# OPTİMİZASYON MODELLERİ VE ÇÖZÜM METODLARI

**Doç.Dr. Metin Türkay**

**Koç Üniversitesi, Endüstri Mühendisliği Bölümü, Rumelifeneri Yolu, Sarıyer 34450 İstanbul**

**Tel: 0 212 338 1586, Fax: 0 212 338 1548, E-Posta: mturkay@ku.edu.tr**

## 1. GİRİŞ

Optimizasyon, bir sistemde varolan kaynakların (işgücü, zaman, kapital, süreçler, hammaddeler, kapasite, ekipman gibi) en verimli şekilde kullanılarak belirli amaçlara (maliyet enazaltılması, kâr ençoklanması, kapasite kullanımının enyükseltilmesi ve verimliliğin ençoklanması gibi) ulaşmayı sağlayan bir teknoloji olarak tanımlanmaktadır (Gass, 2000).

Optimizasyonda modelleme ve çözümleme iki önemli bileşen olarak nitelendirilmektedir. Modelleme gerçek yaşamda karşılaşılan problemin matematiksel olarak ifade edilmesi; çözümleme ise bu modeli sağlayan en iyi çözümün elde edilmesini kapsamaktadır. Optimizasyon teknolojisinin gelişiminde araştırmacılar öncelikli olarak modellemeyle ilgilenmişlerdir. Bu alandaki ilk çalışmalar Leontief tarafından Amerika Birleşik Devletleri'nin dış ticaretini ve ekonomik yapısını modellemek amacıyla yaptığı yayınlardır (Leontief, 1933; 1936). Rus matematikçisi Kantorovich üretim planlamasında en sıklıkla karşılaşılan problemlerin modellenmesine ve elde edilebilecek en iyi sonuçları bulma metodlarını anlattığı makalesiyle modern üretim sistemlerinde optimizasyona olan ihtiyacı ortaya koymuştur (Kantorovich, 1939). Fakat, Kantorovich'in bu çalışması batılı bilim insanları tarafından uzun süre sonra farkedilmiştir (Kantorovich, 1960). Kantorovich, üretim sistemlerinin performansının arttırımına yönelik dokuz farklı optimizasyon problemi tanımlamış ve bu problemlerin çözümüne yönelik olarak her problem için farklı çözüm algoritması geliştirmiştir. Öte yandan, Kantorovich ve Gavurin (1940) ulaşım sektörünün verimliliğini arttırmaya yönelik modelleme çalışmaları da yapmışlardır. Ayrıca, ekonomi alanında optimum kapasite kullanımına yönelik özellikle ulaştırma sistemlerinde modelleme çalışmaları yapılmıştır (Koopmans, 1949; Koopmans, 1951; Koopmans ve Reiter, 1951; Koopmans, 1957; Koopmans ve Bausch, 1959). Bu modelleme çalışmaları bilim dünyası tarafından kabul görmüş ve Leontief 1973 yılında Nodel Ekonomi Ödülünü almıştır, Kantorovich ve Koopmans ise 1975 yılında Nobel Ekonomi Ödülü'nü paylaşmışlardır (Nobel, 2006). Optimizasyon modellerinin özellikle ekonomik sistemlerde kullanılması ve üretim/dağıtım sistemlerinde karşılaşılan problemlerin birçoğunun optimizasyon problemi olarak modellenmesine rağmen optimizasyon modellerinin teorik özelliklerinin araştırılması ve genel çözüm algoritmalarının geliştirilmesi halen devam etmektedir. Optimizasyon problemlerinin çözümüne yönelik olarak ilk önemli çalışma Dantzig tarafından yapılmış ve simpleks algoritması geliştirilmiştir (Dantzig, 1949a). Nobel Ekonomi ödülünü 1975 yılında alan Koopmans, Dantzig'in çalışmalarının önemine inanmış ve Dantzig'in bu ödüle ortak olması gerektiğini Nobel ödül komitesine belirtmiştir. Fakat bu cağrısına cevap alamamış ve Nobel ödülünün üçte birlik kısmını Uluslararası Uygulamalı Sistem Analizi Enstitüsü'nde (*International Institute for Applied Systems Analysis-ILASA*) George Dantzig adına kurulan burs programına bağışlamıştır (Gass ve Assad, 2004). Koopmans, optimizasyon projelerinde modelleme ve çözüm geliştirme çalışmalarının birlikte yürütülmesi gerekliliğine inanmış ve bu konunun önemini Nobel Ödülü'nü aldığının açıklanmasından hemen sonra yaptığı bu jestle ifade etmiştir.

Optimizasyon teknolojisi, karar verme süreçlerini hızlandırmakta ve karar kalitesini arttırmakta kullanılarak gerçek hayatta karşılaşılan problemlerin etkin, doğru ve gerçek zamanlı çözümünde yararlanılmaktadır (Winston, 2003). Optimizasyon, ekonomik açılardan getirdiği kazançların yanında müşteri, işveren ve çalışanların tercih ve kısıtlarının karar sürecinde yer almasında ve sistemde yer alan kaynakların kalitesinin yükseltilmesinde de etkin bir şekilde başvurulan bir yöntem olarak kullanılmaktadır. Optimizasyon teknolojisi kullanılarak gerçekleştirilen projelerin anlatıldığı Interfaces dergisi 1971 yılında yayın hayatına geçmiş ve düzenli olarak uygulamalı projeler için geliştirilen modeller, çözüm yöntemleri ve sonuçları yayınlanmaktadır (INFORMS, 2006). Bu derginin her yıl yayınlanan Ocak ayı sayısı bir önceki yılda tamamlanan en başarılı uygulamaları detaylarıyla tanıtmaktadır.

Birçok temel bilimde karşılaşılan problemlerin çözümünde; endüstriyel, finansal ve servis sistemlerinin performanslarının eniyilenmesinde sık sık kullanılan optimazsyon teknolojisi bir projede kullanıldığında genelde modelleme ve çözümleme ön plana çıkmaktadır (Hillier ve Lieberman, 2005). Optimizasyon modelleri 2. kısımda tanıtılıp sınıflandırılmıştır. Optimizasyon problemleri için çözüm yöntemleri ise 3. kısımda anlatılmıştır.

## 2. OPTİMİZASYON MODELLERİNİN OLUŞTURULMASI

Modeller, temel bilimlerde ve mühendislikte yoğun olarak kullanılan, büyük kapsamlı bir sistemin tüm özelliklerini yansıtacak daha küçük boyutlardaki yapılardır. Modeller, genelde sistemin temel özelliklerini yansıtacak ve modelin kullanım amaçlarını gerçekci olarak içerecek detaylar bulunur. Örneğin, tasarım aşamasında olan bir uçağı düşürdüğümüz zaman, uçağın aero-dinamik yapısını incelerken gerçek uçak yerine uçağın modeli kullanılarak rüzgâr tüneli deneyleri yapılır. Tarımda ise bir bitkinin tüm özellikleri incelenip bitkinin veriminde iyileştirme çalışmaları yapılırken, bitkinin modelleri laboratuvar ortamında değişik parametrelere göre değerlendirilip sonuçlar analiz edilir.

Optimizasyon modelleri ise sistemin işleyişini ve özelliklerini yansıtan, sistemin içindeki ve çevresindeki diğer sistemlerle olan etkileşimleri kapsayan matematiksel ifadelerden oluşmaktadır (Williams, 1999). Aşağıda da görüldüğü gibi, bu matematiksel ifadeler sistemin ölçülebilen özelliklerini belirleyen parametreler, en iyi sonuçları verecek karar değerlerini belirleyen değişkenlerden, sistemin eniyilenecek performans ölçütünden ve sistemin özelliklerini ve sınırlarını belirleyen kısıtlardan oluşmaktadır:

$$
\begin{aligned}
\text{max}\;\; & z = f(x,y) \\
\text{k.s.}\;\; & g(x,y) = 0 \\
& h(x,y) \le 0 \\
& x \in \Re^n \\
& y \in \{0,1,2,\ldots,m\}
\end{aligned}
\tag{1}
$$

Yukarıdaki optimizasyon probleminde sistemin performans ölçütü (amaç fonksiyonu) $z=f(x,y)$ ile ifade edilmiş ve karar değişkenleri $x$ ve $y$'nin bu ölçütü ençoklayacak değerlerinin bulunması hedeflenmektedir. Sistemin özellikleri ise $g(x,y)$ eşitliği ve $h(x,y)$ eşitsizlikleri (kısıtlar) belirlemektedir. Ayrıca, karar değişkenleri iki türlü ifade edilmiştir: $n$ boyutlu uzayda herhangi bir reel değeri alabilen sürekli değişkenler ($x$) ve herhangi bir tamsayı değeri alabilen tamsayılı değişkenler ($y$). Optimizasyon modellerini içerdikleri karar değişkenlerinin, amaç fonksiyonunun ve sistem kısıtlarının özelliklerine göre sistem parametrelerinin bilinen sabit değerlere aldığı durumlarda aşağıdaki gibi sınıflandırılmaktadır (NEOS, 2006). Eğer bir optimizasyon problemleminde $y$ değişkenleri yer almıyorsa ve $f(x)$, $g(x)$ ve $h(x)$ fonksiyonları doğrusalsa o problem bir doğrusal programlama problemi olarak tanımlanır. Bir optimizasyon probleminde $y$ değişkenleri yer almıyorsa ve $f(x)$, $g(x)$ ve $h(x)$ fonksiyonların herhangi birisi doğrusal değilse o problem bir doğrusal olmayan programlama problemidir. Optimizasyon problemlerinde $y$ değişkenleri yer alıyorsa $f(x,y)$, $g(x,y)$ ve $h(x,y)$ fonksiyonlarının doğrusal olması durumda problem tamsayı karışık doğrusal programlama problemi, $f(x,y)$, $g(x,y)$ ve $h(x,y)$ fonksiyonlarından herhangi birisinin doğrusal olmaması durumunda ise tamsayı karışık doğrusal olmayan programlama elde edilir.

### 2.1. Doğrusal Programlama Modelleri

Bu problemlerde sadece sürekli değişkenler ve doğrusal amaç fonsiyonuyla, doğrusal kısıtlar mevcuttur.

$$\begin{aligned} \text{max} \quad & z = c^T x \\ \text{k.s.} \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{2}$$

Doğrusal programlama modelleri en yaygın olarak kullanılan optimizasyon modelleridir ve bilimsel, endüstriyel ve ekonomik problemlerin modellenmesinde yaygın olarak kullanılmıştır. Bu bağlamda, üretim yapan bir işletmede verimliliğin optimizasyonuyla ilgili basit bir doğrusal programlama model aşağıda verilmiştir.

Bir işletmede iki yeni ürünü iki farklı fabrikada üretilebilmek imkanı vardır. Bu işletme için geçerli olan veriler Table 1'de verilmiştir. Her iki fabrikada yeni ürünün üretimi için kullanılabilecek zaman kısıtlıdır. Fabrikaların yapıları farklı olduğu için birim ürün için farklı üretim zamanları vardır. Ayrıca her ürünün kar marjı ve talebi de belirlidir.

Tablo 1. Örnek problem için veriler.

| | Birim ürün için harcanan üretim zamanı | | Kullanılabilir toplam üretim zamanı |
|---|---|---|---|
| | Ürün 1 | Ürün 2 | |
| Fabrika 1 | 1 | 1 | 100 |
| Fabrika 2 | 2 | 1.5 | 170 |
| Kar Marjı | 1.000 | 900 | |
| Talep | 100 | 100 | |

Bu işletmede karı ençoklayan ürün dağılımı aşağıdaki gibi modellenebilir:

*Karar Değişkenleri:*
- $x_1$ : Ürün 1'in üretim miktarı (adet)
- $x_2$ : Ürün 2'nin üretim miktarı (adet)

*Parametreler:*
- $c_1$: Ürün 1'den elde edilen kar miktarı (1.000 TL/adet)

- $c_2$: Ürün 2'den elde edilen kar miktarı (900 TL/adet)
- $b_i$: Fabrika 1'deki kullanılabilir toplam üretim zamanı (100 saat)
- $b_{ii}$: Fabrika 1'deki kullanılabilir toplam üretim zamanı (170 saat)
- $a_{i1}$: Ürün 1'in Fabrika 1'deki üretim zamanı (1 saat/adet)
- $a_{i2}$: Ürün 2'nin Fabrika 1'deki üretim zamanı (1 saat/adet)
- $a_{ii1}$: Ürün 1'in Fabrika 2'deki üretim zamanı (2 saat/adet)
- $a_{ii2}$: Ürün 2'nin Fabrika 2'deki üretim zamanı (1.5 saat/adet)
- $d_1$ : Ürün 1 için belirlenmiş talep miktarı (100 adet)
- $d_2$ : Ürün 2 için belirlenmiş talep miktarı (100 adet)

*Amaç Fonksiyonu:*

- Toplam karın ençoklanması ($z=c_1x_1+c_2x_2$)

*Kısıtlar:*

- Fabrika 1'deki toplam üretim zamanı kısıtlıdır ($a_{i1}x_1+a_{i2}x_2 \leq b_i$)
- Fabrika 2'deki toplam üretim zamanı kısıtlıdır ($a_{ii1}x_1+a_{ii2}x_2 \leq b_i$)
- Ürün 1 için talep miktarı ($x_1 \leq d_1$)
- Ürün 2 için talep miktarı ($x_2 \leq d_2$)
- Her ürün için en düşük üretim miktarı ($x_1 \geq 0$ ve $x_2 \geq 0$)

Bu problem için doğrusal programlama modeli aşağıdaki gibi oluşturulur:

$$
\begin{aligned}
\max\ & z = 1.000x_1 + 900x_2 \\
\text{k.s.}\ & 1x_1 + 1x_2 \leq 100 \\
& 2x_1 + 1.5x_2 \leq 170 \\
& x_1 \leq 100 \\
& x_2 \leq 100 \\
& x_1 \geq 0,\quad x_2 \geq 0
\end{aligned} \tag{3}
$$

### 2.2. Tamsayı-Karışık Doğrusal Programlama Modelleri

Bu modellerde doğrusal programlama özelliklerine ek olarak karar değişkenlerinde bazıları sadece tamsayılı değerler alabilirler.

$$
\begin{aligned}
\max\ & z = c^T x + d^T y \\
\text{k.s.}\ & Ax + Ey \leq b \\
& x \geq 0 \\
& y \in \{0,1,2,\ldots,n\}
\end{aligned} \tag{4}
$$

Yukarıda verdiğimiz doğrusal programlama modelinde iki üründen sadece birisini seçebileceğimizi düşünelim. Eğer birinci ürünü seçersek fabrikalarda gerekli düzenlemelri yapabilmemiz için bir defaya mahsus olmak üzere 1.500 TL masraf yapmamız gerekmektedir ve ikinci ürün içinse 1.200 TL masraf gerekmektedir. Doğrusal programlama modeline aşağıdaki eklemeleri yapmamız gerekmektedir:

*Karar Değişkenleri:*

- $y_1$ = birinci ürün seçimi (=1 eğer birinci ürün seçildiyse, =0 eğer birinci ürün seçilmediyse)
- $y_2$ = ikinci ürün seçimi (=1 eğer ikinci ürün seçildiyse, =0 eğer ikinci ürün seçilmediyse)

*Parametreler:*

- $e_1$: birinci ürün seçildiğinde yapılacak masraf (1.500 TL)
- $e_2$: ikinci ürün seçildiğinde yapılacak masraf (1.200 TL)

*Amaç Fonksiyonu:*

- Toplam karın ençoklanması ($z=c_1x_1+c_2x_2-e_1x_1-e_2x_2$)

*Kısıtlar:*

- Sadece bir ürün seçilebilir ($y_1+y_2=1$)

Bu problem için doğrusal programlama modeli (5):

$$
\begin{array}{ll}
\max & z = 1.000x_1 + 900x_2 - 1.500y_1 - 1.200y_2 \\
\text{k.s.} & x_1 + x_2 \le 100 \\
& 2x_1 + 1{,}5x_2 \le 170 \\
& x_1 - 100y_1 \le 0 \\
& x_2 - 100y_2 \le 0 \\
& y_1 + y_2 = 1 \\
& x_1 \ge 0,\ \ x_2 \ge 0,\ \ y_1 \in \{0,1\},\ y_2 \in \{0,1\}
\end{array} \tag{5}
$$

## 2.3. Doğrusal Olmayan Programlama Modelleri

Bu modellerde amaç fonksiyonu ve/veya kısıtlardan bazıları doğrusal değildir. Optimizasyon sonucunda amaç fonksiyonunu eniyileyecek karar değişkenleri, $x$, $n$ boyutlu uzayda herhangi bir gerçek değeri alabilirler.

$$
\begin{array}{ll}
\max & z = f(x) \\
\text{k.s.} & g(x) = 0 \\
& h(x) \le 0 \\
& x \in \Re^n
\end{array} \tag{6}
$$

Doğrusal programlama örnek modelimizde her dönüm için elde edeceğimiz karın elastik olduğunu düşünelim: tarım arazimizi bir ürün için ne kadar fazla kullanırsak kar her dönümden elde edeceğimiz kar miktarı azalmaktadır. Bu durumda birim kar miktarları, $c_1$ ve $c_2$, (7)'deki gibi ifade edilirler:

$$
c_1 = \frac{1.000}{(x_1)^{1/E_1}},\ \ c_2 = \frac{900}{(x_2)^{1/E_2}} \tag{7}
$$

Bu ifadelerde $E_1$ ve $E_2$ kar elastikliğini temsil etmektedir. Örnek problemimizde birinci ürün için bu değerin 15 ve ikinci ürün içinse 10 olduğu durumda doğrusal olmayan eniyileme problemi (8)'de verildiği şekilde oluşur:

$$
\begin{array}{lll}
\max\ z = \dfrac{1.000}{(x_1)^{1/15}}x_1 + \dfrac{900}{(x_2)^{1/10}}x_2 & & \\
\text{k.s.} \quad x_1 + \quad x_2 \le 100 & & \\
\quad 2x_1 + \quad 1{,}5x_2 \le 170 & & (8) \\
\quad x_1 + \quad \le 100 & & \\
\quad x_2 \le 100 & & \\
\quad x_1 \ge 0, \quad x_2 \ge 0 & &
\end{array}
$$

### 2.4. Tamsayı-Karışık Doğrusal Olmayan Programlama Modelleri

Bu modeller amaç fonksiyonunda ve/veya kısıtlarda doğrusal olmayan ifadeleri ve tamsayılı değişkenleri kapsamaktadır. Bu modellerin genel hali bu bölümün başında verilmiştir. Örnek problemimizde kar esnekliğinin olduğu ve iki üründen sadece birisini seçmemiz gerektiği durumda optimizasyon modeli aşağıdaki gibidir:

$$
\begin{array}{ll}
\max\ z = \dfrac{1.000}{(x_1)^{1/15}}x_1 + \dfrac{900}{(x_2)^{1/10}}x_2 - 1.500y_1 - 1.200y_2 & \\
\text{k.s.} \quad x_1 + x_2 \le 100 & \\
\quad 2x_1 + 1{,}5x_2 \le 170 & \\
\quad x_1 - 100y_1 \le 0 & (9) \\
\quad x_2 - 100y_2 \le 0 & \\
\quad y_1 + y_2 = 1 & \\
\quad x_1 \ge 0, \quad x_2 \ge 0, \quad y_1 \in \{0,1\},\ y_2 \in \{0,1\} &
\end{array}
$$

## 3. OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜM METODLARI

Optimizasyon modellerinin çözümüne yönelik yöntemlerin araştırılması ikinci dünya savaşı yıllarına dayanmaktadır (Dantzig, 2002). Yıllar boyunca yapılan araştırma ve geliştirme faaliyetleri sonucunda özellikle doğrusal programlama problemlerinin çözümünde büyük ilerleme kaydedilmiş olup halen yaygın olarak kullanılmaktadır. Diğer optimizasyon problemlerinin çözümüne yönelik yazılımlar olmakla birlikte, bu problemlerin çözümünü en etkin şekilde elde eden çözüm yöntemleri devamlı geliştirilmektedir. Bu bölümde, daha önce tanımladığımız 4 farklı sınıf optimizasyon problemlemi için çözüm yöntemlerinin örnek problemlere uygulanışını inceleyeceğiz.

### 3.1. Doğrusal Programlama Metodları

Doğrusal programlama modellerinin temeli doğrusal eşitliklerin çözümüne dayanmaktadır. Bu bağlamda, doğrusal programlama modelinde varolan $Ax \le b$ eşitsizlikleri her eşitsizlik için yeni bir değişkenin tanımlanmasıyla eşitliğe dönüştürülür ($Ax+x_s=b$). Bu şekilde eşitliklerle ifade edilen doğrusal programlama modelinde değişken sayısı her eşitlik sayısından daha fazla olur. Doğrusal sistemlerde çözümler bağımsız çözümler kümesi sistemdeki eşitsizlik sayısı ($Ax \le b$), $m$, ve değişken sayısının ($x$ değişkenleri), $n$, bir fonksiyonudur. Olabilecek tüm çözümlerin sayısı ise (10) denkleminden elde edilir.

$$\binom{m+n}{n} = \frac{(m+n)!}{m!\,n!} \tag{10}$$

Tablo 2. Doğrusal Bir Sistemin Eiştlik Sayısı Ve Değişken Sayısına Göre Çözüm Sayısı.

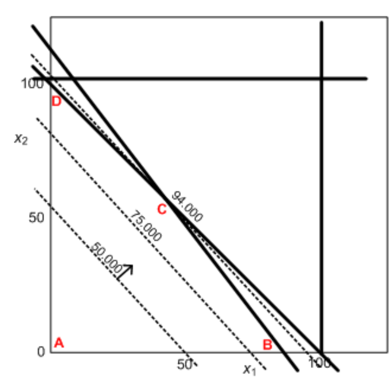
| *Değişken Sayısı* | *Eşitlik Sayısı* | *Çözüm Sayısı* |
|---|---|---|
| 2 | 2 | 6 |
| 5 | 5 | 252 |
| 10 | 10 | 184.756 |
| 25 | 25 | 126.410.606.437.752 |

Tablo 2'de gösterildiği gibi çok küçük boyutlu problemlerde bütün çözümleri teker teker elde edip bu çözümler arasından en iyi sonucu elde etmek mümkün gibi görünse de gerçek hayatta karşılaşılan problemler için en yüksek performanslı bilgisayarlarla bile uygulama için kabul edilebilecek sürelerde çözülemeyecek kadar zordur. Bu durumda bilim dünyası akıllı çözüm algoritmaları arayışına girmiştir. Dantzig (1949a), İkinci Dünya savaşı süresince ve daha sonra yaptığı çalışmaları sonucu doğrusal programlama problemlerinin çözümü için simplex metodunu geliştirmiştir ve metodu 1949 yılında yaptığı yayınla açıklayarak bu alanda öncülük yapmıştır. Simplex metodu doğrusal programlama probleminin ifade edildiği çok boyutlu uzayda problemde varolan kısıt sayısı kadar karar değişkeninin aktif olduğu bir çözümden başlayarak amaç fonksiyonun değerini en iyiye doğru taşıyan aktif olmayan bir değişkenin aktif olan bir değişkenle yer değiştirmesi esasına dayanır. Bu işlem ardışık olarak tekrarlanır ve amaç fonksiyonu değerinde herhangi bir iyilişme elde edilmeyeceği bir durumda ise simplex metodu problemin en iyi çözümü bulduğu için çözüm elde edilmiş olur. Simpleks metodu ile ilgili teorik çalışmalar çeşitli makalelerde verilmiştir (Dantzig ve Wood, 1949; Dantzig, 1949b, Dantzig 1951a, Dantzig, 1951b; Dantzig, 1951c; Dantzig, 1963).

Örnek problem için çözüm alanı Şekil 1'de gösterilmiştir. Modelde iki esas değişken olduğu için çözüm alanı iki boyutlu grafikte göstrerilebilmektedir. Modeldeki birinci kısıt çözüm alanını doğrusal olarak birinci değişken ekseninden 100 değerinde ve ikinci değişken ekseninden yine 100 değerinde kesmekte ve çözüm alanını bu doğrunun altında kalan bölgeyle sınırlamaktadır. İkinci kısıt ise kısıt çözüm alanını doğrusal olarak birinci değişken ekseninden 85 (170÷2) değerinde ve ikinci değişken eklseninden yine 113.33 (170÷1,5) değerinde kesmekte ve çözüm alanını bu doğrunun altında kalan bölgeyle sınırlamaktadır. Tüm kısıtlarıyla birlikte problemin çözüm alanı ise iki kısıtın kesiştikleri çözüm alanıyla sınırlanmıştır. Şekil 1'de, amaç fonsiyonun üç değişik değeri gösterilmiştir. Bu doğru üzerindeki tüm çözümlerin amaç fonksiyonu değeri birbirine eşittir. Örnek problemizde de olduğu gibi tüm doğrusal programlama problemlerinde sonsuz sayıda çözüm olmakla birlikte, en iyi çözüm modeldeki kısıt sayısına eşit sayıdaki değişkenin aktif olduğu noktalarda, yani kısıların kesiştiği noktalarda olabilmektedir (Dantzig, 1963). Örnek problemimizde bu noktalar A, B, C, ve D olarak belirtilmiştir. Dolayısıyla simplex metodu sadece bu çözümleri rasyonel bir şekilde ardışık olarak inceleyerek en iyi çözümü tüm çözümleri elde etmeden bulur. Simplex metoduna en kolay çözüm olan A ($x_1$=$x_2$=0) noktasından başladığımız durumda, ilk iterasyonda bu çözüme komşu olan B veya D çözümlerinden birisini en iyi çözüm adayı olarak incelemek gerekmektedir. Eğer B çözümü tercih edilirse $x_1$ yönünde hereket edilecek ve $x_1$ değişkeninin değerinde yapılacak her birim artışla amaç fonsiyonu değerinde 1.000 birim iyileştirme elde edilecek, D çözümü tercih edilirse $x_2$ yönünde hereket edilecek ve $x_2$ değişkeninin değerinde yapılacak her birim artışla amaç fonsiyonu değerinde 900 birim iyileştirme elde edilecektir. Dolayısıyla amaç fonsiyonu değerini daha hızla artırabilecek olan $x_1$ değişkeni yönünde hereket etmek ve $x_1$ yönündeki çözüm alanının en uç noktası olan B'ye kadar ilerlemek gerekmektedir. Simplex metodunun ikinci iterasyonunda ise B çözümüne komşu olan A veya C çözümlerinden birisini en iyi çözüm adayı olarak incelemek gerekmektedir. B çözümüne ulaşırken A noktasının B'ye göre amaç fonksiyonu değeri bakımından daha kötü olduğunu biliyorduk, dolayısıyla A noktasına geri dönüş mantıklı bir adım değildir ve C çözümünü en iyi çözüm adayı olarak incelemek gerekir. C çözümüne ulaşmak için ikinci kısıt üzerinde hareket ederek $x_1$ değişkenin değeri azaltılırken $x_2$ değişkeninin değeri arttırılır. C çözümünün komşularından D incelendiğinde ise bu noktanın amaç fonksiyonu değerinin C'nin amaç fonsiyonu değerinden daha düşük olacağı görüldüğü için simplex metodu örnek problemimiz için C noktasının en iyi çözüm olduğu sonucuna varır.

Örnek problemimiz için simplex metodunun uygulanışı Tablo 3'de verilen simplex tablocuklarıyla gösterilmiştir. Simpleks metoduna başlangıç çözümü olarak A noktası seçildiğinde ilk tablocuk elde edilir. Amaç fonsiyonu satırında her değişkenin bir birim arttırılması durumunda amaç fonksiyonu değerinde olacak artış gösterilmektedir. Değişkenler arasında $x_1$ en hızlı değişikliği vereceği için seçilir. Yeni bir değişkenin sıfırdan farklı bir değerle yer alabilmesi için iterasyonda verilen tablocukta sıfırdan farklı bir değerle çözümde yer alan değişkenlerden birisinin ($x_{s1}$ veya $x_{s2}$) sıfıra eşitlenmesi gerekmektedir. Eşitsizlikleri eşitliğe çevirmek amacıyla tanımladığımız yeni değişkenler $x_{s1}$ ve $x_{s2}$'nin sıfır değerini alması o kısıtın aktif olduğu anlamına geleceğinden, $x_{s1}$'in sıfırlanmasıyla elde edilecek çözüm birinci kısıtı sağlamamaktadır. Tablo 3'deki ilk tablocukta hesaplanan oran $x_1$ değişkeninin çözümde yer almasıyla hangi değişkenin sıfırlanacağına dair bilgi içermektedir. Bu oranlardan en küçük olanına karşılık gelen değişkenin sıfırlanması durumunda yeni çözüm tüm kısıtları sağlamaktadır. Bu durumda $x_1$ çözümde sıfırdan farklı bir değer alırken, $x_{s2}$ değişkeni sıfıra eşitlenir ve yeni çözüm ikinci tablocukda verildiği gibi elde edilir. İkinci tablocuktan görülebileceği gibi $x_2$ değişkeninin amaç fonsiyonunu ifade eden satırındaki değeri negatiftir ve $x_2$'nin sıfırdan farklı bir değer alarak amaç fonsiyonunu daha yükseltebileceğini göstermektedir. Yeni çözümde $x_2$'nin sıfırdan farklı bir değer alması durumunda $x_{s1}$ değişkeninin sıfırlanması durumunda yeni çözüm tüm kısıtları sağlar ve son tablocuk elde edilir. Son tablocuktanda görülebileceği gibi amaç fonsiyonu satırındaki değeri negatif olan bir değişken olmadığı en iyi çözümün bulunduğu anlamına gelir ve simpleks metodu optimal amaç fonsiyeni değerini 94.000 ve değişkenlerin değerlerini $x_1$=40 ve $x_2$=60 olarak bulur.

Şekil 1. Örnek Doğrusal Problem İçin Çözüm Alanı.



Tablo 3. Örnek Problem İçin Simpleks Metodu Tablocukları.

| **B.V.** | **Kısıt** | **z** | **$x_1$** | **$x_2$** | **$x_{s1}$** | **$x_{s2}$** | **$x_{s3}$** | **$x_{s4}$** | **b** | **Oran** |
|---|---|---|---|---|---|---|---|---|---|---|
| $z$ | Amaç | 1 | -1.000 | -900 | 0 | 0 | 0 | 0 | 0 | |
| $x_{s1}$ | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 100 | 100÷1=100 |
| $x_{s2}$ | 2 | 0 | **2** | 1,5 | 0 | 1 | 0 | 0 | 170 | 170÷2= 85 |
| $x_{s3}$ | 3 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 100 | 100÷1=100 |
| $x_{s4}$ | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 100 | 100÷0=∞ |
| $z$ | Amaç | 1 | 0 | -150 | 0 | 500 | 0 | 0 | 85.00 | |
| $x_{s1}$ | 1 | 0 | 0 | **1/4** | 1 | -1/2 | 0 | 0 | 15 | 15÷1/4=60 |
| $x_1$ | 2 | 0 | 1 | 3/4 | 0 | 1/2 | 0 | 0 | 85 | 85÷3/4=113.33 |
| $x_{s3}$ | 3 | 0 | 0 | -3/4 | 0 | -1/2 | 1 | 0 | 15 | |
| $x_{s4}$ | 4 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 100 | |
| $z$ | Amaç | 1 | 0 | 0 | 600 | 200 | 0 | 0 | 94.00 | |
| $x_2$ | 1 | 0 | 0 | 1 | 4 | -2 | 0 | 0 | 60 | |
| $x_1$ | 2 | 0 | 1 | 0 | -3 | 2 | 0 | 0 | 40 | |
| $x_{s3}$ | 3 | 0 | 0 | 0 | 3 | -2 | 1 | 0 | 60 | |
| $x_{s4}$ | 4 | 0 | 0 | 0 | -4 | 2 | 0 | 1 | 40 | |

Örnek problemde simpleks metodu en iyi çözümü bulurken 6 çözümden sadece 2 tanesini elde ederek en iyi çözümü bulmuştur. Simpleks metodu ile değişken sayısı milyonlara varan problemler çözülebilmektedir.

Doğrusal programlama problemleri için en yaygın olarak simpleks metodu kullanılmasına karşın bu problemlerin çözümü için diğer algoritmalarda geliştirilmiştir. Simpleks metodu çözüm alanının en uç noktalarını sistematik bir şekilde çözerek en iyi çözümü elde eder. Simpleks algoritmasından sonra geliştirilem metodlar ise çözüm alanının en uç noktaları yerine çözüm alanının içerisinden giderek en iyi çözümü elde etmeye göre tasarlanmıştır. Bu metodlar iç-nokta metodlar (interior-point methods) olarak adlandırılırlar. Bu metodlar için temel kavramlar iki Sovyet matematicisi Dikin (1967) ve Khachian (1979) tarafından geliştirilmiştir. Daha sonra Karmarkar (1984) tarafından yapılan araştırmalar ilk iç-nokta metodunun geliştirilmesiyle sonuçlanmıştır. Karmarkar'ın geliştirdiği iç-nokta metoduyla birlikte doğrusal programlama problemlerinin çözümünde yeni bir alan açılmış ve bu metodun teorik detaları incelenmiş ve metodu iyileştirmek için değişiklikler önerilmiştir (Cavalier ve Soyster, 1985; Hooker, 1986; Barnes, 1986; Vanderbei et al., 1986; Mizuno et al., 1993).

### 3.2. Tamsayı-Karışık Doğrusal Programlama Metodları

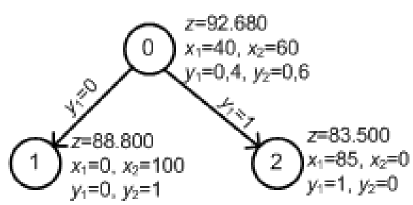
Tamsayı-karışık doğrusal programlama problemlerinin çözümünde en yaygın olarak dal-sınır algoritması kullanılır (Land ve Doig, 1960; Balas, 1965; Dakin, 1965). Dal-sınır algoritması tamsayılı değişkenlerin alabileceği tamsayılar kümesindeki en küçük ve en büyük değerler arasında herhangi bir reel sayıyı alabilcek şekilde gevşetilmesiyle elde edilen problemlerin çözümü esasına dayanır. Her problemin çözümü sonucunda en iyi çözüm için bir sınır elde edilir. Ayrıca, tamsayılı değer alması gereken değişkenler sistematik bir şekilde her problemin soncuna göre değerlendirilerek yeni nodlar oluşturulur. Dal-sınır algoritmasında her dal aşağıdaki durumlarda sonlandırılır:

i. tamsayılı değer alması gereken değişkenlerin tümü tamamen veya kısmen gevşetilmiş problemde tamsayılı değer alır ve bu çözüme tamsayılı çözüm adı verilerek en iyi amaç fonsiyonu değeri gerekli olursa güncellenir,
ii. bir noddaki amaç fonksiyonu değeri güncel olan en iyi amaç fonksiyonu değerinden daha kötüdür,
iii. bir noddaki problemin çözümü kısıtlardan bazılarını sağlamamaktadır.

Bu kriterlere uyularak problem için dal-sınır ağacı oluşturulur ve tüm dallar sonlandırıldıktan sonra en iyi çözüme ulaşılır.

Tamsayı-karışık doğrusal programlama için oluşturduğumuz örnek modelde dal-sınır algoritmasının ilk problemi tamsayılı değişkenler $y_1$ ve $y_2$'nin gevşetilmesiyle oluşturulur. Bu durumda elde edilen doğrusal programlama modeli simpleks metoduyla çözülür ve Şekil 2'de 0 numaralı nodda gösterilen çözüm elde edilir. Bu nodda elde edilen amaç fonsiyonu değeri 92.680'dir ve bu problem için en iyi çözüme bir üst sınır verir. Bu çözümde tamsayılı değişkenlerin değerlerinin 0,4 ve 0,6 olduğunu görürüz. Bu noddan iki yeni dal üzerinde yeni nodlar oluşturmak gerekmektedir. Bu amaçla tamsayılı değişkenlerden birisini ($y_1$ ve $y_2$) seçmemiz gerekir. Bu problemde rastgele bir seçimle $y_1$ değişkeni seçilmiş ve 1 ve 2 numaralı nodlar $y_1$'in 0'a ve 1'e eşitlenmesiyle elde edilmiştir. Yeni nodlarda elde edilen sonuçlar $y_2$ değişkeninin de tamsayılı değer alması sebebiyle sonlandırılır ve en iyi sonucu veren 1 numaralı noddaki çözüm bulunur.

Şekil 2. Tamsayı-Karışık Doğrusal Programlama Örnek Problemi İçin Dal-Sınır Ağacı.




Tamsayı-karışık doğrusal programlama problemlerinin çözümü için diğer bir yaklaşımda ise ilk defa Gomory tarafından önerilen kesen yüzeyler kullanılır (Gomory, 1958). Bu metodda tamsayılı karar değişkenlerinin gevşetilmiş doğrusal programlama problemine ardışık olarak geçerli eşitsizliklerin eklenmesiyle tamsayılı çözüm elde edilmesi hedeflenir. Son yıllarda dal-sınır algoritmalarıyla kesen yüzeyler metodlarının birşleştirilmesi yönünde yapılan çalışmalarda çözümü zor olan yamsayı karışık doğrusal programlama problemlerinde başarılı olunmuştur. Bu kapsamda yapılan gelişmeler Johnson ve çalışma arkadaşları tarafından özetlenmiştir (Johnson ve diğ., 2000).

### 3.3. Doğrulsal Olmayan Programlama Metodları

Doğrusal olmayan programlama problemlerinde eniyi çözümün Karush-Kuhn-Tucker (KKT) şartlarına uyması gereklidir. Bu şartların teorik anlamları iki ayrı çalışmada verilmiştir (Karush, 1939; Kuhn ve Tucker, 1951). Öncelikle, doğrusal olmayan programlama problemlerinde (11) denkleminde verildiği gibi amaç fonsiyonunun gradyanı ile kısıt fonksiyonların gradyanlarının her kısıt için tanımlanmış parametrelerle çarpımlarının toplamı sıfıra eşit olmalıdır. Eğer problemde sadece amaç fonksiyonu varsa (amaç fonsiyonun eniyi değeri kısıtsız olarak belirlenecekse) sadece amaç fonksiyonun gradyanını sıfıra eşitleyecek karar değişkenlerinin değerleri amaç fonksiyonunu eniyilemiş olurlar.

$$\nabla f(x') + \sum_{j=1}^{m} \mu_j \nabla g_j(x') + \sum_{j=1}^{r} \lambda_j \nabla h_j(x') = 0 \tag{11}$$

$$x' \left( \nabla f(x') + \sum_{j=1}^{r} \lambda_j \nabla h_j(x') \right) = 0 \tag{12}$$

$$\begin{aligned} g_j(x') &= 0 \quad j=1,...,m \\ h_j(x') &\le 0 \quad j=1,...,r \end{aligned} \tag{13}$$

$$\left.\begin{aligned} \lambda_j h_j(x') &= 0 \\ \lambda_j &\ge 0 \end{aligned}\right\} \; j=1,...,r \tag{14}$$

Ayrıca karar değişkenleri alacakları değerlerle tamamlayıcı koşuları yerine getirmeli (12), tüm kısıtlara uymalı (13) ve kısıtlar için tanımlanan çarpanlardan $\lambda_j$ için (14)'te verilen ek şartlar sağlanmalıdır.

Doğrusal olmayan problemlerin optimal çözümünü elde etmek için çeşitli metodlar kullanılmaktadır (Bazaraa ve diğ, 2006). Bu metodlar arasında yaygın olarak kullanılan Aktif kısıt kümesi metodunu 5 ana basamaktan oluşmaktadır (Fletcher ve de la Masa, 1989):

Şekil 3: Aktif Kısıt Kümesi Çözüm Algoritması.

1. İterasyon sayacı tanımlanır, $i$=1.
2. Problemdeki tüm kısıtların sağlandığı varsayılarak aktif kısıt kümesi boş küme olarak belirlenir, $J_i=\{j \mid h_j=0\}$. Bu durumda, $J_i=\emptyset$, $\lambda_j=0$, $j$=1,..., $r$.
3. KKT şartları aşağıdaki gibi yazılır ve $x$ değişkenleriyle $\lambda_j$ ($j \in J_i$) çarpanlarının değeri belirlenir.

$$\nabla f(x') + \sum_{j=1}^{r} \lambda_j \nabla h_j(x') = 0$$

$$h_j(x') \le 0 \quad j \in J_1$$

4. Eğer tüm kısıtlar eşitlik olarak sağlanıyorsa ($h_j(x)$=0 ve $\lambda_j$<0, $j$=1,...,$r$) mevcut çözüm KKT şartlarını yerine getiriyor demektir ve en iyi çözüm bulunmuştur. Mevcut çözüm en iyi çözüm olarak raporlanır ve algoritma durdurulur.
5. Eğer herhangi bir kısıt sağlanmıyorsa ($h_j(x)$>0) ve/veya $\lambda_j$<0 ise:
   a. İterasyon sayacı güncellenir, $i=i+1$.
   b. Çarpanı sıfırdan küçük değer alan bir kısıt aktif set kümesinden çıkarılır. Birden fazla çarpanın sıfırdan küçük olması durumunda sıfırdan farkı en büyük olan kısıt seçilerek aktif kısıt kümesinden çıkarılır ve $J_i$ kümesi güncellenir.
   c. Sağlanmayan kısıtlar ($h_j(x)$>0) $J_i$ kümesine eklenerek güncellenir.
   d. 3. basamağa dönülerek yeni bir iterasyon yapılır.

Amaç fonksiyonun eniyilendiği şartları (36.15)'de verildiği gibi yazabiliriz:

$$\begin{aligned} x_1: &\quad (1000)\left(\frac{14}{15}\right)x_1^{-\frac{1}{15}} - \lambda_1 - 2\lambda_2 - \lambda_3 = 0 \\ x_2: &\quad (900)\left(\frac{9}{10}\right)x_2^{-\frac{1}{10}} - \lambda_1 - 1.5\lambda_2 - \lambda_4 = 0 \end{aligned} \tag{15}$$

$$\begin{aligned} x_1: &\quad x_1\left((1000)\left(\frac{14}{15}\right)x_1^{-\frac{1}{15}} - \lambda_1 - 2\lambda_2 - \lambda_3\right) = 0 \\ x_2: &\quad x_2\left((900)\left(\frac{9}{10}\right)x_2^{-\frac{1}{10}} - \lambda_1 - 1.5\lambda_2 - \lambda_4\right) = 0 \end{aligned}$$

$$x_1 + x_2 \leq 100$$
$$2x_1 + 1.5x_2 \leq 170$$
$$x_1 \leq 100$$
$$x_2 \leq 100$$

$$\lambda_1 \left( x_1 + x_2 - 100 \right) = 0$$
$$\lambda_2 \left( 2x_1 + 1.5x_2 - 170 \right) = 0$$
$$\lambda_3 \left( x_1 - 100 \right) = 0$$
$$\lambda_4 \left( x_2 - 100 \right) = 0$$

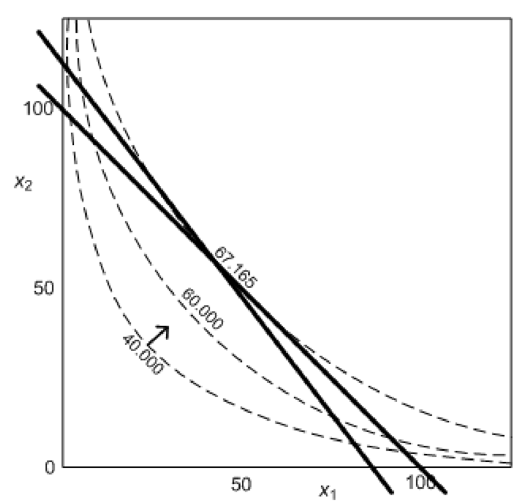
$$x_1, x_2 \geq 0$$
$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$$

Örnek problem için çözüm alanı grafiksel olarak Şekil 4'te verilmiştir.

Şekil 3'te verilen aktif kısıt kümesi metod uygulandığında örnek problemde en iyi çözüm olarak, $x_1$=44.4775, $x_2$= 54.03, $\lambda_1$=0, $\lambda_2$=362.3517, $\lambda_3$=0, $\lambda_4$=0 belirlenir. Bu sonuçtan da görülebileceği gibi en iyi çözümde birinci, içinci ve dördüncü kısıt eşitsizlikler olarak sağlanmakta ($\lambda_1$=0) ve ikinci kısıt ise eşitlik olarak sağlanmaktadır.

Doğrusal olmayan programlama problemlerinin çözümü için diğer yaklaşımlardan en yagın olanları arasında indirgenmiş gradyan, ardışık karesel programlama, güvenli alan ve iç-nokta metodları sayılabilir (Bazaara, ve diğ., 2006).

Şekil 4: Doğrusal Olmayan Programlama Örnek Problemi İçin Çözüm Alanı

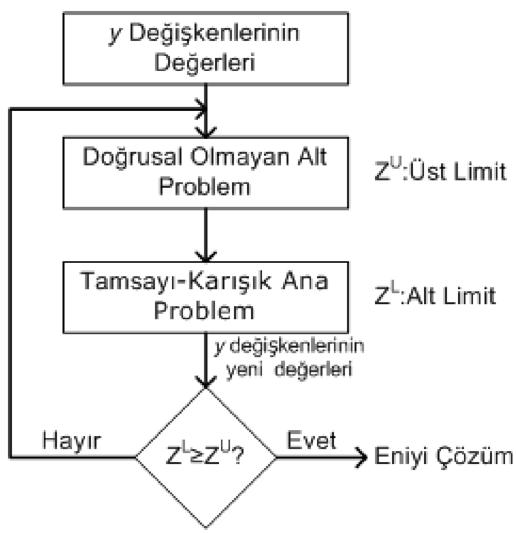


### 3.4. Tamsayı-Karışık Doğrusal Olmayan Programlama Modelleri

Tamsayı-karışık doğrusal olmayan programlama problemlerinde eniyi çözümün hem Karush-Kuhn-Tucker (KKT) şartlarına uyması hemde tamsayılı kısıtları yerine getirmesi gereklidir. Bu problemlerin çözümünde en yaygın olarak kullanılan çözüm metodlarından birisi dışsal yaklaşıklamadır (Duran ve Grossmann, 1986; Viswanathan ve Grossmann, 1990).

Dışsal yaklaşıklama metodu tamsayı-karışık doğrusal olmayan problemin ikiye ayrılması esasına dayanır. Şekil 5'te de gösterildiği gibi tamsayılı değişkenlerin belli değerlere sabitlenmesi sonucu (örneğin 0 veya 1) doğrusal olmayan bir alt problem elde edilir. Bu alt problemin çözülmesiyle eniyi çözüm için bir üst limit elde edilir ve değişkenlerin değerleri belirlenir. Problemde yer alan doğrusal olmayan kısıtların değişkenlerin aldıkları değerlerde teğetlerinin alınmasıyla bu kısıtlar doğrusal olarak yaklaşıklaştırılır. Daha sonra doğrusal olarak ifade edilen probleme tamsayılı değişkenlerin de eklenmesiyle tamsayı-karışık ana problem elde edilir. Tamsayı-karışık ana problemin çözümüyle ise amaç fonksiyonu için alt limit elde edilir ve tamsayılı değişkenler için yeni değerler belirlenir. Doğrusal olmayan alt problemden elde edilen üst limitle tamsayı-karışık ana problemden elde edilen alt limit karşılaştırılarak eniyi çözümün bulunup bulunulmadığı kontrol edilir. Eğer alt limitin değeri üst limitten büyük veya eşitse metod en iyi üst limiti veren doğrusal olmayan programlama problemi sonucunu raprolayarak sonlanır. Eğer üst limit alt limitten küçükse en son çözülen tamsayı-karışık ana problemden elde edilen tamsayılı değişken değerleri sabitelenerk yeni bir doğrusal olmayan programlama problemi elde edilerek döngüye devam edilir.

Şekil 5: Dışsal Yaklaşıklama Metodu




Örnek problemin eniyi .çzümü dışşsal yaklaşıklama metoduyla 3 doğrusal olmayan alt problemin ve 2 tamsayı-karışık ana problemin çözülmesiyle elde edilmiştir. Dışsal yaklaşıklama metoduyla elde edilen çözümde değişkenlerin değerleri, $x_1$=85, $x_2$= 0, $y_1$=1 ve $y_2$=0 olarak belirlenmiştir. Çözümde raporlanan en iyilenmiş amaç fonsiyonmu değeri ise 61.710,76'dır.

Tamsayı-karışık doğrusal olmayan programlama problemlerinin çözümü için diğer yaklaşımlardan en yaygın olanları arasında mantık tabanlı metodlar (Türkay ve Grossmann, 1996) dal-sınır algoritmaları (Gupta ve Ravindran, 1985; Leyffer, 2001), dal-ve-kes algoritmaları (Stubbs ve Mehrotra, 1999), genellendirilmiş Benders ayrıştırma metodu (Geoffrion, 1972) sayılabilir.

## 4. SONUÇ

Bu bölümde işletmelerin verimliliklerini arttırmakta çok önemli rol oynayan optimizasyon teknolojisi ile ilgili detaylı bilgi verilmiştir. Gün geçtikçe gelişen sanayimizin rekabetçi olabilmesi için verimliliğin arttırılması kaçınılmaz bir zorunluluktur. Globalleşen dünya üretim, hizmet ve finans sektörlerinde önemli fırsatları beraberinde getirmektedir. Bu fırsatların en akılcı ve etkin olarak eğerlendirilmesi için iyi bir modelleme, çözümleme ve analiz platformunun gerçekleştirilmesi kaçınılmazdır.

## KAYNAKÇA


Balas E. (1965), An Additive Algorithm for Solving Linear Programs with Zero-One Variables, *Operations Research*, 13, 517-546.

Barnes, E.R. (1986), A variation on Karmarkar's algorithm for solving linear programming problems, *Mathematical Programming*, 36, 2, 174-182.

Bazaraa, M.S., Sherali H.D. ve Shetty, C.M. (2006), *Nonlinear Programming: Theory and Algorithms*, 3. baskı, Wiley-Interscience, New York.

Billington,G., Callioni,G., Crane,B., Ruark,J.D., Rapp, J.U., White,T. ve Willems,S.P. (2004), Accelerating the Profitability of Hewlett-Packard's Supply Chains, *Interfaces* 34, 59–72.

Bollapragada, S., Cheng, H., Phillips, M., Garbiras, M., Scholes, M., Gibbs, T. ve Humphreville, M. (2002), NBC's Optimization Systems Increase Revenues and Productivity, *Interfaces,* 32, 47–60, 2002.

Cavalier, T.M. ve Soyster, A.L. (1985), *Some computational experience and a modification of the Karmarkar algorithm*, ISME Working Paper 85-105, The Pennsylvania State University.

Dakin, R. J. (1965), A Tree Search Algorithm for Mixed Integer Programming Problems, *Computer Journal*, 8,3, 250-255, 1965.

Dantzig, G. B. (1949a), Programming in a linear structure, *Econometrica* 17, 73–74.

Dantzig, G. B. (1949b), Programming of Inter-Dependent Activities II, Mathematical Model, *Econometrica,* 17, 3-4, 200-211, 1949b. (Aynı makale ayrıca Dantzig, G.B. (1951), *Activity Analysis of Production and Allocation* içinde Koopmans, T. C. editör, John Wiley& Sons, NY, 19-32 yayınlanmıştır).

Dantzig, G. B. (1951a), Maximization of a linear function subject to linear inequalities, *Activity Analysis of Production and Allocation* içinde Koopmans, T. C. editör, John Wiley& Sons, NY, 339-347.

Dantzig, G. B. (1951b), Application of the simplex method to a transportation problem, *Activity Analysis of Production and Allocation* içinde Koopmans, T. C. editör, John Wiley& Sons, NY, 359-373.

Dantzig, G. B. (1951c), ,A proof of the equivalence of the programming problem and game problem, *Activity Analysis of Production and Allocation* içinde Koopmans, T. C. editör, John Wiley& Sons, NY, 330-335.

Dantzig, G. B. (1957), Discrete variable extremum problems, *Oper. Res.,* 5, 266–277.

Dantzig, G. B. (1959), Note on solving linear programs in integers, *Naval Res. Logist. Quart.*, 6 75–76.

Dantzig, G. B. (1960), On the significance of solving linear programs with some integer variables, *Econometrica*, 28, 30–44.

Dantzig, G.B. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ.

Dantzig, G.B. (2002), Linear Programming, *Operations Research* 50 (1), 42-47.

Dantzig, G. B. ve Wood, M. (1949), Programming of Inter-Dependent Activities I, General Discussion, *Econometrica*, 17, 3-4, 193-199. (Aynı makale ayrıca Dantzig, G.B. ve Wood, M. (1951), Activity *Analysis of Production and Allocation* içinde Koopmans, T. C. editör, John Wiley& Sons, NY, 15-18 yayınlanmıştır).

Dikin, II. (1967), Iterative solution of problems of linear and quadratic programming, *Doklady Akademiia Nauk SSSR*, 174,747-748. (İngilizce tercümesi *Soviet Mathematics Doklady*, 8, 674-675, 1967.).

Duran, M.A. ve Grossmann, I.E. (1986) An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs, *Mathematical Programming*, 36, 307-339.

Fletcher, R. ve de la Maza, E.S. (1989), Nonlinear programming and nonsmooth optimizatio by successive linear programming, *Mathematical Programming*, 43, 235–256.

Gass, S.I. (2000), Making Decisions with Precision, *Business Week* October 30, 2000 (http://www.businessweek.com/archives/2000/b3705139.arc.htm?campaign_id=search#B3705139), son erişim tarihi: 1 Şubat 2006.

Gass, S.I. ve Assad, A.A. (2004), *Annotated Timeline of Operations Research: An Informal History*, Springer-Verlag New York, NY.

Geoffrion, A.M. (1972), Generalized Benders Decomposition, *Journal of Optimization Theory and Applications*, 10, 237-260.

Gomory, R.E. (1958), Outline of an Algorithm for Integer Solution to Linear Programs, *Bulletin of the American Mathematical Society*, 64, 275-278.

Gupta, O.K. ve Ravindran, V. (1985), Branch and Bound Experiments in Convex Nonlınear Integer Programming, *Management Science*, 31, 1533-1546.

Hillier, F.S. ve Lieberman, G.J. (2005), *Introduction to Operations Research*, 8. baskı, 2005, McGraw Hill, New York, NY.

Hooker, J.N. (1986), Karmarkar's linear programming algorithm, *Interfaces*, 16, 4, 75-90.

INFORMS Interfaces dergisi web sitesi, http://pubsonline.informs.org/, son erişim tarihi: 6 Şubat 2006.

Johnson, E.L., Nemhauser, G.L. ve Savelsbergh, M.W.P. (2000), Progress in Linear Programming Based Branch-and-Bound Algorithms: An Exposition, *INFORMS Journal on Computing*, 12, 2-23.

Ireland, P., Case, R., Fallis, J., Van Dyke, C., Kuehn, J. ve Meketon, M. (2004), "The Canadian Pacific Railway Transforms Operations by Using Models to Develop Its Operating Plans", *Interfaces,* 34, 5–14.

Kantorovich, L. V. (1939), “Mathematical Methods in the Organization and Planning of Production”, Publication House of the Leningrad State University, Leningrad, U.S.S.R., 68 (Rusça).

Kantorovich, L. V. (1960), “Mathematical Methods in the Organization and Planning of Production”, *Management Sci.* 6 ,366–422 (İngilizce).

Kantorovich, L. V. ve Gavurin M. K. (hazırlanış tarihi 1940, yayın tarihi 1949), *Primenenie matematicheskikh metodov v voprosakh analiza gruzopotokov*, in Problemy povysheniia effektivnosti raboty transporta (The Use of Mathematical Methods in Analyzing Problems of Goods Transport, in Problems of Increasing the Efficiency in the Transport Industry, pp. 110-138). Academy of Sciences, U.S.S.R.

Karmarkar N. (1984), A New Polynomial Time Algorithm for Linear Programming, *Combinatorica*, 4, 373–395.

Karush, W. (1939), *Minima of Functions of Several Variables with Inequalities as Side Conditions*, M.Sc. Thesis, Department of Mathematics, University of Chicago, Chicago, IL.

Khachian, L.G. (1979), A polynomial algorithm in linear programming, *Dokklady Akademiia Nauk SSSR*, 244, 1093-1096. (İngilizce tercümesi *Soviet Mathematics Doklady* 20, 191-194, 1979.)

Koopmans, T. C. (1949), Optimum Utilization of the Transportation System, *Proceedings of the International Statistical Conferences*, 5, 136-145.

Koopmans, T. C., editör (1951a), *Activity Analysis of Production and Allocation*, New York, Wiley.

Koopmans, T. C. ve Reiter, S. (1951), *A Model of Transportation*, Koopmans, T. C., editör, Activity Analysis of Production and Allocation, 222-259, New York, Wiley.

Koopmans, T. C. (1957), *Three Essays on the State of Economic Science*, New York, McGraw-Hill.

Koopmans, T. C. ve Bausch, A.F. (1959), Selected Topics in Economics Involving Mathematical Reasoning, *SIAM Review*, 1, 2, 79-148.

Kuhn, H.W. ve Tucker, A.W. (1951) Nonlinear programming. Editör: J. Neyman, *Proc. Second Berkeley Sympos. Math. Statist. Probab*. University of California Press, Berkeley, CA, 481–492.

Land, A.H. ve Doig, A. G. (1960) An automatic method for solving discrete programming problems, *Econometrica* 28, 497-520.

Leontief, W.W. (1933), The Use of Indifference Curves in the Analysis of Foreign Trade, *The Quarterly Journal of Economics,* 47, 493-503.

Leontief, W. W. (1936), Quantitative Input and Output Relations in the Economic System of the United States, *Review of Econ. Statistics*, 18, 3, 105-125.

Leyffer, S. (2001), Integrating SQP and branch-and-bound for Mixed Integer Nonlinear Programming, *Computational Optimization and Applications*, 18, 295-309.

Metty, T., Harlan, R., Samelson, Q., Moore, T., Morris, T., Sorensen, R., Schneur, A., Raskina, O., Schneur, R., Kanner, J., Potts, K. ve Robbins, J. (2005), Reinventing the Supplier Negotiation Process at Motorola, *Interfaces,* 35, 7–23.

Mizuno, S., Todd, M.J. ve Ye, Y. (1993), On adaptive step primal-dual interior-point algorithms for linear programming, *Math. Oper. Res.*, 18, 964-981.

Murty, K.G., Wan, Y.W., Liu, J., Tseng, M.M., Leung, E., Lai, K.K. ve Chiu, H.W.C. (2005), Hongkong International Terminals Gains Elastic Capacity Using a Data-Intensive Decision-Support System, *Interfaces,* 35, 61–75.

NEOS Optimizasyon Rehberi web sitesi, http://www-fp.mcs.anl.gov/otc/Guide/index.html, son erişim tarihi: 7 Şubat 2006.

Nobel Ödülü web sitesi, http://nobelprize.org/economics/laureates/, son erişim tarihi: 5 Şubat 2006.

Patchong, A., Lemoine, T. ve Kern, G. (2003), Improving Car Body Production at PSA Peugeot Citroen, *Interfaces,* 33, 36–49.

Stubbs, R. ve S. Mehrotra, S. (1999), A Branch-and-Cut Method for 0-1 Mixed Convex Programming, *Mathemaitcal Programming*, 86, 515-532.

Türkay, M. ve I.E. Grossmann, I:E: (1996) Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks, *Compt. Chem. Eng.*, 20, 959-978.

Vanderbei, R.J., Meketon, M.S. ve Freedman, B.A. (1986), A modification of karmarkar's linear programming algorithm, *Algorithmica*, 1, 1, 395-407.

Viswanathan, J. ve I.E. Grossmann, I.E. (1990) A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization, *Compt. Chem. Eng.*, 14, 769-782.

Williams, H.P. (1999), *Model Building in Mathematical Programming*, 4. baskı, Wiley, New York, NY.

Winston, W.L. (2003), *Operations Research: Applicatios and Algorithms*, 4. baskı, International Thomson Publishing, Belmont, CA.