontrol ederek araç, kapasitesine eşit veya daha fazla miktarda yüklenir yüklenmez daha ileri gitmeyerek iş yerine en kısa yoldan geri dönüşü sağlamaktadır.
4. Duraklardaki taleplerin bölünmemeleri yani bir araç bir durağa yolcu almak üzere gittiği zaman bu duraktaki bütün elemanların alınması istenmektedir (Araçın bir durağı sadece geçiş için kullanması durumu bunun dışındadır). Çünkü taleplerin bölünmesine izin verilmesi durumunda, elemanların bazı araçları tercih edebilmeleri olasılığı, sözkonusu duraktan yolcu alan araçların kapasite kullanımlarını ve dolayısı ile bütün diğer rotaları etkileyebilecektir. Bu nedenle metod, bir aracın kapasitesi aşılmadıkça duraklardaki taleplerin araçlara bölünmemesini garanti etmektedir.



Şekil 5.2. Tek depo-çok araç rotalama problemi için akış diyagramı

### 5.5. Elde Edilen Sonuçların Yorumlanması

Yukarıda açıklanan ve Quick BASIC'de kodlanan sezgisel metodumuz bir IBM-PC compatible mikrobilgisayarda  $Q=18$ ,  $DK=80.000$  m ve  $P=1000, 1500, 2000, 2500$  ve  $3000$  değerleri ile beş farklı senaryo için koşturulmuş ve yaklaşık olarak 10 saatte mevcut durumdakinden daha iyi ve tatmin edici bir sonuç elde edilmiştir.

Programın temel çıktısı, geliştirilen herbir rotanın içerdiği duraklardan, durakların sıralanmasından, rotanın katetmiş olduğu toplam yolun uzunluğundan ve rota üzerinde taşınan eleman sayısından ibarettir. Sonuçlar  $P=1000, 1500, 2000, 2500$  ve  $3000$  için sırası ile Tablo 5.1, 5.2, 5.3, 5.4 ve 5.5'te sunulmaktadır. Tablo 5.6'da görülen mevcut durum ve herbir senaryoya ilişkin sonuçların karşılaştırılması ile aşağıdaki bilgiler elde edilmiştir.

- Sezgisel metodumuz farklı  $P$  değerleri karşısında büyük hassaslık göstermektedir. Bizim metodumuz ile  $P=1000, 1500, 2000$  ve  $2500$  için elde edilen sonuçlar rota sayısı (gerekli araç sayısı), toplam rota süresi ve atıl kapasite kriterlerinin her üçü açısından da mevcut durumdaki sonuçlardan önemli ölçüde kötü olmaktadır. Tablo 5.6'ya dikkat edildiğinde,  $P=1000$ 'den  $1500$ 'e geçildiğinde rota sayısının yaklaşık % 8,8, toplam rota uzunluğunun % 9,4 ve atıl kapasitenin de % 4,6 azaldığı görülmektedir.  $P=1500$ 'den  $2000$ 'e yükseltildiğinde rota sayısı % 26, toplam rota uzunluğu % 14 ve atıl kapasite % 18,3 azalmaktadır.  $P=2000$ 'den  $2500$ 'e geçişte ise sözkonusu azalmalar sırası ile % 0, % 0,2 ve % 0 olmaktadır. Açıkça görülebileceği gibi  $P$  değeri artırıldıkça iyileşmeler önce yavaş olurken, sonra önemli ölçüde hızlanmakta ve daha sonra da durmaktadır. Bu noktada yapılan inceleme şu andaki iyileşmenin mevcut duruma oldukça yaklaştığını fakat daha iyi bir sonuç elde etme açısından henüz tatmin edici olmadığını göstermektedir.

-  $P$  değerinin  $2500$ 'den  $3000$ 'e çıkartılması durumunda tekrar hızlı bir iyileşme gerçekleşmektedir. Rota sayısı 23'den 17'ye inerek % 26,

toplam rota uzunluđu 946147'den 742414'e inerek % 21,5 ve atıl kapasite de % 25,1 iyileşmektedir.

-  $P=3000$  için elde edilen ve her üç kriter açısından da en iyi olan değerler, mevcut durumla karşılaştırıldığında rota ve dolayısı ile gerekli araç sayısının 21'den 17'ye, toplam rota uzunluğunun 907279'dan 742414'e ve atıl kapasitenin 84'den 12'ye düştüğü görülmektedir. Bu durumda bizim önerdiğimiz sezgisel metod ile elde edilen sonuçların mevcut durumdaki sonuçlardan sırası ile % 19, % 18,17 ve % 18,3 daha iyi oldukları açıktır ve bu sonuçlar yeterince tatmin edicidir.

- Yukarıda  $P$  değeri arttıkça iyileşmenin önce yavaş olduğu sonra hızlandığı ve bir noktada durduktan sonra ani bir sıçrama gösterdiği ortaya konulmuştur. Ancak  $P$  değerinin etkilerinin böyle bir kalıba genelleştirilmesi tam anlamı ile yanlış olacaktır. Çünkü araç rotalama probleminde, duraklar arası mesafelerin dağılımı tamamen tesadüfidir. Yani hiçbir istatistiksel dağılıma uymamaktadır. Bu nedenle bir başka araç rotalama probleminde  $P$  için tamamen farklı bir davranış kalıbı ortaya çıkabilecektir.

- Ek 5'te mevcut durum için ve Ek 7'de bizim sezgisel metodumuz ile geliştirilen rotalar görsel olarak sunulmaktadır. Ancak sözkonusu çizimlerde karayolu ağlarının dikkate alınmamış olması şaşırtıcı olmamalıdır. Çünkü bu hatlar sadece duraklar arası bağlantıları ortaya koymakta ve izlenecek güzergah bu bağlantıların doğal bir sonucu olarak araç sürücöleri tarafından belirlenmektedir.

- Bankada yeni bir elemanın işe başlaması durumunda eđer bu eleman mevcut durakların herhangi birinden servis aracına binmek isterse sözkonusu durağın talebinin 1 birim artırılması ve problemin  $P=3000$  değerinden başlanarak tekrar çözölməsi yeterli olacaktır. Yeni elemanın problemde bulunanlar dışında farklı bir duraktan binmek istemesi durumunda ise yeni durağın 1 birim talebi ile birlikte probleme ilave edilmesi ve problemin gene  $P=3000$ 'den başlanarak çözölməsi gerekmektedir. Herhangi bir elemanın işten ayrılması veya servis aracına

binmekten vazgeçmesi durumunda ise sözkonusu elemana ilişkin durağın talebinin 1 birim eksiltiilerek problemin bu kez  $P=1000$ 'den (veya daha küçük bir değerden) başlanarak ve  $P$  çözüm elde edilinceye kadar arttırılarak çözülmesi uygun olacaktır. Bu durumda  $P=3000$  değerine kadar kabul edilebilir bir çözümün elde edilmesi kesindir.

Görüldüğü gibi mevcut oluşumda herhangi bir değişiklik olması durumunda problemin tekrar çözümünde izlenecek yöntem daha önceki problem çözümünde elde edilen sonuçlardan yararlanmaktadır. Bu, sezgisel araştırmanın çok belirgin ve olağan bir özelliğidir.

- Son olarak, önerilen sezgisel metod ile elde edilen sonuçlarda, toplam seyahat mesafesi, toplam rota sayısı ve dolayısı ile gerekli olan araç ve sürücü sayısı azalmış, koltuk kullanım yüzdesi ise artmıştır. Bunların doğal sonucu olarak araçlar için bakım onarım ve yakıt giderlerinin de azalacağı açıktır.

Tablo 5.1. Önerilen Metod ile P=1000 İçin Oluşturulan Rotalar  
(Q=18, D<sub>k</sub>=80 km.)

| Rota no. | Rota                  | Toplam Rota Uzunluğu (m) | Toplam Rota Yüğü (kişi) |
|----------|-----------------------|--------------------------|-------------------------|
| 1        | 1-65-13-16-34-27-19-1 | 77390                    | 18                      |
| 2        | 1-77-39-21-49-40-2-1  | 67240                    | 17                      |
| 3        | 1-88-37-12-66-7-1     | 67552                    | 18                      |
| 4        | 1-61-4-32-84-1        | 60830                    | 12                      |
| 5        | 1-64-57-29-55-1       | 59379                    | 16                      |
| 6        | 1-89-20-1             | 55579                    | 3                       |
| 7        | 1-80-28-1             | 54671                    | 6                       |
| 8        | 1-38-56-79-62-1       | 54212                    | 18                      |
| 9        | 1-9-33-59-54-83-1     | 51834                    | 15                      |
| 10       | 1-15-22-23-1          | 44649                    | 9                       |
| 11       | 1-81-25-47-31-1       | 40260                    | 15                      |
| 12       | 1-5-45-1              | 37008                    | 4                       |
| 13       | 1-35-52-50-1          | 38046                    | 11                      |
| 14       | 1-73-63-10-1          | 37999                    | 11                      |
| 15       | 1-76-1                | 37366                    | 7                       |
| 16       | 1-26-85-44-1          | 33668                    | 10                      |
| 17       | 1-71-69-1             | 29337                    | 5                       |
| 18       | 1-70-1                | 30800                    | 2                       |
| 19       | 1-30-82-17-75-6-60-1  | 33110                    | 16                      |
| 20       | 1-3-1                 | 30316                    | 3                       |
| 21       | 1-72-87-1             | 30877                    | 5                       |
| 22       | 1-14-1                | 27720                    | 5                       |
| 23       | 1-18-86-67-1          | 25872                    | 9                       |
| 24       | 1-58-1                | 24406                    | 1                       |
| 25       | 1-68-78-53-1          | 22176                    | 9                       |
| 26       | 1-11-1                | 20944                    | 4                       |
| 27       | 1-24-1                | 19712                    | 5                       |
| 28       | 1-48-8-74-1           | 19712                    | 13                      |
| 29       | 1-42-1                | 17094                    | 5                       |
| 30       | 1-41-1                | 15862                    | 4                       |
| 31       | 1-46-1                | 15300                    | 4                       |
| 32       | 1-51-1                | 13552                    | 2                       |
| 33       | 1-36-1                | 12936                    | 8                       |
| 34       | 1-43-1                | 10340                    | 4                       |

Tablo 5.2. Önerilen Metod ile P=1500 İçin Oluşturulan Rotalar  
(Q=18, D<sub>k</sub>=80 km.)

| Rota no. | Rota                  | Toplam Rota Uzunluğu (m) | Toplam Rota Yüğü (kişi) |
|----------|-----------------------|--------------------------|-------------------------|
| 1        | 1-65-13-16-34-27-19-1 | 77390                    | 18                      |
| 2        | 1-77-88-21-49-40-2-1  | 61771                    | 17                      |
| 3        | 1-39-4-32-6-84-1      | 62271                    | 15                      |
| 4        | 1-61-55-1             | 59049                    | 6                       |
| 5        | 1-64-57-12-7-1        | 56837                    | 17                      |
| 6        | 1-89-29-66-67-50-1    | 47433                    | 18                      |
| 7        | 1-80-28-20-1          | 53163                    | 7                       |
| 8        | 1-38-37-56-79-1       | 57284                    | 18                      |
| 9        | 1-9-33-59-30-54-1     | 48985                    | 17                      |
| 10       | 1-15-52-62-1          | 47335                    | 6                       |
| 11       | 1-81-25-22-23-1       | 39666                    | 16                      |
| 12       | 1-5-35-73-47-31-1     | 37940                    | 18                      |
| 13       | 1-76-1                | 37366                    | 7                       |
| 14       | 1-26-85-44-1          | 33668                    | 10                      |
| 15       | 1-71-69-1             | 29337                    | 5                       |
| 16       | 1-63-10-83-1          | 33992                    | 9                       |
| 17       | 1-70-1                | 30800                    | 2                       |
| 18       | 1-3-1                 | 30316                    | 3                       |
| 19       | 1-72-11-87-1          | 32494                    | 9                       |
| 20       | 1-14-1                | 27720                    | 5                       |
| 21       | 1-18-45-86-1          | 26334                    | 8                       |
| 22       | 1-58-17-75-60-1       | 26035                    | 8                       |
| 23       | 1-68-78-53-1          | 22176                    | 9                       |
| 24       | 1-24-82-1             | 19404                    | 6                       |
| 25       | 48-8-74-1             | 19712                    | 13                      |
| 26       | 1-42-1                | 17094                    | 5                       |
| 27       | 1-41-1                | 15862                    | 4                       |
| 28       | 1-46-1                | 15300                    | 4                       |
| 29       | 1-51-1                | 13552                    | 2                       |
| 30       | 1-36-1                | 12936                    | 8                       |
| 31       | 1-43-1                | 10340                    | 4                       |

Tablo 5.3. Önerilen Metod ile P=2000 İçin Oluşturulan Rotalar  
(Q=18, D<sub>k</sub>=80 km.)

| Rota no. | Rota                  | Toplam Rota Uzunluğu (m) | Toplam Rota Yüğü (kişi) |
|----------|-----------------------|--------------------------|-------------------------|
| 1        | 1-65-13-16-34-27-19-1 | 77930                    | 18                      |
| 2        | 1-77-88-21-49-40-2-1  | 61771                    | 17                      |
| 3        | 1-39-4-32-31-45-7-1   | 65274                    | 18                      |
| 4        | 1-61-57-12-60-1       | 59443                    | 17                      |
| 5        | 1-64-89-55-1          | 57894                    | 9                       |
| 6        | 1-80-29-66-74-1       | 48896                    | 17                      |
| 7        | 1-38-37-28-20-1       | 56157                    | 9                       |
| 8        | 1-9-33-59-30-54-1     | 48985                    | 17                      |
| 9        | 1-15-56-52-22-82-84-1 | 49368                    | 18                      |
| 10       | 1-81-25-79-23-1       | 39413                    | 18                      |
| 11       | 1-5-35-73-47-44-1     | 40918                    | 17                      |
| 12       | 1-76-1                | 37366                    | 7                       |
| 13       | 1-26-85-3-62-50-1     | 38038                    | 17                      |
| 14       | 1-71-69-1             | 29337                    | 5                       |
| 15       | 1-63-10-24-75-6-1     | 34957                    | 16                      |
| 16       | 1-70-83-1             | 24983                    | 5                       |
| 17       | 1-72-11-51-87-1       | 33184                    | 11                      |
| 18       | 1-14-36-43-1          | 31424                    | 17                      |
| 19       | 1-18-86-67-1          | 25872                    | 9                       |
| 20       | 1-58-17-1             | 25111                    | 3                       |
| 21       | 1-68-78-53-46-1       | 23974                    | 13                      |
| 22       | 1-48-8-42-1           | 21560                    | 12                      |
| 23       | 1-41-1                | 15862                    | 4                       |

Tablo 5.4. Önerilen Metod ile P=2500 İçin Oluşturulan Rotalar  
(Q=18, D<sub>k</sub>=80 km.)

| Rota no. | Rota                     | Toplam Rota Uzunluğu (m) | Toplam Rota Yüğü (kişi) |
|----------|--------------------------|--------------------------|-------------------------|
| 1        | 1-65-13-16-34-27-19-1    | 77930                    | 18                      |
| 2        | 1-77-88-21-49-40-2-1     | 61771                    | 17                      |
| 3        | 1-39-89-55-1             | 61183                    | 8                       |
| 4        | 1-61-64-57-12-1          | 60665                    | 18                      |
| 5        | 1-80-29-7-67-50-1        | 45980                    | 15                      |
| 6        | 1-38-37-28-20-31-45-86-1 | 62042                    | 17                      |
| 7        | 1-9-33-59-5-35-22-1      | 51931                    | 18                      |
| 8        | 1-4-32-6-84-60-1         | 46461                    | 15                      |
| 9        | 1-15-56-52-79-1          | 46480                    | 18                      |
| 10       | 1-81-25-47-26-44-1       | 43238                    | 18                      |
| 11       | 1-73-85-23-1             | 34362                    | 13                      |
| 12       | 1-76-1                   | 37366                    | 7                       |
| 13       | 1-71-69-1                | 29337                    | 5                       |
| 14       | 1-63-30-54-83-1          | 29680                    | 17                      |
| 15       | 1-70-1                   | 30800                    | 2                       |
| 16       | 1-3-24-82-17-75-62-1     | 35852                    | 15                      |
| 17       | 1-72-11-51-87-1          | 33184                    | 11                      |
| 18       | 1-14-36-43-1             | 31424                    | 17                      |
| 19       | 1-18-41-66-74-1          | 31801                    | 18                      |
| 20       | 1-10-1                   | 24720                    | 1                       |
| 21       | 1-58-1                   | 24406                    | 1                       |
| 22       | 1-68-78-53-46-1          | 23974                    | 13                      |
| 23       | 1-48-8-42-1              | 21560                    | 12                      |

Tablo 5.5. Önerilen Metod ile P=3000 İçin Oluşturulan Rotalar  
(Q=18, D<sub>k</sub>=80 km.)

| Rota no. | Rota                       | Toplam Rota Uzunluğu (m) | Toplam Rota Yüğü (kişi) |
|----------|----------------------------|--------------------------|-------------------------|
| 1        | 1-65-13-16-34-27-19-1      | 77390                    | 18                      |
| 2        | 1-77-88-21-49-40-2-1       | 61771                    | 17                      |
| 3        | 1-39-61-64-89-80-38-37-1   | 78756                    | 18                      |
| 4        | 1-57-12-76-1               | 50209                    | 18                      |
| 5        | 1-9-33-29-55-28-20-1       | 52527                    | 17                      |
| 6        | 1-4-59-5-35-73-1           | 52503                    | 16                      |
| 7        | 1-15-56-52-22-44-1         | 51023                    | 18                      |
| 8        | 1-81-25-79-23-1            | 39413                    | 18                      |
| 9        | 1-26-85-47-3-18-82-1       | 38209                    | 18                      |
| 10       | 1-71-14-69-36-1            | 38788                    | 18                      |
| 11       | 1-63-30-54-83-1            | 29680                    | 17                      |
| 12       | 1-70-10-58-24-75-6-84-60-1 | 40971                    | 18                      |
| 13       | 1-72-11-51-87-62-50-1      | 38002                    | 17                      |
| 14       | 1-68-78-53-46-43-1         | 26973                    | 17                      |
| 15       | 1-48-8-42-86-7-1           | 21637                    | 17                      |
| 16       | 1-17-31-32-45-67-1         | 25928                    | 17                      |
| 17       | 1-41-66-74-1               | 18634                    | 15                      |

Tablo 5.6. Önerilen Metod ile Sağlanan Sonuçlar ve Mevcut Durumunun Karşılaştırılması

| P            | Rota Sayısı | Toplam Rota Uzunluğu (m) | Atıl Kapasite (% , koltuk) |
|--------------|-------------|--------------------------|----------------------------|
| 1000         | 34          | 1217749                  | 51,96                      |
| 1500         | 31          | 1103562                  | 47,31                      |
| 2000         | 23          | 947717                   | 28,98                      |
| 2500         | 23          | 946147                   | 28,98                      |
| 3000         | 17**        | 742414**                 | 3,92*                      |
| Mevcut durum | 21          | 907279                   | 22,22                      |

\* işareti en iyi durumu göstermektedir.

## SONUÇ

Sezgisel araştırma, optimum olarak çözülebilmeleri gerek zaman ve gerek çaba açısından çok zor ve kimi zaman imkansız olan karmaşık gerçek yaşam problemleri için uygun bir süre ve uygun bir çaba ile optimuma yakın çözümler üretilmesine yönelik bir içsel araştırma sürecidir.

Çok karmaşık problemlere çözümler arayan sezgisel araştırma, adından da anlaşılabilirceği gibi temel olarak problem çözücünün sezgilerinden, geçmiş deneyimlerinden ve öngörülerinden yararlanmaktadır. Oluşturulan sezgisel çözüm metodları kendi içlerinde öğretici bir nitelik taşıyarak, her bir aşamada ümitsiz görünen yolları elimine etmekte ve dolayısı ile problem hacmini küçülterek çözüme doğru ilerlemektedirler.

Sezgisel çözüm metodları ile üretilen çözümlerin optimum olmaları şartı bulunmamakta, ancak kimi zaman optimum sonuçlara da ulaşabilmektedir. Ancak bu tarz yöntemler, optimallik bir dereceye kadar feda edilerek zaman ve emek tasarrufu sağlandığı için uzlaştırmacı yapıdadırlar. Bu yapı, gerçek yaşamda birbirleri ile çelişen bir çok amaç arasında bir uzlaşmayı tercih eden insan davranışına uygun düşmektedir.

Sezgisel araştırma, optimalliğe yönelik yöneylem araştırması metodlarına alternatif metodlar üretmektedir ve dolayısı ile uygulama alanı yöneylem araştırmasındaki gibi çok geniştir. Bir başka ifade ile çok karmaşık problemlerin sözkonusu olduğu bütün alanlar, sezgisel araştırmanın faaliyet kapsamı içinde yer almaktadır. Bu nedenle başlangıçta bu çalışmanın ana konusu sezgisel araştırmanın tanıtılması olarak düşünülmüş olmasına karşın, konunun genişliği çalışma kapsamının daraltılmasını zorunlu kılmış ve bunun sonucu olarak, çok

karmaşık problemler sınıfının tipik bir örneği olan araç rotalama probleminde sezgisel çözüm metodları üzerine odaklanılmıştır.

Araç Rotalama Problemi, en basit şekli ile bir merkezi depoda üslenmiş ve kapasiteleri birbirine eşit olan homojen araçlar filosu ile farklı yerleşimlere ve önceden bilinen taleplere sahip olan bir müşteriler setine hizmet verirken toplam seyahat mesafesini ve dolayısı ile seyahat süresini minimize etme problemidir. Bu problem bu en basit şekli ile bile çok karmaşık bir problemidir, yani optimum sonucun elde edilmesi çoğu zaman insani sınırları aşacak ölçüde uzun zaman ve fazla çaba gerektirmektedir. Bu nedenle çözülebilmesi için tek yol sezgisel araştırmayı kullanmaktır.

Literatürde araç rotalama problemi ile kanal rotalama problemi çoğu zaman birbirinden ayrı olarak ele alınmalarına rağmen, kanal rotalama probleminde de esas olarak ulaşım araçlarının rotalanması söz konusu olduğu için, bu çalışmada araç rotalama problemi temel olarak düğüm rotalama ve kanal rotalama problemi olarak iki alt başlık altında incelenmiştir.

Düğüm rotalama problemi, bir araçlar filosu ile bir düğümler setine hizmet sunabilmek amacı ile her bir araç için en düşük maliyetli bir rotanın oluşturulması problemidir. Burada "maliyet" terimi ile sözü edilen, rotanın mesafe veya zaman açısından uzunluğudur. Düğümler yerine belirli hatlar boyunca hizmet vermek söz konusu olduğunda kanal rotalama problemi ortaya çıkmaktadır.

Düğüm rotalama problemlerinin temel versiyonu olan seyyar satıcı probleminde (TSP) amaç, bir tek araç (veya satıcı) için ana şehirde başlayan ve bütün müşterileri kesinlikle ve yalnız bir kere ziyaret ettikten sonra ana şehire geri dönen en düşük maliyetli rotanın bulunmasıdır.

Araç rotalama problemlerinin bir diğer çeşidi, birden fazla merkezden kalkan araçların sözkonusu olması durumudur. Bu çalışmada sözkonusu problem çok depo/araç rotalama problemi olarak

adlandırılmakta ve klasik depo/araç rotalama probleminden daha karmaşık bir problem olarak ortaya çıkmaktadır.

Şimdiye kadar sözü edilen bütün problem türlerinde müşteri talepleri önceden belirli olmasına karşın müşteri taleplerinin önceden kesin olarak bilinmediği ancak bilinen bir ihtimal dağılımına sahip bir rastgele değişken olduğu problemler, stokastik araç rotalama problemleri olarak adlandırılmaktadır.

Araç rotalama probleminin çok önemli bir diğer türü de zaman pencereli araç rotalama problemidir. Burada her bir müşteriye hizmetin o müşteriye ilişkin bir zaman aralığı içinde yerine getirilmesi zorunluluğu bulunmaktadır. Bu durumda probleme rotalama boyutuna ilaveten bir de zaman açısından programlama boyutu dahil olmakta ve bu da problemin karmaşıklığını arttırmaktadır.

Çalışmanın dördüncü bölümünde ayrıca ele alınan kanal rotalama probleminde düğüm rotalama probleminin aksine talepler düğümlerde değil kanal olarak adlandırılan belirli hatlar üzerinde ortaya çıkmaktadır. Daha açık bir ifade ile kesin talep noktaları bilinmedikleri veya birbirlerine çok yakın oldukları için, söz konusu taleplerin bir hat boyunca yer aldıkları kanalların bir uçtan diğerine boydan boya katedilmeleri zorunlu olmaktadır.

Kanal rotalama problemi, problemde bulunan bütün kanallardan geçilmesi gereken Çinli postacı problemi ve yalnız bazı kanallardan geçilmesi gereken kırsal postacı problemi olarak iki alt bölüme ve bunlar da kendi aralarında kanal maliyetlerinin tamamen simetrik, tamamen asimetrik ve bazılarının simetrik diğerlerinin asimetrik olması durumuna göre alt bölümlere ayrılmaktadır. Kanal rotalama problemlerinin son bir versiyonu, her bir kanalın bir pozitif  $q_{ij}$  talebine sahip olduğu ve bu kanallara hizmetin herbiri  $Q$  kapasiteli olan kesinlikle bir tek araç ile verildiği durum olan kapasite kısıtlı kanal rotalama problemidir.

Yukarıda kısaca tanımlanan araç rotalama problemlerinin herbiri için şimdiye kadar geliştirilmiş olan sezgisel çözüm metodlarının en dikkate değer ve etkin olanları bu çalışma boyunca ayrıntılı olarak açıklandıktan sonra klasik araç rotalama probleminin tipik bir örneği olarak, bir iş yeri servis araçları rotalama problemi ele alınmış ve bizim tarafımızdan bir sezgisel çözüm metodu geliştirilerek bu konuya küçük bir katkı yapılmaya çalışılmıştır. Bu amaçla T.Garanti Bankası Genel Müdürlüğü'nde (Güneşli-İstanbul) çalışıp Avrupa yakasında oturmakta olan personele banka tarafından sağlanan ulaştırma hizmetleri incelendikten sonra, geliştirilen sezgisel çözüm metodu ile yeni ve mevcut durumdakinden daha iyi bir ulaşım planı oluşturulmuştur. Ancak sezgisel araştırmanın tanımı gereği bu çalışmada geliştirilen metod ve dolayısı ile ulaşım planı sadece bir öneri niteliğindedir.



## KAYNAKLAR

- Akl, S. An Analysis Of Various Aspects Of The Travelling Salesmen Problem, Ph. D. Thesis, Mc Gill University, 1978.
- Alfa, Attahiru Sule., Heragu, S. Sundresh., Chen, Mingyuan. A 3-opt Based Simulated Annealing Algorithm For Vehicle Routing Problems, Computers and Industrial Engineering, 21(1-4), 1991, S. 635-639.
- Altinkemer, Kemal., Gavish, Bezalel. Parallel Savings Based Heuristics For The Delivery Problem, Operations Research, 39(3), 1991, S. 456-469.
- Anily, S., Federgruen, A. A Class of Euclidean Routing Problems With General Route Cost Functions, Mathematics Of Operations Research, 15(2), 1990, S. 268-275
- Anily, S., Federgruen, A. One Warehouse Multiple Retailer Systems With Vehicle Routing Costs, Management Science, 36(1), 1990, S. 92-114.
- Balakrishnan, Anatharam., Magnanti, Thomas L., Mirchandani, Prakash. Heuristics, LPs And Trees On Trees: Network Design Analysis, Operations Research, 44(3), 1996, S. 478-496.
- Ball, M., Magazine, M. The Design And Analysis Of Heuristics, Networks, 11, 1981, S. 215-219.
- Begur, Sachinanda V. Capacitated Multi Period Vehicle Routing With Fixed Period Length, Ph. Thesis, University Of Alabama, 1994.
- Beltrami, E.L., Bodin, L. D. Networks And Vehicle Routing For Municipal Waste Collection, Networks, 4, 1974, S. 65-94.
- Bertsimas, Dimitris J., Jaillet, Patrick, Odoni Amadeo R. A Priori Optimization, Operations Research, 38(6), 1990, S. 1019-1033.

- Bienstock, Daniel., Bramel, Julien., Levi, David Simchi. A Probabilistic Analysis Of Tour Partitioning Problem With Unsplit Demands, *Mathematics Of Operations Research*, 18(4), 1993, S. 786-802.
- Bodin, L., Golden B. Classification In Vehicle Routing And Scheduling, *Networks*, 11(2), 1981, S. 97-108.
- Bodin, L., Golden, B., Assad, A., Ball, M. Routing and Scheduling Of Vehicles And Crews: The State Of Art, *Computers And Operations Research*, 10(2), 1983, S. 65-195.
- Bowerman, L., Robert., Calamai, H. Paul., Hall, G. Brent. The Spacefilling Curve With Optimal Partitioning Heuristic For The Vehicle Routing Problem, *European Journal Of Operational Research*, 76(1), 1994, S. 128-142.
- Bramel, Julien., Simchi-Levi, David. A Location Based Heuristic For General Routing Problems, *Operations Research*, 43(4), 1995, S. 649-660.
- Breedam, Alex Van. Improvement Heuristics For The Vehicle Routing Problem Based On Simulated Annealing, *European Journal Of Operational Research*, 86(3), 1995, S. 480-490.
- Chapleau, Luc., Ferland, Jacques., Lapalme, Guy., Rousseau, Jean-Marc. A parallel Insert Method For The Capaitated Arc Routing Problem, *Operations Research Letters*, 3(2), 1984, S. 95-99.
- Chapleau, Luc., Ferland, Jacques., Rousseau, Jean-Marc. Clustering For Routing In Densely Populated Areas, *European Journal Of Operational Research*, 20, 1985, S. 48-57.
- Christofides, N. The Optimum Traversal Of A Graph, *Omega*, 1, 1973, S. 719-732.
- Christofides, N. Worst-Case Analysis Of A New Heuristic For The Traveling Salesman Problem, Report 388, Graduate School Of Industrial Administration, Carnegie Mellon University, 1976.

- Christofides, N., Benavent, E., Campos, V., Corberan, A., Mota, E. An Optimal Method For The Mixed Postman Problem. In System Modelling And Optimization, Lecture Notes In Control And Information Sciences, 59, Berlin, 1984.
- Christofides, N., Campos, V., Corberan, A., Mota, E. An Algorithm For The Rural Postman Problem On Directed Graph, Mathematical Programming Study, 26, 1986, S. 155-166.
- Christofides, N., Mingozzi, A., Toth, P. The Vehicle Routing Problem, Combinatorial Optimization, Chap, 11, Wiley, Newyork, 1979.
- Clarke, G., Wright, J. Scheduling Of Vehicles From A Central Depot To A Number Of Delivery Points, Operations Research, 12, 1964, S. 568-581.
- Cloonan, J. A Heuristic Approach To Some Sales Territory Problems, Proc. Fourth Int. Conf. Ops. Res., M.I.T., Cambridge, Mass., 1966, S. 81-84.
- Croes, G. A Method For Solving Traveling Salesman Problems, Operations Research, 6, 1958, S. 791-812.
- Daellenbach, G. Hans., George, A. John. Introduction To Operations Research Techniques, Allyn An Bacon Inc., Boston, 1984, S. 511-545.
- Dell'Amico, Mauro., Fischetti, Matteo., Toth, Paolo. Heuristic Algorithms For The Multiple Depot Vehicle Scheduling Problem, Management Science, 39(1), 1993, S. 115-125.
- Desaulniers, Guy., Desrosiers, Jacques., Dumas, Yvan., Solomon, Marius., Soumis, François. Daily Aircraft Routing And Scheduling, Management Science, 43(6), 1997, S. 841-855.
- Desrochers, M., Verhoog, T. W. A New Heuristic For The Fleet Size And Mix Vehicle Routing Problem, Computers And Operations Research, 18(3), 1991, S. 263-274.

- Dror, M., Levy, L. A Vehicle Routing Improvement Algorithm Comparison Of A Greedy And Matching Implementation For Inventory Routing, *Computers And Operations Research*, 13, 1986, S. 33-45.
- Edmonds, J., Johnson, E. L. Matching, Euler Tours And The Chinese Postman Problem, *Mathematical Programming*, 5, 1973, S. 88-124.
- Eiselt, H. A., Gendreau, Michel., Laporte, Gilbert. Arc Routing Problems, Part I: The Chinese Postman Problem, *Operations Research*, 43(2), 1995, S. 231-241.
- Eiselt, H. A., Gendreau, Michel., Laporte, Gilbert. Arc Routing Problems, Part II: The Rural Postman Problem, *Operations Research*, 43(3), 1995, S. 399-413.
- Federgruen, Avi., Ryzin, Garrett. Probabilistic Analysis Of A Combined Aggregation And Math Programming Heuristic For A General Class Of Vehicle Routing And Scheduling Problems, *Management Science*, 43(8), 1997, S. 1060-1078.
- Fisher, M. Worst-Case Analysis Of Heuristic Algorithms, *Management Science*, 26(1), 1980, S. 1-17.
- Fisher, M., Jaikumar, R. A Generalized Assignment Heuristic For Vehicle Routing, *Networks*, 11(2), 1981, S. 109-124.
- Fleischner, H. Eulerian Graphs And Related Topics, *Annals Of Discrete Mathematics*, 50(2), North-Holland, Amsterdam, 1991.
- Foisy, Christian., Potvin, Jean-Yves. Implementing An Insertion Heuristic For Vehicle Routing On Parallel Hardware, *Computers And Operations Research*, 20, 1993, S. 737-745.
- Ford, L. R., Flukerson, D. R. *Flows In Networks*, Princeton University Press, Princeton, N. J., 1962
- Frederickson, G.N. Approximation Algorithms For Some Postman Problems, *J. Assoc. Comp. Mach*, 26, 1979, S. 538-554.

- Frederickson, G. N., Hecht, M. S., Kim, C.E. Approximation Algorithms For Some Postman Problems, *SIAM J. Computer*, 7, 1978, S. 178-193.
- Frieze, A., Galbiati, G., Maffioli, F. On The Worst-Case Performance Of Some Algorithms For The Asymmetric Traveling Salesman Problem, *Networks*, 12(1), 1982, S. 23-39.
- Garcia, B. Laurent., Potvin, Jean-Yves., Rousseau, Jean-Marc. A Parallel Implementation Of The Tabu Search Heuristic For Vehicle Routing Problems With Time Windows Constraints, *Computers And Operations Research*, 21(9), 1994, S. 1025-1033.
- Gaudio, M., Paletta, G. A Heuristic For The Periodic Vehicle Routing Problem, *Transportation Science*, 26(2), 1992, S. 86-92.
- Gendreau, Michel., Hertz, Alain., Laporte, Gilbert. New Insertion and Post-Optimization Procedures For The Traveling Salesman Problem, *Operations Research*, 40, 1992, S. 1086-1094.
- Gendreau, M., Hertz, A., Laporte, G. A Tabu Search Heuristic For The Vehicle Routing Problem, *Management Science*, 40(10), 1994, S. 1276-1290.
- Gensch, D. An Industrial Application Of The Traveling Salesman's Subtour Problem, *AIIE Trans.*, 10(4), 1978, S. 362-370.
- Gillett, Billy., Johnson, J. Multi-Terminal Vehicle-Dispatch Algorithm, *Omega*, 4, 1976, S. 711-718.
- Gillett, B., Miller, Leland. A Heuristic Algorithm For The Vehicle Dispatch Problem, *Operations Research*, 22, 1974, S. 340-349.
- Glover, F. Tabu Search, Part I, *ORSA J. Computing*, 1, 1989, S. 190-206.
- Glover, F., Laguna, M. Tabu Search, *Modern Heuristic Techniques For Combinatorial Problems*, C. Reeves (Ed.), Blackwell Scientific Publications, Oxford, 1993, S. 70-150.

- Golden, B. Evaluating A Sequential Vehicle Routing Algorithm, *AIIE Trans.*, 9, 1977, S. 204-208.
- Golden, B., Assad, A., Levy, L., Gheysens, F. The Fleet Size And Mix Vehicle Routing Problem, *Computers And Operations Research*, 11(1), 1982, S. 49-66.
- Golden, B., Dearmon, J. S., Baker, E. K. Computational Experiments With Algorithms For A Class Of Routing Problems, *Computers And Operations Research*, 10(1), 1983, S. 47-59.
- Golden, B., Levy, L., Dahl, R. Two Generalizations Of The Traveling Salesman Problem, *Omega*, 9(4), 1981, S. 439-441.
- Golden, B., Magnanti, T., Nguyen, H. Implementing Vehicle Routing Algorithms, *Networks*, 7, 1977, S. 113-148.
- Golden, B., Wong, R. Capacitated Arc Routing Problems, *Networks*, 11, 1981, S. 305-315.
- Graham, R., Lawler, E., Lenstra, J., Kan, A. Rinnooy. Optimization And Approximation In Deterministic Sequencing And Scheduling: A Survey, *Ann. Discrete Math.*, 5, 1979, S. 287-326.
- Hardgrave, W., Nemhauser, G. On The Relation Between The Traveling Salesman Problem And Longest Path Problem, *Operations Research*, 10, 1962, S. 647-657.
- Hertz, A. Finding A Feasible Course Schedule Using Tabu Search, *Discrete Applied Math.*, 35, 1992, S. 255-270.
- Jansen, Klaus. An Approximation Algorithm For The General Routing Problem, *Information Processing Letters*, 41, 1992, S. 333-339.
- Kanellakis, P., Papadimitriou, C. Local Search For The Asymmetric Traveling Salesman Problem, *Operations Research*, 28(5), 1980, S. 1086-1099.

- Karp, R. Probabilistic Analysis Of Partitioning Algorithms For The Traveling Salesman Problem In The Plane, *Mathematical Operations Research*, 2, 1977, S. 209-224.
- Karp, R. A Patching Algorithm For The Nonsymmetric Traveling Salesman Problem, *SIAM J. Computing*, 8, 1979, S. 561-573.
- Kaufmann, A., *Graphs Dynamic Programming And Finite Games*, Academic Press, Newyork, 1967.
- Kim, C. A Minimal Spanning Tree And Approximate Tours For A Traveling Salesman, *Comp. Sci. Tech. Report*, University Of Maryland, 1975.
- Klee, V. Combinatorial Optimization: What is the state of art?, *Mathematical Operations Research*, 5, 1980, S. 1-26.
- Labbe, Martin., Laporte, Gilbert., Mercure, Helen. Capacitated Vehicle Routing On Trees, *Operations Research*, 39(4), 1991, S. 616-622.
- Laporte, Gilbert. The Vehicle Routing Problem: An Overview Of Exact And Approximate Algorithms, *European Journal Of Operational Research*, 59, 1992, S. 345-358.
- Lawler, E. Shortest Path And Network Flow Algorithms, *Ann. Discrete Math*, 4, 1979, S. 251-263.
- Lenstra, J., Rinnooy Kan, A. On General Routing Problems, *Networks*, 6, 1976, S. 273-280.
- Lenstra, J., Rinnooy Kan, A. A Characterization Of Linear Admissible Transformations For The m-Traveling Salesman Problem, *European Journal Of Operational Research*, 3(3), 1979, S. 250-252.
- Li, Chung-Lun., Simchi-Levi, David., Desrochers, Martin. On The Distance Constrained Vehicle Routing Problem, *Operations Research*, 40(4), 1992, S. 790-799.

- Liaw, Ching-Fang. Heuristic Search And Its Transit Applications, Ph. D. Thesis, The University Of Michigan, 1994.
- Lin, S. Computer Solutions Of The Traveling Salesman Problem, Belle System Tech. J., 44, 1965, S. 2245-2269.
- Lin, S., Kernighan, B. An Effective Heuristic Algorithm For The Traveling Salesman Problem, Operations Research, 21, 1973, S. 498-516.
- Lysgaard, Jens. Dynamic Transportation Networks In Vehicle Routing And Scheduling, Interfaces, 22(3), 1992, S. 45-55.
- Moustakas, Clark. Heuristic Research, Sage Publications Inc., Newbury Park, California, 1990.
- Norback, J., Love, R. Geometric Approaches To Solving The Traveling Salesman Problem, Management Science, 23, 1977, S. 1208-1223.
- Ong, H. Design And Analysis Of Heuristics Fore Some Routing And Packing Problems, Ph. D. Thesis, University Of Woterloo, 1981.
- Or, I. Traveling Salesman-Type Combinatorial Problems And Their Relation To The Logistics of Blood Banking, Ph. D. Thesis, Dept Of Industrial Engineering And Management Sciences, Northwestern University, 1976.
- Orloff, C. S. A Fundamental Problem In Vehicle Routing, Networks, 4, 1974, S. 35-64.
- Papadimitriou, C. H. The Euclidean Traveling Salesman Problem is NP-Complete, Theory Of Computer, Science, 4(3), 1977, S. 237-244.
- Papadimitriou, C. H., Steiglitz, K. Combinatorial Optimization, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- Pearn, Wen Lea. Approximate Solutions For The Capacitated Arc Routing Problem, Computers And Operations Research, 16(6), 1989, S. 589-600.

- Pearn, Wen Lea. Augment-Insert Algorithms For The Capacitated Arc Routing Problem, *Computers And Operations Research*, 18(2), 1991, S. 189-198.
- Potvin, Jean-Yves., Lapalme, Guy. A Microcomputer Assistant For The Development Of Vehicle Routing And Scheduling Heuristics, *Decision Support Systems*, 12, 1994, S. 41-56.
- Potvin, Jean-Yves., Lapalme, Guy., Rousseau, Jean-Marc. ALTO: A Computer System For The Design Of Vehicle Routing Algorithms, *Computers And Operations Research*, 16(5), 1989, S. 451-470.
- Potvin, Jean-Yves., Rousseau, Jean Marc. A Parallel Route Building Algorithm For The Vehicle Routing And Scheduling Problem With Time Windows, *European Journal Of Operational Research*, 66, 1993, S. 331-340.
- Preparata, F., Hong, S. Convex Hulls Of Finite Sets Of Points In Two And Tree Dimensions, *Commun. ACM*, 20, 1977, S. 87-92.
- Psaraftis, Harilaos., Solomon, M., Magnanti, Thomas L., Kim, Tai-up. Routing And Scheduling On A Shoreline With Release Times, *Management Science*, 36(2), 1990, S. 212-223.
- Raghavendra, Anirudh., Krishnakumar, T. S., Muralidhar, R., Sarvanan, D. A Practical Heuristic For A Large Scale Vehicle Routing Problem, *European Journal Of Operational Research*, 5(1), 1992, S. 32-38.
- Riordan, J. An Introduction To Combinatorial Analysis, Wiley, New York, 1958.
- Robuste, Francesc., Daganzo, Carlos F., Souleyrette, Reginald R. Implementing Vehicle Routing models, *Transportation Research-B*, 24B(4), 1990, S. 263-286.
- Rosenkrantz, D., Sterns, R., Lewis, P. An Analysis Of Several Heuristics For The Traveling Salesman Problem, *SIAM J. Comp.*, 6, 1977, S. 563-581.

- Sahni, S. General Techniques For Combinatorial Approximation, *Operations Research*, 25(6), 1977, S. 920-936.
- Salhi, Said., Rand, Graham K. Improvements To Vehicle Routing Heuristics, *Journal Of The Operations Research Society*, 38(3), 1987, S. 293-295.
- Salhi, Said., Rand, Graham K. Incorporating Vehicle Routing Int The Vehicle Fleet Composition Problem, *European Journal Of Operational Research*, 66(3), 1993, S. 313-330.
- Scharage, L. Formulation And Structure Of More Complex/Realistic Routing And Scheduling Problems, *Networks*, 11, 1981, S. 229-232.
- Silver, E., Vidal, R., De Werra, D. A Tutorial On Heuristic Methods, *European J. Op. Res.*, 5, 1980, S. 153-162.
- Solomon, Marius. Algorithms For The Vehicle Routing And Scheduling Problems With Time Window Constraints, *Operations Research*, 32(2), 1987, S. 254-265.
- Solomon, Marius., Chalifour, Alain., Desrosiers, Jacques., Boisverst, Jacques. An Application Of Vehicle Routing Methodology. To Large - Scale Larvicide Control, *Interfaces*, 22(3), 1992, S. 88-99.
- Solomon, Marius., Desrosiers, Jacques. Time Window Constrained Routing And Scheduling Problems, *Transportation Science*, 22(1), 1988, S. 1-13.
- Stewart, William., Golden, B. A Lagrangean Relaxation, Heuristic For Vehicle Routing, *European Journal Of Operational Research*, 15, 1984, S. 84-88.
- Tabucanon, Mario., Kovavisaruch, La-ead., Malaivongs. Microcomputer-Based Heuristic Approach To Vehicle Routing For After-Sales Servicing, *Decision Support Systems*, 13, 1995, S. 195-205.

- Taillard, E. Parallel Iterative Search Methods For Vehicle Routing Problem, Working Paper ORWP 92/03, Department de Mathematiques, Ecole Polytechnique Federale de Lausanne, Switzerland, 1992.
- Thompson, Paul., Psaraftis, Harilaos. Cyclic Transfer Algorithms For Multivehicle Routing And Scheduling Problems, Operations Research, 41(5), 1993, S. 935-946.
- Tillman, F., Cain, T. An Upper Bounding Algorithm For The Single And Multiple Terminal Delivery Problem, Management Science, 18, 1972, S. 664-682.
- Ulusoy, G. The Fleet Size And Mix Problem For Capacitated Arc Routing, European J. Op. Res., 22, 1985, s. 329-337.
- Van der Cruyssen, P., Rijkaert, M. Heuristic For The Asymmetric Travelling Salesman Problem, J. Op. Res. Society, 30, 1978, S. 697-701.
- Win, Z. Contributions To Routing Problems, Doctoral Dissertation, Universitat Augsburg, Germany, 1987.
- Wiorkowski, J., Mc Elvain, K. A Rapid Heuristic Algorithm For The Approximate Solution Of The Traveling Salesman Problem, Transportation Research, 9, 1975, S. 181-185.

## Ek 1. Uygulama Problemine Ait Duraklar ve Talepler

| Durak no. | Durak ismi        | Talep (yolcu) |
|-----------|-------------------|---------------|
| 1         | Güneşli           | 0             |
| 2         | Akaretler         | 2             |
| 3         | Aksaray           | 3             |
| 4         | Ayazağa           | 1             |
| 5         | Bademlik          | 2             |
| 6         | Bağcılar-Meydan   | 3             |
| 7         | Bahçelievler-Okul | 2             |
| 8         | Bakırköy          | 5             |
| 9         | Başak konutları   | 1             |
| 10        | Bayrampaşa        | 1             |
| 11        | Bayramtepe yolu   | 4             |
| 12        | Beşiktaş          | 6             |
| 13        | Büyükdere         | 3             |
| 14        | Cami              | 5             |
| 15        | Çağlayan          | 2             |
| 16        | Çayırbaşı         | 3             |
| 17        | Çinçin deresi     | 2             |
| 18        | Davutpaşa         | 3             |
| 19        | Derbent           | 2             |
| 20        | Dolapdere         | 1             |
| 21        | Dördüncü Levent   | 5             |
| 22        | Edirnekapı        | 3             |
| 23        | Emniyet           | 4             |
| 24        | Esenler           | 5             |
| 25        | Eyüp              | 5             |
| 26        | Fatih             | 4             |
| 27        | Fatih Ormanı      | 2             |

|    |                   |   |
|----|-------------------|---|
| 28 | Feriköy           | 4 |
| 29 | Gayrettepe        | 4 |
| 30 | Gaziosmanpaşa     | 5 |
| 31 | Genç Osman        | 3 |
| 32 | Güngören-belediye | 7 |
| 33 | Gürsel            | 4 |
| 34 | Hacı Osman        | 2 |
| 35 | Halıcıoğlu        | 5 |
| 36 | Halkalı           | 8 |
| 37 | Hisar             | 2 |
| 38 | Hisarüstü         | 2 |
| 39 | Huzurevi          | 3 |
| 40 | Ihlamur           | 2 |
| 41 | İlkyuva           | 4 |
| 42 | İncirli           | 5 |
| 43 | İnönü Mahallesi   | 4 |
| 44 | İskenderpaşa      | 2 |
| 45 | İstanbul Evleri   | 2 |
| 46 | Kanarya           | 4 |
| 47 | Karagümrük        | 3 |
| 48 | Kartal Tepe       | 2 |
| 49 | Kışlaönü          | 1 |
| 50 | Koca Sinan        | 4 |
| 51 | Konutbirlik       | 2 |
| 52 | Kulaksız          | 2 |
| 53 | Küçük Çekmece     | 3 |
| 54 | Küçükköy          | 4 |
| 55 | Mecidiyeköy       | 3 |
| 56 | Okmeydanı         | 9 |
| 57 | Ortaköy           | 5 |
| 58 | Otogar            | 1 |
| 59 | Örnektepe         | 3 |
| 60 | Parseller         | 3 |

|    |                        |   |
|----|------------------------|---|
| 61 | Poligon                | 3 |
| 62 | Radarlar               | 2 |
| 63 | Rami                   | 5 |
| 64 | Reşitpaşa              | 4 |
| 65 | Sarıyer                | 6 |
| 66 | Siyavuşpaşa            | 5 |
| 67 | Soğanlı                | 3 |
| 68 | Soğuksu                | 3 |
| 69 | Söğütlü çeşme          | 1 |
| 70 | Su Deposu              | 2 |
| 71 | Sultan Çiftliği        | 4 |
| 72 | Şahin Tepesi           | 2 |
| 73 | Şehitlik               | 5 |
| 74 | Şirinevler             | 6 |
| 75 | Tabya                  | 2 |
| 76 | Taksim                 | 7 |
| 77 | Tarabya                | 4 |
| 78 | Tepe Üstü              | 3 |
| 79 | Ulubatlı Hasan         | 5 |
| 80 | Ulus                   | 2 |
| 81 | Üç Şehitler            | 4 |
| 82 | Üçyüzlü                | 1 |
| 83 | Vefa yolu              | 3 |
| 84 | Yavuz Selim (Bağcılar) | 1 |
| 85 | Yavuz Selim (Fatih)    | 4 |
| 86 | Yayla                  | 3 |
| 87 | Yenidoğan              | 3 |
| 88 | Yeniköy                | 3 |
| 89 | Yeni Levent            | 2 |

## Ek 2. Duraklar Arası En Kısa Mesafeler

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| D(1,1) = 0      | D(1,46) = 7650  | D(2,1) = 15642  | D(2,46) = 28927 |
| D(1,2) = 15642  | D(1,47) = 15587 | D(2,2) = 0      | D(2,47) = 8844  |
| D(1,3) = 15158  | D(1,48) = 9471  | D(2,3) = 8028   | D(2,48) = 20824 |
| D(1,4) = 22437  | D(1,49) = 18919 | D(2,4) = 13046  | D(2,49) = 3036  |
| D(1,5) = 18994  | D(1,50) = 1771  | D(2,5) = 9716   | D(2,50) = 17274 |
| D(1,6) = 6545   | D(1,51) = 6776  | D(2,6) = 13288  | D(2,51) = 23914 |
| D(1,7) = 6006   | D(1,52) = 16577 | D(2,7) = 11319  | D(2,52) = 5709  |
| D(1,8) = 8620   | D(1,53) = 7700  | D(2,8) = 18909  | D(2,53) = 29597 |
| D(1,9) = 24926  | D(1,54) = 9044  | D(2,9) = 7744   | D(2,54) = 13178 |
| D(1,10) = 12360 | D(1,55) = 17854 | D(2,10) = 11792 | D(2,55) = 3003  |
| D(1,11) = 10472 | D(1,56) = 18227 | D(2,11) = 25036 | D(2,56) = 6699  |
| D(1,12) = 18799 | D(1,57) = 26312 | D(2,12) = 726   | D(2,57) = 3036  |
| D(1,13) = 37191 | D(1,58) = 12203 | D(2,13) = 19580 | D(2,58) = 11792 |
| D(1,14) = 13860 | D(1,59) = 19000 | D(2,14) = 15103 | D(2,59) = 5819  |
| D(1,15) = 22385 | D(1,60) = 5390  | D(2,15) = 5819  | D(2,60) = 11946 |
| D(1,16) = 36498 | D(1,61) = 31724 | D(2,16) = 18964 | D(2,61) = 10251 |
| D(1,17) = 8316  | D(1,62) = 2950  | D(2,17) = 16181 | D(2,62) = 21417 |
| D(1,18) = 13090 | D(1,63) = 16093 | D(2,18) = 11022 | D(2,63) = 8283  |
| D(1,19) = 33038 | D(1,64) = 31339 | D(2,19) = 17482 | D(2,64) = 8041  |
| D(1,20) = 16962 | D(1,65) = 38577 | D(2,20) = 3465  | D(2,65) = 21351 |
| D(1,21) = 27159 | D(1,66) = 6768  | D(2,21) = 6611  | D(2,66) = 11994 |
| D(1,22) = 14762 | D(1,67) = 5005  | D(2,22) = 9603  | D(2,67) = 13904 |
| D(1,23) = 11935 | D(1,68) = 10549 | D(2,23) = 9438  | D(2,68) = 30367 |
| D(1,24) = 9856  | D(1,69) = 6930  | D(2,24) = 13871 | D(2,69) = 16720 |
| D(1,25) = 18403 | D(1,70) = 15400 | D(2,25) = 7634  | D(2,70) = 14255 |
| D(1,26) = 16577 | D(1,71) = 16170 | D(2,26) = 7524  | D(2,71) = 21317 |
| D(1,27) = 33957 | D(1,72) = 15092 | D(2,27) = 17821 | D(2,72) = 30041 |
| D(1,28) = 17854 | D(1,73) = 18786 | D(2,28) = 3490  | D(2,73) = 6204  |
| D(1,29) = 18354 | D(1,74) = 6622  | D(2,29) = 4940  | D(2,74) = 17337 |
| D(1,30) = 15169 | D(1,75) = 7546  | D(2,30) = 10637 | D(2,75) = 17798 |
| D(1,31) = 8008  | D(1,76) = 18683 | D(2,31) = 12552 | D(2,76) = 2178  |
| D(1,32) = 7900  | D(1,77) = 37730 | D(2,32) = 14718 | D(2,77) = 15280 |
| D(1,33) = 21538 | D(1,78) = 7854  | D(2,33) = 4433  | D(2,78) = 26807 |
| D(1,34) = 34628 | D(1,79) = 11984 | D(2,34) = 18976 | D(2,79) = 10923 |
| D(1,35) = 18990 | D(1,80) = 28237 | D(2,35) = 8514  | D(2,80) = 3296  |
| D(1,36) = 6468  | D(1,81) = 20405 | D(2,36) = 17490 | D(2,81) = 7128  |
| D(1,37) = 28093 | D(1,82) = 8393  | D(2,37) = 7271  | D(2,82) = 15257 |
| D(1,38) = 28195 | D(1,83) = 7504  | D(2,38) = 8425  | D(2,83) = 13104 |
| D(1,39) = 32549 | D(1,84) = 5929  | D(2,39) = 17022 | D(2,84) = 19800 |
| D(1,40) = 18084 | D(1,85) = 16016 | D(2,40) = 1485  | D(2,85) = 8354  |
| D(1,41) = 7931  | D(1,86) = 7392  | D(2,41) = 22759 | D(2,86) = 14487 |
| D(1,42) = 8547  | D(1,87) = 5544  | D(2,42) = 20349 | D(2,87) = 22769 |
| D(1,43) = 5170  | D(1,88) = 36036 | D(2,43) = 16181 | D(2,88) = 14831 |
| D(1,44) = 14682 | D(1,89) = 28314 | D(2,44) = 8613  | D(2,89) = 8295  |
| D(1,45) = 7850  |                 | D(2,45) = 24864 |                 |

## Ek 2'nin devamı

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| D(3,1) = 15158  | D(3,46) = 20988 | D(4,1) = 22437  | D(4,46) = 30184 |
| D(3,2) = 8028   | D(3,47) = 3036  | D(4,2) = 13046  | D(4,47) = 13130 |
| D(3,3) = 0      | D(3,48) = 12569 | D(4,3) = 15763  | D(4,48) = 20713 |
| D(3,4) = 15763  | D(3,49) = 11305 | D(4,4) = 0      | D(4,49) = 9769  |
| D(3,5) = 7828   | D(3,50) = 14520 | D(4,5) = 10549  | D(4,50) = 22460 |
| D(3,6) = 11106  | D(3,51) = 19910 | D(4,6) = 19789  | D(4,51) = 31955 |
| D(3,7) = 12646  | D(3,52) = 4983  | D(4,7) = 20405  | D(4,52) = 12452 |
| D(3,8) = 11722  | D(3,53) = 19140 | D(4,8) = 19866  | D(4,53) = 28336 |
| D(3,9) = 12683  | D(3,54) = 10747 | D(4,9) = 8855   | D(4,54) = 15092 |
| D(3,10) = 8052  | D(3,55) = 10131 | D(4,10) = 20482 | D(4,55) = 9548  |
| D(3,11) = 20372 | D(3,56) = 8415  | D(4,11) = 30261 | D(4,56) = 13145 |
| D(3,12) = 8754  | D(3,57) = 10932 | D(4,12) = 12320 | D(4,57) = 14498 |
| D(3,13) = 23452 | D(3,58) = 9669  | D(4,13) = 12166 | D(4,58) = 22326 |
| D(3,14) = 17061 | D(3,59) = 10831 | D(4,14) = 18711 | D(4,59) = 6414  |
| D(3,15) = 9603  | D(3,60) = 12749 | D(4,15) = 13475 | D(4,60) = 24328 |
| D(3,16) = 22759 | D(3,61) = 19602 | D(4,16) = 11473 | D(4,61) = 7469  |
| D(3,17) = 9130  | D(3,62) = 13816 | D(4,17) = 24491 | D(4,62) = 25157 |
| D(3,18) = 5044  | D(3,63) = 7227  | D(4,18) = 18476 | D(4,63) = 14630 |
| D(3,19) = 19063 | D(3,64) = 18293 | D(4,19) = 7777  | D(4,64) = 12936 |
| D(3,20) = 6369  | D(3,65) = 24838 | D(4,20) = 12947 | D(4,65) = 13552 |
| D(3,21) = 15213 | D(3,66) = 12184 | D(4,21) = 6082  | D(4,66) = 19943 |
| D(3,22) = 3993  | D(3,67) = 13724 | D(4,22) = 14087 | D(4,67) = 21714 |
| D(3,23) = 1287  | D(3,68) = 20449 | D(4,23) = 15169 | D(4,68) = 30723 |
| D(3,24) = 7513  | D(3,69) = 17061 | D(4,24) = 21868 | D(4,69) = 28871 |
| D(3,25) = 5434  | D(3,70) = 11363 | D(4,25) = 12474 | D(4,70) = 16940 |
| D(3,26) = 1881  | D(3,71) = 15521 | D(4,26) = 11975 | D(4,71) = 21175 |
| D(3,27) = 19910 | D(3,72) = 26378 | D(4,27) = 8624  | D(4,72) = 35343 |
| D(3,28) = 7590  | D(3,73) = 4763  | D(4,28) = 12089 | D(4,73) = 12474 |
| D(3,29) = 13200 | D(3,74) = 10824 | D(4,29) = 6125  | D(4,74) = 23327 |
| D(3,30) = 7979  | D(3,75) = 10131 | D(4,30) = 14399 | D(4,75) = 24405 |
| D(3,31) = 8360  | D(3,76) = 11352 | D(4,31) = 22176 | D(4,76) = 14102 |
| D(3,32) = 8206  | D(3,77) = 23760 | D(4,32) = 12936 | D(4,77) = 12320 |
| D(3,33) = 9064  | D(3,78) = 18370 | D(4,33) = 12936 | D(4,78) = 28644 |
| D(3,34) = 20680 | D(3,79) = 2838  | D(4,34) = 9394  | D(4,79) = 15532 |
| D(3,35) = 6673  | D(3,80) = 17523 | D(4,35) = 11704 | D(4,80) = 10087 |
| D(3,36) = 18062 | D(3,81) = 8844  | D(4,36) = 29872 | D(4,81) = 15862 |
| D(3,37) = 15906 | D(3,82) = 8668  | D(4,37) = 12936 | D(4,82) = 23024 |
| D(3,38) = 16267 | D(3,83) = 9977  | D(4,38) = 12012 | D(4,83) = 15169 |
| D(3,39) = 18601 | D(3,84) = 11517 | D(4,39) = 7315  | D(4,84) = 26022 |
| D(3,40) = 10470 | D(3,85) = 2376  | D(4,40) = 11820 | D(4,85) = 12470 |
| D(3,41) = 12723 | D(3,86) = 10978 | D(4,41) = 20867 | D(4,86) = 22942 |
| D(3,42) = 10875 | D(3,87) = 21065 | D(4,42) = 19019 | D(4,87) = 33187 |
| D(3,43) = 16676 | D(3,88) = 25223 | D(4,43) = 28486 | D(4,88) = 10780 |
| D(3,44) = 502   | D(3,89) = 16830 | D(4,44) = 15994 | D(4,89) = 7699  |
| D(3,45) = 11183 |                 | D(4,45) = 19327 |                 |

## Ek 2'nin devamı

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| D(5,1) = 18994  | D(5,46) = 20328 | D(6,1) = 6545   | D(6,46) = 15323 |
| D(5,2) = 9716   | D(5,47) = 4345  | D(6,2) = 13288  | D(6,47) = 9812  |
| D(5,3) = 7828   | D(5,48) = 12474 | D(6,3) = 11106  | D(6,48) = 7007  |
| D(5,4) = 10549  | D(5,49) = 12993 | D(6,4) = 19789  | D(6,49) = 16565 |
| D(5,5) = 0      | D(5,50) = 13941 | D(6,5) = 14091  | D(6,50) = 7315  |
| D(5,6) = 14091  | D(5,51) = 21329 | D(6,6) = 0      | D(6,51) = 9856  |
| D(5,7) = 13244  | D(5,52) = 3432  | D(6,7) = 5236   | D(6,52) = 14069 |
| D(5,8) = 12320  | D(5,53) = 18480 | D(6,8) = 6853   | D(6,53) = 13475 |
| D(5,9) = 7469   | D(5,54) = 10626 | D(6,9) = 16709  | D(6,54) = 12859 |
| D(5,10) = 9317  | D(5,55) = 7623  | D(6,10) = 7546  | D(6,55) = 16478 |
| D(5,11) = 22946 | D(5,56) = 4125  | D(6,11) = 11319 | D(6,56) = 13195 |
| D(5,12) = 10442 | D(5,57) = 12620 | D(6,12) = 14014 | D(6,57) = 16192 |
| D(5,13) = 20944 | D(5,58) = 13398 | D(6,13) = 29799 | D(6,58) = 6314  |
| D(5,14) = 16654 | D(5,59) = 2079  | D(6,14) = 13937 | D(6,59) = 14704 |
| D(5,15) = 6930  | D(5,60) = 13629 | D(6,15) = 16478 | D(6,60) = 1386  |
| D(5,16) = 20251 | D(5,61) = 14780 | D(6,16) = 29106 | D(6,61) = 27335 |
| D(5,17) = 13415 | D(5,62) = 18865 | D(6,17) = 4021  | D(6,62) = 7161  |
| D(5,18) = 8008  | D(5,63) = 5621  | D(6,18) = 5929  | D(6,63) = 11242 |
| D(5,19) = 16555 | D(5,64) = 13394 | D(6,19) = 25410 | D(6,64) = 25949 |
| D(5,20) = 5148  | D(5,65) = 22330 | D(6,20) = 15488 | D(6,65) = 31185 |
| D(5,21) = 10703 | D(5,66) = 13552 | D(6,21) = 20944 | D(6,66) = 4928  |
| D(5,22) = 3388  | D(5,67) = 15015 | D(6,22) = 8855  | D(6,67) = 7469  |
| D(5,23) = 6501  | D(5,68) = 20867 | D(6,23) = 9779  | D(6,68) = 15477 |
| D(5,24) = 10780 | D(5,69) = 19635 | D(6,24) = 3696  | D(6,69) = 11707 |
| D(5,25) = 2464  | D(5,70) = 11165 | D(6,25) = 9625  | D(6,70) = 14476 |
| D(5,26) = 5500  | D(5,71) = 16247 | D(6,26) = 10967 | D(6,71) = 12782 |
| D(5,27) = 17402 | D(5,72) = 28952 | D(6,27) = 26257 | D(6,72) = 17325 |
| D(5,28) = 4587  | D(5,73) = 3157  | D(6,28) = 16709 | D(6,73) = 8778  |
| D(5,29) = 12320 | D(5,74) = 12628 | D(6,29) = 17098 | D(6,74) = 6468  |
| D(5,30) = 8162  | D(5,75) = 13321 | D(6,30) = 9548  | D(6,75) = 1309  |
| D(5,31) = 10164 | D(5,76) = 7227  | D(6,31) = 4697  | D(6,76) = 16445 |
| D(5,32) = 12166 | D(5,77) = 21252 | D(6,32) = 2387  | D(6,77) = 30107 |
| D(5,33) = 4389  | D(5,78) = 18788 | D(6,33) = 13090 | D(6,78) = 13398 |
| D(5,34) = 18172 | D(5,79) = 4950  | D(6,34) = 27027 | D(6,79) = 8228  |
| D(5,35) = 1155  | D(5,80) = 13013 | D(6,35) = 15246 | D(6,80) = 23254 |
| D(5,36) = 20636 | D(5,81) = 4622  | D(6,36) = 9163  | D(6,81) = 12859 |
| D(5,37) = 14780 | D(5,82) = 11935 | D(6,37) = 27335 | D(6,82) = 2541  |
| D(5,38) = 13317 | D(5,83) = 10703 | D(6,38) = 25872 | D(6,83) = 11858 |
| D(5,39) = 16093 | D(5,84) = 14938 | D(6,39) = 24948 | D(6,84) = 1155  |
| D(5,40) = 12062 | D(5,85) = 5005  | D(6,40) = 15733 | D(6,85) = 10472 |
| D(5,41) = 12242 | D(5,86) = 11088 | D(6,41) = 3234  | D(6,86) = 4697  |
| D(5,42) = 10318 | D(5,87) = 22484 | D(6,42) = 5082  | D(6,87) = 11011 |
| D(5,43) = 19250 | D(5,88) = 20713 | D(6,43) = 11319 | D(6,88) = 30954 |
| D(5,44) = 7326  | D(5,89) = 12320 | D(6,44) = 10604 | D(6,89) = 22561 |
| D(5,45) = 10164 |                 | D(6,45) = 4774  |                 |

## Ek 2'nin devamı

|                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
| D(7,1) = 6006   | D(7,46) = 10780 | D(8,1) = 8620   | D(8,46) = 11935 |
| D(7,2) = 11319  | D(7,47) = 10428 | D(8,2) = 18909  | D(8,47) = 9889  |
| D(7,3) = 12646  | D(7,48) = 5467  | D(8,3) = 11722  | D(8,48) = 1694  |
| D(7,4) = 20405  | D(7,49) = 14596 | D(8,4) = 19866  | D(8,49) = 22186 |
| D(7,5) = 13244  | D(7,50) = 4386  | D(8,5) = 12320  | D(8,50) = 5929  |
| D(7,6) = 5236   | D(7,51) = 13860 | D(8,6) = 6853   | D(8,51) = 15323 |
| D(7,7) = 0      | D(7,52) = 15224 | D(8,7) = 3850   | D(8,52) = 14377 |
| D(7,8) = 3850   | D(7,53) = 8932  | D(8,8) = 0      | D(8,53) = 10087 |
| D(7,9) = 17325  | D(7,54) = 18403 | D(8,9) = 16786  | D(8,54) = 20482 |
| D(7,10) = 11319 | D(7,55) = 17094 | D(8,10) = 12936 | D(8,55) = 16555 |
| D(7,11) = 15554 | D(7,56) = 13811 | D(8,11) = 17710 | D(8,56) = 13272 |
| D(7,12) = 12045 | D(7,57) = 14223 | D(8,12) = 19635 | D(8,57) = 21813 |
| D(7,13) = 30415 | D(7,58) = 13013 | D(8,13) = 29876 | D(8,58) = 11781 |
| D(7,14) = 18634 | D(7,59) = 15092 | D(8,14) = 20867 | D(8,59) = 14014 |
| D(7,15) = 17094 | D(7,60) = 4774  | D(8,15) = 16555 | D(8,60) = 5852  |
| D(7,16) = 29722 | D(7,61) = 27566 | D(8,16) = 29183 | D(8,61) = 26950 |
| D(7,17) = 10720 | D(7,62) = 5698  | D(8,17) = 9488  | D(8,62) = 7546  |
| D(7,18) = 6391  | D(7,63) = 11858 | D(8,18) = 7700  | D(8,63) = 11319 |
| D(7,19) = 26026 | D(7,64) = 26180 | D(8,19) = 25487 | D(8,64) = 25564 |
| D(7,20) = 17028 | D(7,65) = 31801 | D(8,20) = 16104 | D(8,65) = 31262 |
| D(7,21) = 20328 | D(7,66) = 308   | D(8,21) = 20559 | D(8,66) = 4158  |
| D(7,22) = 9471  | D(7,67) = 1925  | D(8,22) = 8932  | D(8,67) = 5467  |
| D(7,23) = 11319 | D(7,68) = 10241 | D(8,23) = 10395 | D(8,68) = 11396 |
| D(7,24) = 10395 | D(7,69) = 10318 | D(8,24) = 9163  | D(8,69) = 12243 |
| D(7,25) = 10703 | D(7,70) = 18788 | D(8,25) = 11858 | D(8,70) = 20944 |
| D(7,26) = 11583 | D(7,71) = 17479 | D(8,26) = 11044 | D(8,71) = 19712 |
| D(7,27) = 26873 | D(7,72) = 21560 | D(8,27) = 26334 | D(8,72) = 23716 |
| D(7,28) = 18249 | D(7,73) = 9394  | D(8,28) = 17325 | D(8,73) = 8855  |
| D(7,29) = 8151  | D(7,74) = 1463  | D(8,29) = 23439 | D(8,74) = 1925  |
| D(7,30) = 12243 | D(7,75) = 6622  | D(8,30) = 13090 | D(8,75) = 8547  |
| D(7,31) = 5775  | D(7,76) = 17985 | D(8,31) = 6545  | D(8,76) = 17061 |
| D(7,32) = 5775  | D(7,77) = 30723 | D(8,32) = 7161  | D(8,77) = 30184 |
| D(7,33) = 13706 | D(7,78) = 8161  | D(8,33) = 13167 | D(8,78) = 9317  |
| D(7,34) = 27643 | D(7,79) = 9768  | D(8,34) = 27104 | D(8,79) = 8844  |
| D(7,35) = 12089 | D(7,80) = 22638 | D(8,35) = 11165 | D(8,80) = 22869 |
| D(7,36) = 12012 | D(7,81) = 13475 | D(8,36) = 13475 | D(8,81) = 12936 |
| D(7,37) = 27566 | D(7,82) = 9240  | D(8,37) = 26950 | D(8,82) = 8008  |
| D(7,38) = 26103 | D(7,83) = 17402 | D(8,38) = 25487 | D(8,83) = 19481 |
| D(7,39) = 25564 | D(7,84) = 6622  | D(8,39) = 25025 | D(8,84) = 8545  |
| D(7,40) = 13761 | D(7,85) = 11088 | D(8,40) = 21651 | D(8,85) = 10549 |
| D(7,41) = 2618  | D(7,86) = 1232  | D(8,41) = 3927  | D(8,86) = 3003  |
| D(7,42) = 2541  | D(7,87) = 15015 | D(8,42) = 1848  | D(8,87) = 16478 |
| D(7,43) = 9933  | D(7,88) = 30338 | D(8,43) = 11858 | D(8,88) = 30569 |
| D(7,44) = 12144 | D(7,89) = 21945 | D(8,44) = 11220 | D(8,89) = 22176 |
| D(7,45) = 2849  |                 | D(8,45) = 4004  |                 |

## Ek 2'nin devamı

|                 |                 |                  |                  |
|-----------------|-----------------|------------------|------------------|
| D(9,1) = 24926  | D(9,46) = 27104 | D(10,1) = 12360  | D(10,46) = 21927 |
| D(9,2) = 7744   | D(9,47) = 11121 | D(10,2) = 11792  | D(10,47) = 7117  |
| D(9,3) = 12683  | D(9,48) = 17633 | D(10,3) = 8052   | D(10,48) = 13783 |
| D(9,4) = 8855   | D(9,49) = 11021 | D(10,4) = 20482  | D(10,49) = 15069 |
| D(9,5) = 7469   | D(9,50) = 20717 | D(10,5) = 9317   | D(10,50) = 14553 |
| D(9,6) = 16709  | D(9,51) = 28875 | D(10,6) = 7546   | D(10,51) = 16247 |
| D(9,7) = 17325  | D(9,52) = 9372  | D(10,7) = 11319  | D(10,52) = 9757  |
| D(9,8) = 16786  | D(9,53) = 25256 | D(10,8) = 12936  | D(10,53) = 20002 |
| D(9,9) = 0      | D(9,54) = 13013 | D(10,9) = 14168  | D(10,54) = 7007  |
| D(9,10) = 14168 | D(9,55) = 7392  | D(10,10) = 0     | D(10,55) = 13860 |
| D(9,11) = 27181 | D(9,56) = 6369  | D(10,11) = 18325 | D(10,56) = 11341 |
| D(9,12) = 8470  | D(9,57) = 10648 | D(10,12) = 12518 | D(10,57) = 18700 |
| D(9,13) = 18711 | D(9,58) = 18481 | D(10,13) = 29491 | D(10,58) = 3137  |
| D(9,14) = 21275 | D(9,59) = 5852  | D(10,14) = 10703 | D(10,59) = 10703 |
| D(9,15) = 5775  | D(9,60) = 20405 | D(10,15) = 14861 | D(10,60) = 8316  |
| D(9,16) = 18018 | D(9,61) = 16478 | D(10,16) = 28798 | D(10,61) = 24948 |
| D(9,17) = 16551 | D(9,62) = 24486 | D(10,17) = 8393  | D(10,62) = 12782 |
| D(9,18) = 14553 | D(9,63) = 11550 | D(10,18) = 6006  | D(10,63) = 4389  |
| D(9,19) = 14322 | D(9,64) = 12397 | D(10,19) = 25641 | D(10,64) = 21078 |
| D(9,20) = 8250  | D(9,65) = 20097 | D(10,20) = 12291 | D(10,65) = 30877 |
| D(9,21) = 8239  | D(9,66) = 16863 | D(10,21) = 18788 | D(10,66) = 11242 |
| D(9,22) = 10164 | D(9,67) = 18634 | D(10,22) = 6160  | D(10,67) = 13475 |
| D(9,23) = 12089 | D(9,68) = 27643 | D(10,23) = 6765  | D(10,68) = 21791 |
| D(9,24) = 16863 | D(9,69) = 24948 | D(10,24) = 4081  | D(10,69) = 18095 |
| D(9,25) = 10087 | D(9,70) = 12474 | D(10,25) = 7392  | D(10,70) = 6034  |
| D(9,26) = 12276 | D(9,71) = 20120 | D(10,26) = 8272  | D(10,71) = 9548  |
| D(9,27) = 15169 | D(9,72) = 32263 | D(10,27) = 27720 | D(10,72) = 23408 |
| D(9,28) = 7293  | D(9,73) = 9394  | D(10,28) = 13552 | D(10,73) = 5621  |
| D(9,29) = 4576  | D(9,74) = 19404 | D(10,29) = 8624  | D(10,74) = 13783 |
| D(9,30) = 10857 | D(9,75) = 19400 | D(10,30) = 5005  | D(10,75) = 8239  |
| D(9,31) = 22253 | D(9,76) = 11516 | D(10,31) = 8468  | D(10,76) = 12694 |
| D(9,32) = 23100 | D(9,77) = 19019 | D(10,32) = 8239  | D(10,77) = 31493 |
| D(9,33) = 3773  | D(9,78) = 25564 | D(10,33) = 12705 | D(10,78) = 20328 |
| D(9,34) = 15939 | D(9,79) = 12452 | D(10,34) = 28490 | D(10,79) = 5214  |
| D(9,35) = 8624  | D(9,80) = 10318 | D(10,35) = 8162  | D(10,80) = 19943 |
| D(9,36) = 25949 | D(9,81) = 12782 | D(10,36) = 16401 | D(10,81) = 8239  |
| D(9,37) = 12089 | D(9,82) = 18018 | D(10,37) = 17787 | D(10,82) = 6853  |
| D(9,38) = 11165 | D(9,83) = 13090 | D(10,38) = 18095 | D(10,83) = 6006  |
| D(9,39) = 13860 | D(9,84) = 21017 | D(10,39) = 25179 | D(10,84) = 9858  |
| D(9,40) = 10176 | D(9,85) = 11781 | D(10,40) = 14234 | D(10,85) = 7777  |
| D(9,41) = 17787 | D(9,86) = 17941 | D(10,41) = 10087 | D(10,86) = 11165 |
| D(9,42) = 15939 | D(9,87) = 30107 | D(10,42) = 10472 | D(10,87) = 17479 |
| D(9,43) = 24563 | D(9,88) = 18249 | D(10,43) = 18480 | D(10,88) = 29799 |
| D(9,44) = 12914 | D(9,89) = 9856  | D(10,44) = 7590  | D(10,89) = 20174 |
| D(9,45) = 16247 |                 | D(10,45) = 9702  |                  |

Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(11,1) = 10472  | D(11,46) = 14784 | D(12,1) = 18799  | D(12,46) = 29653 |
| D(11,2) = 25036  | D(11,47) = 21285 | D(12,2) = 726    | D(12,47) = 9570  |
| D(11,3) = 20372  | D(11,48) = 18557 | D(12,3) = 8754   | D(12,48) = 21550 |
| D(11,4) = 30261  | D(11,49) = 28313 | D(12,4) = 12320  | D(12,49) = 2541  |
| D(11,5) = 22946  | D(11,50) = 13398 | D(12,5) = 10442  | D(12,50) = 18000 |
| D(11,6) = 11319  | D(11,51) = 5387  | D(12,6) = 14014  | D(12,51) = 24640 |
| D(11,7) = 15554  | D(11,52) = 23155 | D(12,7) = 12045  | D(12,52) = 6435  |
| D(11,8) = 17710  | D(11,53) = 12936 | D(12,8) = 19635  | D(12,53) = 30323 |
| D(11,9) = 27181  | D(11,54) = 24178 | D(12,9) = 8470   | D(12,54) = 13904 |
| D(11,10) = 18325 | D(11,55) = 27951 | D(12,10) = 12518 | D(12,55) = 3729  |
| D(11,11) = 0     | D(11,56) = 24668 | D(12,11) = 25762 | D(12,56) = 7425  |
| D(11,12) = 25762 | D(11,57) = 27940 | D(12,12) = 0     | D(12,57) = 2310  |
| D(11,13) = 41272 | D(11,58) = 17479 | D(12,13) = 18854 | D(12,58) = 12518 |
| D(11,14) = 22719 | D(11,59) = 25949 | D(12,14) = 15829 | D(12,59) = 6545  |
| D(11,15) = 27951 | D(11,60) = 13167 | D(12,15) = 6545  | D(12,60) = 12672 |
| D(11,16) = 40579 | D(11,61) = 35416 | D(12,16) = 18238 | D(12,61) = 9525  |
| D(11,17) = 15184 | D(11,62) = 12243 | D(12,17) = 16907 | D(12,62) = 22143 |
| D(11,18) = 17633 | D(11,63) = 18480 | D(12,18) = 11748 | D(12,63) = 9009  |
| D(11,19) = 36883 | D(11,64) = 35801 | D(12,19) = 16756 | D(12,64) = 7315  |
| D(11,20) = 26954 | D(11,65) = 42658 | D(12,20) = 4191  | D(12,65) = 20625 |
| D(11,21) = 29185 | D(11,66) = 16016 | D(12,21) = 5885  | D(12,66) = 12420 |
| D(11,22) = 18711 | D(11,67) = 14322 | D(12,22) = 10329 | D(12,67) = 14630 |
| D(11,23) = 19085 | D(11,68) = 15015 | D(12,23) = 10164 | D(12,68) = 31093 |
| D(11,24) = 14861 | D(11,69) = 12397 | D(12,24) = 14597 | D(12,69) = 17446 |
| D(11,25) = 19789 | D(11,70) = 23254 | D(12,25) = 8360  | D(12,70) = 14981 |
| D(11,26) = 22440 | D(11,71) = 21564 | D(12,26) = 8250  | D(12,71) = 22143 |
| D(11,27) = 37730 | D(11,72) = 6006  | D(12,27) = 17095 | D(12,72) = 30767 |
| D(11,28) = 28175 | D(11,73) = 16324 | D(12,28) = 4290  | D(12,73) = 6930  |
| D(11,29) = 21868 | D(11,74) = 16709 | D(12,29) = 4323  | D(12,74) = 18063 |
| D(11,30) = 19481 | D(11,75) = 12782 | D(12,30) = 11363 | D(12,75) = 18524 |
| D(11,31) = 14245 | D(11,76) = 27911 | D(12,31) = 13278 | D(12,76) = 2904  |
| D(11,32) = 13398 | D(11,77) = 41580 | D(12,32) = 15444 | D(12,77) = 14554 |
| D(11,33) = 24563 | D(11,78) = 12936 | D(12,33) = 5159  | D(12,78) = 27533 |
| D(11,34) = 38500 | D(11,79) = 17534 | D(12,34) = 18250 | D(12,79) = 11649 |
| D(11,35) = 21791 | D(11,80) = 31495 | D(12,35) = 9240  | D(12,80) = 2570  |
| D(11,36) = 9240  | D(11,81) = 19712 | D(12,36) = 18216 | D(12,81) = 7854  |
| D(11,37) = 35955 | D(11,82) = 13706 | D(12,37) = 6545  | D(12,82) = 15983 |
| D(11,38) = 34800 | D(11,83) = 23177 | D(12,38) = 7700  | D(12,83) = 13830 |
| D(11,39) = 36421 | D(11,84) = 10857 | D(12,39) = 16296 | D(12,84) = 20526 |
| D(11,40) = 27478 | D(11,85) = 21945 | D(12,40) = 1782  | D(12,85) = 9080  |
| D(11,41) = 13706 | D(11,86) = 18788 | D(12,41) = 23485 | D(12,86) = 15213 |
| D(11,42) = 16863 | D(11,87) = 5852  | D(12,42) = 21075 | D(12,87) = 23495 |
| D(11,43) = 12251 | D(11,88) = 39195 | D(12,43) = 16907 | D(12,88) = 14105 |
| D(11,44) = 19910 | D(11,89) = 30802 | D(12,44) = 9339  | D(12,89) = 7569  |
| D(11,45) = 18711 |                  | D(12,45) = 25410 |                  |

Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(13,1) = 37191  | D(13,46) = 43043 | D(14,1) = 13860  | D(14,46) = 13706 |
| D(13,2) = 19580  | D(13,47) = 22216 | D(14,2) = 15103  | D(14,47) = 13735 |
| D(13,3) = 23452  | D(13,48) = 30723 | D(14,3) = 17061  | D(14,48) = 22715 |
| D(13,4) = 12166  | D(13,49) = 16303 | D(14,4) = 18711  | D(14,49) = 18380 |
| D(13,5) = 20944  | D(13,50) = 33726 | D(14,5) = 16654  | D(14,50) = 21329 |
| D(13,6) = 29799  | D(13,51) = 41349 | D(14,6) = 13937  | D(14,51) = 10472 |
| D(13,7) = 30415  | D(13,52) = 20416 | D(14,7) = 18634  | D(14,52) = 16709 |
| D(13,8) = 29876  | D(13,53) = 41118 | D(14,8) = 20867  | D(14,53) = 11858 |
| D(13,9) = 18711  | D(13,54) = 25872 | D(14,9) = 21275  | D(14,54) = 10164 |
| D(13,10) = 29491 | D(13,55) = 13321 | D(14,10) = 10703 | D(14,55) = 25090 |
| D(13,11) = 41272 | D(13,56) = 16918 | D(14,11) = 22719 | D(14,56) = 21460 |
| D(13,12) = 18854 | D(13,57) = 16676 | D(14,12) = 15829 | D(14,57) = 25322 |
| D(13,13) = 0     | D(13,58) = 31412 | D(14,13) = 32822 | D(14,58) = 11319 |
| D(13,14) = 32822 | D(13,59) = 17787 | D(14,14) = 0     | D(14,59) = 15545 |
| D(13,15) = 17248 | D(13,60) = 33414 | D(14,15) = 30876 | D(14,60) = 15477 |
| D(13,16) = 693   | D(13,61) = 8624  | D(14,16) = 31878 | D(14,61) = 32516 |
| D(13,17) = 33582 | D(13,62) = 35420 | D(14,17) = 12012 | D(14,62) = 20251 |
| D(13,18) = 27562 | D(13,63) = 23793 | D(14,18) = 16555 | D(14,63) = 22715 |
| D(13,19) = 4389  | D(13,64) = 10934 | D(14,19) = 28413 | D(14,64) = 27489 |
| D(13,20) = 16720 | D(13,65) = 1386  | D(14,20) = 25387 | D(14,65) = 34342 |
| D(13,21) = 12089 | D(13,66) = 29953 | D(14,21) = 22484 | D(14,66) = 18095 |
| D(13,22) = 23173 | D(13,67) = 31724 | D(14,22) = 12782 | D(14,67) = 20482 |
| D(13,23) = 22858 | D(13,68) = 42889 | D(14,23) = 15708 | D(14,68) = 14168 |
| D(13,24) = 30954 | D(13,69) = 37957 | D(14,24) = 11242 | D(14,69) = 7392  |
| D(13,25) = 18942 | D(13,70) = 22561 | D(14,25) = 14245 | D(14,70) = 11165 |
| D(13,26) = 21061 | D(13,71) = 35089 | D(14,26) = 14890 | D(14,71) = 5300  |
| D(13,27) = 3542  | D(13,72) = 46354 | D(14,27) = 29337 | D(14,72) = 32267 |
| D(13,28) = 15862 | D(13,73) = 21791 | D(14,28) = 26707 | D(14,73) = 24255 |
| D(13,29) = 12650 | D(13,74) = 32413 | D(14,29) = 11935 | D(14,74) = 21560 |
| D(13,30) = 22099 | D(13,75) = 33491 | D(14,30) = 10167 | D(14,75) = 12782 |
| D(13,31) = 30415 | D(13,76) = 17875 | D(14,31) = 14091 | D(14,76) = 26773 |
| D(13,32) = 32725 | D(13,77) = 3542  | D(14,32) = 15169 | D(14,77) = 32879 |
| D(13,33) = 16709 | D(13,78) = 41426 | D(14,33) = 28720 | D(14,78) = 12089 |
| D(13,34) = 2772  | D(13,79) = 23221 | D(14,34) = 29953 | D(14,79) = 14223 |
| D(13,35) = 19789 | D(13,80) = 14399 | D(14,35) = 15554 | D(14,80) = 26257 |
| D(13,36) = 38958 | D(13,81) = 23100 | D(14,36) = 9317  | D(14,81) = 26950 |
| D(13,37) = 12243 | D(13,82) = 32109 | D(14,37) = 29260 | D(14,82) = 11627 |
| D(13,38) = 13706 | D(13,83) = 25949 | D(14,38) = 28105 | D(14,83) = 10169 |
| D(13,39) = 4851  | D(13,84) = 35108 | D(14,39) = 27874 | D(14,84) = 13629 |
| D(13,40) = 17978 | D(13,85) = 21556 | D(14,40) = 17545 | D(14,85) = 14395 |
| D(13,41) = 30877 | D(13,86) = 32028 | D(14,41) = 14245 | D(14,86) = 18480 |
| D(13,42) = 29029 | D(13,87) = 42581 | D(14,42) = 19096 | D(14,87) = 12166 |
| D(13,43) = 37572 | D(13,88) = 6545  | D(14,43) = 12089 | D(14,88) = 36520 |
| D(13,44) = 23683 | D(13,89) = 12549 | D(14,44) = 16599 | D(14,89) = 23639 |
| D(13,45) = 29337 |                  | D(14,45) = 16863 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(15,1) = 22385  | D(15,46) = 29722 | D(16,1) = 36498  | D(16,46) = 42350 |
| D(15,2) = 5819   | D(15,47) = 8580  | D(16,2) = 18964  | D(16,47) = 21523 |
| D(15,3) = 9603   | D(15,48) = 17402 | D(16,3) = 22759  | D(16,48) = 30030 |
| D(15,4) = 13475  | D(15,49) = 8360  | D(16,4) = 11473  | D(16,49) = 15561 |
| D(15,5) = 6930   | D(15,50) = 20559 | D(16,5) = 20251  | D(16,50) = 33033 |
| D(15,6) = 16478  | D(15,51) = 28028 | D(16,6) = 29106  | D(16,51) = 40656 |
| D(15,7) = 17094  | D(15,52) = 7194  | D(16,7) = 29722  | D(16,52) = 19723 |
| D(15,8) = 16555  | D(15,53) = 27797 | D(16,8) = 29183  | D(16,53) = 40425 |
| D(15,9) = 5775   | D(15,54) = 11704 | D(16,9) = 18018  | D(16,54) = 25179 |
| D(15,10) = 14861 | D(15,55) = 3927  | D(16,10) = 28798 | D(16,55) = 12628 |
| D(15,11) = 27951 | D(15,56) = 3795  | D(16,11) = 40579 | D(16,56) = 16225 |
| D(15,12) = 6545  | D(15,57) = 8723  | D(16,12) = 18238 | D(16,57) = 16060 |
| D(15,13) = 17248 | D(15,58) = 17706 | D(16,13) = 693   | D(16,58) = 30719 |
| D(15,14) = 30876 | D(15,59) = 5544  | D(16,14) = 31878 | D(16,59) = 17094 |
| D(15,15) = 0     | D(15,60) = 17864 | D(16,15) = 16555 | D(16,60) = 32721 |
| D(15,16) = 16555 | D(15,61) = 13398 | D(16,16) = 0     | D(16,61) = 7931  |
| D(15,17) = 16924 | D(15,62) = 21945 | D(16,17) = 32875 | D(16,62) = 34727 |
| D(15,18) = 12012 | D(15,63) = 10703 | D(16,18) = 26869 | D(16,63) = 23100 |
| D(15,19) = 13706 | D(15,64) = 12089 | D(16,19) = 3996  | D(16,64) = 10241 |
| D(15,20) = 5544  | D(15,65) = 18634 | D(16,20) = 16027 | D(16,65) = 2079  |
| D(15,21) = 6622  | D(15,66) = 16632 | D(16,21) = 11473 | D(16,66) = 29260 |
| D(15,22) = 7623  | D(15,67) = 18403 | D(16,22) = 22480 | D(16,67) = 31031 |
| D(15,23) = 9009  | D(15,68) = 29568 | D(16,23) = 22165 | D(16,68) = 42196 |
| D(15,24) = 17248 | D(15,69) = 20944 | D(16,24) = 30261 | D(16,69) = 37264 |
| D(15,25) = 6700  | D(15,70) = 13167 | D(16,25) = 18249 | D(16,70) = 22176 |
| D(15,26) = 9735  | D(15,71) = 29721 | D(16,26) = 20368 | D(16,71) = 34342 |
| D(15,27) = 14553 | D(15,72) = 33033 | D(16,27) = 2849  | D(16,72) = 45691 |
| D(15,28) = 4323  | D(15,73) = 8547  | D(16,28) = 15169 | D(16,73) = 21098 |
| D(15,29) = 3500  | D(15,74) = 16940 | D(16,29) = 12340 | D(16,74) = 31720 |
| D(15,30) = 10395 | D(15,75) = 19785 | D(16,30) = 21406 | D(16,75) = 32798 |
| D(15,31) = 16709 | D(15,76) = 8228  | D(16,31) = 29722 | D(16,76) = 17182 |
| D(15,32) = 9548  | D(15,77) = 18326 | D(16,32) = 32032 | D(16,77) = 2849  |
| D(15,33) = 2926  | D(15,78) = 28105 | D(16,33) = 16016 | D(16,78) = 40733 |
| D(15,34) = 15323 | D(15,79) = 9372  | D(16,34) = 2079  | D(16,79) = 22528 |
| D(15,35) = 5775  | D(15,80) = 9702  | D(16,35) = 19096 | D(16,80) = 13783 |
| D(15,36) = 21945 | D(15,81) = 9317  | D(16,36) = 38265 | D(16,81) = 22407 |
| D(15,37) = 9702  | D(15,82) = 18403 | D(16,37) = 11550 | D(16,82) = 31416 |
| D(15,38) = 9163  | D(15,83) = 11781 | D(16,38) = 13013 | D(16,83) = 25256 |
| D(15,39) = 13244 | D(15,84) = 21402 | D(16,39) = 4158  | D(16,84) = 34415 |
| D(15,40) = 5133  | D(15,85) = 9240  | D(16,40) = 17359 | D(16,85) = 20863 |
| D(15,41) = 17556 | D(15,86) = 15862 | D(16,41) = 30184 | D(16,86) = 31335 |
| D(15,42) = 15708 | D(15,87) = 29260 | D(16,42) = 28336 | D(16,87) = 41888 |
| D(15,43) = 20559 | D(15,88) = 16632 | D(16,43) = 36879 | D(16,88) = 5852  |
| D(15,44) = 9834  | D(15,89) = 8239  | D(16,44) = 22990 | D(16,89) = 11933 |
| D(15,45) = 16016 |                  | D(16,45) = 28644 |                  |

Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(17,1) = 8316   | D(17,46) = 16011 | D(18,1) = 13090  | D(18,46) = 17479 |
| D(17,2) = 16181  | D(17,47) = 9504  | D(18,2) = 11022  | D(18,47) = 5346  |
| D(17,3) = 9130   | D(17,48) = 10335 | D(18,3) = 5044   | D(18,48) = 8547  |
| D(17,4) = 24491  | D(17,49) = 19458 | D(18,4) = 18476  | D(18,49) = 14299 |
| D(17,5) = 13415  | D(17,50) = 8778  | D(18,5) = 8008   | D(18,50) = 9240  |
| D(17,6) = 4021   | D(17,51) = 14337 | D(18,6) = 5929   | D(18,51) = 18865 |
| D(17,7) = 10720  | D(17,52) = 12144 | D(18,7) = 6391   | D(18,52) = 7986  |
| D(17,8) = 9488   | D(17,53) = 14086 | D(18,8) = 7700   | D(18,53) = 15631 |
| D(17,9) = 16551  | D(17,54) = 11550 | D(18,9) = 14553  | D(18,54) = 9625  |
| D(17,10) = 8393  | D(17,55) = 16962 | D(18,10) = 6006  | D(18,55) = 12089 |
| D(17,11) = 15184 | D(17,56) = 13728 | D(18,11) = 17633 | D(18,56) = 8575  |
| D(17,12) = 16907 | D(17,57) = 21087 | D(18,12) = 11748 | D(18,57) = 13926 |
| D(17,13) = 33582 | D(17,58) = 4592  | D(18,13) = 27562 | D(18,58) = 7777  |
| D(17,14) = 12012 | D(17,59) = 16418 | D(18,14) = 16555 | D(18,59) = 10780 |
| D(17,15) = 16924 | D(17,60) = 5121  | D(18,15) = 12012 | D(18,60) = 8468  |
| D(17,16) = 32875 | D(17,61) = 27342 | D(18,16) = 26869 | D(18,61) = 21710 |
| D(17,17) = 0     | D(17,62) = 9702  | D(18,17) = 5506  | D(18,62) = 11473 |
| D(17,18) = 5506  | D(17,63) = 8870  | D(18,18) = 0     | D(18,63) = 7469  |
| D(17,19) = 29496 | D(17,64) = 25557 | D(18,19) = 23481 | D(18,64) = 22095 |
| D(17,20) = 14685 | D(17,65) = 34977 | D(18,20) = 13607 | D(18,65) = 28948 |
| D(17,21) = 22811 | D(17,66) = 10258 | D(18,21) = 15323 | D(18,66) = 5929  |
| D(17,22) = 8547  | D(17,67) = 12106 | D(18,22) = 4389  | D(18,67) = 7854  |
| D(17,23) = 7843  | D(17,68) = 15857 | D(18,23) = 4763  | D(18,68) = 16709 |
| D(17,24) = 2695  | D(17,69) = 11473 | D(18,24) = 5159  | D(18,69) = 15631 |
| D(17,25) = 12089 | D(17,70) = 11165 | D(18,25) = 5775  | D(18,70) = 18326 |
| D(17,26) = 10659 | D(17,71) = 10857 | D(18,26) = 6501  | D(18,71) = 15400 |
| D(17,27) = 30345 | D(17,72) = 20266 | D(18,27) = 24328 | D(18,72) = 22715 |
| D(17,28) = 15939 | D(17,73) = 8947  | D(18,28) = 14828 | D(18,73) = 5082  |
| D(17,29) = 13013 | D(17,74) = 9644  | D(18,29) = 9012  | D(18,74) = 8085  |
| D(17,30) = 9163  | D(17,75) = 1617  | D(18,30) = 7238  | D(18,75) = 5386  |
| D(17,31) = 3293  | D(17,76) = 15081 | D(18,31) = 2618  | D(18,76) = 14564 |
| D(17,32) = 5005  | D(17,77) = 34116 | D(18,32) = 3388  | D(18,77) = 28101 |
| D(17,33) = 15841 | D(17,78) = 14394 | D(18,33) = 8624  | D(18,78) = 14861 |
| D(17,34) = 31095 | D(17,79) = 6292  | D(18,34) = 25098 | D(18,79) = 3212  |
| D(17,35) = 12260 | D(17,80) = 22022 | D(18,35) = 6853  | D(18,80) = 17633 |
| D(17,36) = 9394  | D(17,81) = 12335 | D(18,36) = 16632 | D(18,81) = 8470  |
| D(17,37) = 27881 | D(17,82) = 616   | D(18,37) = 22249 | D(18,82) = 4004  |
| D(17,38) = 26726 | D(17,83) = 10241 | D(18,38) = 21094 | D(18,83) = 8624  |
| D(17,39) = 29044 | D(17,84) = 3311  | D(18,39) = 23019 | D(18,84) = 7315  |
| D(17,40) = 18623 | D(17,85) = 10164 | D(18,40) = 11853 | D(18,85) = 6006  |
| D(17,41) = 7870  | D(17,86) = 7334  | D(18,41) = 5005  | D(18,86) = 4774  |
| D(17,42) = 7871  | D(17,87) = 15569 | D(18,42) = 6853  | D(18,87) = 20097 |
| D(17,43) = 11396 | D(17,88) = 32422 | D(18,43) = 15246 | D(18,88) = 26407 |
| D(17,44) = 8668  | D(17,89) = 24197 | D(18,44) = 5588  | D(18,89) = 16940 |
| D(17,45) = 6331  |                  | D(18,45) = 3927  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(19,1) = 33038  | D(19,46) = 38654 | D(20,1) = 16962  | D(20,46) = 24959 |
| D(19,2) = 17482  | D(19,47) = 18135 | D(20,2) = 3465   | D(20,47) = 5181  |
| D(19,3) = 19063  | D(19,48) = 26334 | D(20,3) = 6369   | D(20,48) = 16951 |
| D(19,4) = 7777   | D(19,49) = 14205 | D(20,4) = 12947  | D(20,49) = 7920  |
| D(19,5) = 16555  | D(19,50) = 29645 | D(20,5) = 5148   | D(20,50) = 19646 |
| D(19,6) = 25410  | D(19,51) = 36960 | D(20,6) = 15488  | D(20,51) = 27269 |
| D(19,7) = 26026  | D(19,52) = 16027 | D(20,7) = 17028  | D(20,52) = 3795  |
| D(19,8) = 25487  | D(19,53) = 36729 | D(20,8) = 16104  | D(20,53) = 23650 |
| D(19,9) = 14322  | D(19,54) = 21329 | D(20,9) = 8250   | D(20,54) = 13376 |
| D(19,10) = 25641 | D(19,55) = 9779  | D(20,10) = 12291 | D(20,55) = 3399  |
| D(19,11) = 36883 | D(19,56) = 13376 | D(20,11) = 26954 | D(20,56) = 5379  |
| D(19,12) = 16756 | D(19,57) = 14578 | D(20,12) = 4191  | D(20,57) = 6369  |
| D(19,13) = 4389  | D(19,58) = 27331 | D(20,13) = 16720 | D(20,58) = 13915 |
| D(19,14) = 28413 | D(19,59) = 13398 | D(20,14) = 25387 | D(20,59) = 8470  |
| D(19,15) = 13706 | D(19,60) = 29333 | D(20,15) = 5544  | D(20,60) = 16378 |
| D(19,16) = 3996  | D(19,61) = 5313  | D(20,16) = 16027 | D(20,61) = 12870 |
| D(19,17) = 29496 | D(19,62) = 31339 | D(20,17) = 14685 | D(20,62) = 22264 |
| D(19,18) = 23481 | D(19,63) = 19404 | D(20,18) = 13607 | D(20,63) = 9218  |
| D(19,19) = 0     | D(19,64) = 5852  | D(20,19) = 13178 | D(20,64) = 11561 |
| D(19,20) = 13178 | D(19,65) = 5775  | D(20,20) = 0     | D(20,65) = 18106 |
| D(19,21) = 6622  | D(19,66) = 25564 | D(20,21) = 8686  | D(20,66) = 16566 |
| D(19,22) = 19092 | D(19,67) = 27335 | D(20,22) = 6138  | D(20,67) = 18106 |
| D(19,23) = 18469 | D(19,68) = 38500 | D(20,23) = 7556  | D(20,68) = 24808 |
| D(19,24) = 26873 | D(19,69) = 33876 | D(20,24) = 11990 | D(20,69) = 20392 |
| D(19,25) = 15092 | D(19,70) = 18711 | D(20,25) = 7832  | D(20,70) = 12749 |
| D(19,26) = 16980 | D(19,71) = 30877 | D(20,26) = 4026  | D(20,71) = 24232 |
| D(19,27) = 847   | D(19,72) = 41965 | D(20,27) = 14025 | D(20,72) = 32036 |
| D(19,28) = 12320 | D(19,73) = 17402 | D(20,28) = 1221  | D(20,73) = 6908  |
| D(19,29) = 10554 | D(19,74) = 28332 | D(20,29) = 4224  | D(20,74) = 15796 |
| D(19,30) = 18249 | D(19,75) = 29410 | D(20,30) = 10989 | D(20,75) = 14527 |
| D(19,31) = 26334 | D(19,76) = 14333 | D(20,31) = 11451 | D(20,76) = 1419  |
| D(19,32) = 28567 | D(19,77) = 4389  | D(20,32) = 13145 | D(20,77) = 17798 |
| D(19,33) = 13167 | D(19,78) = 37037 | D(20,33) = 5071  | D(20,78) = 22880 |
| D(19,34) = 1617  | D(19,79) = 18832 | D(20,34) = 14795 | D(20,79) = 9207  |
| D(19,35) = 15400 | D(19,80) = 9931  | D(20,35) = 6303  | D(20,80) = 8679  |
| D(19,36) = 34877 | D(19,81) = 18711 | D(20,36) = 21923 | D(20,81) = 8448  |
| D(19,37) = 11319 | D(19,82) = 28028 | D(20,37) = 9174  | D(20,82) = 13145 |
| D(19,38) = 10395 | D(19,83) = 21406 | D(20,38) = 9535  | D(20,83) = 13453 |
| D(19,39) = 462   | D(19,84) = 31027 | D(20,39) = 12716 | D(20,84) = 16148 |
| D(19,40) = 15974 | D(19,85) = 17475 | D(20,40) = 3992  | D(20,85) = 4521  |
| D(19,41) = 26488 | D(19,86) = 27947 | D(20,41) = 17105 | D(20,86) = 15488 |
| D(19,42) = 24640 | D(19,87) = 38192 | D(20,42) = 15257 | D(20,87) = 28501 |
| D(19,43) = 33491 | D(19,88) = 5852  | D(20,43) = 20007 | D(20,88) = 16104 |
| D(19,44) = 19294 | D(19,89) = 7082  | D(20,44) = 6871  | D(20,89) = 10303 |
| D(19,45) = 24948 |                  | D(20,45) = 15565 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(21,1) = 27159  | D(21,46) = 29183 | D(22,1) = 14762  | D(22,46) = 18711 |
| D(21,2) = 6611   | D(21,47) = 11440 | D(22,2) = 9603   | D(22,47) = 957   |
| D(21,3) = 15213  | D(21,48) = 21406 | D(22,3) = 3993   | D(22,48) = 9779  |
| D(21,4) = 6082   | D(21,49) = 1463  | D(22,4) = 14087  | D(22,49) = 12880 |
| D(21,5) = 10703  | D(21,50) = 23793 | D(22,5) = 3388   | D(22,50) = 12936 |
| D(21,6) = 20944  | D(21,51) = 30723 | D(22,6) = 8855   | D(22,51) = 20405 |
| D(21,7) = 20328  | D(21,52) = 13860 | D(22,7) = 9471   | D(22,52) = 3597  |
| D(21,8) = 20559  | D(21,53) = 27708 | D(22,8) = 8932   | D(22,53) = 16786 |
| D(21,9) = 8239   | D(21,54) = 14630 | D(22,9) = 10164  | D(22,54) = 7238  |
| D(21,10) = 18788 | D(21,55) = 2441  | D(22,10) = 6160  | D(22,55) = 8415  |
| D(21,11) = 29185 | D(21,56) = 10017 | D(22,11) = 18711 | D(22,56) = 5181  |
| D(21,12) = 5885  | D(21,57) = 8063  | D(22,12) = 10329 | D(22,57) = 12540 |
| D(21,13) = 12089 | D(21,58) = 20636 | D(22,13) = 23173 | D(22,58) = 7777  |
| D(21,14) = 22484 | D(21,59) = 9933  | D(22,14) = 12782 | D(22,59) = 6391  |
| D(21,15) = 6622  | D(21,60) = 21098 | D(22,15) = 7623  | D(22,60) = 10241 |
| D(21,16) = 11473 | D(21,61) = 6468  | D(22,16) = 22480 | D(22,61) = 17321 |
| D(21,17) = 22811 | D(21,62) = 25179 | D(22,17) = 8547  | D(22,62) = 14322 |
| D(21,18) = 15323 | D(21,63) = 13937 | D(22,18) = 4389  | D(22,63) = 3080  |
| D(21,19) = 6622  | D(21,64) = 5005  | D(22,19) = 19092 | D(22,64) = 17706 |
| D(21,20) = 8686  | D(21,65) = 13860 | D(22,20) = 6138  | D(22,65) = 24559 |
| D(21,21) = 0     | D(21,66) = 19866 | D(22,21) = 12397 | D(22,66) = 9009  |
| D(21,22) = 12397 | D(21,67) = 22484 | D(22,22) = 0     | D(22,67) = 10780 |
| D(21,23) = 16500 | D(21,68) = 29029 | D(22,23) = 2706  | D(22,68) = 18557 |
| D(21,24) = 20174 | D(21,69) = 28259 | D(22,24) = 5852  | D(22,69) = 20559 |
| D(21,25) = 14091 | D(21,70) = 23177 | D(22,25) = 1694  | D(22,70) = 10780 |
| D(21,26) = 10285 | D(21,71) = 24948 | D(22,26) = 2112  | D(22,71) = 12551 |
| D(21,27) = 6961  | D(21,72) = 34264 | D(22,27) = 19939 | D(22,72) = 23793 |
| D(21,28) = 8587  | D(21,73) = 11781 | D(22,28) = 7392  | D(22,73) = 924   |
| D(21,29) = 3278  | D(21,74) = 20713 | D(22,29) = 9570  | D(22,74) = 9317  |
| D(21,30) = 17248 | D(21,75) = 22711 | D(22,30) = 4851  | D(22,75) = 8393  |
| D(21,31) = 19635 | D(21,76) = 9973  | D(22,31) = 5313  | D(22,76) = 6534  |
| D(21,32) = 21868 | D(21,77) = 11242 | D(22,32) = 7007  | D(22,77) = 23712 |
| D(21,33) = 6314  | D(21,78) = 27566 | D(22,33) = 6776  | D(22,78) = 17094 |
| D(21,34) = 8116  | D(21,79) = 18051 | D(22,34) = 20709 | D(22,79) = 1155  |
| D(21,35) = 9548  | D(21,80) = 3542  | D(22,35) = 2233  | D(22,80) = 13475 |
| D(21,36) = 29799 | D(21,81) = 15169 | D(22,36) = 18480 | D(22,81) = 4312  |
| D(21,37) = 5852  | D(21,82) = 21329 | D(22,37) = 17860 | D(22,82) = 7007  |
| D(21,38) = 4928  | D(21,83) = 14707 | D(22,38) = 16705 | D(22,83) = 7315  |
| D(21,39) = 6162  | D(21,84) = 24328 | D(22,39) = 18630 | D(22,84) = 10010 |
| D(21,40) = 5687  | D(21,85) = 10780 | D(22,40) = 12045 | D(22,85) = 1617  |
| D(21,41) = 21560 | D(21,86) = 19250 | D(22,41) = 9933  | D(22,86) = 8239  |
| D(21,42) = 19712 | D(21,87) = 31955 | D(22,42) = 8085  | D(22,87) = 21637 |
| D(21,43) = 28366 | D(21,88) = 1617  | D(22,43) = 20482 | D(22,88) = 22018 |
| D(21,44) = 15675 | D(21,89) = 1617  | D(22,44) = 3066  | D(22,89) = 13779 |
| D(21,45) = 18172 |                  | D(22,45) = 8393  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(23,1) = 11935  | D(23,46) = 19701 | D(24,1) = 9856   | D(24,46) = 15708 |
| D(23,2) = 9438   | D(23,47) = 1760  | D(24,2) = 13871  | D(24,47) = 6809  |
| D(23,3) = 1287   | D(23,48) = 11242 | D(24,3) = 7513   | D(24,48) = 10010 |
| D(23,4) = 15169  | D(23,49) = 12715 | D(24,4) = 21868  | D(24,49) = 17148 |
| D(23,5) = 6501   | D(23,50) = 13233 | D(24,5) = 10780  | D(24,50) = 11473 |
| D(23,6) = 9779   | D(23,51) = 18623 | D(24,6) = 3696   | D(24,51) = 14014 |
| D(23,7) = 11319  | D(23,52) = 6369  | D(24,7) = 10395  | D(24,52) = 10689 |
| D(23,8) = 10395  | D(23,53) = 17853 | D(24,8) = 9163   | D(24,53) = 13783 |
| D(23,9) = 12089  | D(23,54) = 9460  | D(24,9) = 16863  | D(24,54) = 9625  |
| D(23,10) = 6765  | D(23,55) = 9537  | D(24,10) = 4081  | D(24,55) = 14267 |
| D(23,11) = 19085 | D(23,56) = 8052  | D(24,11) = 14861 | D(24,56) = 11033 |
| D(23,12) = 10164 | D(23,57) = 12342 | D(24,12) = 14597 | D(24,57) = 18392 |
| D(23,13) = 22858 | D(23,58) = 8362  | D(24,13) = 30954 | D(24,58) = 3850  |
| D(23,14) = 15708 | D(23,59) = 9504  | D(24,14) = 11242 | D(24,59) = 13783 |
| D(23,15) = 9009  | D(23,60) = 11462 | D(24,15) = 17248 | D(24,60) = 4158  |
| D(23,16) = 22165 | D(23,61) = 17721 | D(24,16) = 30261 | D(24,61) = 25102 |
| D(23,17) = 7843  | D(23,62) = 12463 | D(24,17) = 2695  | D(24,62) = 12397 |
| D(23,18) = 4763  | D(23,63) = 5940  | D(24,18) = 5159  | D(24,63) = 6237  |
| D(23,19) = 18469 | D(23,64) = 16412 | D(24,19) = 26873 | D(24,64) = 25487 |
| D(23,20) = 7556  | D(23,65) = 24244 | D(24,20) = 11990 | D(24,65) = 32340 |
| D(23,21) = 16500 | D(23,66) = 10857 | D(24,21) = 20174 | D(24,66) = 9933  |
| D(23,22) = 2706  | D(23,67) = 12397 | D(24,22) = 5852  | D(24,67) = 11781 |
| D(23,23) = 0     | D(23,68) = 19162 | D(24,23) = 6206  | D(24,68) = 15554 |
| D(23,24) = 6206  | D(23,69) = 15774 | D(24,24) = 0     | D(24,69) = 14168 |
| D(23,25) = 4147  | D(23,70) = 10109 | D(24,25) = 7469  | D(24,70) = 9779  |
| D(23,26) = 1023  | D(23,71) = 14234 | D(24,26) = 7469  | D(24,71) = 10087 |
| D(23,27) = 19316 | D(23,72) = 25091 | D(24,27) = 27720 | D(24,72) = 19943 |
| D(23,28) = 8877  | D(23,73) = 3476  | D(24,28) = 13244 | D(24,73) = 6314  |
| D(23,29) = 11946 | D(23,74) = 9537  | D(24,29) = 10703 | D(24,74) = 9317  |
| D(23,30) = 6688  | D(23,75) = 8844  | D(24,30) = 7161  | D(24,75) = 2540  |
| D(23,31) = 7073  | D(23,76) = 10065 | D(24,31) = 2541  | D(24,76) = 12386 |
| D(23,32) = 6919  | D(23,77) = 23166 | D(24,32) = 3927  | D(24,77) = 31493 |
| D(23,33) = 8470  | D(23,78) = 17083 | D(24,33) = 16170 | D(24,78) = 14091 |
| D(23,34) = 20086 | D(23,79) = 1551  | D(24,34) = 28490 | D(24,79) = 4655  |
| D(23,35) = 5346  | D(23,80) = 18810 | D(24,35) = 9625  | D(24,80) = 19327 |
| D(23,36) = 16775 | D(23,81) = 7557  | D(24,36) = 12089 | D(24,81) = 9702  |
| D(23,37) = 14025 | D(23,82) = 7361  | D(24,37) = 25641 | D(24,82) = 1155  |
| D(23,38) = 14386 | D(23,83) = 8730  | D(24,38) = 24486 | D(24,83) = 8624  |
| D(23,39) = 18007 | D(23,84) = 10230 | D(24,39) = 26411 | D(24,84) = 4157  |
| D(23,40) = 11880 | D(23,85) = 1100  | D(24,40) = 16313 | D(24,85) = 7469  |
| D(23,41) = 11396 | D(23,86) = 9691  | D(24,41) = 7546  | D(24,86) = 7007  |
| D(23,42) = 9548  | D(23,87) = 19778 | D(24,42) = 8316  | D(24,87) = 15246 |
| D(23,43) = 15389 | D(23,88) = 26510 | D(24,43) = 14091 | D(24,88) = 29799 |
| D(23,44) = 825   | D(23,89) = 18117 | D(24,44) = 7051  | D(24,89) = 21560 |
| D(23,45) = 9856  |                  | D(24,45) = 6006  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(25,1) = 18403  | D(25,46) = 20002 | D(26,1) = 16577  | D(26,46) = 24211 |
| D(25,2) = 7634   | D(25,47) = 2651  | D(26,2) = 7524   | D(26,47) = 1155  |
| D(25,3) = 5434   | D(25,48) = 12715 | D(26,3) = 1881   | D(26,48) = 11891 |
| D(25,4) = 12474  | D(25,49) = 10911 | D(26,4) = 11975  | D(26,49) = 10831 |
| D(25,5) = 2464   | D(25,50) = 13090 | D(26,5) = 5500   | D(26,50) = 15048 |
| D(25,6) = 9625   | D(25,51) = 17711 | D(26,6) = 10967  | D(26,51) = 22517 |
| D(25,7) = 10703  | D(25,52) = 5291  | D(26,7) = 11583  | D(26,52) = 5709  |
| D(25,8) = 11858  | D(25,53) = 18077 | D(26,8) = 11044  | D(26,53) = 22286 |
| D(25,9) = 10087  | D(25,54) = 8855  | D(26,9) = 12276  | D(26,54) = 9350  |
| D(25,10) = 7392  | D(25,55) = 10109 | D(26,10) = 8272  | D(26,55) = 10527 |
| D(25,11) = 19789 | D(25,56) = 6875  | D(26,11) = 22440 | D(26,56) = 7293  |
| D(25,12) = 8360  | D(25,57) = 14234 | D(26,12) = 8250  | D(26,57) = 10428 |
| D(25,13) = 18942 | D(25,58) = 8085  | D(26,13) = 21061 | D(26,58) = 9889  |
| D(25,14) = 14245 | D(25,59) = 4235  | D(26,14) = 14890 | D(26,59) = 8503  |
| D(25,15) = 6700  | D(25,60) = 11319 | D(26,15) = 9735  | D(26,60) = 12353 |
| D(25,16) = 18249 | D(25,61) = 20251 | D(26,16) = 20368 | D(26,61) = 15209 |
| D(25,17) = 12089 | D(25,62) = 14399 | D(26,17) = 10659 | D(26,62) = 16434 |
| D(25,18) = 5775  | D(25,63) = 4312  | D(26,18) = 6501  | D(26,63) = 5192  |
| D(25,19) = 15092 | D(25,64) = 18863 | D(26,19) = 16980 | D(26,64) = 15594 |
| D(25,20) = 7832  | D(25,65) = 20328 | D(26,20) = 4026  | D(26,65) = 22447 |
| D(25,21) = 14091 | D(25,66) = 9779  | D(26,21) = 10285 | D(26,66) = 11121 |
| D(25,22) = 1694  | D(25,67) = 12012 | D(26,22) = 2112  | D(26,67) = 12892 |
| D(25,23) = 4147  | D(25,68) = 19866 | D(26,23) = 1023  | D(26,68) = 24057 |
| D(25,24) = 7469  | D(25,69) = 20405 | D(26,24) = 7469  | D(26,69) = 22671 |
| D(25,25) = 0     | D(25,70) = 8932  | D(26,25) = 3846  | D(26,70) = 12892 |
| D(25,26) = 3846  | D(25,71) = 13090 | D(26,26) = 0     | D(26,71) = 14659 |
| D(25,27) = 17171 | D(25,72) = 24871 | D(26,27) = 17827 | D(26,72) = 27522 |
| D(25,28) = 9086  | D(25,73) = 2541  | D(26,28) = 5280  | D(26,73) = 3036  |
| D(25,29) = 4466  | D(25,74) = 10395 | D(26,29) = 11550 | D(26,74) = 11429 |
| D(25,30) = 5698  | D(25,75) = 11935 | D(26,30) = 6963  | D(26,75) = 10501 |
| D(25,31) = 8085  | D(25,76) = 8228  | D(26,31) = 7425  | D(26,76) = 4422  |
| D(25,32) = 8932  | D(25,77) = 26796 | D(26,32) = 9119  | D(26,77) = 21600 |
| D(25,33) = 5390  | D(25,78) = 18403 | D(26,33) = 8888  | D(26,78) = 22594 |
| D(25,34) = 17941 | D(25,79) = 2596  | D(26,34) = 18597 | D(26,79) = 2233  |
| D(25,35) = 1655  | D(25,80) = 15169 | D(26,35) = 4345  | D(26,80) = 11363 |
| D(25,36) = 18711 | D(25,81) = 2926  | D(26,36) = 20592 | D(26,81) = 6424  |
| D(25,37) = 19558 | D(25,82) = 10549 | D(26,37) = 15748 | D(26,82) = 9119  |
| D(25,38) = 18557 | D(25,83) = 8932  | D(26,38) = 14593 | D(26,83) = 9424  |
| D(25,39) = 14630 | D(25,84) = 13552 | D(26,39) = 16518 | D(26,84) = 12118 |
| D(25,40) = 10079 | D(25,85) = 3351  | D(26,40) = 9965  | D(26,85) = 495   |
| D(25,41) = 9933  | D(25,86) = 10010 | D(26,41) = 12045 | D(26,86) = 10351 |
| D(25,42) = 9394  | D(25,87) = 18943 | D(26,42) = 10197 | D(26,87) = 23749 |
| D(25,43) = 20482 | D(25,88) = 25102 | D(26,43) = 22594 | D(26,88) = 19906 |
| D(25,44) = 4972  | D(25,89) = 15477 | D(26,44) = 1419  | D(26,89) = 11667 |
| D(25,45) = 8624  |                  | D(26,45) = 11505 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(27,1) = 33957  | D(27,46) = 39501 | D(28,1) = 17854  | D(28,46) = 26180 |
| D(27,2) = 17821  | D(27,47) = 18982 | D(28,2) = 3490   | D(28,47) = 6435  |
| D(27,3) = 19910  | D(27,48) = 27181 | D(28,3) = 7590   | D(28,48) = 18172 |
| D(27,4) = 8624   | D(27,49) = 14544 | D(28,4) = 12089  | D(28,49) = 4785  |
| D(27,5) = 17402  | D(27,50) = 30492 | D(28,5) = 4587   | D(28,50) = 20867 |
| D(27,6) = 26257  | D(27,51) = 37807 | D(28,6) = 16709  | D(28,51) = 28490 |
| D(27,7) = 26873  | D(27,52) = 16874 | D(28,7) = 18249  | D(28,52) = 4309  |
| D(27,8) = 26334  | D(27,53) = 37576 | D(28,8) = 17325  | D(28,53) = 24871 |
| D(27,9) = 15169  | D(27,54) = 22176 | D(28,9) = 7293   | D(28,54) = 14630 |
| D(27,10) = 27720 | D(27,55) = 10626 | D(28,10) = 13552 | D(28,55) = 2541  |
| D(27,11) = 37730 | D(27,56) = 14223 | D(28,11) = 28175 | D(28,56) = 4158  |
| D(27,12) = 17095 | D(27,57) = 14917 | D(28,12) = 4290  | D(28,57) = 6468  |
| D(27,13) = 3542  | D(27,58) = 6699  | D(28,13) = 15862 | D(28,58) = 15169 |
| D(27,14) = 29337 | D(27,59) = 14245 | D(28,14) = 26707 | D(28,59) = 6424  |
| D(27,15) = 14553 | D(27,60) = 30180 | D(28,15) = 4323  | D(28,60) = 17633 |
| D(27,16) = 2849  | D(27,61) = 6160  | D(28,16) = 15169 | D(28,61) = 12012 |
| D(27,17) = 30345 | D(27,62) = 32186 | D(28,17) = 15939 | D(28,62) = 23485 |
| D(27,18) = 24328 | D(27,63) = 20251 | D(28,18) = 14828 | D(28,63) = 10472 |
| D(27,19) = 847   | D(27,64) = 6699  | D(28,19) = 12320 | D(28,64) = 10703 |
| D(27,20) = 14025 | D(27,65) = 4928  | D(28,20) = 1221  | D(28,65) = 17248 |
| D(27,21) = 6961  | D(27,66) = 26411 | D(28,21) = 8587  | D(28,66) = 17787 |
| D(27,22) = 19939 | D(27,67) = 28182 | D(28,22) = 7392  | D(28,67) = 19327 |
| D(27,23) = 19316 | D(27,68) = 39347 | D(28,23) = 8877  | D(28,68) = 26029 |
| D(27,24) = 27720 | D(27,69) = 34723 | D(28,24) = 13244 | D(28,69) = 22176 |
| D(27,25) = 17171 | D(27,70) = 19943 | D(28,25) = 9086  | D(28,70) = 14135 |
| D(27,26) = 17827 | D(27,71) = 31801 | D(28,26) = 5280  | D(28,71) = 25552 |
| D(27,27) = 0     | D(27,72) = 42812 | D(28,27) = 13167 | D(28,72) = 33257 |
| D(27,28) = 13167 | D(27,73) = 18249 | D(28,28) = 0     | D(28,73) = 8162  |
| D(27,29) = 11443 | D(27,74) = 29179 | D(28,29) = 3696  | D(28,74) = 17017 |
| D(27,30) = 30328 | D(27,75) = 30257 | D(28,30) = 12243 | D(28,75) = 15781 |
| D(27,31) = 27181 | D(27,76) = 15180 | D(28,31) = 12705 | D(28,76) = 4026  |
| D(27,32) = 29414 | D(27,77) = 4620  | D(28,32) = 14399 | D(28,77) = 16940 |
| D(27,33) = 14014 | D(27,78) = 37884 | D(28,33) = 3850  | D(28,78) = 24101 |
| D(27,34) = 770   | D(27,79) = 19679 | D(28,34) = 13937 | D(28,79) = 10428 |
| D(27,35) = 16247 | D(27,80) = 10270 | D(28,35) = 5742  | D(28,80) = 8580  |
| D(27,36) = 35724 | D(27,81) = 19558 | D(28,36) = 23177 | D(28,81) = 9702  |
| D(27,37) = 12166 | D(27,82) = 28875 | D(28,37) = 8316  | D(28,82) = 14399 |
| D(27,38) = 11242 | D(27,83) = 22253 | D(28,38) = 8677  | D(28,83) = 14707 |
| D(27,39) = 1309  | D(27,84) = 31874 | D(28,39) = 11858 | D(28,84) = 17402 |
| D(27,40) = 16365 | D(27,85) = 18322 | D(28,40) = 3986  | D(28,85) = 5775  |
| D(27,41) = 27335 | D(27,86) = 28794 | D(28,41) = 18326 | D(28,86) = 16709 |
| D(27,42) = 25487 | D(27,87) = 30939 | D(28,42) = 16478 | D(28,87) = 29722 |
| D(27,43) = 34338 | D(27,88) = 5852  | D(28,43) = 21791 | D(28,88) = 15246 |
| D(27,44) = 20141 | D(27,89) = 7421  | D(28,44) = 8092  | D(28,89) = 10204 |
| D(27,45) = 25795 |                  | D(28,45) = 16786 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(29,1) = 18354  | D(29,46) = 25759 | D(30,1) = 15169  | D(30,46) = 26393 |
| D(29,2) = 4940   | D(29,47) = 9966  | D(30,2) = 10637  | D(30,47) = 5808  |
| D(29,3) = 13200  | D(29,48) = 25434 | D(30,3) = 7979   | D(30,48) = 13947 |
| D(29,4) = 6125   | D(29,49) = 1881  | D(30,4) = 14399  | D(30,49) = 13914 |
| D(29,5) = 12320  | D(29,50) = 14106 | D(30,5) = 8162   | D(30,50) = 16247 |
| D(29,6) = 17098  | D(29,51) = 20746 | D(30,6) = 9548   | D(30,51) = 17403 |
| D(29,7) = 8151   | D(29,52) = 7920  | D(30,7) = 12243  | D(30,52) = 8448  |
| D(29,8) = 23439  | D(29,53) = 26429 | D(30,8) = 13090  | D(30,53) = 24468 |
| D(29,9) = 4576   | D(29,54) = 10010 | D(30,9) = 10857  | D(30,54) = 2926  |
| D(29,10) = 8624  | D(29,55) = 990   | D(30,10) = 5005  | D(30,55) = 13266 |
| D(29,11) = 21868 | D(29,56) = 4455  | D(30,11) = 19481 | D(30,56) = 10032 |
| D(29,12) = 4323  | D(29,57) = 4059  | D(30,12) = 11363 | D(30,57) = 17391 |
| D(29,13) = 12650 | D(29,58) = 8624  | D(30,13) = 22099 | D(30,58) = 7700  |
| D(29,14) = 11935 | D(29,59) = 10349 | D(30,14) = 10167 | D(30,59) = 5313  |
| D(29,15) = 3500  | D(29,60) = 8778  | D(30,15) = 10395 | D(30,60) = 12859 |
| D(29,16) = 12340 | D(29,61) = 11480 | D(30,16) = 21406 | D(30,61) = 23408 |
| D(29,17) = 13013 | D(29,62) = 18249 | D(30,17) = 9163  | D(30,62) = 15554 |
| D(29,18) = 9012  | D(29,63) = 13090 | D(30,18) = 7238  | D(30,63) = 2079  |
| D(29,19) = 10554 | D(29,64) = 9270  | D(30,19) = 18249 | D(30,64) = 22020 |
| D(29,20) = 4224  | D(29,65) = 14420 | D(30,20) = 10989 | D(30,65) = 23485 |
| D(29,21) = 3278  | D(29,66) = 8526  | D(30,21) = 17248 | D(30,66) = 12166 |
| D(29,22) = 9570  | D(29,67) = 10736 | D(30,22) = 4851  | D(30,67) = 14399 |
| D(29,23) = 11946 | D(29,68) = 27199 | D(30,23) = 6688  | D(30,68) = 26257 |
| D(29,24) = 10703 | D(29,69) = 13552 | D(30,24) = 7161  | D(30,69) = 18711 |
| D(29,25) = 4466  | D(29,70) = 11087 | D(30,25) = 5698  | D(30,70) = 3850  |
| D(29,26) = 11550 | D(29,71) = 18249 | D(30,26) = 6963  | D(30,71) = 9012  |
| D(29,27) = 11443 | D(29,72) = 26873 | D(30,27) = 30328 | D(30,72) = 24563 |
| D(29,28) = 3696  | D(29,73) = 11011 | D(30,28) = 12243 | D(30,73) = 3311  |
| D(29,29) = 0     | D(29,74) = 14169 | D(30,29) = 7469  | D(30,74) = 14123 |
| D(29,30) = 7469  | D(29,75) = 14630 | D(30,30) = 0     | D(30,75) = 9009  |
| D(29,31) = 9384  | D(29,76) = 5610  | D(30,31) = 9086  | D(30,76) = 11385 |
| D(29,32) = 11550 | D(29,77) = 8334  | D(30,32) = 9933  | D(30,77) = 29953 |
| D(29,33) = 2114  | D(29,78) = 23639 | D(30,33) = 7854  | D(30,78) = 24794 |
| D(29,34) = 12048 | D(29,79) = 10164 | D(30,34) = 21098 | D(30,79) = 5137  |
| D(29,35) = 7260  | D(29,80) = 3740  | D(30,35) = 7007  | D(30,80) = 18326 |
| D(29,36) = 14322 | D(29,81) = 11935 | D(30,36) = 17017 | D(30,81) = 5929  |
| D(29,37) = 8500  | D(29,82) = 12089 | D(30,37) = 22715 | D(30,82) = 7623  |
| D(29,38) = 9655  | D(29,83) = 9936  | D(30,38) = 21714 | D(30,83) = 3003  |
| D(29,39) = 10006 | D(29,84) = 16632 | D(30,39) = 17787 | D(30,84) = 10626 |
| D(29,40) = 2607  | D(29,85) = 10956 | D(30,40) = 14994 | D(30,85) = 6468  |
| D(29,41) = 27289 | D(29,86) = 11313 | D(30,41) = 10241 | D(30,86) = 11550 |
| D(29,42) = 24879 | D(29,87) = 19601 | D(30,42) = 10626 | D(30,87) = 18635 |
| D(29,43) = 13013 | D(29,88) = 8118  | D(30,43) = 20251 | D(30,88) = 28259 |
| D(29,44) = 12606 | D(29,89) = 4279  | D(30,44) = 7513  | D(30,89) = 18634 |
| D(29,45) = 29214 |                  | D(30,45) = 9856  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(31,1) = 8008   | D(31,46) = 22042 | D(32,1) = 7900   | D(32,46) = 20482 |
| D(31,2) = 12552  | D(31,47) = 6270  | D(32,2) = 14718  | D(32,47) = 7964  |
| D(31,3) = 8360   | D(31,48) = 7392  | D(32,3) = 8206   | D(32,48) = 8008  |
| D(31,4) = 22176  | D(31,49) = 15829 | D(32,4) = 12936  | D(32,49) = 17995 |
| D(31,5) = 10164  | D(31,50) = 9163  | D(32,5) = 12166  | D(32,50) = 8624  |
| D(31,6) = 4697   | D(31,51) = 13321 | D(32,6) = 2387   | D(32,51) = 12474 |
| D(31,7) = 5775   | D(31,52) = 10073 | D(32,7) = 5775   | D(32,52) = 12075 |
| D(31,8) = 6545   | D(31,53) = 20117 | D(32,8) = 7161   | D(32,53) = 18557 |
| D(31,9) = 22253  | D(31,54) = 12166 | D(32,9) = 23100  | D(32,54) = 13629 |
| D(31,10) = 8468  | D(31,55) = 13728 | D(32,10) = 8239  | D(32,55) = 15422 |
| D(31,11) = 14245 | D(31,56) = 10494 | D(32,11) = 13398 | D(32,56) = 12188 |
| D(31,12) = 13278 | D(31,57) = 17853 | D(32,12) = 15444 | D(32,57) = 19547 |
| D(31,13) = 30415 | D(31,58) = 6083  | D(32,13) = 32725 | D(32,58) = 7623  |
| D(31,14) = 14091 | D(31,59) = 11088 | D(32,14) = 15169 | D(32,59) = 13090 |
| D(31,15) = 16709 | D(31,60) = 4697  | D(32,15) = 9548  | D(32,60) = 3388  |
| D(31,16) = 29722 | D(31,61) = 24563 | D(32,16) = 32032 | D(32,61) = 26796 |
| D(31,17) = 3293  | D(31,62) = 10241 | D(32,17) = 5005  | D(32,62) = 10318 |
| D(31,18) = 2618  | D(31,63) = 9741  | D(32,18) = 3388  | D(32,63) = 9702  |
| D(31,19) = 26334 | D(31,64) = 24407 | D(32,19) = 28567 | D(32,64) = 26640 |
| D(31,20) = 11451 | D(31,65) = 31801 | D(32,20) = 13145 | D(32,65) = 34111 |
| D(31,21) = 19635 | D(31,66) = 5313  | D(32,21) = 21868 | D(32,66) = 5313  |
| D(31,22) = 5313  | D(31,67) = 7700  | D(32,22) = 7007  | D(32,67) = 7084  |
| D(31,23) = 7073  | D(31,68) = 21888 | D(32,23) = 6919  | D(32,68) = 20328 |
| D(31,24) = 2541  | D(31,69) = 15708 | D(32,24) = 3927  | D(32,69) = 14168 |
| D(31,25) = 8085  | D(31,70) = 12243 | D(32,25) = 8932  | D(32,70) = 13090 |
| D(31,26) = 7425  | D(31,71) = 12936 | D(32,26) = 9119  | D(32,71) = 14014 |
| D(31,27) = 27181 | D(31,72) = 19327 | D(32,27) = 29414 | D(32,72) = 18480 |
| D(31,28) = 12705 | D(31,73) = 7893  | D(32,28) = 14399 | D(32,73) = 7854  |
| D(31,29) = 9384  | D(31,74) = 6391  | D(32,29) = 11550 | D(32,74) = 6699  |
| D(31,30) = 9086  | D(31,75) = 3003  | D(32,30) = 9933  | D(32,75) = 2772  |
| D(31,31) = 0     | D(31,76) = 11847 | D(32,31) = 1540  | D(32,76) = 13541 |
| D(31,32) = 1540  | D(31,77) = 30954 | D(32,32) = 0     | D(32,77) = 33880 |
| D(31,33) = 15631 | D(31,78) = 20425 | D(32,33) = 6853  | D(32,78) = 18865 |
| D(31,34) = 27951 | D(31,79) = 5522  | D(32,34) = 30184 | D(32,79) = 5368  |
| D(31,35) = 9009  | D(31,80) = 18788 | D(32,35) = 11011 | D(32,80) = 20482 |
| D(31,36) = 12705 | D(31,81) = 11666 | D(32,36) = 11165 | D(32,81) = 11627 |
| D(31,37) = 25102 | D(31,82) = 1848  | D(32,37) = 27335 | D(32,82) = 3465  |
| D(31,38) = 24101 | D(31,83) = 11165 | D(32,38) = 26334 | D(32,83) = 12628 |
| D(31,39) = 25872 | D(31,84) = 4158  | D(32,39) = 28114 | D(32,84) = 2772  |
| D(31,40) = 14994 | D(31,85) = 6930  | D(32,40) = 17160 | D(32,85) = 8624  |
| D(31,41) = 2233  | D(31,86) = 4081  | D(32,41) = 3696  | D(32,86) = 4697  |
| D(31,42) = 4081  | D(31,87) = 14553 | D(32,42) = 5544  | D(32,87) = 13706 |
| D(31,43) = 16093 | D(31,88) = 29260 | D(32,43) = 14553 | D(32,88) = 31570 |
| D(31,44) = 7898  | D(31,89) = 21021 | D(32,44) = 7744  | D(32,89) = 23321 |
| D(31,45) = 2079  |                  | D(32,45) = 3000  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(33,1) = 21538  | D(33,46) = 26334 | D(34,1) = 34628  | D(34,46) = 40271 |
| D(33,2) = 4433   | D(33,47) = 7733  | D(34,2) = 18976  | D(34,47) = 19752 |
| D(33,3) = 9064   | D(33,48) = 14014 | D(34,3) = 20680  | D(34,48) = 27951 |
| D(33,4) = 12936  | D(33,49) = 6974  | D(34,4) = 9394   | D(34,49) = 15699 |
| D(33,5) = 4389   | D(33,50) = 17171 | D(34,5) = 18172  | D(34,50) = 31262 |
| D(33,6) = 13090  | D(33,51) = 24640 | D(34,6) = 27027  | D(34,51) = 38577 |
| D(33,7) = 13706  | D(33,52) = 5753  | D(34,7) = 27643  | D(34,52) = 17644 |
| D(33,8) = 13167  | D(33,53) = 24409 | D(34,8) = 27104  | D(34,53) = 38346 |
| D(33,9) = 3773   | D(33,54) = 8547  | D(34,9) = 15939  | D(34,54) = 22946 |
| D(33,10) = 12705 | D(33,55) = 3388  | D(34,10) = 28490 | D(34,55) = 11396 |
| D(33,11) = 24563 | D(33,56) = 3113  | D(34,11) = 38500 | D(34,56) = 14993 |
| D(33,12) = 5159  | D(33,57) = 7337  | D(34,12) = 18250 | D(34,57) = 16072 |
| D(33,13) = 16709 | D(33,58) = 16628 | D(34,13) = 2772  | D(34,58) = 28948 |
| D(33,14) = 28720 | D(33,59) = 3003  | D(34,14) = 29953 | D(34,59) = 15015 |
| D(33,15) = 2926  | D(33,60) = 17017 | D(34,15) = 15323 | D(34,60) = 30950 |
| D(33,16) = 16016 | D(33,61) = 12859 | D(34,16) = 2079  | D(34,61) = 6930  |
| D(33,17) = 15841 | D(33,62) = 7315  | D(34,17) = 31095 | D(34,62) = 32956 |
| D(33,18) = 8624  | D(33,63) = 18557 | D(34,18) = 25098 | D(34,63) = 21021 |
| D(33,19) = 13167 | D(33,64) = 11550 | D(34,19) = 1617  | D(34,64) = 7469  |
| D(33,20) = 5071  | D(33,65) = 18095 | D(34,20) = 14795 | D(34,65) = 4158  |
| D(33,21) = 6314  | D(33,66) = 13244 | D(34,21) = 8116  | D(34,66) = 27181 |
| D(33,22) = 6776  | D(33,67) = 15015 | D(34,22) = 20709 | D(34,67) = 28952 |
| D(33,23) = 8470  | D(33,68) = 26180 | D(34,23) = 20086 | D(34,68) = 40117 |
| D(33,24) = 16170 | D(33,69) = 20174 | D(34,24) = 28490 | D(34,69) = 35493 |
| D(33,25) = 5390  | D(33,70) = 10010 | D(34,25) = 17941 | D(34,70) = 20790 |
| D(33,26) = 8888  | D(33,71) = 27565 | D(34,26) = 18597 | D(34,71) = 32417 |
| D(33,27) = 14014 | D(33,72) = 29645 | D(34,27) = 770   | D(34,72) = 43582 |
| D(33,28) = 3850  | D(33,73) = 6699  | D(34,28) = 13937 | D(34,73) = 19019 |
| D(33,29) = 2114  | D(33,74) = 13552 | D(34,29) = 12048 | D(34,74) = 29949 |
| D(33,30) = 7854  | D(33,75) = 18707 | D(34,30) = 21098 | D(34,75) = 31027 |
| D(33,31) = 15631 | D(33,76) = 7898  | D(34,31) = 27951 | D(34,76) = 15950 |
| D(33,32) = 6853  | D(33,77) = 17787 | D(34,32) = 30184 | D(34,77) = 3850  |
| D(33,33) = 0     | D(33,78) = 24717 | D(34,33) = 14784 | D(34,78) = 38654 |
| D(33,34) = 14784 | D(33,79) = 8833  | D(34,34) = 0     | D(34,79) = 20449 |
| D(33,35) = 4774  | D(33,80) = 9163  | D(34,35) = 17017 | D(34,80) = 11425 |
| D(33,36) = 21175 | D(33,81) = 8008  | D(34,36) = 36494 | D(34,81) = 20328 |
| D(33,37) = 9163  | D(33,82) = 17325 | D(34,37) = 12936 | D(34,82) = 29645 |
| D(33,38) = 8624  | D(33,83) = 8625  | D(34,38) = 12012 | D(34,83) = 23023 |
| D(33,39) = 12705 | D(33,84) = 20324 | D(34,39) = 2079  | D(34,84) = 32644 |
| D(33,40) = 4216  | D(33,85) = 8393  | D(34,40) = 17662 | D(34,85) = 19092 |
| D(33,41) = 14168 | D(33,86) = 12474 | D(34,41) = 28105 | D(34,86) = 29564 |
| D(33,42) = 12320 | D(33,87) = 25872 | D(34,42) = 26257 | D(34,87) = 39809 |
| D(33,43) = 19789 | D(33,88) = 16093 | D(34,43) = 35108 | D(34,88) = 5082  |
| D(33,44) = 9295  | D(33,89) = 7931  | D(34,44) = 20911 | D(34,89) = 8626  |
| D(33,45) = 12628 |                  | D(34,45) = 26565 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(35,1) = 18990  | D(35,46) = 19173 | D(36,1) = 6468   | D(36,46) = 7238  |
| D(35,2) = 8514   | D(35,47) = 3190  | D(36,2) = 17490  | D(36,47) = 19437 |
| D(35,3) = 6673   | D(35,48) = 13629 | D(36,3) = 18062  | D(36,48) = 14322 |
| D(35,4) = 11704  | D(35,49) = 11791 | D(36,4) = 29872  | D(36,49) = 20767 |
| D(35,5) = 1155   | D(35,50) = 12786 | D(36,5) = 20636  | D(36,50) = 8470  |
| D(35,6) = 15246  | D(35,51) = 20174 | D(36,6) = 9163   | D(36,51) = 2849  |
| D(35,7) = 12089  | D(35,52) = 3135  | D(36,7) = 12012  | D(36,52) = 20845 |
| D(35,8) = 11165  | D(35,53) = 17325 | D(36,8) = 13475  | D(36,53) = 5852  |
| D(35,9) = 8624   | D(35,54) = 9471  | D(36,9) = 25949  | D(36,54) = 18711 |
| D(35,10) = 8162  | D(35,55) = 6468  | D(36,10) = 16401 | D(36,55) = 22330 |
| D(35,11) = 21791 | D(35,56) = 2970  | D(36,11) = 9240  | D(36,56) = 20966 |
| D(35,12) = 9240  | D(35,57) = 11418 | D(36,12) = 18216 | D(36,57) = 20394 |
| D(35,13) = 19789 | D(35,58) = 12243 | D(36,13) = 38958 | D(36,58) = 14707 |
| D(35,14) = 15554 | D(35,59) = 3003  | D(36,14) = 9317  | D(36,59) = 23639 |
| D(35,15) = 5775  | D(35,60) = 12474 | D(36,15) = 21945 | D(36,60) = 9163  |
| D(35,16) = 19096 | D(35,61) = 15935 | D(36,16) = 38265 | D(36,61) = 33106 |
| D(35,17) = 12260 | D(35,62) = 17325 | D(36,17) = 9394  | D(36,62) = 6083  |
| D(35,18) = 6853  | D(35,63) = 4466  | D(36,18) = 16632 | D(36,63) = 16170 |
| D(35,19) = 15400 | D(35,64) = 14549 | D(36,19) = 34877 | D(36,64) = 33491 |
| D(35,20) = 6303  | D(35,65) = 21175 | D(36,20) = 21923 | D(36,65) = 40344 |
| D(35,21) = 9548  | D(35,66) = 12397 | D(36,21) = 29799 | D(36,66) = 12474 |
| D(35,22) = 2233  | D(35,67) = 13860 | D(36,22) = 18480 | D(36,67) = 9317  |
| D(35,23) = 5346  | D(35,68) = 19712 | D(36,23) = 16775 | D(36,68) = 8624  |
| D(35,24) = 9625  | D(35,69) = 18480 | D(36,24) = 12089 | D(36,69) = 3458  |
| D(35,25) = 1655  | D(35,70) = 9933  | D(36,25) = 18711 | D(36,70) = 21868 |
| D(35,26) = 4345  | D(35,71) = 14399 | D(36,26) = 20592 | D(36,71) = 8162  |
| D(35,27) = 16247 | D(35,72) = 27797 | D(36,27) = 35724 | D(36,72) = 12243 |
| D(35,28) = 5742  | D(35,73) = 1632  | D(36,28) = 23177 | D(36,73) = 14014 |
| D(35,29) = 7260  | D(35,74) = 11473 | D(36,29) = 14322 | D(36,74) = 9394  |
| D(35,30) = 7007  | D(35,75) = 12162 | D(36,30) = 17017 | D(36,75) = 9394  |
| D(35,31) = 9009  | D(35,76) = 6072  | D(36,31) = 12705 | D(36,76) = 22319 |
| D(35,32) = 11011 | D(35,77) = 20097 | D(36,32) = 11165 | D(36,77) = 39497 |
| D(35,33) = 4774  | D(35,78) = 17633 | D(36,33) = 21175 | D(36,78) = 5775  |
| D(35,34) = 17017 | D(35,79) = 3795  | D(36,34) = 36494 | D(36,79) = 15224 |
| D(35,35) = 0     | D(35,80) = 11858 | D(36,35) = 19481 | D(36,80) = 32109 |
| D(35,36) = 19481 | D(35,81) = 3465  | D(36,36) = 0     | D(36,81) = 17402 |
| D(35,37) = 15935 | D(35,82) = 10780 | D(36,37) = 33645 | D(36,82) = 10934 |
| D(35,38) = 14472 | D(35,83) = 9848  | D(36,38) = 32490 | D(36,83) = 17710 |
| D(35,39) = 14938 | D(35,84) = 13779 | D(36,39) = 33415 | D(36,84) = 8085  |
| D(35,40) = 10870 | D(35,85) = 3850  | D(36,40) = 19926 | D(36,85) = 20097 |
| D(35,41) = 13397 | D(35,86) = 9933  | D(36,41) = 9471  | D(36,86) = 12012 |
| D(35,42) = 11473 | D(35,87) = 21329 | D(36,42) = 12628 | D(36,87) = 3542  |
| D(35,43) = 18095 | D(35,88) = 19558 | D(36,43) = 3077  | D(36,88) = 37803 |
| D(35,44) = 6171  | D(35,89) = 11165 | D(36,44) = 17600 | D(36,89) = 31416 |
| D(35,45) = 11319 |                  | D(36,45) = 14476 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(37,1) = 28093  | D(37,46) = 34800 | D(38,1) = 28195  | D(38,46) = 33645 |
| D(37,2) = 7271   | D(37,47) = 16903 | D(38,2) = 8425   | D(38,47) = 15720 |
| D(37,3) = 15906  | D(37,48) = 27797 | D(38,3) = 16267  | D(38,48) = 26334 |
| D(37,4) = 12936  | D(37,49) = 3994  | D(38,4) = 12012  | D(38,49) = 5145  |
| D(37,5) = 14780  | D(37,50) = 28413 | D(38,5) = 13317  | D(38,50) = 27258 |
| D(37,6) = 27335  | D(37,51) = 34338 | D(38,6) = 25872  | D(38,51) = 33183 |
| D(37,7) = 27566  | D(37,52) = 20247 | D(38,7) = 26103  | D(38,52) = 18784 |
| D(37,8) = 26950  | D(37,53) = 32952 | D(38,8) = 25487  | D(38,53) = 31797 |
| D(37,9) = 12089  | D(37,54) = 20097 | D(38,9) = 11165  | D(38,54) = 18942 |
| D(37,10) = 17787 | D(37,55) = 5775  | D(38,10) = 18095 | D(38,55) = 5236  |
| D(37,11) = 35955 | D(37,56) = 9372  | D(38,11) = 34800 | D(38,56) = 8833  |
| D(37,12) = 6545  | D(37,57) = 4367  | D(38,12) = 7700  | D(38,57) = 5522  |
| D(37,13) = 12243 | D(37,58) = 26099 | D(38,13) = 13706 | D(38,58) = 24944 |
| D(37,14) = 29260 | D(37,59) = 16320 | D(38,14) = 28105 | D(38,59) = 14857 |
| D(37,15) = 9702  | D(37,60) = 28101 | D(38,15) = 9163  | D(38,60) = 26946 |
| D(37,16) = 11550 | D(37,61) = 5698  | D(38,16) = 13013 | D(38,61) = 6468  |
| D(37,17) = 27881 | D(37,62) = 30107 | D(38,17) = 26726 | D(38,62) = 28952 |
| D(37,18) = 22249 | D(37,63) = 20478 | D(38,18) = 21094 | D(38,63) = 19015 |
| D(37,19) = 11319 | D(37,64) = 6622  | D(38,19) = 10395 | D(38,64) = 5698  |
| D(37,20) = 9174  | D(37,65) = 13629 | D(38,20) = 9535  | D(38,65) = 15092 |
| D(37,21) = 5852  | D(37,66) = 27104 | D(38,21) = 4928  | D(38,66) = 25641 |
| D(37,22) = 17860 | D(37,67) = 28875 | D(38,22) = 16705 | D(38,67) = 27412 |
| D(37,23) = 14025 | D(37,68) = 25285 | D(38,23) = 14386 | D(38,68) = 34184 |
| D(37,24) = 25641 | D(37,69) = 32644 | D(38,24) = 24486 | D(38,69) = 31489 |
| D(37,25) = 19558 | D(37,70) = 20020 | D(38,25) = 18557 | D(38,70) = 18403 |
| D(37,26) = 15748 | D(37,71) = 31724 | D(38,26) = 14593 | D(38,71) = 30569 |
| D(37,27) = 12166 | D(37,72) = 41961 | D(38,27) = 11242 | D(38,72) = 40806 |
| D(37,28) = 8316  | D(37,73) = 18014 | D(38,28) = 8677  | D(38,73) = 16551 |
| D(37,29) = 8500  | D(37,74) = 27100 | D(38,29) = 9655  | D(38,74) = 25945 |
| D(37,30) = 22715 | D(37,75) = 28178 | D(38,30) = 21714 | D(38,75) = 27023 |
| D(37,31) = 25102 | D(37,76) = 10329 | D(38,31) = 24101 | D(38,76) = 10690 |
| D(37,32) = 27335 | D(37,77) = 9009  | D(38,32) = 26334 | D(38,77) = 10395 |
| D(37,33) = 9163  | D(37,78) = 33260 | D(38,33) = 8624  | D(38,78) = 32105 |
| D(37,34) = 12936 | D(37,79) = 15576 | D(38,34) = 12012 | D(38,79) = 15937 |
| D(37,35) = 15935 | D(37,80) = 4466  | D(38,35) = 14472 | D(38,80) = 2568  |
| D(37,36) = 33645 | D(37,81) = 18553 | D(38,36) = 32490 | D(38,81) = 17090 |
| D(37,37) = 0     | D(37,82) = 26796 | D(38,37) = 1463  | D(38,82) = 25641 |
| D(37,38) = 1463  | D(37,83) = 20174 | D(38,38) = 0     | D(38,83) = 19019 |
| D(37,39) = 10857 | D(37,84) = 29795 | D(38,39) = 9933  | D(38,84) = 28640 |
| D(37,40) = 5845  | D(37,85) = 16243 | D(38,40) = 7060  | D(38,85) = 15142 |
| D(37,41) = 27951 | D(37,86) = 26715 | D(38,41) = 26488 | D(38,86) = 25560 |
| D(37,42) = 26103 | D(37,87) = 35493 | D(38,42) = 24640 | D(38,87) = 34338 |
| D(37,43) = 32259 | D(37,88) = 6237  | D(38,43) = 31113 | D(38,88) = 7931  |
| D(37,44) = 13200 | D(37,89) = 5621  | D(38,44) = 13561 | D(38,89) = 4697  |
| D(37,45) = 24563 |                  | D(38,45) = 23100 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(39,1) = 32549  | D(39,46) = 38192 | D(40,1) = 18084  | D(40,46) = 31393 |
| D(39,2) = 17022  | D(39,47) = 17613 | D(40,2) = 1485   | D(40,47) = 11286 |
| D(39,3) = 18601  | D(39,48) = 25872 | D(40,3) = 10470  | D(40,48) = 23266 |
| D(39,4) = 7315   | D(39,49) = 13741 | D(40,4) = 11820  | D(40,49) = 831   |
| D(39,5) = 16093  | D(39,50) = 29183 | D(40,5) = 12062  | D(40,50) = 19483 |
| D(39,6) = 24948  | D(39,51) = 36498 | D(40,6) = 15733  | D(40,51) = 26356 |
| D(39,7) = 25564  | D(39,52) = 15565 | D(40,7) = 13761  | D(40,52) = 8115  |
| D(39,8) = 25025  | D(39,53) = 36267 | D(40,8) = 21651  | D(40,53) = 32105 |
| D(39,9) = 13860  | D(39,54) = 20867 | D(40,9) = 10176  | D(40,54) = 15620 |
| D(39,10) = 25179 | D(39,55) = 9317  | D(40,10) = 14234 | D(40,55) = 2228  |
| D(39,11) = 36421 | D(39,56) = 12914 | D(40,11) = 27478 | D(40,56) = 5709  |
| D(39,12) = 16296 | D(39,57) = 14118 | D(40,12) = 1782  | D(40,57) = 4028  |
| D(39,13) = 4851  | D(39,58) = 26869 | D(40,13) = 17978 | D(40,58) = 14234 |
| D(39,14) = 27874 | D(39,59) = 12936 | D(40,14) = 17545 | D(40,59) = 8261  |
| D(39,15) = 13244 | D(39,60) = 28871 | D(40,15) = 5133  | D(40,60) = 14388 |
| D(39,16) = 4158  | D(39,61) = 3795  | D(40,16) = 17359 | D(40,61) = 8785  |
| D(39,17) = 29044 | D(39,62) = 30877 | D(40,17) = 18623 | D(40,62) = 23655 |
| D(39,18) = 23019 | D(39,63) = 18942 | D(40,18) = 11853 | D(40,63) = 10715 |
| D(39,19) = 462   | D(39,64) = 5390  | D(40,19) = 15974 | D(40,64) = 7264  |
| D(39,20) = 12716 | D(39,65) = 6237  | D(40,20) = 3992  | D(40,65) = 19796 |
| D(39,21) = 6162  | D(39,66) = 25102 | D(40,21) = 5687  | D(40,66) = 14136 |
| D(39,22) = 18630 | D(39,67) = 26873 | D(40,22) = 12045 | D(40,67) = 16346 |
| D(39,23) = 18007 | D(39,68) = 38038 | D(40,23) = 11880 | D(40,68) = 32843 |
| D(39,24) = 26411 | D(39,69) = 33414 | D(40,24) = 16313 | D(40,69) = 19162 |
| D(39,25) = 14630 | D(39,70) = 17941 | D(40,25) = 10079 | D(40,70) = 16697 |
| D(39,26) = 16518 | D(39,71) = 30338 | D(40,26) = 9965  | D(40,71) = 23859 |
| D(39,27) = 1309  | D(39,72) = 41503 | D(40,27) = 16365 | D(40,72) = 32483 |
| D(39,28) = 11858 | D(39,73) = 16940 | D(40,28) = 3986  | D(40,73) = 8646  |
| D(39,29) = 10006 | D(39,74) = 27870 | D(40,29) = 2607  | D(40,74) = 19544 |
| D(39,30) = 17787 | D(39,75) = 28948 | D(40,30) = 14994 | D(40,75) = 20240 |
| D(39,31) = 25872 | D(39,76) = 13871 | D(40,31) = 14994 | D(40,76) = 4629  |
| D(39,32) = 28114 | D(39,77) = 3927  | D(40,32) = 17160 | D(40,77) = 13912 |
| D(39,33) = 12705 | D(39,78) = 36575 | D(40,33) = 4216  | D(40,78) = 29134 |
| D(39,34) = 2079  | D(39,79) = 18370 | D(40,34) = 17662 | D(40,79) = 13410 |
| D(39,35) = 14938 | D(39,80) = 9471  | D(40,35) = 10870 | D(40,80) = 2713  |
| D(39,36) = 33415 | D(39,81) = 18249 | D(40,36) = 19926 | D(40,81) = 9570  |
| D(39,37) = 10857 | D(39,82) = 27566 | D(40,37) = 5845  | D(40,82) = 17699 |
| D(39,38) = 9933  | D(39,83) = 20944 | D(40,38) = 7060  | D(40,83) = 15546 |
| D(39,39) = 0     | D(39,84) = 30565 | D(40,39) = 15765 | D(40,84) = 22242 |
| D(39,40) = 15765 | D(39,85) = 17013 | D(40,40) = 0     | D(40,85) = 10796 |
| D(39,41) = 26026 | D(39,86) = 27485 | D(40,41) = 25201 | D(40,86) = 16694 |
| D(39,42) = 24178 | D(39,87) = 37730 | D(40,42) = 22791 | D(40,87) = 25211 |
| D(39,43) = 33029 | D(39,88) = 5390  | D(40,43) = 18623 | D(40,88) = 13523 |
| D(39,44) = 18832 | D(39,89) = 6622  | D(40,44) = 11055 | D(40,89) = 7374  |
| D(39,45) = 24486 |                  | D(40,45) = 27126 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(41,1) = 7931   | D(41,46) = 13398 | D(42,1) = 8547   | D(42,46) = 12166 |
| D(41,2) = 22759  | D(41,47) = 10890 | D(42,2) = 20349  | D(42,47) = 9042  |
| D(41,3) = 12723  | D(41,48) = 4158  | D(42,3) = 10875  | D(42,48) = 2310  |
| D(41,4) = 20867  | D(41,49) = 26036 | D(42,4) = 19019  | D(42,49) = 23626 |
| D(41,5) = 12242  | D(41,50) = 6545  | D(42,5) = 10318  | D(42,50) = 6006  |
| D(41,6) = 3234   | D(41,51) = 11319 | D(42,6) = 5082   | D(42,51) = 14476 |
| D(41,7) = 2618   | D(41,52) = 12760 | D(42,7) = 2541   | D(42,52) = 13530 |
| D(41,8) = 3927   | D(41,53) = 11550 | D(42,8) = 1848   | D(42,53) = 10318 |
| D(41,9) = 17787  | D(41,54) = 15015 | D(42,9) = 15939  | D(42,54) = 18865 |
| D(41,10) = 10087 | D(41,55) = 17556 | D(42,10) = 10472 | D(42,55) = 15708 |
| D(41,11) = 13706 | D(41,56) = 14273 | D(42,11) = 16863 | D(42,56) = 12425 |
| D(41,12) = 23485 | D(41,57) = 25663 | D(42,12) = 21075 | D(42,57) = 23253 |
| D(41,13) = 30877 | D(41,58) = 10164 | D(42,13) = 29029 | D(42,58) = 10934 |
| D(41,14) = 14245 | D(41,59) = 15015 | D(42,14) = 19096 | D(42,59) = 13167 |
| D(41,15) = 17556 | D(41,60) = 1925  | D(42,15) = 15708 | D(42,60) = 4158  |
| D(41,16) = 30184 | D(41,61) = 27951 | D(42,16) = 28336 | D(42,61) = 26103 |
| D(41,17) = 7870  | D(41,62) = 8547  | D(42,17) = 7871  | D(42,62) = 6776  |
| D(41,18) = 5005  | D(41,63) = 12320 | D(42,18) = 6853  | D(42,63) = 10472 |
| D(41,19) = 26488 | D(41,64) = 26565 | D(42,19) = 24640 | D(42,64) = 24717 |
| D(41,20) = 17105 | D(41,65) = 32263 | D(42,20) = 15257 | D(42,65) = 30415 |
| D(41,21) = 21560 | D(41,66) = 2310  | D(42,21) = 19712 | D(42,66) = 2233  |
| D(41,22) = 9933  | D(41,67) = 4697  | D(42,22) = 8085  | D(42,67) = 4312  |
| D(41,23) = 11396 | D(41,68) = 12859 | D(42,23) = 9548  | D(42,68) = 11627 |
| D(41,24) = 7546  | D(41,69) = 8470  | D(42,24) = 8316  | D(42,69) = 11396 |
| D(41,25) = 9933  | D(41,70) = 17864 | D(42,25) = 9394  | D(42,70) = 17787 |
| D(41,26) = 12045 | D(41,71) = 13090 | D(42,26) = 10197 | D(42,71) = 17941 |
| D(41,27) = 27335 | D(41,72) = 19712 | D(42,27) = 25487 | D(42,72) = 22869 |
| D(41,28) = 18326 | D(41,73) = 9856  | D(42,28) = 16478 | D(42,73) = 8008  |
| D(41,29) = 27289 | D(41,74) = 4389  | D(42,29) = 24879 | D(42,74) = 2541  |
| D(41,30) = 10241 | D(41,75) = 4004  | D(42,30) = 10626 | D(42,75) = 6930  |
| D(41,31) = 2233  | D(41,76) = 18062 | D(42,31) = 4081  | D(42,76) = 16214 |
| D(41,32) = 3696  | D(41,77) = 31185 | D(42,32) = 5544  | D(42,77) = 29337 |
| D(41,33) = 14168 | D(41,78) = 10780 | D(42,33) = 12320 | D(42,78) = 9548  |
| D(41,34) = 28105 | D(41,79) = 9845  | D(42,34) = 26257 | D(42,79) = 7997  |
| D(41,35) = 13397 | D(41,80) = 23870 | D(42,35) = 11473 | D(42,80) = 22022 |
| D(41,36) = 9471  | D(41,81) = 13937 | D(42,36) = 12628 | D(42,81) = 12089 |
| D(41,37) = 27951 | D(41,82) = 6391  | D(42,37) = 26103 | D(42,82) = 7161  |
| D(41,38) = 26488 | D(41,83) = 14014 | D(42,38) = 24640 | D(42,83) = 17864 |
| D(41,39) = 26026 | D(41,84) = 4004  | D(42,39) = 24178 | D(42,84) = 6930  |
| D(41,40) = 25201 | D(41,85) = 11550 | D(42,40) = 22791 | D(42,85) = 9702  |
| D(41,41) = 0     | D(41,86) = 1771  | D(42,41) = 2310  | D(42,86) = 1386  |
| D(41,42) = 2310  | D(41,87) = 12474 | D(42,42) = 0     | D(42,87) = 15631 |
| D(41,43) = 8085  | D(41,88) = 31570 | D(42,43) = 11011 | D(42,88) = 29722 |
| D(41,44) = 12221 | D(41,89) = 23177 | D(42,44) = 10373 | D(42,89) = 21329 |
| D(41,45) = 1825  |                  | D(42,45) = 2849  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(43,1) = 5170   | D(43,46) = 5479  | D(44,1) = 14682  | D(44,46) = 20526 |
| D(43,2) = 16181  | D(43,47) = 21439 | D(44,2) = 8613   | D(44,47) = 2574  |
| D(43,3) = 16676  | D(43,48) = 12705 | D(44,3) = 502    | D(44,48) = 12067 |
| D(43,4) = 28486  | D(43,49) = 19458 | D(44,4) = 15994  | D(44,49) = 11890 |
| D(43,5) = 19250  | D(43,50) = 12397 | D(44,5) = 7326   | D(44,50) = 14058 |
| D(43,6) = 11319  | D(43,51) = 7777  | D(44,6) = 10604  | D(44,51) = 19448 |
| D(43,7) = 9933   | D(43,52) = 19459 | D(44,7) = 12144  | D(44,52) = 5643  |
| D(43,8) = 11858  | D(43,53) = 5390  | D(44,8) = 11220  | D(44,53) = 18678 |
| D(43,9) = 24563  | D(43,54) = 22869 | D(44,9) = 12914  | D(44,54) = 10285 |
| D(43,10) = 18480 | D(43,55) = 20944 | D(44,10) = 7590  | D(44,55) = 10362 |
| D(43,11) = 12251 | D(43,56) = 19580 | D(44,11) = 19910 | D(44,56) = 8870  |
| D(43,12) = 16907 | D(43,57) = 19085 | D(44,12) = 9339  | D(44,57) = 11517 |
| D(43,13) = 37572 | D(43,58) = 16709 | D(44,13) = 23683 | D(44,58) = 9207  |
| D(43,14) = 12089 | D(43,59) = 22253 | D(44,14) = 16599 | D(44,59) = 10329 |
| D(43,15) = 20559 | D(43,60) = 12089 | D(44,15) = 9834  | D(44,60) = 12287 |
| D(43,16) = 36879 | D(43,61) = 31720 | D(44,16) = 22990 | D(44,61) = 16896 |
| D(43,17) = 11396 | D(43,62) = 10010 | D(44,17) = 8668  | D(44,62) = 13354 |
| D(43,18) = 15246 | D(43,63) = 14784 | D(44,18) = 5588  | D(44,63) = 6765  |
| D(43,19) = 33491 | D(43,64) = 32105 | D(44,19) = 19294 | D(44,64) = 15587 |
| D(43,20) = 20007 | D(43,65) = 38958 | D(44,20) = 6871  | D(44,65) = 25069 |
| D(43,21) = 28366 | D(43,66) = 1095  | D(44,21) = 15675 | D(44,66) = 11682 |
| D(43,22) = 20482 | D(43,67) = 12320 | D(44,22) = 3066  | D(44,67) = 13222 |
| D(43,23) = 15389 | D(43,68) = 7931  | D(44,23) = 825   | D(44,68) = 19987 |
| D(43,24) = 14091 | D(43,69) = 4815  | D(44,24) = 7051  | D(44,69) = 16599 |
| D(43,25) = 20482 | D(43,70) = 23254 | D(44,25) = 4972  | D(44,70) = 10868 |
| D(43,26) = 22594 | D(43,71) = 10934 | D(44,26) = 1419  | D(44,71) = 15059 |
| D(43,27) = 34338 | D(43,72) = 17171 | D(44,27) = 20141 | D(44,72) = 25916 |
| D(43,28) = 21791 | D(43,73) = 12628 | D(44,28) = 8092  | D(44,73) = 4301  |
| D(43,29) = 13013 | D(43,74) = 8008  | D(44,29) = 12606 | D(44,74) = 10362 |
| D(43,30) = 20251 | D(43,75) = 12089 | D(44,30) = 7513  | D(44,75) = 9669  |
| D(43,31) = 16093 | D(43,76) = 20933 | D(44,31) = 7898  | D(44,76) = 10890 |
| D(43,32) = 14553 | D(43,77) = 38111 | D(44,32) = 7744  | D(44,77) = 23991 |
| D(43,33) = 19789 | D(43,78) = 5467  | D(44,33) = 9295  | D(44,78) = 17908 |
| D(43,34) = 35108 | D(43,79) = 13838 | D(44,34) = 20911 | D(44,79) = 2376  |
| D(43,35) = 18095 | D(43,80) = 30646 | D(44,35) = 6171  | D(44,80) = 17985 |
| D(43,36) = 3077  | D(43,81) = 16016 | D(44,36) = 17600 | D(44,81) = 8382  |
| D(43,37) = 32259 | D(43,82) = 12936 | D(44,37) = 13200 | D(44,82) = 8206  |
| D(43,38) = 31113 | D(43,83) = 21868 | D(44,38) = 13561 | D(44,83) = 9515  |
| D(43,39) = 33029 | D(43,84) = 10241 | D(44,39) = 18832 | D(44,84) = 11055 |
| D(43,40) = 18623 | D(43,85) = 22099 | D(44,40) = 11055 | D(44,85) = 1914  |
| D(43,41) = 8085  | D(43,86) = 10626 | D(44,41) = 12221 | D(44,86) = 10516 |
| D(43,42) = 11011 | D(43,87) = 9009  | D(44,42) = 10373 | D(44,87) = 20613 |
| D(43,43) = 0     | D(43,88) = 36417 | D(44,43) = 16174 | D(44,88) = 25685 |
| D(43,44) = 16174 | D(43,89) = 29983 | D(44,44) = 0     | D(44,89) = 17292 |
| D(43,45) = 12859 |                  | D(44,45) = 10681 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(45,1) = 7850   | D(45,46) = 13475 | D(46,1) = 7650   | D(46,46) = 0     |
| D(45,2) = 24864  | D(45,47) = 10350 | D(46,2) = 28927  | D(46,47) = 23056 |
| D(45,3) = 11183  | D(45,48) = 4312  | D(46,3) = 20988  | D(46,48) = 14245 |
| D(45,4) = 19327  | D(45,49) = 27961 | D(46,4) = 30184  | D(46,49) = 32204 |
| D(45,5) = 10164  | D(45,50) = 7084  | D(46,5) = 20328  | D(46,50) = 13013 |
| D(45,6) = 4774   | D(45,51) = 16324 | D(46,6) = 15323  | D(46,51) = 10164 |
| D(45,7) = 2849   | D(45,52) = 12375 | D(46,7) = 10780  | D(46,52) = 20537 |
| D(45,8) = 4004   | D(45,53) = 11627 | D(46,8) = 11935  | D(46,53) = 1848  |
| D(45,9) = 16247  | D(45,54) = 17325 | D(46,9) = 27104  | D(46,54) = 24486 |
| D(45,10) = 9702  | D(45,55) = 16016 | D(46,10) = 21927 | D(46,55) = 29722 |
| D(45,11) = 18711 | D(45,56) = 12733 | D(46,11) = 14784 | D(46,56) = 26439 |
| D(45,12) = 25410 | D(45,57) = 27588 | D(46,12) = 29653 | D(46,57) = 31831 |
| D(45,13) = 29337 | D(45,58) = 8624  | D(46,13) = 43043 | D(46,58) = 18326 |
| D(45,14) = 16863 | D(45,59) = 14475 | D(46,14) = 13706 | D(46,59) = 23331 |
| D(45,15) = 16016 | D(45,60) = 3619  | D(46,15) = 29722 | D(46,60) = 15631 |
| D(45,16) = 28644 | D(45,61) = 24563 | D(46,16) = 42350 | D(46,61) = 34261 |
| D(45,17) = 6331  | D(45,62) = 7931  | D(46,17) = 16011 | D(46,62) = 11165 |
| D(45,18) = 3927  | D(45,63) = 10780 | D(46,18) = 17479 | D(46,63) = 21483 |
| D(45,19) = 24948 | D(45,64) = 23177 | D(46,19) = 38654 | D(46,64) = 34646 |
| D(45,20) = 15565 | D(45,65) = 30723 | D(46,20) = 24959 | D(46,65) = 44429 |
| D(45,21) = 18172 | D(45,66) = 2541  | D(46,21) = 29183 | D(46,66) = 11242 |
| D(45,22) = 8393  | D(45,67) = 4774  | D(46,22) = 18711 | D(46,67) = 14399 |
| D(45,23) = 9856  | D(45,68) = 12936 | D(46,23) = 19701 | D(46,68) = 5313  |
| D(45,24) = 6006  | D(45,69) = 13244 | D(46,24) = 15708 | D(46,69) = 6699  |
| D(45,25) = 8624  | D(45,70) = 20328 | D(46,25) = 20002 | D(46,70) = 25025 |
| D(45,26) = 11505 | D(45,71) = 15708 | D(46,26) = 24211 | D(46,71) = 12551 |
| D(45,27) = 25795 | D(45,72) = 24717 | D(46,27) = 39501 | D(46,72) = 19866 |
| D(45,28) = 16786 | D(45,73) = 8316  | D(46,28) = 26180 | D(46,73) = 18788 |
| D(45,29) = 29214 | D(45,74) = 3696  | D(46,29) = 25759 | D(46,74) = 10164 |
| D(45,30) = 9856  | D(45,75) = 5621  | D(46,30) = 26393 | D(46,75) = 14091 |
| D(45,31) = 2079  | D(45,76) = 16522 | D(46,31) = 22042 | D(46,76) = 25916 |
| D(45,32) = 3000  | D(45,77) = 29645 | D(46,32) = 20482 | D(46,77) = 43351 |
| D(45,33) = 12628 | D(45,78) = 10857 | D(46,33) = 26334 | D(46,78) = 2618  |
| D(45,34) = 26565 | D(45,79) = 8305  | D(46,34) = 40271 | D(46,79) = 18150 |
| D(45,35) = 11319 | D(45,80) = 20482 | D(46,35) = 19173 | D(46,80) = 31493 |
| D(45,36) = 14476 | D(45,81) = 12397 | D(46,36) = 7238  | D(46,81) = 22946 |
| D(45,37) = 24563 | D(45,82) = 4851  | D(46,37) = 34800 | D(46,82) = 14553 |
| D(45,38) = 23100 | D(45,83) = 16324 | D(46,38) = 33645 | D(46,83) = 23485 |
| D(45,39) = 24486 | D(45,84) = 5621  | D(46,39) = 38192 | D(46,84) = 12628 |
| D(45,40) = 27126 | D(45,85) = 11010 | D(46,40) = 31393 | D(46,85) = 23716 |
| D(45,41) = 1825  | D(45,86) = 1925  | D(46,41) = 13398 | D(46,86) = 12089 |
| D(45,42) = 2849  | D(45,87) = 17479 | D(46,42) = 12166 | D(46,87) = 12012 |
| D(45,43) = 12859 | D(45,88) = 28182 | D(46,43) = 5479  | D(46,88) = 39193 |
| D(45,44) = 10681 | D(45,89) = 19789 | D(46,44) = 20526 | D(46,89) = 30800 |
| D(45,45) = 0     |                  | D(46,45) = 13475 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(47,1) = 15587  | D(47,46) = 23056 | D(48,1) = 9471   | D(48,46) = 14245 |
| D(47,2) = 8844   | D(47,47) = 0     | D(48,2) = 20824  | D(48,47) = 10736 |
| D(47,3) = 3036   | D(47,48) = 10736 | D(48,3) = 12569  | D(48,48) = 0     |
| D(47,4) = 13130  | D(47,49) = 12121 | D(48,4) = 20713  | D(48,49) = 24101 |
| D(47,5) = 4345   | D(47,50) = 13893 | D(48,5) = 12474  | D(48,50) = 7854  |
| D(47,6) = 9812   | D(47,51) = 21362 | D(48,6) = 7007   | D(48,51) = 16170 |
| D(47,7) = 10428  | D(47,52) = 4554  | D(48,7) = 5467   | D(48,52) = 15224 |
| D(47,8) = 9889   | D(47,53) = 21131 | D(48,8) = 1694   | D(48,53) = 12397 |
| D(47,9) = 11121  | D(47,54) = 8195  | D(48,9) = 17633  | D(48,54) = 21329 |
| D(47,10) = 7117  | D(47,55) = 9372  | D(48,10) = 13783 | D(48,55) = 17402 |
| D(47,11) = 21285 | D(47,56) = 6138  | D(48,11) = 18557 | D(48,56) = 14119 |
| D(47,12) = 9570  | D(47,57) = 11583 | D(48,12) = 21550 | D(48,57) = 23728 |
| D(47,13) = 22216 | D(47,58) = 8734  | D(48,13) = 30723 | D(48,58) = 12628 |
| D(47,14) = 13735 | D(47,59) = 7348  | D(48,14) = 22715 | D(48,59) = 14861 |
| D(47,15) = 8580  | D(47,60) = 11198 | D(48,15) = 17402 | D(48,60) = 6699  |
| D(47,16) = 21523 | D(47,61) = 16364 | D(48,16) = 30030 | D(48,61) = 27797 |
| D(47,17) = 9504  | D(47,62) = 15275 | D(48,17) = 10335 | D(48,62) = 8624  |
| D(47,18) = 5346  | D(47,63) = 4037  | D(48,18) = 8547  | D(48,63) = 12166 |
| D(47,19) = 18135 | D(47,64) = 16745 | D(48,19) = 26334 | D(48,64) = 26411 |
| D(47,20) = 5181  | D(47,65) = 23602 | D(48,20) = 16951 | D(48,65) = 32109 |
| D(47,21) = 11440 | D(47,66) = 9965  | D(48,21) = 21406 | D(48,66) = 5159  |
| D(47,22) = 957   | D(47,67) = 11737 | D(48,22) = 9779  | D(48,67) = 6314  |
| D(47,23) = 1760  | D(47,68) = 22902 | D(48,23) = 11242 | D(48,68) = 13706 |
| D(47,24) = 6809  | D(47,69) = 21516 | D(48,24) = 10010 | D(48,69) = 13090 |
| D(47,25) = 2651  | D(47,70) = 11572 | D(48,25) = 12715 | D(48,70) = 20636 |
| D(47,26) = 1155  | D(47,71) = 13504 | D(48,26) = 11891 | D(48,71) = 21560 |
| D(47,27) = 18982 | D(47,72) = 26367 | D(48,27) = 27181 | D(48,72) = 24563 |
| D(47,28) = 6435  | D(47,73) = 1881  | D(48,28) = 18172 | D(48,73) = 9702  |
| D(47,29) = 9966  | D(47,74) = 10274 | D(48,29) = 25434 | D(48,74) = 3773  |
| D(47,30) = 5808  | D(47,75) = 9346  | D(48,30) = 13947 | D(48,75) = 8855  |
| D(47,31) = 6270  | D(47,76) = 5577  | D(48,31) = 7392  | D(48,76) = 17908 |
| D(47,32) = 7964  | D(47,77) = 22755 | D(48,32) = 8008  | D(48,77) = 31031 |
| D(47,33) = 7733  | D(47,78) = 21439 | D(48,33) = 14014 | D(48,78) = 11627 |
| D(47,34) = 19752 | D(47,79) = 1078  | D(48,34) = 27951 | D(48,79) = 9691  |
| D(47,35) = 3190  | D(47,80) = 12518 | D(48,35) = 13629 | D(48,80) = 23716 |
| D(47,36) = 19437 | D(47,81) = 5269  | D(48,36) = 14322 | D(48,81) = 13783 |
| D(47,37) = 16903 | D(47,82) = 7964  | D(48,37) = 27797 | D(48,82) = 8855  |
| D(47,38) = 15720 | D(47,83) = 8272  | D(48,38) = 26334 | D(48,83) = 20328 |
| D(47,39) = 17613 | D(47,84) = 10963 | D(48,39) = 25872 | D(48,84) = 8851  |
| D(47,40) = 11286 | D(47,85) = 660   | D(48,40) = 23266 | D(48,85) = 11396 |
| D(47,41) = 10890 | D(47,86) = 9196  | D(48,41) = 4158  | D(48,86) = 3850  |
| D(47,42) = 9042  | D(47,87) = 22594 | D(48,42) = 2310  | D(48,87) = 17325 |
| D(47,43) = 21439 | D(47,88) = 21061 | D(48,43) = 12705 | D(48,88) = 31416 |
| D(47,44) = 2574  | D(47,89) = 12822 | D(48,44) = 12067 | D(48,89) = 23023 |
| D(47,45) = 10350 |                  | D(48,45) = 4312  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(49,1) = 18919  | D(49,46) = 32204 | D(50,1) = 1771   | D(50,46) = 13013 |
| D(49,2) = 3036   | D(49,47) = 12121 | D(50,2) = 17274  | D(50,47) = 13893 |
| D(49,3) = 11305  | D(49,48) = 24101 | D(50,3) = 14520  | D(50,48) = 7854  |
| D(49,4) = 9769   | D(49,49) = 0     | D(50,4) = 22460  | D(50,49) = 20551 |
| D(49,5) = 12993  | D(49,50) = 20551 | D(50,5) = 13941  | D(50,50) = 0     |
| D(49,6) = 16565  | D(49,51) = 27191 | D(50,6) = 7315   | D(50,51) = 8624  |
| D(49,7) = 14596  | D(49,52) = 8986  | D(50,7) = 4386   | D(50,52) = 14150 |
| D(49,8) = 22186  | D(49,53) = 32874 | D(50,8) = 5929   | D(50,53) = 11704 |
| D(49,9) = 11021  | D(49,54) = 16455 | D(50,9) = 20717  | D(50,54) = 18480 |
| D(49,10) = 15069 | D(49,55) = 2475  | D(50,10) = 14553 | D(50,55) = 20559 |
| D(49,11) = 28313 | D(49,56) = 6270  | D(50,11) = 13398 | D(50,56) = 17276 |
| D(49,12) = 2541  | D(49,57) = 4861  | D(50,12) = 18000 | D(50,57) = 20178 |
| D(49,13) = 16303 | D(49,58) = 15069 | D(50,13) = 33726 | D(50,58) = 14091 |
| D(49,14) = 18380 | D(49,59) = 9096  | D(50,14) = 21329 | D(50,59) = 16944 |
| D(49,15) = 8360  | D(49,60) = 15223 | D(50,15) = 20559 | D(50,60) = 7931  |
| D(49,16) = 15561 | D(49,61) = 6974  | D(50,16) = 33033 | D(50,61) = 27874 |
| D(49,17) = 19458 | D(49,62) = 24694 | D(50,17) = 8778  | D(50,62) = 3003  |
| D(49,18) = 14299 | D(49,63) = 11560 | D(50,18) = 9240  | D(50,63) = 15015 |
| D(49,19) = 14205 | D(49,64) = 8459  | D(50,19) = 29645 | D(50,64) = 28259 |
| D(49,20) = 7920  | D(49,65) = 18074 | D(50,20) = 19646 | D(50,65) = 35112 |
| D(49,21) = 1463  | D(49,66) = 14971 | D(50,21) = 23793 | D(50,66) = 4851  |
| D(49,22) = 12880 | D(49,67) = 17181 | D(50,22) = 12936 | D(50,67) = 2156  |
| D(49,23) = 12715 | D(49,68) = 33644 | D(50,23) = 13233 | D(50,68) = 13013 |
| D(49,24) = 17148 | D(49,69) = 19997 | D(50,24) = 11473 | D(50,69) = 12782 |
| D(49,25) = 10911 | D(49,70) = 17532 | D(50,25) = 13090 | D(50,70) = 20328 |
| D(49,26) = 10831 | D(49,71) = 24694 | D(50,26) = 15048 | D(50,71) = 20174 |
| D(49,27) = 14544 | D(49,72) = 33318 | D(50,27) = 30492 | D(50,72) = 18711 |
| D(49,28) = 4785  | D(49,73) = 4981  | D(50,28) = 20867 | D(50,73) = 12859 |
| D(49,29) = 1881  | D(49,74) = 20614 | D(50,29) = 14106 | D(50,74) = 4389  |
| D(49,30) = 13914 | D(49,75) = 21075 | D(50,30) = 16247 | D(50,75) = 8162  |
| D(49,31) = 15829 | D(49,76) = 6435  | D(50,31) = 9163  | D(50,76) = 20603 |
| D(49,32) = 17995 | D(49,77) = 12003 | D(50,32) = 8624  | D(50,77) = 34265 |
| D(49,33) = 6974  | D(49,78) = 30084 | D(50,33) = 17171 | D(50,78) = 11011 |
| D(49,34) = 15699 | D(49,79) = 13900 | D(50,34) = 31262 | D(50,79) = 11682 |
| D(49,35) = 11791 | D(49,80) = 1925  | D(50,35) = 12786 | D(50,80) = 26103 |
| D(49,36) = 20767 | D(49,81) = 10405 | D(50,36) = 8470  | D(50,81) = 16247 |
| D(49,37) = 3994  | D(49,82) = 18534 | D(50,37) = 28413 | D(50,82) = 10318 |
| D(49,38) = 5145  | D(49,83) = 16381 | D(50,38) = 27258 | D(50,83) = 17479 |
| D(49,39) = 13741 | D(49,84) = 23077 | D(50,39) = 29183 | D(50,84) = 6776  |
| D(49,40) = 831   | D(49,85) = 11631 | D(50,40) = 19483 | D(50,85) = 14553 |
| D(49,41) = 26036 | D(49,86) = 17764 | D(50,41) = 6545  | D(50,86) = 5313  |
| D(49,42) = 23626 | D(49,87) = 26046 | D(50,42) = 6006  | D(50,87) = 9856  |
| D(49,43) = 19458 | D(49,88) = 11554 | D(50,43) = 12397 | D(50,88) = 32571 |
| D(49,44) = 11890 | D(49,89) = 2464  | D(50,44) = 14058 | D(50,89) = 25410 |
| D(49,45) = 27961 |                  | D(50,45) = 7084  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(51,1) = 6776   | D(51,46) = 10164 | D(52,1) = 16577  | D(52,46) = 20537 |
| D(51,2) = 23914  | D(51,47) = 21362 | D(52,2) = 5709   | D(52,47) = 4554  |
| D(51,3) = 19910  | D(51,48) = 16170 | D(52,3) = 4983   | D(52,48) = 15224 |
| D(51,4) = 31955  | D(51,49) = 27191 | D(52,4) = 12452  | D(52,49) = 8986  |
| D(51,5) = 21329  | D(51,50) = 8624  | D(52,5) = 3432   | D(52,50) = 14150 |
| D(51,6) = 9856   | D(51,51) = 0     | D(52,6) = 14069  | D(52,51) = 21538 |
| D(51,7) = 13860  | D(51,52) = 21538 | D(52,7) = 15224  | D(52,52) = 0     |
| D(51,8) = 15323  | D(51,53) = 8316  | D(52,8) = 14377  | D(52,53) = 18689 |
| D(51,9) = 28875  | D(51,54) = 19712 | D(52,9) = 9372   | D(52,54) = 10835 |
| D(51,10) = 16247 | D(51,55) = 28028 | D(52,10) = 9757  | D(52,55) = 7095  |
| D(51,11) = 5387  | D(51,56) = 24745 | D(52,11) = 23155 | D(52,56) = 3498  |
| D(51,12) = 24640 | D(51,57) = 26818 | D(52,12) = 6435  | D(52,57) = 8613  |
| D(51,13) = 41349 | D(51,58) = 16632 | D(52,13) = 20416 | D(52,58) = 11374 |
| D(51,14) = 10472 | D(51,59) = 24322 | D(52,14) = 16709 | D(52,59) = 6138  |
| D(51,15) = 28028 | D(51,60) = 10318 | D(52,15) = 7194  | D(52,60) = 13838 |
| D(51,16) = 40656 | D(51,61) = 33799 | D(52,16) = 19723 | D(52,61) = 20247 |
| D(51,17) = 14337 | D(51,62) = 7315  | D(52,17) = 12144 | D(52,62) = 14806 |
| D(51,18) = 18865 | D(51,63) = 16863 | D(52,18) = 7986  | D(52,63) = 6842  |
| D(51,19) = 36960 | D(51,64) = 34184 | D(52,19) = 16027 | D(52,64) = 18861 |
| D(51,20) = 27269 | D(51,65) = 42735 | D(52,20) = 3795  | D(52,65) = 21802 |
| D(51,21) = 30723 | D(51,66) = 14322 | D(52,21) = 13860 | D(52,66) = 15532 |
| D(51,22) = 20405 | D(51,67) = 11165 | D(52,22) = 3597  | D(52,67) = 15224 |
| D(51,23) = 18623 | D(51,68) = 10395 | D(52,23) = 6369  | D(52,68) = 21076 |
| D(51,24) = 14014 | D(51,69) = 7623  | D(52,24) = 10689 | D(52,69) = 19844 |
| D(51,25) = 17711 | D(51,70) = 20790 | D(52,25) = 5291  | D(52,70) = 10824 |
| D(51,26) = 22517 | D(51,71) = 9317  | D(52,26) = 5709  | D(52,71) = 20305 |
| D(51,27) = 37807 | D(51,72) = 9625  | D(52,27) = 16874 | D(52,72) = 29161 |
| D(51,28) = 28490 | D(51,73) = 14707 | D(52,28) = 4309  | D(52,73) = 4378  |
| D(51,29) = 20746 | D(51,74) = 11550 | D(52,29) = 7920  | D(52,74) = 12837 |
| D(51,30) = 17403 | D(51,75) = 11319 | D(52,30) = 8448  | D(52,75) = 11990 |
| D(51,31) = 13321 | D(51,76) = 28226 | D(52,31) = 10073 | D(52,76) = 3762  |
| D(51,32) = 12474 | D(51,77) = 41657 | D(52,32) = 12075 | D(52,77) = 20724 |
| D(51,33) = 24640 | D(51,78) = 8316  | D(52,33) = 5753  | D(52,78) = 18997 |
| D(51,34) = 38577 | D(51,79) = 17072 | D(52,34) = 17644 | D(52,79) = 4818  |
| D(51,35) = 20174 | D(51,80) = 33033 | D(52,35) = 3135  | D(52,80) = 16170 |
| D(51,36) = 2849  | D(51,81) = 18095 | D(52,36) = 20845 | D(52,81) = 5852  |
| D(51,37) = 34338 | D(51,82) = 12859 | D(52,37) = 20247 | D(52,82) = 11844 |
| D(51,38) = 33183 | D(51,83) = 18711 | D(52,38) = 18784 | D(52,83) = 10912 |
| D(51,39) = 36498 | D(51,84) = 9394  | D(52,39) = 15565 | D(52,84) = 13607 |
| D(51,40) = 26356 | D(51,85) = 22022 | D(52,40) = 8115  | D(52,85) = 5214  |
| D(51,41) = 11319 | D(51,86) = 13783 | D(52,41) = 12760 | D(52,86) = 11836 |
| D(51,42) = 14476 | D(51,87) = 1155  | D(52,42) = 13530 | D(52,87) = 22693 |
| D(51,43) = 7777  | D(51,88) = 40733 | D(52,43) = 19459 | D(52,88) = 23870 |
| D(51,44) = 19448 | D(51,89) = 32340 | D(52,44) = 5643  | D(52,89) = 15477 |
| D(51,45) = 16324 |                  | D(52,45) = 12375 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(53,1) = 7700   | D(53,46) = 1848  | D(54,1) = 9044   | D(54,46) = 24486 |
| D(53,2) = 29597  | D(53,47) = 21131 | D(54,2) = 13178  | D(54,47) = 8195  |
| D(53,3) = 19140  | D(53,48) = 12397 | D(54,3) = 10747  | D(54,48) = 21329 |
| D(53,4) = 28336  | D(53,49) = 32874 | D(54,4) = 15092  | D(54,49) = 16455 |
| D(53,5) = 18480  | D(53,50) = 11704 | D(54,5) = 10626  | D(54,50) = 18480 |
| D(53,6) = 13475  | D(53,51) = 8316  | D(54,6) = 12859  | D(54,51) = 19712 |
| D(53,7) = 8932   | D(53,52) = 18689 | D(54,7) = 18403  | D(54,52) = 10835 |
| D(53,8) = 10087  | D(53,53) = 0     | D(54,8) = 20482  | D(54,53) = 22561 |
| D(53,9) = 25256  | D(53,54) = 22561 | D(54,9) = 13013  | D(54,54) = 0     |
| D(53,10) = 20002 | D(53,55) = 27797 | D(54,10) = 7007  | D(54,55) = 15653 |
| D(53,11) = 12936 | D(53,56) = 24478 | D(54,11) = 24178 | D(54,56) = 12419 |
| D(53,12) = 30323 | D(53,57) = 32501 | D(54,12) = 13904 | D(54,57) = 19778 |
| D(53,13) = 41118 | D(53,58) = 16401 | D(54,13) = 25872 | D(54,58) = 11825 |
| D(53,14) = 11858 | D(53,59) = 21483 | D(54,14) = 10164 | D(54,59) = 6468  |
| D(53,15) = 27797 | D(53,60) = 14091 | D(54,15) = 11704 | D(54,60) = 6115  |
| D(53,16) = 40425 | D(53,61) = 32413 | D(54,16) = 25179 | D(54,61) = 19558 |
| D(53,17) = 14086 | D(53,62) = 9625  | D(54,17) = 11550 | D(54,62) = 16247 |
| D(53,18) = 15631 | D(53,63) = 19635 | D(54,18) = 9625  | D(54,63) = 4928  |
| D(53,19) = 36729 | D(53,64) = 32798 | D(54,19) = 21329 | D(54,64) = 19943 |
| D(53,20) = 23650 | D(53,65) = 42504 | D(54,20) = 13376 | D(54,65) = 27258 |
| D(53,21) = 27708 | D(53,66) = 9394  | D(54,21) = 14630 | D(54,66) = 17941 |
| D(53,22) = 16786 | D(53,67) = 12551 | D(54,22) = 7238  | D(54,67) = 19789 |
| D(53,23) = 17853 | D(53,68) = 3465  | D(54,23) = 9460  | D(54,68) = 24332 |
| D(53,24) = 13783 | D(53,69) = 5236  | D(54,24) = 9625  | D(54,69) = 23177 |
| D(53,25) = 18077 | D(53,70) = 23100 | D(54,25) = 8855  | D(54,70) = 2772  |
| D(53,26) = 22286 | D(53,71) = 10703 | D(54,26) = 9350  | D(54,71) = 9009  |
| D(53,27) = 37576 | D(53,72) = 18018 | D(54,27) = 22176 | D(54,72) = 29260 |
| D(53,28) = 24871 | D(53,73) = 16940 | D(54,28) = 14630 | D(54,73) = 6160  |
| D(53,29) = 26429 | D(53,74) = 8855  | D(54,29) = 10010 | D(54,74) = 22561 |
| D(53,30) = 24468 | D(53,75) = 12166 | D(54,30) = 2926  | D(54,75) = 11396 |
| D(53,31) = 20117 | D(53,76) = 24607 | D(54,31) = 12166 | D(54,76) = 13772 |
| D(53,32) = 18557 | D(53,77) = 41426 | D(54,32) = 13629 | D(54,77) = 26642 |
| D(53,33) = 24409 | D(53,78) = 770   | D(54,33) = 8547  | D(54,78) = 22869 |
| D(53,34) = 38346 | D(53,79) = 16302 | D(54,34) = 22946 | D(54,79) = 7909  |
| D(53,35) = 17325 | D(53,80) = 30018 | D(54,35) = 9471  | D(54,80) = 20713 |
| D(53,36) = 5852  | D(53,81) = 21098 | D(54,36) = 18711 | D(54,81) = 9317  |
| D(53,37) = 32952 | D(53,82) = 12628 | D(54,37) = 20097 | D(54,82) = 10010 |
| D(53,38) = 31797 | D(53,83) = 21560 | D(54,38) = 18942 | D(54,83) = 1078  |
| D(53,39) = 36267 | D(53,84) = 10703 | D(54,39) = 20867 | D(54,84) = 13013 |
| D(53,40) = 32105 | D(53,85) = 21791 | D(54,40) = 15620 | D(54,85) = 8855  |
| D(53,41) = 11550 | D(53,86) = 10164 | D(54,41) = 15015 | D(54,86) = 19712 |
| D(53,42) = 10318 | D(53,87) = 10164 | D(54,42) = 18865 | D(54,87) = 20944 |
| D(53,43) = 5390  | D(53,88) = 37718 | D(54,43) = 22869 | D(54,88) = 24332 |
| D(53,44) = 18678 | D(53,89) = 29325 | D(54,44) = 10285 | D(54,89) = 16624 |
| D(53,45) = 11627 |                  | D(54,45) = 17325 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(55,1) = 17854  | D(55,46) = 29722 | D(56,1) = 18227  | D(56,46) = 26439 |
| D(55,2) = 3003   | D(55,47) = 9372  | D(56,2) = 6699   | D(56,47) = 6138  |
| D(55,3) = 10131  | D(55,48) = 17402 | D(56,3) = 8415   | D(56,48) = 14119 |
| D(55,4) = 9548   | D(55,49) = 2475  | D(56,4) = 13145  | D(56,49) = 6270  |
| D(55,5) = 7623   | D(55,50) = 20559 | D(56,5) = 4125   | D(56,50) = 17276 |
| D(55,6) = 16478  | D(55,51) = 28028 | D(56,6) = 13195  | D(56,51) = 24745 |
| D(55,7) = 17094  | D(55,52) = 7095  | D(56,7) = 13811  | D(56,52) = 3498  |
| D(55,8) = 16555  | D(55,53) = 27797 | D(56,8) = 13272  | D(56,53) = 24478 |
| D(55,9) = 7392   | D(55,54) = 15653 | D(56,9) = 6369   | D(56,54) = 12419 |
| D(55,10) = 13860 | D(55,55) = 0     | D(56,10) = 11341 | D(56,55) = 3597  |
| D(55,11) = 27951 | D(55,56) = 3597  | D(56,11) = 24668 | D(56,56) = 0     |
| D(55,12) = 3729  | D(55,57) = 5907  | D(56,12) = 7425  | D(56,57) = 9603  |
| D(55,13) = 13321 | D(55,58) = 16192 | D(56,13) = 16918 | D(56,58) = 12958 |
| D(55,14) = 25090 | D(55,59) = 6622  | D(56,14) = 21460 | D(56,59) = 3839  |
| D(55,15) = 3927  | D(55,60) = 18656 | D(56,15) = 3795  | D(56,60) = 15422 |
| D(55,16) = 12628 | D(55,61) = 9471  | D(56,16) = 16225 | D(56,61) = 13068 |
| D(55,17) = 16962 | D(55,62) = 21945 | D(56,17) = 13728 | D(56,62) = 18662 |
| D(55,18) = 12089 | D(55,63) = 10472 | D(56,18) = 8575  | D(56,63) = 8190  |
| D(55,19) = 9779  | D(55,64) = 8162  | D(56,19) = 13376 | D(56,64) = 11759 |
| D(55,20) = 3399  | D(55,65) = 14707 | D(56,20) = 5379  | D(56,65) = 18304 |
| D(55,21) = 2441  | D(55,66) = 16632 | D(56,21) = 10017 | D(56,66) = 13349 |
| D(55,22) = 8415  | D(55,67) = 18403 | D(56,22) = 5181  | D(56,67) = 15120 |
| D(55,23) = 9537  | D(55,68) = 29568 | D(56,23) = 8052  | D(56,68) = 26285 |
| D(55,24) = 14267 | D(55,69) = 21329 | D(56,24) = 11033 | D(56,69) = 19965 |
| D(55,25) = 10109 | D(55,70) = 15950 | D(56,25) = 6875  | D(56,70) = 13376 |
| D(55,26) = 10527 | D(55,71) = 23935 | D(56,26) = 7293  | D(56,71) = 20305 |
| D(55,27) = 10626 | D(55,72) = 33033 | D(56,27) = 14223 | D(56,72) = 29750 |
| D(55,28) = 2541  | D(55,73) = 8470  | D(56,28) = 4158  | D(56,73) = 5110  |
| D(55,29) = 990   | D(55,74) = 16940 | D(56,29) = 4455  | D(56,74) = 13657 |
| D(55,30) = 13266 | D(55,75) = 16804 | D(56,30) = 10032 | D(56,75) = 13570 |
| D(55,31) = 13728 | D(55,76) = 4554  | D(56,31) = 10494 | D(56,76) = 5214  |
| D(55,32) = 15422 | D(55,77) = 14399 | D(56,32) = 12188 | D(56,77) = 17996 |
| D(55,33) = 3388  | D(55,78) = 28105 | D(56,33) = 3113  | D(56,78) = 24822 |
| D(55,34) = 11396 | D(55,79) = 9900  | D(56,34) = 14993 | D(56,79) = 6270  |
| D(55,35) = 6468  | D(55,80) = 5929  | D(56,35) = 2970  | D(56,80) = 10010 |
| D(55,36) = 22330 | D(55,81) = 9779  | D(56,36) = 20966 | D(56,81) = 6650  |
| D(55,37) = 5775  | D(55,82) = 15422 | D(56,37) = 9372  | D(56,82) = 12188 |
| D(55,38) = 5236  | D(55,83) = 15730 | D(56,38) = 8833  | D(56,83) = 12496 |
| D(55,39) = 9317  | D(55,84) = 18425 | D(56,39) = 12914 | D(56,84) = 15191 |
| D(55,40) = 2228  | D(55,85) = 10032 | D(56,40) = 5709  | D(56,85) = 6798  |
| D(55,41) = 17556 | D(55,86) = 15862 | D(56,41) = 14273 | D(56,86) = 12579 |
| D(55,42) = 15708 | D(55,87) = 29260 | D(56,42) = 12425 | D(56,87) = 25977 |
| D(55,43) = 20944 | D(55,88) = 12705 | D(56,43) = 19580 | D(56,88) = 16302 |
| D(55,44) = 10362 | D(55,89) = 4158  | D(56,44) = 8870  | D(56,89) = 11634 |
| D(55,45) = 16016 |                  | D(56,45) = 12733 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(57,1) = 26312  | D(57,46) = 31831 | D(58,1) = 12203  | D(58,46) = 18326 |
| D(57,2) = 3036   | D(57,47) = 11583 | D(58,2) = 11792  | D(58,47) = 8734  |
| D(57,3) = 10932  | D(57,48) = 23728 | D(58,3) = 9669   | D(58,48) = 12628 |
| D(57,4) = 14498  | D(57,49) = 4861  | D(58,4) = 22326  | D(58,49) = 15069 |
| D(57,5) = 12620  | D(57,50) = 20178 | D(58,5) = 13398  | D(58,50) = 14091 |
| D(57,6) = 16192  | D(57,51) = 26818 | D(58,6) = 6314   | D(58,51) = 16632 |
| D(57,7) = 14223  | D(57,52) = 8613  | D(58,7) = 13013  | D(58,52) = 11374 |
| D(57,8) = 21813  | D(57,53) = 32501 | D(58,8) = 11781  | D(58,53) = 16401 |
| D(57,9) = 10648  | D(57,54) = 19778 | D(58,9) = 18481  | D(58,54) = 11825 |
| D(57,10) = 18700 | D(57,55) = 5907  | D(58,10) = 3137  | D(58,55) = 16192 |
| D(57,11) = 27940 | D(57,56) = 9603  | D(58,11) = 17479 | D(58,56) = 12958 |
| D(57,12) = 2310  | D(57,57) = 0     | D(58,12) = 12518 | D(58,57) = 20317 |
| D(57,13) = 16676 | D(57,58) = 20317 | D(58,13) = 31412 | D(58,58) = 0     |
| D(57,14) = 25322 | D(57,59) = 8723  | D(58,14) = 11319 | D(58,59) = 16401 |
| D(57,15) = 8723  | D(57,60) = 14850 | D(58,15) = 17706 | D(58,60) = 6776  |
| D(57,16) = 16060 | D(57,61) = 7347  | D(58,16) = 30719 | D(58,61) = 25560 |
| D(57,17) = 21087 | D(57,62) = 24321 | D(58,17) = 4592  | D(58,62) = 15015 |
| D(57,18) = 13926 | D(57,63) = 11187 | D(58,18) = 7777  | D(58,63) = 8855  |
| D(57,19) = 14578 | D(57,64) = 5137  | D(58,19) = 27331 | D(58,64) = 25945 |
| D(57,20) = 6369  | D(57,65) = 18447 | D(58,20) = 13915 | D(58,65) = 32798 |
| D(57,21) = 8063  | D(57,66) = 14598 | D(58,21) = 20636 | D(58,66) = 12551 |
| D(57,22) = 12540 | D(57,67) = 16808 | D(58,22) = 7777  | D(58,67) = 14399 |
| D(57,23) = 12342 | D(57,68) = 33271 | D(58,23) = 8362  | D(58,68) = 18172 |
| D(57,24) = 18392 | D(57,69) = 19624 | D(58,24) = 3850  | D(58,69) = 16789 |
| D(57,25) = 14234 | D(57,70) = 23320 | D(58,25) = 8085  | D(58,70) = 11935 |
| D(57,26) = 10428 | D(57,71) = 25091 | D(58,26) = 9889  | D(58,71) = 10164 |
| D(57,27) = 14917 | D(57,72) = 32945 | D(58,27) = 6699  | D(58,72) = 22561 |
| D(57,28) = 6468  | D(57,73) = 9108  | D(58,28) = 15169 | D(58,73) = 8932  |
| D(57,29) = 4059  | D(57,74) = 20241 | D(58,29) = 8624  | D(58,74) = 11935 |
| D(57,30) = 17391 | D(57,75) = 20929 | D(58,30) = 7700  | D(58,75) = 6545  |
| D(57,31) = 17853 | D(57,76) = 5082  | D(58,31) = 6083  | D(58,76) = 14311 |
| D(57,32) = 19547 | D(57,77) = 12376 | D(58,32) = 7623  | D(58,77) = 31951 |
| D(57,33) = 7337  | D(57,78) = 29711 | D(58,33) = 16628 | D(58,78) = 16709 |
| D(57,34) = 16072 | D(57,79) = 13827 | D(58,34) = 28948 | D(58,79) = 6811  |
| D(57,35) = 11418 | D(57,80) = 4748  | D(58,35) = 12243 | D(58,80) = 19789 |
| D(57,36) = 20394 | D(57,81) = 10032 | D(58,36) = 14707 | D(58,81) = 12320 |
| D(57,37) = 4367  | D(57,82) = 19547 | D(58,37) = 26099 | D(58,82) = 5159  |
| D(57,38) = 5522  | D(57,83) = 19855 | D(58,38) = 24944 | D(58,83) = 10824 |
| D(57,39) = 14118 | D(57,84) = 22546 | D(58,39) = 26869 | D(58,84) = 8162  |
| D(57,40) = 4028  | D(57,85) = 10923 | D(58,40) = 14234 | D(58,85) = 9394  |
| D(57,41) = 25663 | D(57,86) = 17391 | D(58,41) = 10164 | D(58,86) = 9625  |
| D(57,42) = 23253 | D(57,87) = 25673 | D(58,42) = 10934 | D(58,87) = 17864 |
| D(57,43) = 19085 | D(57,88) = 11927 | D(58,43) = 16709 | D(58,88) = 30257 |
| D(57,44) = 11517 | D(57,89) = 9774  | D(58,44) = 9207  | D(58,89) = 22018 |
| D(57,45) = 27588 |                  | D(58,45) = 8624  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(59,1) = 19000  | D(59,46) = 23331 | D(60,1) = 5390   | D(60,46) = 15631 |
| D(59,2) = 5819   | D(59,47) = 7348  | D(60,2) = 11946  | D(60,47) = 11198 |
| D(59,3) = 10831  | D(59,48) = 14861 | D(60,3) = 12749  | D(60,48) = 6699  |
| D(59,4) = 6414   | D(59,49) = 9096  | D(60,4) = 24328  | D(60,49) = 15223 |
| D(59,5) = 2079   | D(59,50) = 16944 | D(60,5) = 13629  | D(60,50) = 7931  |
| D(59,6) = 14704  | D(59,51) = 24322 | D(60,6) = 1386   | D(60,51) = 10318 |
| D(59,7) = 15092  | D(59,52) = 6138  | D(60,7) = 4774   | D(60,52) = 13838 |
| D(59,8) = 14014  | D(59,53) = 21483 | D(60,8) = 5852   | D(60,53) = 14091 |
| D(59,9) = 5852   | D(59,54) = 6468  | D(60,9) = 20405  | D(60,54) = 6115  |
| D(59,10) = 10703 | D(59,55) = 6622  | D(60,10) = 8316  | D(60,55) = 18656 |
| D(59,11) = 25949 | D(59,56) = 3839  | D(60,11) = 13167 | D(60,56) = 15422 |
| D(59,12) = 6545  | D(59,57) = 8723  | D(60,12) = 12672 | D(60,57) = 14850 |
| D(59,13) = 17787 | D(59,58) = 16401 | D(60,13) = 33414 | D(60,58) = 6776  |
| D(59,14) = 15545 | D(59,59) = 0     | D(60,14) = 15477 | D(60,59) = 16632 |
| D(59,15) = 5544  | D(59,60) = 16632 | D(60,15) = 17864 | D(60,60) = 0     |
| D(59,16) = 17094 | D(59,61) = 16320 | D(60,16) = 32721 | D(60,61) = 27562 |
| D(59,17) = 16418 | D(59,62) = 19635 | D(60,17) = 5121  | D(60,62) = 9856  |
| D(59,18) = 10780 | D(59,63) = 8239  | D(60,18) = 8468  | D(60,63) = 15246 |
| D(59,19) = 13398 | D(59,64) = 14934 | D(60,19) = 29333 | D(60,64) = 27947 |
| D(59,20) = 8470  | D(59,65) = 19173 | D(60,20) = 16378 | D(60,65) = 34800 |
| D(59,21) = 9933  | D(59,66) = 15400 | D(60,21) = 21098 | D(60,66) = 4312  |
| D(59,22) = 6391  | D(59,67) = 18018 | D(60,22) = 10241 | D(60,67) = 6237  |
| D(59,23) = 9504  | D(59,68) = 23870 | D(60,23) = 11462 | D(60,68) = 15246 |
| D(59,24) = 13783 | D(59,69) = 22638 | D(60,24) = 4158  | D(60,69) = 13475 |
| D(59,25) = 4235  | D(59,70) = 7315  | D(60,25) = 11319 | D(60,70) = 14861 |
| D(59,26) = 8503  | D(59,71) = 14390 | D(60,26) = 12353 | D(60,71) = 14322 |
| D(59,27) = 14245 | D(59,72) = 31955 | D(60,27) = 30180 | D(60,72) = 18249 |
| D(59,28) = 6424  | D(59,73) = 5775  | D(60,28) = 17633 | D(60,73) = 14322 |
| D(59,29) = 10349 | D(59,74) = 15631 | D(60,29) = 8778  | D(60,74) = 6006  |
| D(59,30) = 5313  | D(59,75) = 16324 | D(60,30) = 12859 | D(60,75) = 2233  |
| D(59,31) = 11088 | D(59,76) = 7502  | D(60,31) = 4697  | D(60,76) = 16775 |
| D(59,32) = 13090 | D(59,77) = 18095 | D(60,32) = 3388  | D(60,77) = 33953 |
| D(59,33) = 3003  | D(59,78) = 21791 | D(60,33) = 17017 | D(60,78) = 12936 |
| D(59,34) = 15015 | D(59,79) = 7953  | D(60,34) = 30950 | D(60,79) = 9911  |
| D(59,35) = 3003  | D(59,80) = 12243 | D(60,35) = 12474 | D(60,80) = 23408 |
| D(59,36) = 23639 | D(59,81) = 5621  | D(60,36) = 9163  | D(60,81) = 16478 |
| D(59,37) = 16320 | D(59,82) = 14938 | D(60,37) = 28101 | D(60,82) = 3619  |
| D(59,38) = 14857 | D(59,83) = 6545  | D(60,38) = 26946 | D(60,83) = 5852  |
| D(59,39) = 12936 | D(59,84) = 17941 | D(60,39) = 28871 | D(60,84) = 2156  |
| D(59,40) = 8261  | D(59,85) = 8008  | D(60,40) = 14388 | D(60,85) = 11858 |
| D(59,41) = 15015 | D(59,86) = 13860 | D(60,41) = 1925  | D(60,86) = 3773  |
| D(59,42) = 13167 | D(59,87) = 25487 | D(60,42) = 4158  | D(60,87) = 11627 |
| D(59,43) = 22253 | D(59,88) = 19943 | D(60,43) = 12089 | D(60,88) = 32259 |
| D(59,44) = 10329 | D(59,89) = 11550 | D(60,44) = 12287 | D(60,89) = 22715 |
| D(59,45) = 14475 |                  | D(60,45) = 3619  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(61,1) = 31724  | D(61,46) = 34261 | D(62,1) = 2950   | D(62,46) = 11165 |
| D(61,2) = 10251  | D(61,47) = 16364 | D(62,2) = 21417  | D(62,47) = 15275 |
| D(61,3) = 19602  | D(61,48) = 27797 | D(62,3) = 13816  | D(62,48) = 8624  |
| D(61,4) = 7469   | D(61,49) = 6974  | D(62,4) = 25157  | D(62,49) = 24694 |
| D(61,5) = 14780  | D(61,50) = 27874 | D(62,5) = 18865  | D(62,50) = 3003  |
| D(61,6) = 27335  | D(61,51) = 33799 | D(62,6) = 7161   | D(62,51) = 7315  |
| D(61,7) = 27566  | D(61,52) = 20247 | D(62,7) = 5698   | D(62,52) = 14806 |
| D(61,8) = 26950  | D(61,53) = 32413 | D(62,8) = 7546   | D(62,53) = 9625  |
| D(61,9) = 16478  | D(61,54) = 19558 | D(62,9) = 24486  | D(62,54) = 16247 |
| D(61,10) = 24948 | D(61,55) = 9471  | D(62,10) = 12782 | D(62,55) = 21945 |
| D(61,11) = 35416 | D(61,56) = 13068 | D(62,11) = 12243 | D(62,56) = 18662 |
| D(61,12) = 9525  | D(61,57) = 7347  | D(62,12) = 22143 | D(62,57) = 24321 |
| D(61,13) = 8624  | D(61,58) = 25560 | D(62,13) = 35420 | D(62,58) = 15015 |
| D(61,14) = 32516 | D(61,59) = 16320 | D(62,14) = 20251 | D(62,59) = 19635 |
| D(61,15) = 13398 | D(61,60) = 27562 | D(62,15) = 21945 | D(62,60) = 9856  |
| D(61,16) = 7931  | D(61,61) = 0     | D(62,16) = 34727 | D(62,61) = 29568 |
| D(61,17) = 27342 | D(61,62) = 29568 | D(62,17) = 9702  | D(62,62) = 0     |
| D(61,18) = 21710 | D(61,63) = 19939 | D(62,18) = 11473 | D(62,63) = 15862 |
| D(61,19) = 5313  | D(61,64) = 2695  | D(62,19) = 31339 | D(62,64) = 29953 |
| D(61,20) = 12870 | D(61,65) = 10010 | D(62,20) = 22264 | D(62,65) = 36806 |
| D(61,21) = 6468  | D(61,66) = 27104 | D(62,21) = 25179 | D(62,66) = 6160  |
| D(61,22) = 17321 | D(61,67) = 28875 | D(62,22) = 14322 | D(62,67) = 3619  |
| D(61,23) = 17721 | D(61,68) = 34800 | D(62,23) = 12463 | D(62,68) = 12320 |
| D(61,24) = 25102 | D(61,69) = 32105 | D(62,24) = 12397 | D(62,69) = 10395 |
| D(61,25) = 20251 | D(61,70) = 18172 | D(62,25) = 14399 | D(62,70) = 19558 |
| D(61,26) = 15209 | D(61,71) = 34980 | D(62,26) = 16434 | D(62,71) = 19096 |
| D(61,27) = 6160  | D(61,72) = 41422 | D(62,27) = 32186 | D(62,72) = 16863 |
| D(61,28) = 12012 | D(61,73) = 17475 | D(62,28) = 23485 | D(62,73) = 18942 |
| D(61,29) = 11480 | D(61,74) = 26561 | D(62,29) = 18249 | D(62,74) = 5929  |
| D(61,30) = 23408 | D(61,75) = 27639 | D(62,30) = 15554 | D(62,75) = 6776  |
| D(61,31) = 24563 | D(61,76) = 14025 | D(62,31) = 10241 | D(62,76) = 23221 |
| D(61,32) = 26796 | D(61,77) = 8085  | D(62,32) = 10318 | D(62,77) = 35959 |
| D(61,33) = 12859 | D(61,78) = 32721 | D(62,33) = 7315  | D(62,78) = 9471  |
| D(61,34) = 6930  | D(61,79) = 19272 | D(62,34) = 32956 | D(62,79) = 7964  |
| D(61,35) = 15935 | D(61,80) = 7700  | D(62,35) = 17325 | D(62,80) = 27489 |
| D(61,36) = 33106 | D(61,81) = 18014 | D(62,36) = 6083  | D(62,81) = 20174 |
| D(61,37) = 5698  | D(61,82) = 26257 | D(62,37) = 30107 | D(62,82) = 11242 |
| D(61,38) = 6468  | D(61,83) = 19635 | D(62,38) = 28952 | D(62,83) = 15246 |
| D(61,39) = 3795  | D(61,84) = 29256 | D(62,39) = 30877 | D(62,84) = 5698  |
| D(61,40) = 8785  | D(61,85) = 15704 | D(62,40) = 23655 | D(62,85) = 15939 |
| D(61,41) = 27951 | D(61,86) = 26176 | D(62,41) = 8547  | D(62,86) = 7007  |
| D(61,42) = 26103 | D(61,87) = 34954 | D(62,42) = 6776  | D(62,87) = 5588  |
| D(61,43) = 31720 | D(61,88) = 5544  | D(62,43) = 10010 | D(62,88) = 34265 |
| D(61,44) = 16896 | D(61,89) = 6699  | D(62,44) = 13354 | D(62,89) = 26796 |
| D(61,45) = 24563 |                  | D(62,45) = 7931  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(63,1) = 16093  | D(63,46) = 21483 | D(64,1) = 31339  | D(64,46) = 34646 |
| D(63,2) = 8283   | D(63,47) = 4037  | D(64,2) = 8041   | D(64,47) = 16745 |
| D(63,3) = 7227   | D(63,48) = 12166 | D(64,3) = 18293  | D(64,48) = 26411 |
| D(63,4) = 14630  | D(63,49) = 11560 | D(64,4) = 12936  | D(64,49) = 8459  |
| D(63,5) = 5621   | D(63,50) = 15015 | D(64,5) = 13394  | D(64,50) = 28259 |
| D(63,6) = 11242  | D(63,51) = 16863 | D(64,6) = 25949  | D(64,51) = 34184 |
| D(63,7) = 11858  | D(63,52) = 6842  | D(64,7) = 26180  | D(64,52) = 18861 |
| D(63,8) = 11319  | D(63,53) = 19635 | D(64,8) = 25564  | D(64,53) = 32798 |
| D(63,9) = 11550  | D(63,54) = 4928  | D(64,9) = 12397  | D(64,54) = 19943 |
| D(63,10) = 4389  | D(63,55) = 10472 | D(64,10) = 21078 | D(64,55) = 8162  |
| D(63,11) = 18480 | D(63,56) = 8190  | D(64,11) = 35801 | D(64,56) = 11759 |
| D(63,12) = 9009  | D(63,57) = 11187 | D(64,12) = 7315  | D(64,57) = 5137  |
| D(63,13) = 23793 | D(63,58) = 8855  | D(64,13) = 10934 | D(64,58) = 25945 |
| D(63,14) = 22715 | D(63,59) = 8239  | D(64,14) = 27489 | D(64,59) = 14934 |
| D(63,15) = 10703 | D(63,60) = 15246 | D(64,15) = 12089 | D(64,60) = 27947 |
| D(63,16) = 23100 | D(63,61) = 19939 | D(64,16) = 10241 | D(64,61) = 2695  |
| D(63,17) = 8870  | D(63,62) = 15862 | D(64,17) = 25557 | D(64,62) = 29953 |
| D(63,18) = 7469  | D(63,63) = 0     | D(64,18) = 22095 | D(64,63) = 19092 |
| D(63,19) = 19404 | D(63,64) = 19092 | D(64,19) = 5852  | D(64,64) = 0     |
| D(63,20) = 9218  | D(63,65) = 25179 | D(64,20) = 11561 | D(64,65) = 12320 |
| D(63,21) = 13937 | D(63,66) = 11396 | D(64,21) = 5005  | D(64,66) = 25718 |
| D(63,22) = 3080  | D(63,67) = 13167 | D(64,22) = 17706 | D(64,67) = 27489 |
| D(63,23) = 5940  | D(63,68) = 20944 | D(64,23) = 16412 | D(64,68) = 35185 |
| D(63,24) = 6237  | D(63,69) = 15169 | D(64,24) = 25487 | D(64,69) = 32490 |
| D(63,25) = 4312  | D(63,70) = 5544  | D(64,25) = 18863 | D(64,70) = 18865 |
| D(63,26) = 5192  | D(63,71) = 21560 | D(64,26) = 15594 | D(64,71) = 29953 |
| D(63,27) = 20251 | D(63,72) = 24486 | D(64,27) = 6699  | D(64,72) = 41807 |
| D(63,28) = 10472 | D(63,73) = 2464  | D(64,28) = 10703 | D(64,73) = 16628 |
| D(63,29) = 13090 | D(63,74) = 11242 | D(64,29) = 9270  | D(64,74) = 26946 |
| D(63,30) = 2079  | D(63,75) = 8774  | D(64,30) = 22020 | D(64,75) = 28024 |
| D(63,31) = 9741  | D(63,76) = 9614  | D(64,31) = 24407 | D(64,76) = 12716 |
| D(63,32) = 9702  | D(63,77) = 24101 | D(64,32) = 26640 | D(64,77) = 9625  |
| D(63,33) = 18557 | D(63,78) = 18865 | D(64,33) = 11550 | D(64,78) = 33106 |
| D(63,34) = 21021 | D(63,79) = 4389  | D(64,34) = 7469  | D(64,79) = 17963 |
| D(63,35) = 4466  | D(63,80) = 16247 | D(64,35) = 14549 | D(64,80) = 5775  |
| D(63,36) = 16170 | D(63,81) = 5852  | D(64,36) = 33491 | D(64,81) = 17167 |
| D(63,37) = 20478 | D(63,82) = 7392  | D(64,37) = 6622  | D(64,82) = 26642 |
| D(63,38) = 19015 | D(63,83) = 5005  | D(64,38) = 5698  | D(64,83) = 20020 |
| D(63,39) = 18942 | D(63,84) = 10391 | D(64,39) = 5390  | D(64,84) = 29641 |
| D(63,40) = 10715 | D(63,85) = 4697  | D(64,40) = 7264  | D(64,85) = 16085 |
| D(63,41) = 12320 | D(63,86) = 10703 | D(64,41) = 26565 | D(64,86) = 26561 |
| D(63,42) = 10472 | D(63,87) = 18018 | D(64,42) = 24717 | D(64,87) = 35339 |
| D(63,43) = 14784 | D(63,88) = 23947 | D(64,43) = 32105 | D(64,88) = 5775  |
| D(63,44) = 6765  | D(63,89) = 15554 | D(64,44) = 15587 | D(64,89) = 4543  |
| D(63,45) = 10780 |                  | D(64,45) = 23177 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(65,1) = 38577  | D(65,46) = 44429 | D(66,1) = 6768   | D(66,46) = 11242 |
| D(65,2) = 21351  | D(65,47) = 23602 | D(66,2) = 11994  | D(66,47) = 9965  |
| D(65,3) = 24838  | D(65,48) = 32109 | D(66,3) = 12184  | D(66,48) = 5159  |
| D(65,4) = 13552  | D(65,49) = 18074 | D(66,4) = 19943  | D(66,49) = 14971 |
| D(65,5) = 22330  | D(65,50) = 35112 | D(66,5) = 13552  | D(66,50) = 4851  |
| D(65,6) = 31185  | D(65,51) = 42735 | D(66,6) = 4928   | D(66,51) = 14322 |
| D(65,7) = 31801  | D(65,52) = 21802 | D(66,7) = 308    | D(66,52) = 15532 |
| D(65,8) = 31262  | D(65,53) = 42504 | D(66,8) = 4158   | D(66,53) = 9394  |
| D(65,9) = 20097  | D(65,54) = 27258 | D(66,9) = 16863  | D(66,54) = 17941 |
| D(65,10) = 30877 | D(65,55) = 14707 | D(66,10) = 11242 | D(66,55) = 16632 |
| D(65,11) = 42658 | D(65,56) = 18304 | D(66,11) = 16016 | D(66,56) = 13349 |
| D(65,12) = 20625 | D(65,57) = 18447 | D(66,12) = 12420 | D(66,57) = 14598 |
| D(65,13) = 1386  | D(65,58) = 32798 | D(66,13) = 29953 | D(66,58) = 12551 |
| D(65,14) = 34342 | D(65,59) = 19173 | D(66,14) = 18095 | D(66,59) = 15400 |
| D(65,15) = 18634 | D(65,60) = 34800 | D(66,15) = 16632 | D(66,60) = 4312  |
| D(65,16) = 2079  | D(65,61) = 10010 | D(66,16) = 29260 | D(66,61) = 27104 |
| D(65,17) = 34977 | D(65,62) = 36806 | D(66,17) = 10258 | D(66,62) = 6160  |
| D(65,18) = 28948 | D(65,63) = 25179 | D(66,18) = 5929  | D(66,63) = 11396 |
| D(65,19) = 5775  | D(65,64) = 12320 | D(66,19) = 25564 | D(66,64) = 25718 |
| D(65,20) = 18106 | D(65,65) = 0     | D(66,20) = 16566 | D(66,65) = 31339 |
| D(65,21) = 13860 | D(65,66) = 31339 | D(66,21) = 19866 | D(66,66) = 0     |
| D(65,22) = 24559 | D(65,67) = 33110 | D(66,22) = 9009  | D(66,67) = 2387  |
| D(65,23) = 24244 | D(65,68) = 44275 | D(66,23) = 10857 | D(66,68) = 10703 |
| D(65,24) = 32340 | D(65,69) = 39343 | D(66,24) = 9933  | D(66,69) = 10780 |
| D(65,25) = 20328 | D(65,70) = 24278 | D(66,25) = 9779  | D(66,70) = 18326 |
| D(65,26) = 22447 | D(65,71) = 36806 | D(66,26) = 11121 | D(66,71) = 16940 |
| D(65,27) = 4928  | D(65,72) = 47740 | D(66,27) = 26411 | D(66,72) = 22022 |
| D(65,28) = 17248 | D(65,73) = 23177 | D(66,28) = 17787 | D(66,73) = 8932  |
| D(65,29) = 14420 | D(65,74) = 33799 | D(66,29) = 8526  | D(66,74) = 1771  |
| D(65,30) = 23485 | D(65,75) = 34877 | D(66,30) = 12166 | D(66,75) = 6160  |
| D(65,31) = 31801 | D(65,76) = 19261 | D(66,31) = 5313  | D(66,76) = 17523 |
| D(65,32) = 34111 | D(65,77) = 4928  | D(66,32) = 5313  | D(66,77) = 30261 |
| D(65,33) = 18095 | D(65,78) = 42812 | D(66,33) = 13244 | D(66,78) = 8624  |
| D(65,34) = 4158  | D(65,79) = 24607 | D(66,34) = 27181 | D(66,79) = 9306  |
| D(65,35) = 21175 | D(65,80) = 16170 | D(66,35) = 12397 | D(66,80) = 22176 |
| D(65,36) = 40344 | D(65,81) = 24486 | D(66,36) = 12474 | D(66,81) = 13013 |
| D(65,37) = 13629 | D(65,82) = 33495 | D(66,37) = 27104 | D(66,82) = 8778  |
| D(65,38) = 15092 | D(65,83) = 27335 | D(66,38) = 25641 | D(66,83) = 16940 |
| D(65,39) = 6237  | D(65,84) = 36494 | D(66,39) = 25102 | D(66,84) = 6314  |
| D(65,40) = 19796 | D(65,85) = 22942 | D(66,40) = 14136 | D(66,85) = 10625 |
| D(65,41) = 32263 | D(65,86) = 33414 | D(66,41) = 2310  | D(66,86) = 1232  |
| D(65,42) = 30415 | D(65,87) = 43967 | D(66,42) = 2233  | D(66,87) = 15477 |
| D(65,43) = 38958 | D(65,88) = 7931  | D(66,43) = 1095  | D(66,88) = 29876 |
| D(65,44) = 25069 | D(65,89) = 14322 | D(66,44) = 11682 | D(66,89) = 21483 |
| D(65,45) = 30723 |                  | D(66,45) = 2541  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(67,1) = 5005   | D(67,46) = 14399 | D(68,1) = 10549  | D(68,46) = 5313  |
| D(67,2) = 13904  | D(67,47) = 11737 | D(68,2) = 30367  | D(68,47) = 22902 |
| D(67,3) = 13724  | D(67,48) = 6314  | D(68,3) = 20449  | D(68,48) = 13706 |
| D(67,4) = 21714  | D(67,49) = 17181 | D(68,4) = 30723  | D(68,49) = 33644 |
| D(67,5) = 15015  | D(67,50) = 2156  | D(68,5) = 20867  | D(68,50) = 13013 |
| D(67,6) = 7469   | D(67,51) = 11165 | D(68,6) = 15477  | D(68,51) = 10395 |
| D(67,7) = 1925   | D(67,52) = 15224 | D(68,7) = 10241  | D(68,52) = 21076 |
| D(67,8) = 5467   | D(67,53) = 12551 | D(68,8) = 11396  | D(68,53) = 3465  |
| D(67,9) = 18634  | D(67,54) = 19789 | D(68,9) = 27643  | D(68,54) = 24332 |
| D(67,10) = 13475 | D(67,55) = 18403 | D(68,10) = 21791 | D(68,55) = 29568 |
| D(67,11) = 14322 | D(67,56) = 15120 | D(68,11) = 15015 | D(68,56) = 26285 |
| D(67,12) = 14630 | D(67,57) = 16808 | D(68,12) = 31093 | D(68,57) = 33271 |
| D(67,13) = 31724 | D(67,58) = 14399 | D(68,13) = 42889 | D(68,58) = 18172 |
| D(67,14) = 20482 | D(67,59) = 18018 | D(68,14) = 14168 | D(68,59) = 23870 |
| D(67,15) = 18403 | D(67,60) = 6237  | D(68,15) = 29568 | D(68,60) = 15246 |
| D(67,16) = 31031 | D(67,61) = 28875 | D(68,16) = 42196 | D(68,61) = 34800 |
| D(67,17) = 12106 | D(67,62) = 3619  | D(68,17) = 15857 | D(68,62) = 12320 |
| D(67,18) = 7854  | D(67,63) = 13167 | D(68,18) = 16709 | D(68,63) = 20944 |
| D(67,19) = 27335 | D(67,64) = 27489 | D(68,19) = 38500 | D(68,64) = 35185 |
| D(67,20) = 18106 | D(67,65) = 33110 | D(68,20) = 24808 | D(68,65) = 44275 |
| D(67,21) = 22484 | D(67,66) = 2387  | D(68,21) = 29029 | D(68,66) = 10703 |
| D(67,22) = 10780 | D(67,67) = 0     | D(68,22) = 18557 | D(68,67) = 13860 |
| D(67,23) = 12397 | D(67,68) = 13860 | D(68,23) = 19162 | D(68,68) = 0     |
| D(67,24) = 11781 | D(67,69) = 12705 | D(68,24) = 15554 | D(68,69) = 7777  |
| D(67,25) = 12012 | D(67,70) = 22484 | D(68,25) = 19866 | D(68,70) = 26488 |
| D(67,26) = 12892 | D(67,71) = 19327 | D(68,26) = 24057 | D(68,71) = 13013 |
| D(67,27) = 28182 | D(67,72) = 20328 | D(68,27) = 39347 | D(68,72) = 20097 |
| D(67,28) = 19327 | D(67,73) = 10703 | D(68,28) = 26029 | D(68,73) = 18249 |
| D(67,29) = 10736 | D(67,74) = 3696  | D(68,29) = 27199 | D(68,74) = 10164 |
| D(67,30) = 14399 | D(67,75) = 8162  | D(68,30) = 26257 | D(68,75) = 15554 |
| D(67,31) = 7700  | D(67,76) = 19063 | D(68,31) = 21888 | D(68,76) = 25765 |
| D(67,32) = 7084  | D(67,77) = 32032 | D(68,32) = 20328 | D(68,77) = 43197 |
| D(67,33) = 15015 | D(67,78) = 11781 | D(68,33) = 26180 | D(68,78) = 3157  |
| D(67,34) = 28952 | D(67,79) = 10846 | D(68,34) = 40117 | D(68,79) = 17611 |
| D(67,35) = 13860 | D(67,80) = 24794 | D(68,35) = 19712 | D(68,80) = 31339 |
| D(67,36) = 9317  | D(67,81) = 14784 | D(68,36) = 8624  | D(68,81) = 22407 |
| D(67,37) = 28875 | D(67,82) = 10626 | D(68,37) = 25285 | D(68,82) = 14399 |
| D(67,38) = 27412 | D(67,83) = 18788 | D(68,38) = 34184 | D(68,83) = 23331 |
| D(67,39) = 26873 | D(67,84) = 8470  | D(68,39) = 38038 | D(68,84) = 12474 |
| D(67,40) = 16346 | D(67,85) = 12397 | D(68,40) = 32843 | D(68,85) = 23562 |
| D(67,41) = 4697  | D(67,86) = 3003  | D(68,41) = 12859 | D(68,86) = 11704 |
| D(67,42) = 4312  | D(67,87) = 12320 | D(68,42) = 11627 | D(68,87) = 12243 |
| D(67,43) = 12320 | D(67,88) = 32494 | D(68,43) = 7931  | D(68,88) = 39039 |
| D(67,44) = 13222 | D(67,89) = 30569 | D(68,44) = 19987 | D(68,89) = 30646 |
| D(67,45) = 4774  |                  | D(68,45) = 12936 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(69,1) = 6930   | D(69,46) = 6699  | D(70,1) = 15400  | D(70,45) = 20328 |
| D(69,2) = 16720  | D(69,47) = 21516 | D(70,2) = 14255  | D(70,47) = 11572 |
| D(69,3) = 17061  | D(69,48) = 13090 | D(70,3) = 11363  | D(70,48) = 20636 |
| D(69,4) = 28871  | D(69,49) = 19997 | D(70,4) = 16940  | D(70,49) = 17532 |
| D(69,5) = 19635  | D(69,50) = 12782 | D(70,5) = 11165  | D(70,50) = 20328 |
| D(69,6) = 11707  | D(69,51) = 7623  | D(70,6) = 14476  | D(70,51) = 20790 |
| D(69,7) = 10318  | D(69,52) = 19844 | D(70,7) = 18788  | D(70,52) = 10824 |
| D(69,8) = 12243  | D(69,53) = 5236  | D(70,8) = 20944  | D(70,53) = 23100 |
| D(69,9) = 24948  | D(69,54) = 23177 | D(70,9) = 12474  | D(70,54) = 2772  |
| D(69,10) = 18095 | D(69,55) = 21329 | D(70,10) = 6034  | D(70,55) = 15950 |
| D(69,11) = 12397 | D(69,56) = 19965 | D(70,11) = 23254 | D(70,56) = 13376 |
| D(69,12) = 17446 | D(69,57) = 19624 | D(70,12) = 14981 | D(70,57) = 23320 |
| D(69,13) = 37957 | D(69,58) = 16789 | D(70,13) = 22561 | D(70,58) = 11935 |
| D(69,14) = 7392  | D(69,59) = 22638 | D(70,14) = 11165 | D(70,59) = 7315  |
| D(69,15) = 20944 | D(69,60) = 13475 | D(70,15) = 13167 | D(70,60) = 14861 |
| D(69,16) = 37264 | D(69,61) = 32105 | D(70,16) = 22176 | D(70,61) = 18172 |
| D(69,17) = 11473 | D(69,62) = 10395 | D(70,17) = 11165 | D(70,62) = 19558 |
| D(69,18) = 15631 | D(69,63) = 15169 | D(70,18) = 18326 | D(70,63) = 5544  |
| D(69,19) = 33876 | D(69,64) = 32490 | D(70,19) = 18711 | D(70,64) = 18865 |
| D(69,20) = 20392 | D(69,65) = 39343 | D(70,20) = 12749 | D(70,65) = 24278 |
| D(69,21) = 28259 | D(69,66) = 10780 | D(70,21) = 23177 | D(70,66) = 18326 |
| D(69,22) = 20559 | D(69,67) = 12705 | D(70,22) = 10780 | D(70,67) = 22484 |
| D(69,23) = 15774 | D(69,68) = 7777  | D(70,23) = 10109 | D(70,68) = 26488 |
| D(69,24) = 14168 | D(69,69) = 0     | D(70,24) = 9779  | D(70,69) = 26103 |
| D(69,25) = 20405 | D(69,70) = 26103 | D(70,25) = 8932  | D(70,70) = 0     |
| D(69,26) = 22671 | D(69,71) = 6237  | D(70,26) = 12892 | D(70,71) = 10010 |
| D(69,27) = 34723 | D(69,72) = 17017 | D(70,27) = 19943 | D(70,72) = 27566 |
| D(69,28) = 22176 | D(69,73) = 13013 | D(70,28) = 14135 | D(70,73) = 7161  |
| D(69,29) = 13552 | D(69,74) = 8393  | D(70,29) = 11087 | D(70,74) = 20636 |
| D(69,30) = 18711 | D(69,75) = 12936 | D(70,30) = 3850  | D(70,75) = 11858 |
| D(69,31) = 15708 | D(69,76) = 21318 | D(70,31) = 12243 | D(70,76) = 11363 |
| D(69,32) = 14168 | D(69,77) = 38496 | D(70,32) = 13090 | D(70,77) = 24101 |
| D(69,33) = 20174 | D(69,78) = 5314  | D(70,33) = 10010 | D(70,78) = 23408 |
| D(69,34) = 35493 | D(69,79) = 14223 | D(70,34) = 20790 | D(70,79) = 8690  |
| D(69,35) = 18480 | D(69,80) = 30569 | D(70,35) = 9933  | D(70,80) = 24640 |
| D(69,36) = 3458  | D(69,81) = 16401 | D(70,36) = 21868 | D(70,81) = 11935 |
| D(69,37) = 32644 | D(69,82) = 13013 | D(70,37) = 20020 | D(70,82) = 10549 |
| D(69,38) = 31489 | D(69,83) = 22176 | D(70,38) = 18403 | D(70,83) = 2079  |
| D(69,39) = 33414 | D(69,84) = 12166 | D(70,39) = 17941 | D(70,84) = 13552 |
| D(69,40) = 19162 | D(69,85) = 22176 | D(70,40) = 16697 | D(70,85) = 12298 |
| D(69,41) = 8470  | D(69,86) = 11011 | D(70,41) = 17864 | D(70,86) = 19250 |
| D(69,42) = 11396 | D(69,87) = 8855  | D(70,42) = 17787 | D(70,87) = 22099 |
| D(69,43) = 4815  | D(69,88) = 36802 | D(70,43) = 23254 | D(70,88) = 24486 |
| D(69,44) = 16599 | D(69,89) = 29876 | D(70,44) = 10868 | D(70,89) = 24563 |
| D(69,45) = 13244 |                  | D(70,45) = 20328 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(71,1) = 16170  | D(71,46) = 12551 | D(72,1) = 15092  | D(72,46) = 19866 |
| D(71,2) = 21317  | D(71,47) = 13504 | D(72,2) = 30041  | D(72,47) = 26367 |
| D(71,3) = 15521  | D(71,48) = 21560 | D(72,3) = 26378  | D(72,48) = 24563 |
| D(71,4) = 21175  | D(71,49) = 24694 | D(72,4) = 35343  | D(72,49) = 33318 |
| D(71,5) = 16247  | D(71,50) = 20174 | D(72,5) = 28952  | D(72,50) = 18711 |
| D(71,6) = 12782  | D(71,51) = 9317  | D(72,6) = 17325  | D(72,51) = 9625  |
| D(71,7) = 17479  | D(71,52) = 20305 | D(72,7) = 21560  | D(72,52) = 29161 |
| D(71,8) = 19712  | D(71,53) = 10703 | D(72,8) = 23716  | D(72,53) = 18018 |
| D(71,9) = 20120  | D(71,54) = 9009  | D(72,9) = 32263  | D(72,54) = 29260 |
| D(71,10) = 9548  | D(71,55) = 23935 | D(72,10) = 23408 | D(72,55) = 33033 |
| D(71,11) = 21564 | D(71,56) = 20305 | D(72,11) = 6006  | D(72,56) = 29750 |
| D(71,12) = 22143 | D(71,57) = 25091 | D(72,12) = 30767 | D(72,57) = 32945 |
| D(71,13) = 35089 | D(71,58) = 10164 | D(72,13) = 46354 | D(72,58) = 22561 |
| D(71,14) = 5300  | D(71,59) = 14390 | D(72,14) = 32267 | D(72,59) = 31955 |
| D(71,15) = 29721 | D(71,60) = 14322 | D(72,15) = 33033 | D(72,60) = 18249 |
| D(71,16) = 34342 | D(71,61) = 34980 | D(72,16) = 45691 | D(72,61) = 41422 |
| D(71,17) = 10857 | D(71,62) = 19096 | D(72,17) = 20266 | D(72,62) = 16863 |
| D(71,18) = 15400 | D(71,63) = 21560 | D(72,18) = 22715 | D(72,63) = 24486 |
| D(71,19) = 30877 | D(71,64) = 29953 | D(72,19) = 41965 | D(72,64) = 41807 |
| D(71,20) = 24232 | D(71,65) = 36806 | D(72,20) = 32036 | D(72,65) = 47740 |
| D(71,21) = 24948 | D(71,66) = 16940 | D(72,21) = 34264 | D(72,66) = 22022 |
| D(71,22) = 12551 | D(71,67) = 19327 | D(72,22) = 23793 | D(72,67) = 20328 |
| D(71,23) = 14234 | D(71,68) = 13013 | D(72,23) = 25091 | D(72,68) = 20097 |
| D(71,24) = 10087 | D(71,69) = 6237  | D(72,24) = 19943 | D(72,69) = 17017 |
| D(71,25) = 13090 | D(71,70) = 10010 | D(72,25) = 24871 | D(72,70) = 27566 |
| D(71,26) = 14659 | D(71,71) = 0     | D(72,26) = 27522 | D(72,71) = 31112 |
| D(71,27) = 31801 | D(71,72) = 31112 | D(72,27) = 42812 | D(72,72) = 0     |
| D(71,28) = 25552 | D(71,73) = 23100 | D(72,28) = 33257 | D(72,73) = 22330 |
| D(71,29) = 18249 | D(71,74) = 20405 | D(72,29) = 26873 | D(72,74) = 22022 |
| D(71,30) = 9012  | D(71,75) = 11627 | D(72,30) = 24563 | D(72,75) = 17864 |
| D(71,31) = 12936 | D(71,76) = 23308 | D(72,31) = 19327 | D(72,76) = 32993 |
| D(71,32) = 14014 | D(71,77) = 35343 | D(72,32) = 18480 | D(72,77) = 46662 |
| D(71,33) = 27565 | D(71,78) = 10934 | D(72,33) = 29645 | D(72,78) = 18018 |
| D(71,34) = 32417 | D(71,79) = 12638 | D(72,34) = 43582 | D(72,79) = 23540 |
| D(71,35) = 14399 | D(71,80) = 26026 | D(72,35) = 27797 | D(72,80) = 36577 |
| D(71,36) = 8162  | D(71,81) = 25795 | D(72,36) = 12243 | D(72,81) = 25718 |
| D(71,37) = 31724 | D(71,82) = 10472 | D(72,37) = 41961 | D(72,82) = 18788 |
| D(71,38) = 30569 | D(71,83) = 9014  | D(72,38) = 40806 | D(72,83) = 28259 |
| D(71,39) = 30338 | D(71,84) = 12474 | D(72,39) = 41503 | D(72,84) = 15939 |
| D(71,40) = 23859 | D(71,85) = 14164 | D(72,40) = 32483 | D(72,85) = 27027 |
| D(71,41) = 13090 | D(71,86) = 17325 | D(72,41) = 19712 | D(72,86) = 23870 |
| D(71,42) = 17941 | D(71,87) = 11011 | D(72,42) = 22869 | D(72,87) = 10241 |
| D(71,43) = 10934 | D(71,88) = 38984 | D(72,43) = 17171 | D(72,88) = 44277 |
| D(71,44) = 15059 | D(71,89) = 26103 | D(72,44) = 25916 | D(72,89) = 35884 |
| D(71,45) = 15708 |                  | D(72,45) = 24717 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(73,1) = 18786  | D(73,46) = 18788 | D(74,1) = 6622   | D(74,46) = 10164 |
| D(73,2) = 6204   | D(73,47) = 1881  | D(74,2) = 17337  | D(74,47) = 10274 |
| D(73,3) = 4763   | D(73,48) = 9702  | D(74,3) = 10824  | D(74,48) = 3773  |
| D(73,4) = 12474  | D(73,49) = 4981  | D(74,4) = 23327  | D(74,49) = 20614 |
| D(73,5) = 3157   | D(73,50) = 12859 | D(74,5) = 12628  | D(74,50) = 4389  |
| D(73,6) = 8778   | D(73,51) = 14707 | D(74,6) = 6468   | D(74,51) = 11550 |
| D(73,7) = 9394   | D(73,52) = 4378  | D(74,7) = 1463   | D(74,52) = 12837 |
| D(73,8) = 8855   | D(73,53) = 16940 | D(74,8) = 1925   | D(74,53) = 8855  |
| D(73,9) = 9394   | D(73,54) = 6160  | D(74,9) = 19404  | D(74,54) = 22561 |
| D(73,10) = 5621  | D(73,55) = 8470  | D(74,10) = 13783 | D(74,55) = 16940 |
| D(73,11) = 16324 | D(73,56) = 5110  | D(74,11) = 16709 | D(74,56) = 13657 |
| D(73,12) = 6930  | D(73,57) = 9108  | D(74,12) = 18063 | D(74,57) = 20241 |
| D(73,13) = 21791 | D(73,58) = 8932  | D(74,13) = 32413 | D(74,58) = 11935 |
| D(73,14) = 24255 | D(73,59) = 5775  | D(74,14) = 21560 | D(74,59) = 15631 |
| D(73,15) = 8547  | D(73,60) = 14322 | D(74,15) = 16940 | D(74,60) = 6006  |
| D(73,16) = 21098 | D(73,61) = 17475 | D(74,16) = 31720 | D(74,61) = 26561 |
| D(73,17) = 8947  | D(73,62) = 18942 | D(74,17) = 9644  | D(74,62) = 5929  |
| D(73,18) = 5082  | D(73,63) = 2464  | D(74,18) = 8085  | D(74,63) = 11242 |
| D(73,19) = 17402 | D(73,64) = 16628 | D(74,19) = 28332 | D(74,64) = 26946 |
| D(73,20) = 6908  | D(73,65) = 23177 | D(74,20) = 15796 | D(74,65) = 33799 |
| D(73,21) = 11781 | D(73,66) = 8932  | D(74,21) = 20713 | D(74,66) = 1771  |
| D(73,22) = 924   | D(73,67) = 10703 | D(74,22) = 9317  | D(74,67) = 3696  |
| D(73,23) = 3476  | D(73,68) = 18249 | D(74,23) = 9537  | D(74,68) = 10164 |
| D(73,24) = 6314  | D(73,69) = 13013 | D(74,24) = 9317  | D(74,69) = 8393  |
| D(73,25) = 2541  | D(73,70) = 7161  | D(74,25) = 10395 | D(74,70) = 20636 |
| D(73,26) = 3036  | D(73,71) = 23100 | D(74,26) = 11429 | D(74,71) = 20405 |
| D(73,27) = 18249 | D(73,72) = 22330 | D(74,27) = 29179 | D(74,72) = 22022 |
| D(73,28) = 8162  | D(73,73) = 0     | D(74,28) = 17017 | D(74,73) = 9086  |
| D(73,29) = 11011 | D(73,74) = 9086  | D(74,29) = 14169 | D(74,74) = 0     |
| D(73,30) = 3311  | D(73,75) = 8851  | D(74,30) = 14123 | D(74,75) = 8701  |
| D(73,31) = 7893  | D(73,76) = 7458  | D(74,31) = 6391  | D(74,76) = 16573 |
| D(73,32) = 7854  | D(73,77) = 22099 | D(74,32) = 6699  | D(74,77) = 32952 |
| D(73,33) = 6699  | D(73,78) = 16170 | D(74,33) = 13552 | D(74,78) = 8162  |
| D(73,34) = 19019 | D(73,79) = 1925  | D(74,34) = 29949 | D(74,79) = 7986  |
| D(73,35) = 1632  | D(73,80) = 14091 | D(74,35) = 11473 | D(74,80) = 23023 |
| D(73,36) = 14014 | D(73,81) = 4081  | D(74,36) = 9394  | D(74,81) = 12474 |
| D(73,37) = 18014 | D(73,82) = 7469  | D(74,37) = 27100 | D(74,82) = 8162  |
| D(73,38) = 16551 | D(73,83) = 6237  | D(74,38) = 25945 | D(74,83) = 21510 |
| D(73,39) = 16940 | D(73,84) = 10468 | D(74,39) = 27870 | D(74,84) = 8470  |
| D(73,40) = 8646  | D(73,85) = 2541  | D(74,40) = 19544 | D(74,85) = 10934 |
| D(73,41) = 9856  | D(73,86) = 8316  | D(74,41) = 4389  | D(74,86) = 2541  |
| D(73,42) = 8008  | D(73,87) = 15862 | D(74,42) = 2541  | D(74,87) = 12782 |
| D(73,43) = 12628 | D(73,88) = 21791 | D(74,43) = 8008  | D(74,88) = 31258 |
| D(73,44) = 4301  | D(73,89) = 13398 | D(74,44) = 10362 | D(74,89) = 22330 |
| D(73,45) = 8316  |                  | D(74,45) = 3696  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(75,1) = 7546   | D(75,46) = 14091 | D(76,1) = 18683  | D(76,46) = 25916 |
| D(75,2) = 17798  | D(75,47) = 9346  | D(76,2) = 2178   | D(76,47) = 5577  |
| D(75,3) = 10131  | D(75,48) = 8855  | D(76,3) = 11352  | D(76,48) = 17908 |
| D(75,4) = 24405  | D(75,49) = 21075 | D(76,4) = 14102  | D(76,49) = 6435  |
| D(75,5) = 13321  | D(75,50) = 8162  | D(76,5) = 7227   | D(76,50) = 20603 |
| D(75,6) = 1309   | D(75,51) = 11319 | D(76,6) = 16445  | D(76,51) = 28226 |
| D(75,7) = 6622   | D(75,52) = 11990 | D(76,7) = 17985  | D(76,52) = 3762  |
| D(75,8) = 8547   | D(75,53) = 12166 | D(76,8) = 17061  | D(76,53) = 24607 |
| D(75,9) = 19400  | D(75,54) = 11396 | D(76,9) = 11516  | D(76,54) = 13772 |
| D(75,10) = 8239  | D(75,55) = 16804 | D(76,10) = 12694 | D(76,55) = 4554  |
| D(75,11) = 12782 | D(75,56) = 13570 | D(76,11) = 27911 | D(76,56) = 5214  |
| D(75,12) = 18524 | D(75,57) = 20929 | D(76,12) = 2904  | D(76,57) = 5082  |
| D(75,13) = 33491 | D(75,58) = 6545  | D(76,13) = 17875 | D(76,58) = 14311 |
| D(75,14) = 12782 | D(75,59) = 16324 | D(76,14) = 26773 | D(76,59) = 7502  |
| D(75,15) = 19785 | D(75,60) = 2233  | D(76,15) = 8228  | D(76,60) = 16775 |
| D(75,16) = 32798 | D(75,61) = 27639 | D(76,16) = 17182 | D(76,61) = 14025 |
| D(75,17) = 1617  | D(75,62) = 6776  | D(76,17) = 15081 | D(76,62) = 23221 |
| D(75,18) = 5386  | D(75,63) = 8774  | D(76,18) = 14564 | D(76,63) = 9614  |
| D(75,19) = 29410 | D(75,64) = 28024 | D(76,19) = 14333 | D(76,64) = 12716 |
| D(75,20) = 14527 | D(75,65) = 34877 | D(76,20) = 1419  | D(76,65) = 19261 |
| D(75,21) = 22711 | D(75,66) = 6160  | D(76,21) = 9973  | D(76,66) = 17523 |
| D(75,22) = 8393  | D(75,67) = 8162  | D(76,22) = 6534  | D(76,67) = 19063 |
| D(75,23) = 8844  | D(75,68) = 15554 | D(76,23) = 10065 | D(76,68) = 25765 |
| D(75,24) = 2540  | D(75,69) = 12936 | D(76,24) = 12386 | D(76,69) = 21318 |
| D(75,25) = 11935 | D(75,70) = 11858 | D(76,25) = 8228  | D(76,70) = 11363 |
| D(75,26) = 10501 | D(75,71) = 11627 | D(76,26) = 4422  | D(76,71) = 23308 |
| D(75,27) = 30257 | D(75,72) = 17864 | D(76,27) = 15180 | D(76,72) = 32993 |
| D(75,28) = 15781 | D(75,73) = 8851  | D(76,28) = 4026  | D(76,73) = 7458  |
| D(75,29) = 14630 | D(75,74) = 8701  | D(76,29) = 5610  | D(76,74) = 16573 |
| D(75,30) = 9009  | D(75,75) = 0     | D(76,30) = 11385 | D(76,75) = 14923 |
| D(75,31) = 3003  | D(75,76) = 14923 | D(76,31) = 11847 | D(76,76) = 0     |
| D(75,32) = 2772  | D(75,77) = 34030 | D(76,32) = 13541 | D(76,77) = 18953 |
| D(75,33) = 18707 | D(75,78) = 12474 | D(76,33) = 7898  | D(76,78) = 23837 |
| D(75,34) = 31027 | D(75,79) = 7293  | D(76,34) = 15950 | D(76,79) = 8514  |
| D(75,35) = 12162 | D(75,80) = 21868 | D(76,35) = 6072  | D(76,80) = 9966  |
| D(75,36) = 9394  | D(75,81) = 12239 | D(76,36) = 22319 | D(76,81) = 10846 |
| D(75,37) = 28178 | D(75,82) = 1463  | D(76,37) = 10329 | D(76,82) = 13541 |
| D(75,38) = 27023 | D(75,83) = 10164 | D(76,38) = 10690 | D(76,83) = 13849 |
| D(75,39) = 28948 | D(75,84) = 2079  | D(76,39) = 13871 | D(76,84) = 16544 |
| D(75,40) = 20240 | D(75,85) = 10006 | D(76,40) = 4629  | D(76,85) = 4917  |
| D(75,41) = 4004  | D(75,86) = 5698  | D(76,41) = 18062 | D(76,86) = 16445 |
| D(75,42) = 6930  | D(75,87) = 12551 | D(76,42) = 16214 | D(76,87) = 29458 |
| D(75,43) = 12089 | D(75,88) = 32336 | D(76,43) = 20933 | D(76,88) = 17259 |
| D(75,44) = 9669  | D(75,89) = 24097 | D(76,44) = 10890 | D(76,89) = 11590 |
| D(75,45) = 5621  |                  | D(76,45) = 16522 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(77,1) = 37730  | D(77,46) = 43351 | D(78,1) = 7854   | D(78,46) = 2618  |
| D(77,2) = 15280  | D(77,47) = 22755 | D(78,2) = 26807  | D(78,47) = 21439 |
| D(77,3) = 23760  | D(77,48) = 31031 | D(78,3) = 18370  | D(78,48) = 11627 |
| D(77,4) = 12320  | D(77,49) = 12003 | D(78,4) = 28644  | D(78,49) = 30084 |
| D(77,5) = 21252  | D(77,50) = 34265 | D(78,5) = 18788  | D(78,50) = 11011 |
| D(77,6) = 30107  | D(77,51) = 41657 | D(78,6) = 13398  | D(78,51) = 8316  |
| D(77,7) = 30723  | D(77,52) = 20724 | D(78,7) = 8161   | D(78,52) = 18997 |
| D(77,8) = 30184  | D(77,53) = 41426 | D(78,8) = 9317   | D(78,53) = 770   |
| D(77,9) = 19019  | D(77,54) = 26642 | D(78,9) = 25564  | D(78,54) = 22869 |
| D(77,10) = 31493 | D(77,55) = 14399 | D(78,10) = 20328 | D(78,55) = 28105 |
| D(77,11) = 41580 | D(77,56) = 17996 | D(78,11) = 12936 | D(78,56) = 24822 |
| D(77,12) = 14554 | D(77,57) = 12376 | D(78,12) = 27533 | D(78,57) = 29711 |
| D(77,13) = 3542  | D(77,58) = 31951 | D(78,13) = 41426 | D(78,58) = 16709 |
| D(77,14) = 32879 | D(77,59) = 18095 | D(78,14) = 12089 | D(78,59) = 21791 |
| D(77,15) = 18326 | D(77,60) = 33953 | D(78,15) = 28105 | D(78,60) = 12936 |
| D(77,16) = 2849  | D(77,61) = 8085  | D(78,16) = 40733 | D(78,61) = 32721 |
| D(77,17) = 34116 | D(77,62) = 35959 | D(78,17) = 14394 | D(78,62) = 9471  |
| D(77,18) = 28101 | D(77,63) = 24101 | D(78,18) = 14861 | D(78,63) = 18865 |
| D(77,19) = 4389  | D(77,64) = 9625  | D(78,19) = 37037 | D(78,64) = 33106 |
| D(77,20) = 17798 | D(77,65) = 4928  | D(78,20) = 22880 | D(78,65) = 42812 |
| D(77,21) = 11242 | D(77,66) = 30261 | D(78,21) = 27566 | D(78,66) = 8624  |
| D(77,22) = 23712 | D(77,67) = 32032 | D(78,22) = 17094 | D(78,67) = 11781 |
| D(77,23) = 23166 | D(77,68) = 43197 | D(78,23) = 17083 | D(78,68) = 3157  |
| D(77,24) = 31493 | D(77,69) = 38496 | D(78,24) = 14091 | D(78,69) = 5314  |
| D(77,25) = 26796 | D(77,70) = 24101 | D(78,25) = 18403 | D(78,70) = 23408 |
| D(77,26) = 21600 | D(77,71) = 35343 | D(78,26) = 22594 | D(78,71) = 10934 |
| D(77,27) = 4620  | D(77,72) = 46662 | D(78,27) = 37884 | D(78,72) = 18018 |
| D(77,28) = 16940 | D(77,73) = 22099 | D(78,28) = 24101 | D(78,73) = 16170 |
| D(77,29) = 8334  | D(77,74) = 32952 | D(78,29) = 23639 | D(78,74) = 8162  |
| D(77,30) = 29953 | D(77,75) = 34030 | D(78,30) = 24794 | D(78,75) = 12474 |
| D(77,31) = 30954 | D(77,76) = 18953 | D(78,31) = 20425 | D(78,76) = 23837 |
| D(77,32) = 33880 | D(77,77) = 0     | D(78,32) = 18865 | D(78,77) = 41734 |
| D(77,33) = 17787 | D(77,78) = 41734 | D(78,33) = 24717 | D(78,78) = 0     |
| D(77,34) = 3850  | D(77,79) = 23559 | D(78,34) = 38654 | D(78,79) = 15532 |
| D(77,35) = 20097 | D(77,80) = 10151 | D(78,35) = 17633 | D(78,80) = 29876 |
| D(77,36) = 39497 | D(77,81) = 23403 | D(78,36) = 5775  | D(78,81) = 20328 |
| D(77,37) = 9009  | D(77,82) = 32648 | D(78,37) = 33260 | D(78,82) = 12936 |
| D(77,38) = 10395 | D(77,83) = 26719 | D(78,38) = 32105 | D(78,83) = 21868 |
| D(77,39) = 3927  | D(77,84) = 35647 | D(78,39) = 36575 | D(78,84) = 11011 |
| D(77,40) = 13912 | D(77,85) = 22095 | D(78,40) = 29134 | D(78,85) = 22099 |
| D(77,41) = 31185 | D(77,86) = 32567 | D(78,41) = 10780 | D(78,86) = 9394  |
| D(77,42) = 29337 | D(77,87) = 42889 | D(78,42) = 9548  | D(78,87) = 10164 |
| D(77,43) = 38111 | D(77,88) = 3003  | D(78,43) = 5467  | D(78,88) = 37576 |
| D(77,44) = 23991 | D(77,89) = 12859 | D(78,44) = 17908 | D(78,89) = 29183 |
| D(77,45) = 29645 |                  | D(78,45) = 10857 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(79,1) = 11984  | D(79,46) = 18150 | D(80,1) = 28237  | D(80,46) = 31493 |
| D(79,2) = 10923  | D(79,47) = 1078  | D(80,2) = 3296   | D(80,47) = 12518 |
| D(79,3) = 2838   | D(79,48) = 9691  | D(80,3) = 17523  | D(80,48) = 23716 |
| D(79,4) = 15532  | D(79,49) = 13900 | D(80,4) = 10087  | D(80,49) = 1925  |
| D(79,5) = 4950   | D(79,50) = 11682 | D(80,5) = 13013  | D(80,50) = 26103 |
| D(79,6) = 8228   | D(79,51) = 17072 | D(80,6) = 23254  | D(80,51) = 33033 |
| D(79,7) = 9768   | D(79,52) = 4818  | D(80,7) = 22638  | D(80,52) = 16170 |
| D(79,8) = 8844   | D(79,53) = 16302 | D(80,8) = 22869  | D(80,53) = 30018 |
| D(79,9) = 12452  | D(79,54) = 7909  | D(80,9) = 10318  | D(80,54) = 20713 |
| D(79,10) = 5214  | D(79,55) = 9900  | D(80,10) = 19943 | D(80,55) = 5929  |
| D(79,11) = 17534 | D(79,56) = 6270  | D(80,11) = 31495 | D(80,56) = 10010 |
| D(79,12) = 11649 | D(79,57) = 13827 | D(80,12) = 2570  | D(80,57) = 4748  |
| D(79,13) = 23221 | D(79,58) = 6811  | D(80,13) = 14399 | D(80,58) = 19789 |
| D(79,14) = 14223 | D(79,59) = 7953  | D(80,14) = 26257 | D(80,59) = 12243 |
| D(79,15) = 9372  | D(79,60) = 9911  | D(80,15) = 9702  | D(80,60) = 23408 |
| D(79,16) = 22528 | D(79,61) = 19272 | D(80,16) = 13783 | D(80,61) = 7700  |
| D(79,17) = 6292  | D(79,62) = 7964  | D(80,17) = 22022 | D(80,62) = 27489 |
| D(79,18) = 3212  | D(79,63) = 4389  | D(80,18) = 17633 | D(80,63) = 16247 |
| D(79,19) = 18832 | D(79,64) = 17963 | D(80,19) = 9931  | D(80,64) = 5775  |
| D(79,20) = 9207  | D(79,65) = 24607 | D(80,20) = 8679  | D(80,65) = 16170 |
| D(79,21) = 18051 | D(79,66) = 9306  | D(80,21) = 3542  | D(80,66) = 22176 |
| D(79,22) = 1155  | D(79,67) = 10846 | D(80,22) = 13475 | D(80,67) = 24794 |
| D(79,23) = 1551  | D(79,68) = 17611 | D(80,23) = 18810 | D(80,68) = 31339 |
| D(79,24) = 4655  | D(79,69) = 14223 | D(80,24) = 19327 | D(80,69) = 30569 |
| D(79,25) = 2596  | D(79,70) = 8690  | D(80,25) = 15169 | D(80,70) = 24640 |
| D(79,26) = 2233  | D(79,71) = 12638 | D(80,26) = 11363 | D(80,71) = 26026 |
| D(79,27) = 19679 | D(79,72) = 23540 | D(80,27) = 10270 | D(80,72) = 36577 |
| D(79,28) = 10428 | D(79,73) = 1925  | D(80,28) = 8580  | D(80,73) = 14091 |
| D(79,29) = 10164 | D(79,74) = 7986  | D(80,29) = 3740  | D(80,74) = 23023 |
| D(79,30) = 5137  | D(79,75) = 7293  | D(80,30) = 18326 | D(80,75) = 21868 |
| D(79,31) = 5522  | D(79,76) = 8514  | D(80,31) = 18788 | D(80,76) = 9966  |
| D(79,32) = 5368  | D(79,77) = 23559 | D(80,32) = 20482 | D(80,77) = 10151 |
| D(79,33) = 8833  | D(79,78) = 15532 | D(80,33) = 9163  | D(80,78) = 29876 |
| D(79,34) = 20449 | D(79,79) = 0     | D(80,34) = 11425 | D(80,79) = 20361 |
| D(79,35) = 3795  | D(79,80) = 20361 | D(80,35) = 11858 | D(80,80) = 0     |
| D(79,36) = 15224 | D(79,81) = 6006  | D(80,36) = 32109 | D(80,81) = 17479 |
| D(79,37) = 15576 | D(79,82) = 5810  | D(80,37) = 4466  | D(80,82) = 20482 |
| D(79,38) = 15937 | D(79,83) = 7119  | D(80,38) = 2568  | D(80,83) = 20790 |
| D(79,39) = 18370 | D(79,84) = 7392  | D(80,39) = 9471  | D(80,84) = 23485 |
| D(79,40) = 13410 | D(79,85) = 1738  | D(80,40) = 2713  | D(80,85) = 11858 |
| D(79,41) = 9845  | D(79,86) = 8140  | D(80,41) = 23870 | D(80,86) = 21560 |
| D(79,42) = 7997  | D(79,87) = 18227 | D(80,42) = 22022 | D(80,87) = 34265 |
| D(79,43) = 13838 | D(79,88) = 28061 | D(80,43) = 30646 | D(80,88) = 9702  |
| D(79,44) = 2376  | D(79,89) = 19668 | D(80,44) = 17985 | D(80,89) = 3050  |
| D(79,45) = 8305  |                  | D(80,45) = 20482 |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(81,1) = 20405  | D(81,46) = 22946 | D(82,1) = 8393   | D(82,46) = 14553 |
| D(81,2) = 7128   | D(81,47) = 5269  | D(82,2) = 15257  | D(82,47) = 7964  |
| D(81,3) = 8844   | D(81,48) = 13783 | D(82,3) = 8668   | D(82,48) = 8855  |
| D(81,4) = 15862  | D(81,49) = 10405 | D(82,4) = 23024  | D(82,49) = 18534 |
| D(81,5) = 4622   | D(81,50) = 16247 | D(82,5) = 11935  | D(82,50) = 10318 |
| D(81,6) = 12859  | D(81,51) = 18095 | D(82,6) = 2541   | D(82,51) = 12859 |
| D(81,7) = 13475  | D(81,52) = 5852  | D(82,7) = 9240   | D(82,52) = 11844 |
| D(81,8) = 12936  | D(81,53) = 21098 | D(82,8) = 8008   | D(82,53) = 12628 |
| D(81,9) = 12782  | D(81,54) = 9317  | D(82,9) = 18018  | D(82,54) = 10010 |
| D(81,10) = 8239  | D(81,55) = 9779  | D(82,10) = 6853  | D(82,55) = 15422 |
| D(81,11) = 19712 | D(81,56) = 6650  | D(82,11) = 13706 | D(82,56) = 12188 |
| D(81,12) = 7854  | D(81,57) = 10032 | D(82,12) = 15983 | D(82,57) = 19547 |
| D(81,13) = 23100 | D(81,58) = 12320 | D(82,13) = 32109 | D(82,58) = 5159  |
| D(81,14) = 26950 | D(81,59) = 5621  | D(82,14) = 11627 | D(82,59) = 14938 |
| D(81,15) = 9317  | D(81,60) = 16478 | D(82,15) = 18403 | D(82,60) = 3619  |
| D(81,16) = 22407 | D(81,61) = 18014 | D(82,16) = 31416 | D(82,61) = 26257 |
| D(81,17) = 12335 | D(81,62) = 20174 | D(82,17) = 616   | D(82,62) = 11242 |
| D(81,18) = 8470  | D(81,63) = 5852  | D(82,18) = 4004  | D(82,63) = 7392  |
| D(81,19) = 18711 | D(81,64) = 17167 | D(82,19) = 28028 | D(82,64) = 26642 |
| D(81,20) = 8448  | D(81,65) = 24486 | D(82,20) = 13145 | D(82,65) = 33495 |
| D(81,21) = 15169 | D(81,66) = 13013 | D(82,21) = 21329 | D(82,66) = 8778  |
| D(81,22) = 4312  | D(81,67) = 14784 | D(82,22) = 7007  | D(82,67) = 10626 |
| D(81,23) = 7557  | D(81,68) = 22407 | D(82,23) = 7361  | D(82,68) = 14399 |
| D(81,24) = 9702  | D(81,69) = 16401 | D(82,24) = 1155  | D(82,69) = 13013 |
| D(81,25) = 2926  | D(81,70) = 11935 | D(82,25) = 10549 | D(82,70) = 10549 |
| D(81,26) = 6424  | D(81,71) = 25795 | D(82,26) = 9119  | D(82,71) = 10472 |
| D(81,27) = 19558 | D(81,72) = 25718 | D(82,27) = 28875 | D(82,72) = 18788 |
| D(81,28) = 9702  | D(81,73) = 4081  | D(82,28) = 14399 | D(82,73) = 7469  |
| D(81,29) = 11935 | D(81,74) = 12474 | D(82,29) = 12089 | D(82,74) = 8162  |
| D(81,30) = 5929  | D(81,75) = 12239 | D(82,30) = 7623  | D(82,75) = 1463  |
| D(81,31) = 11666 | D(81,76) = 10846 | D(82,31) = 1848  | D(82,76) = 13541 |
| D(81,32) = 11627 | D(81,77) = 23403 | D(82,32) = 3465  | D(82,77) = 32648 |
| D(81,33) = 8008  | D(81,78) = 20328 | D(82,33) = 17325 | D(82,78) = 12936 |
| D(81,34) = 20328 | D(81,79) = 6006  | D(82,34) = 29645 | D(82,79) = 5810  |
| D(81,35) = 3465  | D(81,80) = 17479 | D(82,35) = 10780 | D(82,80) = 20482 |
| D(81,36) = 17402 | D(81,81) = 0     | D(82,36) = 10934 | D(82,81) = 10857 |
| D(81,37) = 18553 | D(81,82) = 10857 | D(82,37) = 26796 | D(82,82) = 0     |
| D(81,38) = 17090 | D(81,83) = 9394  | D(82,38) = 25641 | D(82,83) = 8778  |
| D(81,39) = 18249 | D(81,84) = 13856 | D(82,39) = 27566 | D(82,84) = 3157  |
| D(81,40) = 9570  | D(81,85) = 5929  | D(82,40) = 17699 | D(82,85) = 8624  |
| D(81,41) = 13937 | D(81,86) = 11704 | D(82,41) = 6391  | D(82,86) = 5852  |
| D(81,42) = 12089 | D(81,87) = 19250 | D(82,42) = 7161  | D(82,87) = 14091 |
| D(81,43) = 16016 | D(81,88) = 25179 | D(82,43) = 12936 | D(82,88) = 30954 |
| D(81,44) = 8382  | D(81,89) = 16786 | D(82,44) = 8206  | D(82,89) = 22715 |
| D(81,45) = 12397 |                  | D(82,45) = 4851  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(83,1) = 7504   | D(83,46) = 23485 | D(84,1) = 5929   | D(84,46) = 12628 |
| D(83,2) = 13104  | D(83,47) = 8272  | D(84,2) = 19800  | D(84,47) = 10963 |
| D(83,3) = 9977   | D(83,48) = 20328 | D(84,3) = 11517  | D(84,48) = 8851  |
| D(83,4) = 15169  | D(83,49) = 16381 | D(84,4) = 26022  | D(84,49) = 23077 |
| D(83,5) = 10703  | D(83,50) = 17479 | D(84,5) = 14938  | D(84,50) = 6776  |
| D(83,6) = 11858  | D(83,51) = 18711 | D(84,6) = 1155   | D(84,51) = 9394  |
| D(83,7) = 17402  | D(83,52) = 10912 | D(84,7) = 6622   | D(84,52) = 13607 |
| D(83,8) = 19481  | D(83,53) = 21560 | D(84,8) = 8545   | D(84,53) = 10703 |
| D(83,9) = 13090  | D(83,54) = 1078  | D(84,9) = 21017  | D(84,54) = 13013 |
| D(83,10) = 6006  | D(83,55) = 15730 | D(84,10) = 9858  | D(84,55) = 18425 |
| D(83,11) = 23177 | D(83,56) = 12496 | D(84,11) = 10857 | D(84,56) = 15191 |
| D(83,12) = 13830 | D(83,57) = 19855 | D(84,12) = 20526 | D(84,57) = 22546 |
| D(83,13) = 25949 | D(83,58) = 10824 | D(84,13) = 35108 | D(84,58) = 8162  |
| D(83,14) = 10169 | D(83,59) = 6545  | D(84,14) = 13629 | D(84,59) = 17941 |
| D(83,15) = 11781 | D(83,60) = 5852  | D(84,15) = 21402 | D(84,60) = 2156  |
| D(83,16) = 25256 | D(83,61) = 19635 | D(84,16) = 34415 | D(84,61) = 29256 |
| D(83,17) = 10241 | D(83,62) = 15246 | D(84,17) = 3311  | D(84,62) = 5698  |
| D(83,18) = 8624  | D(83,63) = 5005  | D(84,18) = 7315  | D(84,63) = 10391 |
| D(83,19) = 21406 | D(83,64) = 20020 | D(84,19) = 31027 | D(84,64) = 29641 |
| D(83,20) = 13453 | D(83,65) = 27335 | D(84,20) = 16148 | D(84,65) = 36494 |
| D(83,21) = 14707 | D(83,66) = 16940 | D(84,21) = 24328 | D(84,66) = 6314  |
| D(83,22) = 7315  | D(83,67) = 18788 | D(84,22) = 10010 | D(84,67) = 8470  |
| D(83,23) = 8730  | D(83,68) = 23331 | D(84,23) = 10230 | D(84,68) = 12474 |
| D(83,24) = 8624  | D(83,69) = 22176 | D(84,24) = 4157  | D(84,69) = 12166 |
| D(83,25) = 8932  | D(83,70) = 2079  | D(84,25) = 13552 | D(84,70) = 13552 |
| D(83,26) = 9424  | D(83,71) = 9014  | D(84,26) = 12118 | D(84,71) = 12474 |
| D(83,27) = 22253 | D(83,72) = 28259 | D(84,27) = 31874 | D(84,72) = 15939 |
| D(83,28) = 14707 | D(83,73) = 6237  | D(84,28) = 17402 | D(84,73) = 10468 |
| D(83,29) = 9936  | D(83,74) = 21510 | D(84,29) = 16632 | D(84,74) = 8470  |
| D(83,30) = 3003  | D(83,75) = 10164 | D(84,30) = 10626 | D(84,75) = 2079  |
| D(83,31) = 11165 | D(83,76) = 13849 | D(84,31) = 4158  | D(84,76) = 16544 |
| D(83,32) = 12628 | D(83,77) = 26719 | D(84,32) = 2772  | D(84,77) = 35647 |
| D(83,33) = 8625  | D(83,78) = 21868 | D(84,33) = 20324 | D(84,78) = 11011 |
| D(83,34) = 23023 | D(83,79) = 7119  | D(84,34) = 32644 | D(84,79) = 7392  |
| D(83,35) = 9848  | D(83,80) = 20790 | D(84,35) = 13779 | D(84,80) = 23485 |
| D(83,36) = 17710 | D(83,81) = 9394  | D(84,36) = 8085  | D(84,81) = 13856 |
| D(83,37) = 20174 | D(83,82) = 8778  | D(84,37) = 29795 | D(84,82) = 3157  |
| D(83,38) = 19019 | D(83,83) = 0     | D(84,38) = 28640 | D(84,83) = 11781 |
| D(83,39) = 20944 | D(83,84) = 11781 | D(84,39) = 30565 | D(84,84) = 0     |
| D(83,40) = 15546 | D(83,85) = 8932  | D(84,40) = 22242 | D(84,85) = 11623 |
| D(83,41) = 14014 | D(83,86) = 18711 | D(84,41) = 4004  | D(84,86) = 5775  |
| D(83,42) = 17864 | D(83,87) = 19943 | D(84,42) = 6930  | D(84,87) = 10626 |
| D(83,43) = 21868 | D(83,88) = 24409 | D(84,43) = 10241 | D(84,88) = 33953 |
| D(83,44) = 9515  | D(83,89) = 16701 | D(84,44) = 11055 | D(84,89) = 25714 |
| D(83,45) = 16324 |                  | D(84,45) = 5621  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(85,1) = 16016  | D(85,46) = 23716 | D(86,1) = 7392   | D(86,46) = 12089 |
| D(85,2) = 8354   | D(85,47) = 660   | D(86,2) = 14487  | D(86,47) = 9196  |
| D(85,3) = 2376   | D(85,48) = 11396 | D(86,3) = 10978  | D(86,48) = 3850  |
| D(85,4) = 12470  | D(85,49) = 11631 | D(86,4) = 22942  | D(86,49) = 17764 |
| D(85,5) = 5005   | D(85,50) = 14553 | D(86,5) = 11088  | D(86,50) = 5313  |
| D(85,6) = 10472  | D(85,51) = 22022 | D(86,6) = 4697   | D(86,51) = 13783 |
| D(85,7) = 11088  | D(85,52) = 5214  | D(86,7) = 1232   | D(86,52) = 11836 |
| D(85,8) = 10549  | D(85,53) = 21791 | D(86,8) = 3003   | D(86,53) = 10164 |
| D(85,9) = 11781  | D(85,54) = 8855  | D(86,9) = 17941  | D(86,54) = 19712 |
| D(85,10) = 7777  | D(85,55) = 10032 | D(86,10) = 11165 | D(86,55) = 15862 |
| D(85,11) = 21945 | D(85,56) = 6798  | D(86,11) = 18788 | D(86,56) = 12579 |
| D(85,12) = 9080  | D(85,57) = 10923 | D(86,12) = 15213 | D(86,57) = 17391 |
| D(85,13) = 21556 | D(85,58) = 9394  | D(86,13) = 32028 | D(86,58) = 9625  |
| D(85,14) = 14395 | D(85,59) = 8008  | D(86,14) = 18480 | D(86,59) = 13860 |
| D(85,15) = 9240  | D(85,60) = 11858 | D(86,15) = 15862 | D(86,60) = 3773  |
| D(85,16) = 20863 | D(85,61) = 15704 | D(86,16) = 31335 | D(86,61) = 26176 |
| D(85,17) = 10164 | D(85,62) = 15939 | D(86,17) = 7334  | D(86,62) = 7007  |
| D(85,18) = 6006  | D(85,63) = 4697  | D(86,18) = 4774  | D(86,63) = 10703 |
| D(85,19) = 17475 | D(85,64) = 16085 | D(86,19) = 27947 | D(86,64) = 26561 |
| D(85,20) = 4521  | D(85,65) = 22942 | D(86,20) = 15488 | D(86,65) = 33414 |
| D(85,21) = 10780 | D(85,66) = 10625 | D(86,21) = 19250 | D(86,66) = 1232  |
| D(85,22) = 1617  | D(85,67) = 12397 | D(86,22) = 8239  | D(86,67) = 3003  |
| D(85,23) = 1100  | D(85,68) = 23562 | D(86,23) = 9691  | D(86,68) = 11704 |
| D(85,24) = 7469  | D(85,69) = 22176 | D(86,24) = 7007  | D(86,69) = 11011 |
| D(85,25) = 3351  | D(85,70) = 12298 | D(86,25) = 10010 | D(86,70) = 19250 |
| D(85,26) = 495   | D(85,71) = 14164 | D(86,26) = 10351 | D(86,71) = 17325 |
| D(85,27) = 18322 | D(85,72) = 27027 | D(86,27) = 28794 | D(86,72) = 23870 |
| D(85,28) = 5775  | D(85,73) = 2541  | D(86,28) = 16709 | D(86,73) = 8316  |
| D(85,29) = 10956 | D(85,74) = 10934 | D(86,29) = 11313 | D(86,74) = 2541  |
| D(85,30) = 6468  | D(85,75) = 10006 | D(86,30) = 11550 | D(86,75) = 5698  |
| D(85,31) = 6930  | D(85,76) = 4917  | D(86,31) = 4081  | D(86,76) = 16445 |
| D(85,32) = 8624  | D(85,77) = 22095 | D(86,32) = 4697  | D(86,77) = 32567 |
| D(85,33) = 8393  | D(85,78) = 22099 | D(86,33) = 12474 | D(86,78) = 9394  |
| D(85,34) = 19092 | D(85,79) = 1738  | D(86,34) = 29564 | D(86,79) = 8140  |
| D(85,35) = 3850  | D(85,80) = 11858 | D(86,35) = 9933  | D(86,80) = 21560 |
| D(85,36) = 20097 | D(85,81) = 5929  | D(86,36) = 12012 | D(86,81) = 11704 |
| D(85,37) = 16243 | D(85,82) = 8624  | D(86,37) = 26715 | D(86,82) = 5852  |
| D(85,38) = 15142 | D(85,83) = 8932  | D(86,38) = 25560 | D(86,83) = 18711 |
| D(85,39) = 17013 | D(85,84) = 11623 | D(86,39) = 27485 | D(86,84) = 5775  |
| D(85,40) = 10796 | D(85,85) = 0     | D(86,40) = 16694 | D(86,85) = 9856  |
| D(85,41) = 11550 | D(85,86) = 9856  | D(86,41) = 1771  | D(86,86) = 0     |
| D(85,42) = 9702  | D(85,87) = 23254 | D(86,42) = 1386  | D(86,87) = 15015 |
| D(85,43) = 22099 | D(85,88) = 20401 | D(86,43) = 10626 | D(86,88) = 30873 |
| D(85,44) = 1914  | D(85,89) = 12162 | D(86,44) = 10516 | D(86,89) = 20867 |
| D(85,45) = 11010 |                  | D(86,45) = 1925  |                  |

## Ek 2'nin devamı

|                  |                  |                  |                  |
|------------------|------------------|------------------|------------------|
| D(87,1) = 5544   | D(87,46) = 12012 | D(88,1) = 36036  | D(88,46) = 39193 |
| D(87,2) = 22769  | D(87,47) = 22594 | D(88,2) = 14831  | D(88,47) = 21061 |
| D(87,3) = 21065  | D(87,48) = 17325 | D(88,3) = 25223  | D(88,48) = 31416 |
| D(87,4) = 33187  | D(87,49) = 26046 | D(88,4) = 10780  | D(88,49) = 11554 |
| D(87,5) = 22484  | D(87,50) = 9856  | D(88,5) = 20713  | D(88,50) = 32571 |
| D(87,6) = 11011  | D(87,51) = 1155  | D(88,6) = 30954  | D(88,51) = 40733 |
| D(87,7) = 15015  | D(87,52) = 22693 | D(88,7) = 30338  | D(88,52) = 23870 |
| D(87,8) = 16478  | D(87,53) = 10164 | D(88,8) = 30569  | D(88,53) = 37718 |
| D(87,9) = 30107  | D(87,54) = 20944 | D(88,9) = 18249  | D(88,54) = 24332 |
| D(87,10) = 17479 | D(87,55) = 29260 | D(88,10) = 29799 | D(88,55) = 12705 |
| D(87,11) = 5852  | D(87,56) = 25977 | D(88,11) = 39195 | D(88,56) = 16302 |
| D(87,12) = 23495 | D(87,57) = 25673 | D(88,12) = 14105 | D(88,57) = 11927 |
| D(87,13) = 42581 | D(87,58) = 17864 | D(88,13) = 6545  | D(88,58) = 30257 |
| D(87,14) = 12166 | D(87,59) = 25487 | D(88,14) = 36520 | D(88,59) = 19943 |
| D(87,15) = 29260 | D(87,60) = 11627 | D(88,15) = 16632 | D(88,60) = 32259 |
| D(87,16) = 41888 | D(87,61) = 34954 | D(88,16) = 5852  | D(88,61) = 5544  |
| D(87,17) = 15569 | D(87,62) = 5588  | D(88,17) = 32422 | D(88,62) = 34265 |
| D(87,18) = 20097 | D(87,63) = 18018 | D(88,18) = 26407 | D(88,63) = 23947 |
| D(87,19) = 38192 | D(87,64) = 35339 | D(88,19) = 5852  | D(88,64) = 5775  |
| D(87,20) = 28501 | D(87,65) = 43967 | D(88,20) = 16104 | D(88,65) = 7931  |
| D(87,21) = 31955 | D(87,66) = 15477 | D(88,21) = 1617  | D(88,66) = 29876 |
| D(87,22) = 21637 | D(87,67) = 12320 | D(88,22) = 22018 | D(88,67) = 32494 |
| D(87,23) = 19778 | D(87,68) = 12243 | D(88,23) = 26510 | D(88,68) = 39039 |
| D(87,24) = 15246 | D(87,69) = 8855  | D(88,24) = 29799 | D(88,69) = 36802 |
| D(87,25) = 18943 | D(87,70) = 22099 | D(88,25) = 25102 | D(88,70) = 24486 |
| D(87,26) = 23749 | D(87,71) = 11011 | D(88,26) = 19906 | D(88,71) = 38984 |
| D(87,27) = 30939 | D(87,72) = 10241 | D(88,27) = 5852  | D(88,72) = 44277 |
| D(87,28) = 29722 | D(87,73) = 15862 | D(88,28) = 15246 | D(88,73) = 21791 |
| D(87,29) = 19601 | D(87,74) = 12782 | D(88,29) = 8118  | D(88,74) = 31258 |
| D(87,30) = 18635 | D(87,75) = 12551 | D(88,30) = 28259 | D(88,75) = 32336 |
| D(87,31) = 14553 | D(87,76) = 29458 | D(88,31) = 29260 | D(88,76) = 17259 |
| D(87,32) = 13706 | D(87,77) = 42889 | D(88,32) = 31570 | D(88,77) = 3003  |
| D(87,33) = 25872 | D(87,78) = 10164 | D(88,33) = 16093 | D(88,78) = 37576 |
| D(87,34) = 39809 | D(87,79) = 18227 | D(88,34) = 5082  | D(88,79) = 28061 |
| D(87,35) = 21329 | D(87,80) = 34265 | D(88,35) = 19558 | D(88,80) = 9702  |
| D(87,36) = 3542  | D(87,81) = 19250 | D(88,36) = 37803 | D(88,81) = 25179 |
| D(87,37) = 35493 | D(87,82) = 14091 | D(88,37) = 6237  | D(88,82) = 30954 |
| D(87,38) = 34338 | D(87,83) = 19943 | D(88,38) = 7931  | D(88,83) = 24409 |
| D(87,39) = 37730 | D(87,84) = 10626 | D(88,39) = 5390  | D(88,84) = 33953 |
| D(87,40) = 25211 | D(87,85) = 23254 | D(88,40) = 13523 | D(88,85) = 20401 |
| D(87,41) = 12474 | D(87,86) = 15015 | D(88,41) = 31570 | D(88,86) = 30873 |
| D(87,42) = 15631 | D(87,87) = 0     | D(88,42) = 29722 | D(88,87) = 41965 |
| D(87,43) = 9009  | D(87,88) = 41965 | D(88,43) = 36417 | D(88,88) = 0     |
| D(87,44) = 20613 | D(87,89) = 33572 | D(88,44) = 25685 | D(88,89) = 10857 |
| D(87,45) = 17479 |                  | D(88,45) = 28182 |                  |

## Ek 2'nin devamı

|            |       |            |       |
|------------|-------|------------|-------|
| D(89,1) =  | 28314 | D(89,46) = | 30800 |
| D(89,2) =  | 8295  | D(89,47) = | 12822 |
| D(89,3) =  | 16830 | D(89,48) = | 23023 |
| D(89,4) =  | 7699  | D(89,49) = | 2464  |
| D(89,5) =  | 12320 | D(89,50) = | 25410 |
| D(89,6) =  | 22561 | D(89,51) = | 32340 |
| D(89,7) =  | 21945 | D(89,52) = | 15477 |
| D(89,8) =  | 22176 | D(89,53) = | 29325 |
| D(89,9) =  | 9856  | D(89,54) = | 16624 |
| D(89,10) = | 20174 | D(89,55) = | 4158  |
| D(89,11) = | 30802 | D(89,56) = | 11634 |
| D(89,12) = | 7569  | D(89,57) = | 9774  |
| D(89,13) = | 12549 | D(89,58) = | 22018 |
| D(89,14) = | 23639 | D(89,59) = | 11550 |
| D(89,15) = | 8239  | D(89,60) = | 22715 |
| D(89,16) = | 11933 | D(89,61) = | 6699  |
| D(89,17) = | 24197 | D(89,62) = | 26796 |
| D(89,18) = | 16940 | D(89,63) = | 15554 |
| D(89,19) = | 7082  | D(89,64) = | 4543  |
| D(89,20) = | 10303 | D(89,65) = | 14322 |
| D(89,21) = | 1617  | D(89,66) = | 21483 |
| D(89,22) = | 13779 | D(89,67) = | 30569 |
| D(89,23) = | 18117 | D(89,68) = | 30646 |
| D(89,24) = | 21560 | D(89,69) = | 29876 |
| D(89,25) = | 15477 | D(89,70) = | 24563 |
| D(89,26) = | 11667 | D(89,71) = | 26103 |
| D(89,27) = | 7421  | D(89,72) = | 35884 |
| D(89,28) = | 10204 | D(89,73) = | 13398 |
| D(89,29) = | 4279  | D(89,74) = | 22330 |
| D(89,30) = | 18634 | D(89,75) = | 24097 |
| D(89,31) = | 21021 | D(89,76) = | 11590 |
| D(89,32) = | 23321 | D(89,77) = | 12859 |
| D(89,33) = | 7931  | D(89,78) = | 29183 |
| D(89,34) = | 8626  | D(89,79) = | 19668 |
| D(89,35) = | 11165 | D(89,80) = | 3050  |
| D(89,36) = | 31416 | D(89,81) = | 16786 |
| D(89,37) = | 5621  | D(89,82) = | 22715 |
| D(89,38) = | 4697  | D(89,83) = | 16701 |
| D(89,39) = | 6622  | D(89,84) = | 25714 |
| D(89,40) = | 7374  | D(89,85) = | 12162 |
| D(89,41) = | 23177 | D(89,86) = | 20867 |
| D(89,42) = | 21329 | D(89,87) = | 33572 |
| D(89,43) = | 29983 | D(89,88) = | 10857 |
| D(89,44) = | 17292 | D(89,89) = | 0     |
| D(89,45) = | 19789 |            |       |

## Ek 4. Uygulama Problemine Ait Mevcut Rotalar

| Rota no. | Rota                    | Toplam Rota Uzunluğu (m) | Toplam Rota Yüğü (kişi) |
|----------|-------------------------|--------------------------|-------------------------|
| 1        | 1-49-12-29-40-2-1       | 45517                    | 15                      |
| 2        | 1-25-30-10-58-24-1      | 45949                    | 17                      |
| 3        | 1-26-85-47-22-1         | 43307                    | 14                      |
| 4        | 1-89-80-21-57-1         | 69281                    | 14                      |
| 5        | 1-69-36-43-1            | 18685                    | 13                      |
| 6        | 1-76-28-20-1            | 40892                    | 12                      |
| 7        | 1-15-33-56-55-1         | 49875                    | 18                      |
| 8        | 1-65-13-16-19-4-1       | 74566                    | 15                      |
| 9        | 1-77-88-34-27-39-1      | 80443                    | 14                      |
| 10       | 1-68-78-53-46-1         | 23974                    | 13                      |
| 11       | 1-9-72-11-51-87-1       | 75281                    | 12                      |
| 12       | 1-3-44-79-23-1          | 31522                    | 14                      |
| 13       | 1-61-64-38-37-1         | 69673                    | 11                      |
| 14       | 1-14-66-7-67-1          | 39193                    | 15                      |
| 15       | 1-59-35-5-52-1          | 43167                    | 12                      |
| 16       | 1-48-8-42-1             | 21560                    | 12                      |
| 17       | 1-45-32-41-60-1         | 21861                    | 16                      |
| 18       | 1-81-73-63-1            | 43043                    | 14                      |
| 19       | 1-71-82-17-31-75-6-84-1 | 41947                    | 16                      |
| 20       | 1-18-86-74-62-50-1      | 31108                    | 18                      |
| 21       | 1-70-54-81-1            | 26754                    | 9                       |

## Ek 6. İş Yeri Servis Araçlarını Rotalama Programı

Akış diyagramı Şekil 5.2'de sunulan sezgisel metodumuz için Quick BASIC programlama dili ile geliştirilmiş olan araç rotalama programı aşağıda listelenmektedir.

```

10  REM -----
20  REM İŞ YERİ SERVİS ARAÇLARI İÇİN ROTA İNŞA PROSEDÜRÜ
30  REM -----
40  PRINT "Kalkış ve biniş noktalarının sayısını giriniz"
50  INPUT N
60  PRINT "Maksimum mesafe kısıtını giriniz"
70  INPUT DK
80  PRINT "Araç kapasitesini giriniz"
90  INPUT Q
100 PRINT "Maksimum sapma kısıtını giriniz"
110 INPUT P
120 OPTION BASE 1
130 DIM A(N), AT(N), C$(N), B(N), CT$(N)
140 OPTION BASE 1
150 DIM D(N, N), AD(N,N), ROT$(N)
160 PRINT "Biniş noktalarındaki talepleri giriniz"
170 FOR I=1 TO N
180  INPUT A(I)
190 NEXT I
200 PRINT "Noktalar arası mesafeleri giriniz"
210 FOR I=1 TO N
220  FOR J=1 TO N
230  INPUT D(I, J)
240  NEXT J
250  NEXT I
260  FOR I=1 TO N

```

```
270 C$(I)=STR$(I)
280 NEXT I
290 J=0
300 FOR I=2 TO N
310 IF A(I)<Q THEN GO TO 400
320 J=J+1
330 AT(J)=A(I)
340 A(I)=0
350 AD((1, J)=D(1,I)
360 CT$(J)=C$(I)
370 PRINT "Aşırı talepli nokta=";CT$(J)
380 PRINT "Talepli miktarı=";AT(J)
390 PRINT "Depoya uzaklık=";AD(1,J)
400 NEXT I
410 IF J=0 THEN GO TO 570
420 S=0
430 FOR I=1 TO J
440 S=S+1
450 PRINT "ROTA (";S;")=1"; CT$(I)
460 PRINT "Toplam yük="; Q
470 PRINT "Toplam mesafe="; D(1,VAL(CT$(I)))
480 PRINT
490 IF AT(I)-Q=0 THEN GO TO 520
500 IF AT(I)-Q<Q THEN A(VAL(CT$(I)))=AT(I)-Q:GO TO 520
510 AT(I)=AT(I)-Q: GO TO 440
520 NEXT I
530 B(1)=0
540 FOR I=2 TO N
550 B(I)=1
560 NEXT I
570 L=1
580 L=L+1
```

```
590 IF L>N THEN GOTO 1120
600 IF A(L)=0 THEN GOTO 580
610 BU=D (1,L)
620 Y=L
630 FOR I=1 TO N
640 IF A(I)=0 THEN GOTO 680
650 IF D(1,I)≤ BU THEN GOTO 680
660 BU=D(1,I)
670 Y=I
680 NEXT I
690 YK=A(Y)
700 MES=D(1,Y)
710 S=S+1
720 ROT$(S)=C$(1)+C$(Y)
730 A(Y)=0
740 IF MES<DK THEN GOTO 800
750 PRINT "ROTA (";S;")=1"; Y
760 PRINT "Toplam yük="; YK
770 PRINT "Toplam mesafe="; MES
780 PRINT
790 GOTO 570
800 F=1
810 F=F+1
820 IF F>N THEN GOTO 750
830 IF F=Y THEN GOTO 810
840 IF B(F)=0 THEN GOTO 810
850 IF A(F)=0 THEN GOTO 810
860 KU=D(F,Y)
870 G=F
880 FOR I=2 TO N
890 IF I=Y THEN GOTO 950
900 IF A(I)=0 THEN GOTO 950
```

```

910 IF B(I)=0 THEN GOTO 950
920 IF D(I,Y)≥KU THEN GOTO 950
930 KU=D(I,Y)
940 G=I
950 NEXT I
960 IF YK+A(G)>Q THEN B(G)=0:GOTO 800
970 IF MES+D(Y,G)>DK THEN B(G)=0:GOTO 800
980 IF D(Y,1)-D(G,1)<0 THEN B(G)=0:GOTO 800
990 IF D(Y,G)+D(G,1)-D(Y,1)>P THEN B(G)=0:GOTO 800
1000 IF YK+A(G)=Q THEN ROT$(S)=ROT$(S)+C$(G): YK=YK+A(G): MES=
MES+D(Y,G): A(G)= 0: GOTO 1080
1010 H=0
1020 FOR I=2 TO N
1030 IF A(I)= 0 THEN GOTO 1050
1040 IF Q-(YK+A(G))≥A(I) THEN H=H+1
1050 NEXT I
1060 ROT$(S)=ROT$(S)+C$(G):YK=YK+A(G):MES=MES+D(Y,G):A(G)=0
1070 IF H≥1 THEN Y=G: GOTO 800
1080 PRINT "ROTA(";S;")="; ROT$(S)
1090 PRINT "Toplam yük="; YK
1100 PRINT "Toplam mesafe="; MES
1110 GOTO 540
1120 STOP

```

Program listemizin 150 no. lu satırında görülen ve rota dizisini temsil eden ROT\$ dizisinin boyutu için aşağıdaki formda bir formülün kullanılması uygun olacaktır.

$$\text{Rota dizisinin boyutu} = \left\lceil \frac{\text{Max. durak talebi}}{\text{Kapasite limiti}} \right\rceil \times \text{Durak sayısı}$$

## ÖZGEÇMİŞ

1964 yılında Rize’de doğdu. 1974 yılında ilkokulu, 1977 yılında ortaokulu, 1980 yılında liseyi ve 1985 yılında İstanbul Teknik Üniversitesi İşletme Mühendisliği bölümünü bitirdi. 1990 yılında Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü İşletme ana bilim dalında “Diskriminant Analizi” adlı tezi ile yüksek lisans öğrenimini tamamladıktan sonra Karadeniz Teknik Üniversitesi İnşaat Mühendisliği bölümünde başladığı öğrenimini 1992 yılında tamamladı. Evli olan araştırmacı 1994 yılından beri Karadeniz Teknik Üniversitesi Ünye İktisadi ve İdari Bilimler Fakültesinde Araştırma görevlisi olarak görev yapmaktadır.

T.C. YÜKSEKÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ