Uludağ Üniversitesi
İktisadi ve İdari Bilimler Fakültesi Dergisi
Cilt XXIV, Sayı 1, 2005, s. 101-114

TAMSAYILI PROGRAMLAMADA DAL KESME YÖNTEMİ VE BİR EKMEK FABRİKASINDA OLUŞTURULAN ARAÇ ROTALAMA PROBLEMİNE UYGULANMASI

Zehra BAŞKAYA[*]
Burcu AVCI ÖZTÜRK[**]

Özet

Günümüzde rekabetin artması ve teknolojinin hızla ilerlemesi ile işletmelerin kendilerini sürekli yenilemeleri gerekmektedir. Bu durumda müşteri taleplerinin zamanında ve eksiksiz olarak minimum maliyetle karşılanması büyük önem taşımaktadır. Müşteri taleplerinin zamanında ve en az maliyetle karşılanmasının planlanması için karmaşık bir optimizasyon problemi olan araç rotalama kullanılmaktadır. Klasik araç rotalama problemleri, bir merkez depodan müşterilere minimum maliyetle ürün taşınmasına dayanmaktadır. Maliyetin minimum olması için de, araçların kat ettikleri toplam yolun minimum olması gerekmektedir. Araç rotalama problemleri, 112 acil servis ambulanslarının en uygun yollarının saptanmasında, telefonla çağrılan taksilerde, toplu taşıma sisteminde, eve teslim hizmetlerinde, çöp toplama araçlarının rotalarının belirlenmesinde ve bunlar gibi daha bir çok alanda kullanılmaktadır. Bu araştırmada, bir ekmek fabrikasının 5 satış şubesine ekmek dağıtımı problemi, araç rotalama kullanılarak dal-kesme yöntemi ile çözülmüş, araçlar için en kısa yollar ve rotalar belirlenmiştir.

Anahtar Kelimeler: *Tamsayılı Programlama, Dal-kesme Yöntemi, Araç Rotalama Problemi.*

---

[*] *Doç. Dr.,Uludağ Üniversitesi, İİBF, İşletme Bölümü.*
[**] *Arş. Gör., Uludağ Üniversitesi, İİBF, İşletme Bölümü.*

Abstract

With increasing of competition and the rapid growth of technology and its increasing usage, firms need to innovate themselves. The objective of vehicle routing is to provide a high level of customer demands while keeping the operating and investment costs as low as possible. The Vehicle Routing is a complex combinatorial optimization problem which has been used in order to plan with overall minimum route cost which service all the demands. Typical Vehicle Routing Problem depends on least cost routes from one depot to a set of geographically scattered points (cities, stores, warehouses, customers). All routes need to minimum in order to minimize customer support costs. Vehicle Routing Problem has been used as designing routes for 112 emergency service ambulance and garbage collection, in calling taxis, mass transportation, home delivery and any other areas like these. In this study, delivery problem to five sales agency in a bread factory has been dissolved using Vehicle Routing and determined the shortest routes for vehicles with branch and cut approach.

Key Words: *Integer Programming, Branch And Cut Method, Vehicle Routing Problem.*

1. GİRİŞ

Tamsayılı programlama çözüm metodolojisi içerisinde gerçekleştirilen çarpıcı gelişim nedeniyle 1980'lerin ortalarından beri yöneylem araştırmasının özellikle ilgi çekici bir dalı olmuştur. Bu gelişimi ele alabilmek için tarihsel gelişimi göz önünde bulundurmak gerekmektedir. Yöntemde bir büyük atılım, 1960'larda ve 1970'lerin başında, dal-sınır yönteminin geliştirilmesi ve yönteme eklemeler yapılması ile gerçekleştirilmiştir. 100 değişkenin altındaki küçük çaptaki problemlerin oldukça etkin bir şekilde çözülebildiği fakat problemin büyüklüğü arttıkça hesaplama sürelerinde kabul edilebilir limitlerin üzerine çıkıldığı gözlenmiştir. Önce kesme düzlemi yaklaşımı ve daha sonra 1980'lerin ortasında dal-kesme yaklaşımı geliştirilmiştir. Bu iki yaklaşım kullanılarak büyük çaptaki problemlerin çözümünde önemli bir yol kat edilmiştir (Hillier ve Lieberman, 2005: 521).

Araç rotalama problemleri de matematiksel ve sezgisel bazı yöntemler ile çözülebilmektedir. Araç rotalama problemi, kısıtlı kapasiteler olan araçlar ile toplam yolculuk maliyetini minimum yapacak şekilde her aracın hangi müşterilere hangi sırada hizmet vereceğinin belirlenmesi olarak tanımlanmaktadır. Problemin NP-zor yapıda olması ve uygulama alanının genişliği uzun yıllardır hem araştırmacıların hem de uygulamacıların ilgisini çeken bir konu olmasına yol açmıştır (Alabaş ve Dengiz, 2004: 1).

Bu araştırmada, tamsayılı programlama çözüm yöntemlerinden dal-kesme yöntemi açıklanmış ve bir ekmek fabrikasının 5 satış şubesine yaptığı ekmek dağıtımı için minimum maliyetli rotaların belirlenmesini sağlayan bir araç rotalama modelinin çözümünde kullanılmıştır.

2. DAL-KESME YÖNTEMİ

Dal-kesme yöntemi tamsayılı programlama problemleri için oldukça etkili bir yöntemdir. Bu yöntem kesme düzlemi algoritması ve dal-sınır yöntemlerinin bir birleşimidir. Dal-kesme yöntemi de diğer tamsayılı programlama algoritmalarıyla (Dal-sınır, Kesme düzlemi) benzer olarak tamsayılı programlama probleminin, doğrusal programlama ile yapılacak çözümü ile başlar.

Genel bir tamsayılı programlama problemini sadece kesme düzlemi yaklaşımı ile verimli olarak çözebilmek mümkün değildir, alternatif optimum çözümleri bulmak için dallandırma yapmak da ayrıca gereklidir. Dal-sınır yaklaşımı, kesme düzlemi algoritmasının uygulanması ile oldukça hızlandırılabilir. Dallandırma yapılmadan kesme eklenebileceği gibi ağacın her düğümünün çözüm aşamasında da kesmeler kullanılabilir (Mitchell, 1998: 1–2).

2.1. Problemin Dallara Ayrılması:

Dallandırma yapılarak çözüme götürmeyen bazı çözüm seçenekleri önceden elimine edilmektedir. Bu nedenle gerekli değerlendirmelerin sayısı, genellikle, çözüm alanının boyutlarından oldukça küçüktür. Bu verimli arama yöntemi, çözüm alanını küçük alt problemlere böler. Bu alt problemlere *'dallandırma noktaları'* adı verilir. Her alt problem, daha fazla araştırma gerekip gerekmediği belirlenmek üzere değerlendirilir.

Değerlendirme, amaç fonksiyon değerlerini alt ve üst sınırlarla karşılaştırarak gerçekleştirilir. Minimizasyon sorunlarında alt problemin olurlu çözümleri için amaç fonksiyon değerlerine bir alt sınır bulunur. Başlangıçta ∞ kullanılmak üzere, eğer alt sınır $\geq$ bir üst sınır ise tüm alt problem elimine edilir. Yöntem her değerlendirmeden sonra üst sınırın yeniden düzenlenmesini olanaklı kılacak biçimde tasarlanmıştır. Dallandırma elimine edilmemiş ve en küçük alt sınırlı alt problemlerde yapılır. Sınırlar arasındaki fark, daha iyi çözümler bulundukça azalır ve optimal çözüm bulunur.

Maksimizasyon sorunlarında ise yöntem biraz değiştirilir. Her alt problem için amaç fonksiyonunun değerine üst sınırlar bulunur. Her alt problem için bu üst sınırlar, optimum amaç fonksiyon değerine ilişkin bir alt sınır ile karşılaştırılır. Başlangıçta sıfır değeri kullanılmak üzere, eğer üst sınır ≤ alt sınır ise o alt problem elimine edilir. Buraya kadar en iyi çözüm şimdiye kadar yapılan değerlendirmelerden en yüksek amaç fonksiyon değerini verendir ve o andaki alt sınır bu değerdir. Dallandırma genellikle en yüksek üst sınırı olan ve elimine edilmemiş alt problemde yapılır (Tütek ve Gümüşoğlu, 1994: 260–261).

Sonlandırılmamış alt problemlerde, en yüksek sınıra sahip olan seçilir. Bu alt problemin düğümünden, bir sonraki değişken kararlaştırılarak dallandırma yapılarak iki yeni alt problem oluşturulur (Hiller ve Lieberman, 1990: 472).

2.2. Kesmelerin Oluşturulması

Eğer dallarda elde edilen çözümler tamsayılı değilse, doğrusal programlama problemine ek bir doğrusal kısıtlayıcı eklenir. Bu ek kısıtlayıcı Gomory'nin geliştirdiği 'Kesim Düzlemi' kuralına göre elde edilir. Kesim düzlemi kuralında doğrusal programlama çözümü ile elde edilen optimal çözüm değerlerinden en büyük kesir değerli karar değişkeni seçilir. Daha sonra, bu değişkenin satırında bulunan değişkenlerin katsayıları tamsayılı ve kesirli olarak yazılır ve tamsayılı değişkenler denklemin sağ tarafında toplanır. Sağ tarafta yer alan tamsayılı değişkenler atılır ve sadece kesirli eleman bırakılır. Tamsayılı değişkenler atıldığına göre eşitlik halindeki denklem eşitsizliğe dönüşecek ve sol taraftaki kesirli kısım sağ taraftaki elemanın değerinden büyük veya eşit olacaktır. Böylece ek kısıtlayıcı elde edilmiştir ve bu ek kısıtlayıcı denklemi standart hale getirmek için yeni bir değişken eklenir. Sonra da bu ek kısıtlayıcıya tamsayılı olmayan optimal çözüm tablosunda yer verilerek tekrar doğrusal programlama çözüm işlemlerine geçilir (Öztürk, 2001: 168). Eğer çözümler yine tamsayılı olarak bulunamazsa, tekrar kesme eklenebilir ya da problem tekrar dallara ayrılabilir.

3. ARAÇ ROTALAMA PROBLEMİ (ARP)

İşletmeler, ürünlerinin müşterilerin yerleşim yerlerine ulaşımını sağlamak amacıyla birkaç araçtan oluşan araç filoları kullanırlar. Dağıtım için kullanılacak olan araç filolarının etkin bir şekilde yönetilmesi gerekliliği birtakım problemleri ortaya çıkarmaktadır. Araçların ziyaret etmesi gereken yerlerin sırasını içeren bu problemler rotalama problemleri olarak adlandırılabilir (Başkaya, 2005: 203). Bu problemlerden biri de araç rotalama problemidir (ARP). ARP, NP (deterministik olmayan polinominal) karmaşıklığına sahip ve çözülmesi zaman alan bir problemdir.

Araç rotalama problemlerinin NP-zor tipi problemler olması optimal sonuca ulaşma zamanının çok fazla olmasına sebep olmaktadır. (Avriel ve

Golany, 1996: 581). Araç rotalama problemleri tamsayılı programlama çözüm yöntemleri ile çözülebildiği gibi sezgisel ve optimuma yakın sonuç veren yöntemlerle de çözülebilmektedir. Bu araştırmada, klasik bir araç rotalama probleminin tamsayılı programlama çözüm yöntemlerinden dal-kesme yöntemi kullanılarak çözümü aranacaktır.

Araç rotalama problemi, mamullerin birden fazla araç ile bir veya birden fazla üretim merkezinden ilgili talep yerlerine minimum mesafe veya minimum maliyetle taşınmasını hedefleyen bir problem türüdür. Bu problem türünde dağıtıcı belirli kapasitelere sahip araçlardan oluşan bir filoya sahiptir (Prasanna ve Stanislaw, 2006: 1).

Araç rotalama problemlerinin önemli ve yaygın bir uygulama alanı vardır. Bir araç filosu, tek bir depoda stoklanan malları müşterilerin günlük siparişleri doğrultusunda dağıtmaktadır. İşletmelerin bazılarında, müşteriler her günün başlangıcında siparişlerini merkez depoya bildirmekte ve böylece merkez depo tarafından tahsis edilen araçlar için taleplere göre bir dağıtım programı hazırlanmaktadır. Taşıma yolları belirlenirken toplam mesafeyi minimize edecek bir rota belirlenmektedir. Tahsis edilen her aracın sabit bir kapasitesi vardır ve talepler belirli olup her sipariş için araç kapasitesinin sabit bir bölümü kullanılmaktadır (Robeson ve House, 1985: 400).

Klasik bir araç rotalama problemi aşağıdaki özellikleri taşımaktadır (Caccetta ve Hill, 2001: 524):

1. Bir çeşit ürün, tek bir depodan, bilinen taleplerle müşterilere dağıtılmaktadır.
2. Her müşterinin talebi bir araç tarafından karşılanmaktadır.
3. Her aracın kapasitesi aynıdır ve her araç tek bir tur yapmaktadır.
4. Gidilen toplam uzaklıkta bir aracın kapasitesi aşılamamaktadır.
5. Her müşteriye belirli bir zaman aralığında hizmet sunulmaktadır.
6. Araç rotalama probleminde amaç, tüm araçlar tarafından kat edilecek olan toplam uzaklığı minimize etmektir.

Araç rotalama problemlerinin sonuçları bir maliyet değerleme aracı olarak stratejik amaçlarla kullanılabilmektedir. Böyle bir çalışma için ilk önce müşteri talepleri ile ilgili önceden planlanmış bilgiler, maliyet bilgileri ve dağıtım işlemleri ile ilgili bilgiler elde edilmekte, sonra bu bilgilere dayanılarak uygun dağıtım programının belirlenmesi sağlanmaktadır (Vanrijn, 1988: 42-43).

3.1. Araç Rotalama Probleminin Matematiksel Modeli

Araç rotalama problemlerinin tamsayılı programlama çözüm yöntemleri ile çözüme ulaştırılabilmesi için öncelikle doğrusal programlama modeli olarak ifade edilmesi gerekmektedir. Tüm tamsayılı programlama problemlerinde olduğu gibi algoritmaların uygulanabilmesi için problemin doğrusal programlama çözümüne ihtiyaç duyulmaktadır.

Problemin çözümünde uzaklık matrisi $c = (c_{ij})$ şehirler arasındaki uzaklıkları vermektedir. Simetrik bir problemde $c_{ij} = c_{ji}$ olarak belirlenmektedir (Laporte ve Nobelt, 1980:2). n şehirli bir araç rotalama problemi, $n.(n - 1)/2$ elemandan oluşan bir uzaklık vektörüne sahiptir ve aslında uzaklıkları veren vektörde n^2 kadar eleman olması gerekirken $c_{ij} = c_{ji}$ ve $c_{ii} = \infty$ olduğundan aynı şehirler arasındaki uzaklıklar dikkate alınmamaktadır (Dantzing ve Johnson, 2001:1).

n toplam şehir veya depo sayısını, Q araç kapasitelerini, c_{ij} şehirler veya depolar arasındaki uzaklıkları, K araç sayısını ve D talep miktarlarını göstermek üzere araç rotalama probleminin matematiksel modeli aşağıdaki şekilde kurulmaktadır (Caccetta ve Hill, 2001: 525):

$$X_{ij} = \begin{cases} 1 \text{ Bir yerden diğerine geçiş varsa;} \\ \\ 0 \text{ Bir yerden diğerine geçiş yoksa.} \end{cases}$$

$$\text{Minimum } \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} X_{ij} \qquad c_{ii} = \infty \qquad i = 0,1,....., n \qquad j = 0,1,....., n$$

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad i \neq j \qquad i = 0,1,....., n$$

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad i \neq j \qquad j = 0,1,....., n$$

$$\sum_{j=1}^{n} X_{0j} = K$$

$$\sum_{i=1}^{n} X_{i0} = K$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} D_i \leq Q \qquad i \neq j \qquad i = 1,2,....., n \qquad j = 1,2,....., n$$

$$X_{ij} \geq 0 \text{ ve tamsayı}$$

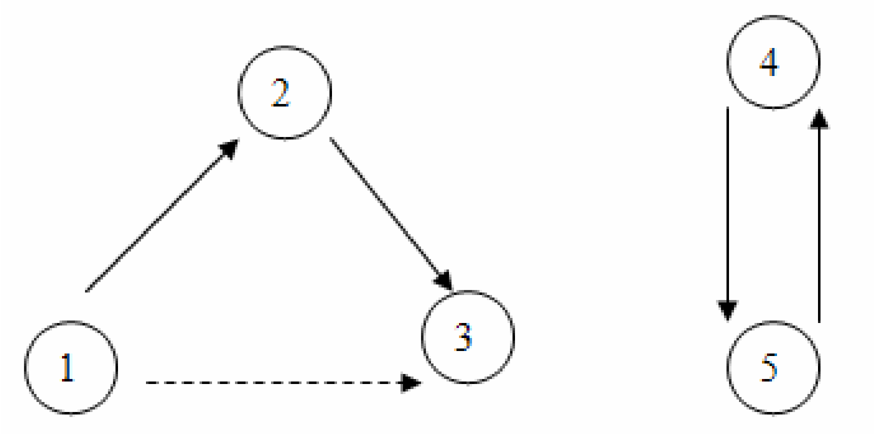
3.2. Araç Rotalama Probleminin Dal- Kesme Algoritması İle Çözümü

Dal-kesme algoritması, araç rotalama probleminin doğrusal programlama çözümü yapıldıktan sonra uygulanmaya başlamaktadır.

Araç rotalama probleminin doğasından dolayı optimum çözümün gerçekleşebilmesi için alt turların oluşmaması gerekmektedir. Alt tur örneğin 5 şehirli bir problemde tek araç olduğu durumda $X_{12} = X_{23} = X_{31}$ ve $X_{45} = X_{54}$ şeklinde oluşabilir. Bu turların şekilsel ifadesi **Şekil 3.1**'de gösterilmektedir (Fischetti ve diğerleri, 1997: 379).

Çözüm sonucunda eğer alt tur oluşmuş ise öncelikle seçilen bir alt turun oluştuğu yerler arasındaki uzaklıklar sonsuza eşitlenerek başka bir deyişle alt turu oluşturan değişkenler teker teker sıfıra eşitlenerek problem dallara ayrılmaktadır (Wagner, 1972: 471). Örneğin alt tur $X_{45} = X_{54}$ şeklinde oluşmuş ise; problem $X_{45} = 0$ ve $X_{54} = 0$ olarak iki dala ayrılmaktadır.

Problemin çözümüne yine ulaşılmadığı taktirde dallara alt tur engelleme kısıtları kesme olarak eklenmektedir.



Şekil 3.1. Alt Tur Örneği

Örneğin $X_{23} + X_{32} \leq 1$ 2. ve 3. şehirler arasındaki alt turu engellerken, $X_{23} + X_{34} + X_{42} \leq 2$ 2., 3. ve 4. şehirler arasındaki alt turu engellemektedir (Wagner, 1972: 472). Dolayısıyla alt tur engelleme kısıtları n alt turu oluşturan şehir sayısını göstermek üzere;

$X_{ij} + X_{ji} \leq 1$

$X_{ij} + X_{jk} + X_{ki} \leq 2$

$X_{ij} + X_{jk} + X_{kl} + X_{li} \leq 3$

.

.

.

.

$X_{ij} + \ldots\ldots\ldots\ldots\ldots\ldots.. + X_{ni} \leq (n - 1)$ şeklinde oluşmaktadır.

Problem eğer kısıtlayıcılar eklendiğinde de alt tur oluşmadan sonlanmazsa, tekrar dallara ayrılabilir ya da alt tur engelleme kısıtlayıcıları tekrar eklenerek alt turlar elimine edilebilir.

4. ARAŞTIRMANIN UYGULAMASI

4.1. Araştırmanın Evreni ve Örneklemi

Bir ulaştırma sorununun araç rotalama problemi olarak ifade edilip çözümünün yapılabilmesi için birden çok araç ve talep yerine sahip olan bir işletmeye ihtiyaç duyulmaktadır. Bu nedenle araştırma modeline uygun olan 5 şubeye ve 2 araca sahip bir ekmek fabrikasının ulaştırma sorunu örneklem olarak seçilmiştir.

Araştırmada örneklem olarak seçilen ekmek fabrikasının 5 satış şubesine her şubenin ihtiyacı olan ekmek sayısını karşılayarak, iki aracının, minimum maliyeti gerçekleştirebilmek için hangi rotaları izlemesi gerektiği incelenmeye çalışılmıştır. Bu amaçla, fabrikanın dağıtım sorunu bir araç rotalama problemi olarak ifade edilmiş ve dal-kesme yöntemi ile çözülmüştür.

4.2. Elde Edilen Veriler

Araştırma ile ilgili veriler fabrikanın merkez sorumlusundan alınmıştır. Ekmek fabrikasının elinde ekmekleri taşıyabileceği iki aracı bulunmaktadır, araçların kapasiteleri eşittir ve her birinin 2.800 adet ekmek taşıma kapasitesi vardır. Satış şubelerinin günlük ekmek ihtiyacı **Tablo**

4.1'de, merkez ve şubeler arasındaki uzaklıklar da km olarak **Tablo 4.2**'de gösterilmektedir.

Tablo 4.1. Şubelerin Günlük Ekmek İhtiyacı

Şubeler	Günlük Ekmek İhtiyacı (adet)
1. Şube (1)	550
2. Şube (2)	630
3. Şube (3)	750
4. Şube (4)	650
5. Şube (5)	740
Toplam	3.320

Tablo 4.2. Şubeler Arasındaki Uzaklıklar

	Merkez	1. Şube	2. Şube	3. Şube	4. Şube	5. Şube
Merkez (0)	0	46	51	35	60	22
1. Şube (1)	46	0	95	24	66	32
2. Şube (2)	51	95	0	86	92	67
3. Şube (3)	35	24	86	0	46	16
4. Şube (4)	60	66	92	46	0	49
5. Şube (5)	22	32	67	16	49	0

4.3. Araştırmanın Modeli

Elde edilen veriler ışığında model araç rotalama problemi biçiminde ifade edilecektir. Ekmek fabrikasının şubeleri arasındaki ekmek dağıtımındaki rotasını belirleyecek olan matematiksel model aşağıda gösterilmektedir.

Amaç fonksiyonu, maliyetin dolayısıyla gidilecek mesafenin minimum olarak hesaplanmasına uygun şekilde düzenlenmiştir, bu problemde amaç fonksiyonunun açılımı aşağıda yer almaktadır:

$Z_{MİN} = 46X_{01} + 51X_{02} + 35X_{03} + 60X_{04} + 22X_{05} + 46X_{10} + 95X_{12} + 24X_{13} + 66X_{14} + 32X_{15} + 51X_{20} + 95X_{21} + 86X_{23} + 92X_{24} + 67X_{25} + 35X_{30} + 24X_{31} + 86X_{32} + 46X_{34} + 16X_{35} + 60X_{40} + 66X_{41} + 92X_{42} + 46X_{43} + 49X_{45} + 22X_{50} + 32X_{51} + 67X_{52} + 16X_{53} + 49X_{54}$

Fabrikanın elinde 2 aracı vardır. Araçların taşıyabilecekleri maksimum kapasiteler toplam talebi karşılayabilecek durumdadır. Toplam Kapasite/Araç Kapasiteleri = 3.320/2.800 = 1,15. Bu durumda fabrikanın toplam kapasiteyi karşılayabilmek için iki araca ihtiyacı olduğu görülmektedir. Modelin kısıtlayıcıları ise aşağıda belirtilmiştir.

Depodan 2 adet araç çıkmaktadır.

$X_{01} + X_{02} + X_{03} + X_{04} + X_{05} = 2$

Depoya 2 adet araç dönmektedir.

$X_{10} + X_{20} + X_{30} + X_{40} + X_{50} = 2$

Bir talep merkezinden yalnızca bir talep merkezine gidilebilmektedir.

$X_{10} + X_{12} + X_{13} + X_{14} + X_{15} = 1$

$X_{20} + X_{21} + X_{23} + X_{24} + X_{25} = 1$

$X_{30} + X_{31} + X_{32} + X_{34} + X_{35} = 1$

$X_{40} + X_{41} + X_{42} + X_{43} + X_{45} = 1$

$X_{50} + X_{51} + X_{52} + X_{53} + X_{54} = 1$

Bir talep merkezine yalnızca bir talep merkezinden gelinebilmektedir.

$X_{01} + X_{21} + X_{31} + X_{41} + X_{51} = 1$

$X_{02} + X_{12} + X_{32} + X_{42} + X_{52} = 1$

$X_{03} + X_{13} + X_{23} + X_{43} + X_{53} = 1$

$X_{04} + X_{14} + X_{24} + X_{34} + X_{54} = 1$

$X_{05} + X_{15} + X_{25} + X_{35} + X_{45} = 1$

Rota üzerindeki talepler toplamı, kapasiteyi aşmamalıdır.

$630X_{21} + 750X_{31} + 650X_{41} + 740X_{51} \leq 2.800$

$550X_{12} + 750X_{32} + 650X_{42} + 740X_{52} \leq 2.800$

$550X_{13} + 630X_{23} + 650X_{43} + 740X_{53} \leq 2.800$

$550X_{14} + 630X_{24} + 750X_{34} + 740X_{54} \leq 2.800$

$550X_{15} + 630X_{25} + 750X_{35} + 650X_{45} \leq 2.800$

$X_{ij} \geq 0$ ve tamsayı olmalıdır.

4.4. Bulgular

4.4.1. Araç Rotalama Probleminin Doğrusal Programlama Çözümü

Matematiksel modelin Win QSB bilgisayar programı kullanılarak elde edilen doğrusal programlama çözümü, $X_{02} = X_{20} = 1$, $X_{04} = X_{45} = X_{50} = 1$, $X_{13} = X_{31} = 1$ ve Minimum Z = 281 km olarak bulunmuştur. Bu çözüm, problem için uygun çözüm değildir çünkü, ekmek fabrikasının iki aracı vardır ve optimum çözümün de iki rotadan oluşması gerekmektedir. Çözümde üç rota bulunmaktadır. Bir araç merkez depodan 2. şubeye gitmekte ve tekrar merkez depoya dönmektedir. Bir araç, yine merkez depodan çıkıp 4. şubeye, daha sonra 5. şubeye gitmekte ve oradan merkez depoya dönmektedir. Diğer bir araç ta, 1. şubeden çıkıp 3. şubeye gitmekte ve 1. şubeye geri dönmektedir. $X_{13} = X_{31} = 1$ turu, araç rotalama problemlerinin özellikleri ile çelişmektedir. Her araç, merkez depodan çıkmalı ve yine merkez depoya dönmelidir. Fakat bu tur bir aracın 1. şubeden, 3. şubeye gideceğini ve 3. şubeden yine 1. şubeye döneceğini ifade etmektedir. Bu ifade bir alt turu göstermektedir. Bu nedenle problem alt turu oluşturan değişkenler teker teker sıfıra eşitlenerek dallara ayrılacaktır. Birinci dal, $X_{13} = 0$ ve ikinci dal $X_{31} = 0$ şeklinde oluşacaktır.

4.4.2. Birinci Alt Problem

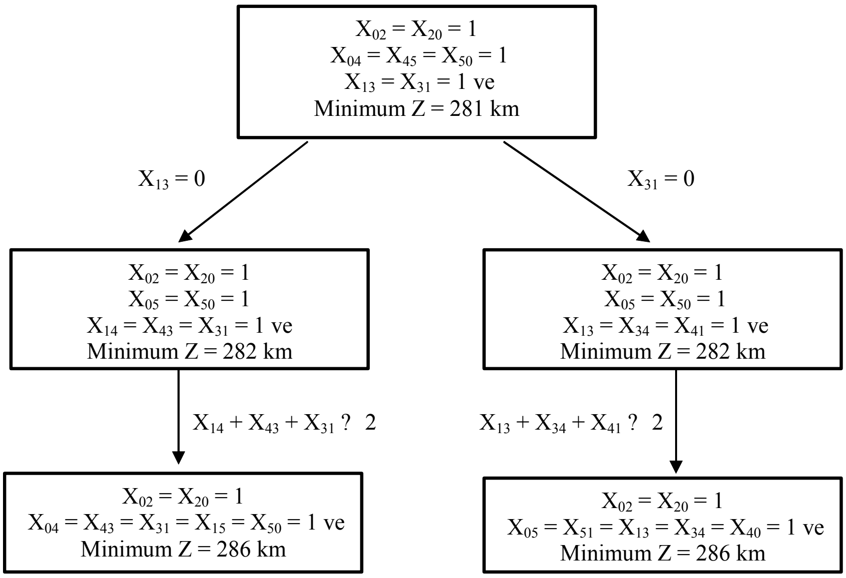
Modelde $X_{13} = 0$ yazılarak birinci alt problem oluşturulmuştur. Bu ifade, problemin doğrusal programlama çözümünde elde edilen alt turu engelleyecektir. Model tekrar çözüldüğünde alt problemin sonucu, $X_{02} = X_{20} = 1$, $X_{05} = X_{50} = 1$, $X_{14} = X_{43} = X_{31} = 1$ ve Minimum Z = 282 km olarak bulunmaktadır. Görüldüğü gibi 3 adet tur oluşmuştur ve 3 araca ihtiyaç duyulmaktadır. Optimum çözüme ulaşılamamıştır ve oluşan alt turun elimine edilebilmesi için modele alt tur engelleme kısıtlayıcısı eklenecektir. Alt turu oluşturan tur, $X_{14} = X_{43} = X_{31} = 1$ rotasıdır. Çünkü herhangi bir aracın 1. şubeden çıkarak yine 1. şubeye dönmesi mümkün değildir. Bu tur için alt tur engelleme kısıtlayıcısı $X_{14} + X_{43} + X_{31} \leq 2$ olmaktadır.

Kısıtlayıcı modele eklendikten sonra, doğrusal programlama çözümü yapılmış ve $X_{02} = X_{20} = 1$, $X_{04} = X_{43} = X_{31} = X_{15} = X_{50} = 1$ ve Minimum Z = 286 km sonucu elde edilmiştir. Burada 2 tur oluşmuştur ve 2 araca ihtiyaç duyulmaktadır. Alt tur oluşmadığından problem bu dalda sonlandırılacaktır. Bir araç, merkez depodan çıkacak 2. şubeye gidecek ve yine merkez depoya geri dönecektir. Diğer bir araç ise, merkez depodan 4. şubeye, oradan 3. Şubeye, sonra 1. şubeye, daha sonra 5. şubeye gidecek ve merkez depoya geri dönecektir. Toplam kat ettikleri minimum uzaklık ta 286 km olacaktır. Bu aşamadan sonra ikinci alt problem çözülecek ve daha uygun bir çözüm aranacaktır. Daha uygun bir çözüm bulunmadığı taktirde, Minimum Z = 286 km optimum çözüm olarak belirlenecektir.

4.4.3. İkinci Alt Problem

Modelde $X_{31} = 0$ yazılarak ikinci alt problem oluşturulmuştur. Model tekrar çözüldüğünde alt problemin sonucu, $X_{02} = X_{20} = 1$, $X_{05} = X_{50} = 1$, $X_{13} = X_{34} = X_{41} = 1$ ve Minimum Z = 282 km olarak bulunmaktadır. Bu çözümde de görüldüğü gibi yine 3 adet tur oluşmuştur ve 3 araca ihtiyaç duyulmaktadır. Optimum çözüme ulaşılamamıştır ve oluşan alt turun elimine edilebilmesi için modele alt tur engelleme kısıtlayıcısı eklenecektir. Alt turu oluşturan tur, $X_{13} = X_{34} = X_{41} = 1$ rotasıdır. Çünkü herhangi bir aracın 1. şubeden çıkarak yine 1. şubeye dönmesi mümkün değildir. Bu tur için alt tur engelleme kısıtlayıcısı $X_{13} + X_{34} + X_{41} \leq 2$ olmaktadır.

Kısıtlayıcı modele eklendikten sonra, tekrar doğrusal programlama çözümü yapılmış ve $X_{02} = X_{20} = 1$, $X_{05} = X_{51} = X_{13} = X_{34} = X_{40} = 1$ ve Minimum Z = 286 km sonucu elde edilmiştir. Burada da birinci alt problemin sonucunda olduğu gibi yine 2 tur oluşmuştur ve 2 araca ihtiyaç duyulmaktadır. Alt tur oluşmadığından problem bu dalda da sonlandırılacaktır. Bir araç, merkez depodan çıkacak 2. şubeye gidecek ve yine merkez depoya geri dönecektir. Diğer bir araç ise, merkez depodan 5. şubeye, oradan 1. şubeye, sonra 3. şubeye, daha sonra 4. şubeye gidecek ve merkez depoya geri dönecektir. Toplam kat ettikleri minimum uzaklık ta yine 286 km olacaktır. Bu sonuca göre, daha uygun bir çözüm yoktur. Çünkü, iki alt problemin sonucu da 286 km olarak bulunmuştur ve alternatif bir tur oluşmuştur. Her iki alt problemde bulunan rotalar da optimumdur. Problemin çözümü **Şekil 4.1**'de özetlenmiştir.



Şekil 4.1. Araç Rotalama Probleminin Çözümü

5. SONUÇ

Araç rotalama problemleri, öncelikle müşteri taleplerinin eldeki araçlarla karşılanıp karşılanamayacağının ölçümünü sağlamaktadır. Eğer bu talepleri en iyi şekilde karşılamak mümkün ise, bu durumun en az maliyetle nasıl sağlanacağının cevabını da araç rotalama probleminin çözümü vermektedir. Araçlar için en uygun rotanın saptanması, hem teslim sürecinin işleyişinin hem de işletmenin taşıma maliyetlerinin kontrolünün etkin bir şekilde gerçekleştirilmesini sağlamaktadır.

Bu araştırmada, bir ekmek fabrikasının iki aracı için 5 şubesine yapacağı dağıtımın toplam uzaklığını minimum yapacak araç rotalama modeli kurulmuş ve tamsayılı programlama çözüm yöntemlerinden dal-kesme yöntemi ile çözülmeye çalışılmıştır. Dal-kesme yöntemi uygulamada süre tasarrufu sağlamakta ve alternatif rotaların belirlenmesini kolaylaştırmaktadır.

Modelin çözümü sonucuna göre ekmek fabrikasının 2 aracının kat etmesi gereken optimum uzaklık 286 km'dir. Bu minimum uzaklığı sağlayan iki farklı rota tespit edilmiştir.

1. Rota:

1. Aracın rotası, Merkez Depo – 2. Şube – Merkez Depo ($X_{02} = X_{20} = 1$) olarak belirlenmiştir. Taşınan miktar ise 630 adettir.

2. Aracın rotası ise, Merkez Depo – 4. Şube – 3. Şube – 1. Şube – 5. Şube – Merkez Depo ($X_{04} = X_{43} = X_{31} = X_{15} = X_{50} = 1$) olarak belirlenmiştir. Taşınan miktar ise, 2.690 adettir.

2. Rota:

1. Aracın rotası, ilk rota ile aynıdır ve Merkez Depo – 2. Şube – Merkez Depo ($X_{02} = X_{20} = 1$) olarak belirlenmiştir. Taşınan miktar ise 630 adettir.

2. Aracın rotası ise, Merkez Depo – 5. Şube – 1. Şube – 3. Şube – 4. Şube – Merkez Depo ($X_{05} = X_{51} = X_{13} = X_{34} = X_{40} = 1$) olarak belirlenmiştir. Görüldüğü gibi sadece ikinci araçların rotaları farklıdır ve alternatif rota oluşmuştur. Gidilecek yol aynıdır fakat önce gidilecek olan şube değişiklik göstermektedir. Taşınan miktar ise yine, 2.690 adettir.

1. araç ile toplam 630 adet ekmek taşınacak ve toplam 102 km yol kat edilecektir. 2. araç ile toplam 2.690 adet ekmek taşınacak ve toplam 184 km yol kat edilecektir.

Ekmek fabrikası araçların rotalarını problemin optimum sonucunda belirtilen şekilde ayarladığında minimum yol olan 286 km' yi elde edecektir. Bu durum da minimum taşıma maliyetinin gerçekleştirmesini sağlayacaktır.

KAYNAKÇA

Alabaş, Çiğdem – Dengiz, Berna, (2004), "Yerel Arama Yöntemlerinde Yöre Yapısı: Araç Rotalama Problemine Bir Uygulama", Yöneylem Araştırması/Endüstri Mühendisliği-XXIV Ulusal Kongresi, 15–18 Haziran 2004, Gaziantep-Adana.

Avriel, Mordecai – Golany, Boaz, (1996), Mathematical Programming for Industrial Engineers, Marcel Dekker Inc., New York.

Başkaya, Zehra (2005), Tamsayılı Programlama Algoritmaları ve Bilgisayar Uygulamalı Problem Çözümleri, Ekin Kitabevi, Bursa.

Caccetta L, – Hill S.P., (2001), "Branch And Cut Methods For Network Optimization", Mathematical And Computer Modeling, 33, ss: 517-532.

Dantzig, Fulkerson and Johnson's Cutting Plane Method, (05.12.2006) http://www.tsp.gatech.edu/methods/dfj/index.html

Fischetti, Matteo – Gonzalez, Juan Jose Salazar – Toth, Paolo, (1997), "A Branch-And-Cut Algorithm for the Symmetric Generalized Travelling Salesman Problem", Operations Research, Volume 45, Issue 3.

Hillier, Frederick S. Hillier – Gerald J. Lieberman, (1990), Introduction To Operations Research, Mc Graw-Hill Publishing Company, USA.

Hillier, Frederick S. Hillier – Gerald J. Lieberman,(2005), Introduction To Operations Research, Eighth Edition, McGraw - Hill Publishing Company, Boston.

Laporte, Gilbert – Nobert, Yves, (1880), "A Cutting Plane Algorithm for the M-Salesman Problem", Operations Research Society, Vol 31.

Mitchell, John E., (1998), Branch-And-Cut Algorithms for Integer Programming, Mathematical Sciences, Rensselaer Polytechnic Institute, USA.

Öztürk, Ahmet, (2001), Yöneylem Araştırması, Ekin Kitabevi, Bursa.

Prasanna, Peiris – Stanislaw H. Zak, "Solving Vehicle Routing Problem Using Genetic Algorithms", (07.12.2006)

http://www.ece.purdue.edu/ECE/Research/ARS/ARS2000/PART_I/Section1/1_19.whtml

Robeson, James F. ROBESON – House, Robert G., (1985), The Distribution Handbook, Collier Macmillan Publishers, London.

Taha, Hamdy A., (1992), Operations Research, Macmillan Publishing Company, New York.

Tütek, Hülya H. – Gümüşoğlu, Şevkinaz, (1994), Sayısal Yöntemler,Beta Basım Yayım, İstanbul.

Vanrijn, C.F.H., (1988), Logistics:Where Ends Have To Meet, Pergamon Press, New York.

Wagner, Harvey M., (1972), Principles of Operations Research, Prentice Hall, London.