

Yöneylem Araştırması II

Öğr. Gör. Dr. Hakan
ÇERÇİOĞLU
cercioglu@gazi.edu.tr

ENM 305- Yöneylem Araştırması II



ENM 305 YÖNEYLEM ARAŞTIRMASI II		ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ	
Yarıyıl	Kredi Dağılımı		
	Teori	Uygulama	Laboratuvar
5	4	0	0
Ders Dili	Türkçe		
Zorunlu / Seçmeli	Zorunlu		
Önşartlar	ENM 202		
Katalog Tanımı	Tamsayılı programlama ve çözüm teknikleri (dal-sınır ve kesirli kesme algoritmaları). Şebeke modelleri: En az yayılan ağaç, En kısa yol, Maksimum akış, Ulaştırma ve Atama modelleri, Proje Yönetim Teknikleri, Doğrusal olmayan programlama.		
Dersin Amacı	Tamsayılı programlama, Şebeke analizi ve Doğrusal olmayan programlama problemlerinin yöneylem araştırması içindeki yeri ve öneminin vurgulanması, genel kavramları ve uygulamalarının tanıtılması, modelleme ve çözüm tekniklerinin öğretilmesi.		

[ENM 305- Yöneylem Araştırması II]



ENM 305 YÖNEYLEM ARAŞTIRMASI II		ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ
Dersin Kazanımları	Tamsayıli programlama, Şebeke analizi ve Doğrusal olmayan programlama problemlerinin gerçek hayattaki uygulamaları ve çözüm yöntemleri hakkında altyapının edinilmesi ve yönetim problemlerinde kullanımının öğrenilmesi.	
Ders Kitabı ve/veya Kaynaklar	1. Winston, W. L., (2003), Operations Research: Applications and Algorithms,. Duxbury press, 4 th.edition. 2. Taha, H.A., (2006), Operations Research an Introduction, Prentice hall, 8 th.edition. 3. Taha, H.A., (1975), Integer Programming Theory, Applications and Computations, Academic press.	

Yöneylem Araştırması II

3

[ENM 305- Yöneylem Araştırması II]



ENM 305 YÖNEYLEM ARAŞTIRMASI II		ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ	
Değerlendirme Ölçütleri		Adet	Yüzde
	Ara Sınavlar	2	40
	Kısa Sınavlar	-	-
	Ödevler	-	-
	Projeler	-	-
	Dönem Ödevi	-	-
	Laboratuvar	-	-
	Diğer (Devamlılık)	-	10
	Dönem Sonu Sınavı	1	50
İçerik Ağırlıkları Yüzdesi (%)	Matematik ve Temel Bilimler	25	
	Mühendislik Bilimleri	35	
	Mühendislik Tasarımı	40	
	Sosyal Bilimler	0	
Ders Sorumluları	Prof. Dr. Cevriye GENCER, Prof. Dr. Bilal TOKLU, Öğr. Gör. Dr. Hakan ÇERÇİOĞLU		

Yöneylem Araştırması II

4

[Ders Planı]



DERS PLANI	
Hafta	Konular
1	Giriş Tanımlar, Kesikli optimizasyon, Tamsayılı optimizasyon, Matematiksel modeller. Sürekli Optimizasyonun Hatırlatılması Grafik çözüm, Primal simpleks, Dual simpleks
2	Tamsayılı Optimizasyona Giriş Tamsayılı optimizasyonun tanımı, Saf tamsayılı ve karma tamsayılı modellerin tanımı tamsayılı programlama problemlerinin formülasyonu Ya-yada kısıtları,
3	İse-O zaman kısıtları Tamsayılı Optimizasyon İçin Çözüm Metodları Dal-sınır metodu, Gomory (kesme) metodu
4	Dal-Sınır Metodu Tanımı, dal-sınır metodu kuralları ve dallanma stratejileri ve örnek
5	Grafik çözüm ve tablo çözüm ve örnekler

Yöneylem Araştırması II

5

[Ders Planı]



DERS PLANI	
Hafta	Konular
6	Kesme Düzlemi Metodu Gomory (kesme) metodu tanımı, Kesme grafikleri ve örnekler
7	0-1 Modelleme 0-1 değişkenlerin tanımı, 0-1 tamsayılı modelleme, Küme kaplama problemi, Üretim planlama problemi, 0-1 kargo yükleme (sırt çantası) problemi çözümü
8	Dal-sınır metodu ile sırt çantası problemlerinin çözümü 1. ARA SINAV
9	Şebeke Modelleri Giriş ve genel kavramlar, En kısa yol problemi, Çevrimsiz ve çevrimli şebekelerde en kısa yol algoritmaları,
10	Minimum yayılan ağaç problemi, Prim ve Kruskal algoritmaları Max akış problemi, Maksimum akış- min kesi teoremi, Etiketleme algoritması.

Yöneylem Araştırması II

6

[Ders Planı]



DERS PLANI	
Hafta	Konular
11	Ulaştırma Problemi Başlangıç çözümleri; Kuzeybatı, Minimum maliyet ve VAM Metodu, Optimalite testi Atama Problemi Macar metodu
12	Proje Yönetim Teknikleri PERT/CPM, Öncelik Diyagramı, Proje tamamlanma ve bolluk zamanların hesaplanması
13	Doğrusal Olmayan Programlama Genel kavramlar, Tek ve çok değişkenli problemlerin çözümü
14	2. ARA SINAV Eşitlik kısıtı altında optimizasyon-Lagrange çarpanları
15	Eşitlik kısıtı altında optimizasyon-Kuhn-Tucker şartları

Yöneylem Araştırması II

7

[Bölüm Ders Programı İle Ders Arasındaki İlişki]



	Program Kazanımları	1	2	3
1	Matematik, fen ve mühendislik bilgilerini uygulama becerisi			X
2	Deney tasarımı ve yapma ile deney sonuçlarını analiz etme ve yorumlama becerisi	X		
3	İstenen gereksinimleri karşılayacak biçimde bir sistemi, parçayı ya da süreci tasarımı becerisi		X	
4	Disiplinlerarası takımlarda çalışabilme becerisi	X		
5	Mühendislik problemlerini tanımlama, formüle etme ve çözme becerisi			X
6	Mesleki ve etik sorumluluk bilinci	X		
7	İngilizce ve Türkçe etkin iletişim kurma becerisi	X		
8	Mühendislik çözümlerinin, evrensel ve toplumsal boyutlarda etkilerini anlamak için gerekli genişlikte eğitim		X	
9	Yaşam boyu öğrenmenin gerekliliği bilinci	X		
10	Çağın sorunları hakkında bilgi	X		
11	Mühendislik uygulamaları için gerekli olan teknikleri, yetenekleri ve modern araçları kullanma becerisi		X	
Dersin katkısı : 1:Hiç 2:Kısmi 3:Tümüyle				

Yöneylem Araştırması II

8

BÖLÜM I: Yöneylem Araştırmasına Giriş

- Yöneylem Araştırmasının Tarihçesi
- Yöneylem Araştırmasının Tanımı
- Uygulama Alanları
- Yöneylem Araştırması Modelleri

[Yöneylem Araştırmasının Tarihçesi]



- Yöneylem araştırması çalışmalarına ilk kez **1940**'da, 2. Dünya Savaşında Alman hava hücumlarının İngiltere'yi büyük kayıplara uğrattığı günlerde **İngiltere tarafından başlanmıştır**.
- Oluşturulan bir araştırma ekibi, Alman hava hücumlarının dağılımını olasılık kurallarına göre saptayarak, eldeki olanaklarla oluşturduğu hava savunma sistemi beklenenin çok üstünde bir başarıya ulaşmıştır.
- Buna paralel olarak A.B.D. 'de İngiltere'nin geliştirdiği bu yöntemlerden yararlanarak kendi kara, hava ve deniz kuvvetlerinde yöneylem araştırması ekipleri oluşturmuş ve problemlerini çözmüştür. [Esin, 1988]

[Yöneylem Araştırmasının Tarihçesi]



- Ülkemizde oluşturulan ilk Yöneylem Araştırması birimi 19 Ağustos 1954 tarihinde Genel Kurmay Başkanlığı bünyesinde kurulan İlmi İstişare Kurulu Müdürlüğü olup, gerçek anlamda ilk Yöneylem Araştırması grubu 1 Haziran 1956 tarihinde yaklaşık 10 yedek subaydan oluşturulmuştur.
- Sivil kesimde ise ilk olarak 1 Eylül 1965 tarihinde TÜBİTAK bünyesinde bir Yöneylem Araştırması ünitesi kurulmuş, 1973 yılında “Gebze Marmara Bilimsel ve Endüstriyel Araştırma Enstitüsü” nün bir ünitesi olarak faaliyetlerine devam etmiş ve 1992 yılında “Sistem Analizi” adı verilerek yeni bir birime dönüştürülmüştür.
- Eğitim alanında ilk uygulamalar İTÜ Makine Fakültesinde başlamış, 1964-1965 öğretim yılında ODTÜ Matematik Bölümünde başlayan Doğrusal Programlama dersleri izlemiştir [Esin, 1988].

[Yöneylem Araştırmasının Tanımı]



- Organizasyonların faaliyetleri ile ilgili karar verme sürecine yönelik bilimsel bir yaklaşımdır.
- Endüstri, iş dünyası, yönetim ve savunma alanlarında; insan, makine, malzeme, zaman, bilgi ve paradan oluşan büyük sistemlerin yönetiminde ortaya çıkan karmaşık problemlerin çözümünde bilimsel metotların uygulanmasıdır.

[Yöneylem Araştırmasının Tanımı]



- Sistemlerin karşılaştıkları problemlerde, disiplinler arası bir ekiple, **bilimsel metotları** kullanarak ve **problemin kontrol edilebilir unsurları** ile ilgili alternatifleri değerlendirmek suretiyle optimal çözümü bulmayı amaçlar.
- Gerçek hayattan kaynaklanan ve çoğunlukla sınırlı kaynakların tahsis edildiği deterministik ve olasılıklı problemlerin modellenmesi ve optimal kararın verilmesi ile ilgilenir.

[Bilimsel Yöntemin Aşamaları]



-
- Problemin tanımlanması (formüle edilmesi) ve gerekli verilerin toplanması
- Modelin kurulması
- Modelden çözüm elde edilmesi
- Modelin ve çözümün test edilmesi (kanıtlanması)
- Çözümün uygulanması

[Uygulama Alanları]



- Üretim Planlama
- Üretim Çizelgeleme
- Verimlilik Analizi
- Toplam Kalite Yönetimi
- Proje Yönetimi
- Taşıma/Ulaşım
- Stratejik Planlama
- Kent Hizmetleri Yönetimi
- Yatırım Planlama
- Savunma Uygulamaları
- Optimizasyon
- Banzetim
- Bilgisayarla Bütünleşik İmalat
- Tam Zamanında Üretim
- Karar Destek Ve Uzman Sistemler
- Malzeme Envanter Yönetimi
- Tahmin Ve Kestirme Yöntemleri
- Esnek İmalat Sistemleri
- Karar Modelleri
- Rassal Süreçler
- Tesis, Yersicimi Ve Dağıtım
- Maliyet Analizi
- Finansal Planlama
- Bütçe Planlama Ve Kontrol
- Bakım Planlaması
- Enerji Planlaması
- Performans Ölçümü
- Reorganizasyon
- İnsangücü Planlaması
- Yönetim Bilişim Sistemleri

[Yöneylem Araştırması Modelleri]



- Doğrusal Programlama
- Tamsayılı Programlama
- Hedef Programlama
- Ulaştırma Ve Atama Modelleri
- Doğrusal Olmayan Programlama
- Oyun Teorisi
- Deterministik Dinamik Programlama
- Deterministik Stok Modelleri
- Şebeke (Ağ) Analizi
- CPM Ve PERT İle Proje Planlama
- Markov Zincirleri
- Kuyruk Teorisi
- Karar Analizi
- Simülasyon
- Tahmin Modelleri
- Güvenilirlik Analizi
- Olasılıklı Dinamik Programlama
- Olasılıklı Stok Modelleri

BÖLÜM II: Doğrusal Programlama Tekrarı

- Doğrusal Programlama Tanımı
- Doğrusal Programlama Varsayımları
- Grafik Çözüm Metodu
- Simpleks Metodu
- Simpleks Metodunda Özel Durumlar
- Farklı Kısıtları Olan Modellerin Çözümü
- Büyük-M Metodu
- İki Safha Metodu
- Dual Simpleks Metodu

[Doğrusal Programlama Tanımı]



- **Doğrusal** : Modelde kullanılan bütün matematiksel fonksiyonların **lineer** olduğunu,
- **Programlama** : Çözümün bulunması için yapılacak faaliyetlerin akış sırasını ifade eder.
- Doğrusal Programlama en yaygın olarak sınırlı miktardaki kaynakların çeşitli faaliyetlere en iyi (yani optimal) şekilde paylaştırılmasına yönelik problemlerde kullanılır.

[Doğrusal Programlama Varsayımları]

- Her doğrusal programlama modelinde aşağıdaki varsayımların geçerli olduğu kabul edilir:
 - Oranlılık (Proportionality)
 - Toplanabilirlik (Additivity)
 - Bölünebilirlik (Divisibility)
 - Belirlilik (Certainty)

[Grafik Çözüm Metodu]

- Grafik çözüm, **iki değişkeni** olan doğrusal programlama modellerinin **koordinat eksen** üzerinde gözle izlenerek optimal çözümünün bulunmasını sağlayan bir tekniktir.
- Genel olarak iki aşamadan oluşur:
 - Bütün kısıtları sağlayan **uygun çözüm bölgesinin** koordinat düzleminde belirlenmesi,
 - Bu bölge içerisinde optimal amaç fonksiyonu değerini veren **optimal çözümün** tespiti.

[Örnek: Grafik Çözüm Metodu]



$$\begin{aligned} \text{MAKS } Z &= 3X_1 + 5X_2 \\ \text{S.T.} \quad 3X_1 + 2X_2 &\leq 18 \\ X_1 &\leq 4 \\ 2X_2 &\leq 12 \\ X_1 &\geq 0 \\ X_2 &\geq 0 \end{aligned}$$

Yöneylem Araştırması II

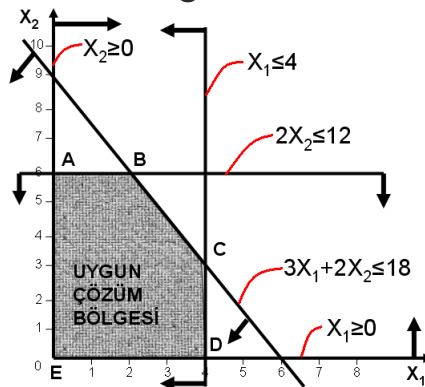
21

[Örnek: Grafik Çözüm Metodu]



$$\begin{aligned} \text{MAKS } Z &= 3X_1 + 5X_2 \\ \text{S.T.} \quad 3X_1 + 2X_2 &\leq 18 \\ X_1 &\leq 4 \\ 2X_2 &\leq 12 \\ X_1 &\geq 0 \\ X_2 &\geq 0 \end{aligned}$$

■ Uygun Çözüm Bölgesi



Yöneylem Araştırması II

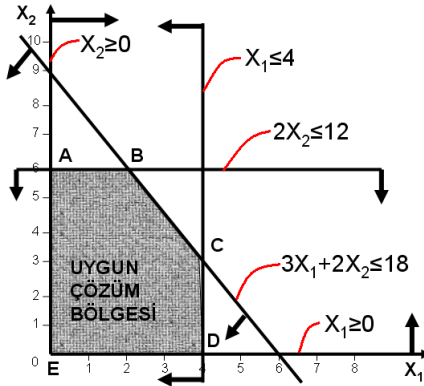
22

Örnek: Grafik Çözüm Metodu



Optimal Çözümün Tespiti (Birinci Yöntem)

- Optimal çözüm köşe noktalarından birinde oluşur. Tüm köşe noktaları içerisinde, amaç fonksiyonu değerini en iyileyen nokta, optimal çözüm noktasıdır.



$$\text{MAKS } Z = 3X_1 + 5X_2$$

NOKTA	(X1,X2)	Z
A	0, 6	30
B	2, 6	36
C	4, 3	27
D	4, 0	12
E	0, 0	0

OPTİMAL ÇÖZÜM

$$\begin{aligned} X_1 &= 2, \\ X_2 &= 6, \\ Z &= 36 \end{aligned}$$

Yöneylem Araştırması II

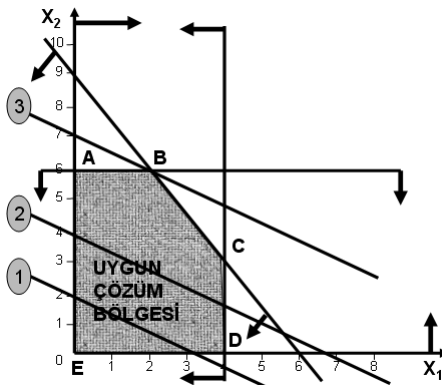
23

Örnek: Grafik Çözüm Metodu



Optimal Çözümün Tespiti (İkinci Yöntem)

- Amaç fonksiyonunu temsil eden doğru, amaç fonksiyonu değerinin iyileştiği istikamette kaydırıldığında uygun çözüm bölgesini terk ettiği son nokta optimal çözüm noktasıdır.



$$(1) Z = 10 = 3X_1 + 5X_2$$

$$(2) Z = 20 = 3X_1 + 5X_2$$

$$(3) Z = ? = 3X_1 + 5X_2$$

$3X_1 + 2X_2 = 18$ VE $2X_2 = 12$
DENKLEMLERİ ORTAK
ÇÖZÜLÜRSE:

OPTİMAL ÇÖZÜM

$$\begin{aligned} X_1 &= 2, \\ X_2 &= 6, \\ Z &= 36 \end{aligned}$$

Yöneylem Araştırması II

24

[Grafik Çözüm Özel Durumlar]



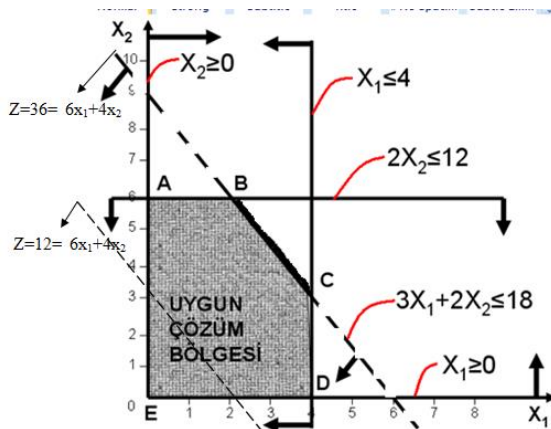
- Bir önceki örneğin amaç fonksiyonu $z = 6x_1 + 4x_2$ şeklinde değiştirildiği takdirde, yeni model;

$$\begin{aligned} \text{Max } Z &= 6x_1 + 4x_2 \\ \text{S.T. } 3x_1 + 2x_2 &\leq 18 \\ x_1 &\leq 4 \\ 2x_2 &\leq 12 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

[Grafik Çözüm Özel Durumlar]



- Alternatif Çözüm mevcuttur. BC doğru parçası üzerindeki her nokta çözümdür.



[Grafik Çözüm Özel Durumlar]



- Modelimiz üzerinde 2. kısıt $x_1 \geq 4$ ve 3. kısıt $2x_2 \geq 12$ şeklinde değiştirildiği takdirde, yeni model;

$$\begin{aligned} \text{Max } Z &= 3x_1 + 5x_2 \\ \text{S.T. } 3x_1 + 2x_2 &\leq 18 \\ x_1 &\geq 4 \\ 2x_2 &\geq 12 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

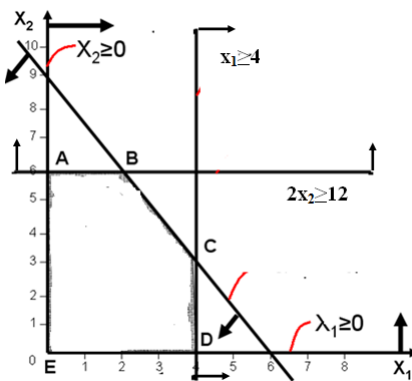
Yöneylem Araştırması II

27

[Grafik Çözüm Özel Durumlar]



- Bu durumda problemin tüm kısıtlarını sağlayan uygun bir çözüm alanı oluşmamaktadır. Dolayısıyla problem **çözüm­süz­dür**.



Yöneylem Araştırması II

28

[Grafik Çözüm Özel Durumlar]



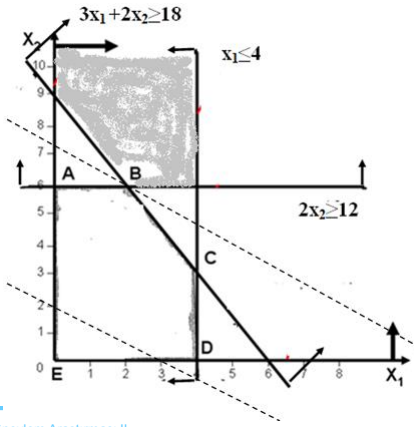
- Modelimiz üzerinde 1. kısıt $3x_1 + 2x_2 \geq 18$ ve 3. kısıt $2x_2 \geq 12$ şeklinde değiştirildiği takdirde, yeni model;

$$\begin{aligned} \text{Max } Z &= 3x_1 + 5x_2 \\ \text{S.T. } 3x_1 + 2x_2 &\geq 18 \\ x_1 &\leq 4 \\ 2x_2 &\geq 12 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \end{aligned}$$

[Grafik Çözüm Özel Durumlar]



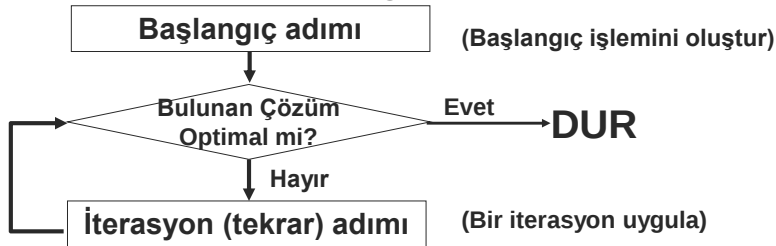
- Bu durumda problemin tüm kısıtlarını sağlayan uygun bir çözüm alanı kapalı bir alan içerisinde değildir. Dolayısıyla problemin çözümü **sınırsızdır**



[Simpleks Metodu]



- Simpleks metodu, gerek değişken sayısı gerekse de kısıt sayısı açısından her boyuttaki doğrusal programlama modellerinin çözümünün, cebirsel işlemler uygulayarak bulunmasını sağlayan bir algoritmadır.
- Simpleks metodu algoritması:



Yöneylem Araştırması II

31

[Simpleks Metodu (Başlangıç İşlemleri)]



- Simpleks metoduyla çözülecek problem öncelikle **Standart (genişletilmiş) Yapıya** dönüştürülür.
- Standart yapıda:
 - Kısıtların sağ taraf değerleri **negatif olmayan sabit sayılar** olmalı,
 - Bütün kısıtlar **eşitlik** şeklinde ifade edilmeli,
 - Bütün değişkenler **negatif olmayan değişken** olarak tanımlanmalı.
- Standart yapıya dönüştürmek için:
 - " $\leq$ " şeklindeki kısıtların sol tarafına, kullanılmayan kapasiteyi ifade eden S_i notasyonlu "**Gevşek Değişken**" eklenerek kısıt " $=$ " şekline çevrilir.
 - " $\geq$ " şeklindeki kısıtların sol tarafından, ihtiyaç fazlasını ifade eden e_i notasyonlu "**Artık Değişken**" çıkartılarak kısıt " $=$ " şekline çevrilir.

Yöneylem Araştırması II

32

[Simpleks Metodu (Optimallik Testi)]



- Her hangi bir iterasyon aşamasındaki simpleks tablosu aşağıdaki koşulları sağlıyorsa **optimaldir**.
 - **Maksimizasyon** probleminde amaç fonksiyonu katsayılarının (0-satırı) tamamı **negatif olmayan** sayılardır.
 - **Minimizasyon** probleminde amaç fonksiyonu katsayılarının (0-satırı) tamamı **pozitif olmayan** sayılardır.

[Simpleks Metodu (İterasyon Safhası)]



- Üç aşamadan oluşur:
 - **Giren Değişkenin Belirlenmesi** : Amaç fonksiyonu satırında, maksimizasyon problemi için en küçük değerli negatif sayıya, minimizasyon problemi için en büyük değerli pozitif sayıya sahip değişkendir. Bu değişkenin bulunduğu sütun "**Anahtar Sütun**" dur.
 - **Çıkan Değişkenin Belirlenmesi** : Anahtar sütununda pozitif katsayıya sahip satırlar için, sağ taraf değerleri anahtar sütun değerlerine bölündüğünde bulunan oranlar içinde minimum değere sahip olan satırdaki değişkendir. Bu işleme "**Minimum Oran Testi**", bulunan satıra da "**Anahtar Satır**" denir. Anahtar satır ve anahtar sütunun kesiştiği noktadaki sayı "**Anahtar Sayı**" dır.
 - **Yeni uygun temel çözümün bulunması** : **Gauss** eleme yöntemi kullanılır. Bu yönetime göre:
 - Yeni anahtar satır = (anahtar satır) / (anahtar sayı)
 - Yeni satır = (eski satır) - [(anahtar sütun katsayısı) x (yeni anahtar satır)]

[Simpleks Metodunda Özel Durumlar]



- Alternatif Optimal Çözüm
 - Optimal çözüm tablosunun amaç fonksiyonu satırında en az bir **temel olmayan değişkenin katsayısı sıfırdır**.
 - Alternatif çözümün bulunabilmesi için amaç fonksiyonu satırında 0 (sıfır) sayısına sahip herhangi bir temel olmayan değişken giren değişken olarak seçilir ve iterasyon işlemleri uygulanır.
- Sınırsız Çözüm
 - Herhangi bir iterasyonda hiç bir temel değişken, çıkan değişken olarak seçilemiyorsa (yani **minimum oran testi yapılamıyorsa**), amaç fonksiyonunun iyileştiği yönde çözüm bölgesi sınırsızdır.

[Farklı Kısıtları Olan Modellerin Çözümü]



- “ $\geq$ ” Kısıtlarının sol tarafından “ e_i ” notasyonlu artık değişken çıkartılıp “ a_j ” notasyonlu yapay değişken ilave edilir.

$$X_1 - 3X_2 \geq 4 \rightarrow X_1 + 3X_2 - e_1 + a_1 = 4$$
- “=” Kısıtlarına “ a_j ” notasyonlu yapay değişken ilave edilir.

$$3X_1 - 2X_2 = 7 \rightarrow 3X_1 - 2X_2 + a_1 = 7$$

[Farklı Kısıtları Olan Modellerin Çözümü]



- “=” ve “≥” kısıtlarına eklenen yapay değişkenlerin gerçekte bir anlamı olmayıp, sadece **başlangıç temel çözümü oluşturabilmek için** kullanılır. Bu yüzden, bu yapay değişkenlerin ileri safhalarda, bir daha temele girmemek üzere, **temel çözümü terk etmesi** gerekir. Bunu sağlamak amacıyla aşağıdaki iki metot kullanılmaktadır.
 - Büyük M (Big-M) metodu
 - İki safha (Two Phase) metodu

Yöneylem Araştırması II

37

[Büyük-M Metodu]



- Yapay değişkenlerin uygun temel çözümden çıkmasını sağlayacak şekilde amaç fonksiyonuna yapay değişkenlerle ilgili **büyük cezalar eklenmesine dayalı** bir yöntemdir.
- Bu amaçla her bir yapay değişken (a_j) için amaç fonksiyonuna;
 - amaç **maksimizasyon** ise; $-Ma_j$
 - amaç **minimizasyon** ise; Ma_j
 değerleri ilave edilir.
- Bu değerlerdeki **M** katsayısı çok büyük bir pozitif sayıyı temsil eder.

Yöneylem Araştırması II

38

[ÖRNEK: Büyük-M Metodu]



DOĞRUSAL MODEL

$$\text{MIN } Z = 2 X_1 + 3 X_2$$

S.T.

$$\begin{aligned} \frac{1}{2} X_1 + \frac{1}{4} X_2 &\leq 4 \\ X_1 + 3 X_2 &\geq 20 \\ X_1 + X_2 &= 10 \\ X_1, X_2 &\geq 0 \end{aligned}$$

STANDART YAPI

$$\begin{aligned} Z \quad & -2 X_1 - 3 X_2 \quad \quad \quad -M a_2 \quad -M a_3 = 0 \\ & \frac{1}{2} X_1 + \frac{1}{4} X_2 + S_1 \quad \quad \quad = 4 \\ & X_1 + 3 X_2 \quad \quad -e_2 \quad + a_2 = 20 \\ & X_1 + X_2 \quad \quad \quad + a_3 = 10 \\ & X_1, X_2, S_1, e_2, a_2, a_3 \geq 0 \end{aligned}$$

Yöneylem Araştırması II

39

[ÖRNEK: Büyük-M Metodu]



STANDART YAPI

$$\begin{aligned} Z \quad & -2 X_1 - 3 X_2 \quad \quad \quad -M a_2 \quad -M a_3 = 0 \\ & \frac{1}{2} X_1 + \frac{1}{4} X_2 + S_1 \quad \quad \quad = 4 \\ & X_1 + 3 X_2 \quad \quad -e_2 \quad + a_2 = 20 \\ & X_1 + X_2 \quad \quad \quad + a_3 = 10 \\ & X_1, X_2, S_1, e_2, a_2, a_3 \geq 0 \end{aligned}$$

İte. No	TD	Değ. No	Katsayılar							STD
			Z	X ₁	X ₂	S ₁	e ₂	a ₂	a ₃	
0	Z	0	1	-2	-3	0	0	-M	-M	0
	S ₁	1	0	1/2	1/4	1	0	0	0	4
	a ₂	2	0	1	3	0	-1	1	0	20
	a ₃	3	0	1	1	0	0	0	1	10

xM

xM

Yöneylem Araştırması II

40

[ÖRNEK: Büyük-M Metodu]



İte. No	TD	Değ. No	Katsayılar							STD
			Z	X_1	X_2	S_1	e_2	a_2	a_3	
0	Z	0	1	$2M-2$	$4M-3$	0	$-M$	0	0	$30M$
	S_1	1	0	$\frac{1}{2}$	$\frac{1}{4}$	1	0	0	0	4
	a_2	2	0	1	3	0	-1	1	0	20
	a_3	3	0	1	1	0	0	0	1	10
1	Z	0	1	$(2M-3)/3$	0	0	$(M-3)/3$	$(3-4M)/3$	0	$(60+10M)/3$
	S_1	1	0	$5/12$	0	1	$1/12$	$-1/12$	0	$7/3$
	X_2	2	0	$1/3$	1	0	$-1/3$	$1/3$	0	$20/3$
	a_3	3	0	$2/3$	0	0	$1/3$	$-1/3$	1	$10/3$
2	Z	0	1	0	0	0	$-1/2$	$(-M+1)/2$	$(-2M+3)/2$	25
	S_1	1	0	0	0	1	$-1/8$	$1/8$	$5/8$	$1/4$
	X_2	2	0	0	1	0	$-1/2$	$\frac{1}{2}$	$-1/2$	5
	X_1	3	0	1	0	0	$1/2$	$-1/2$	$3/2$	5

OPTİMAL ÇÖZÜM

$Z=25$, $X_1=5$, $X_2=5$

Yöneylem Araştırması II

41

[İki Safha Metodu]



- Yapay değişkenlerin ilave edildiği amaç fonksiyonunu, iki farklı amaç fonksiyonuna dönüştürerek, çözüme **iki aşamada** ulaşır.
- Birinci safha
 - Amaç fonksiyonu yapay değişkenlerin **toplamı**dır.
 - Simpleks metoduyla çözülürse amaç fonksiyonu değeri sıfır ($z=0$) elde edilir.
- İkinci safha
 - Birinci safha sonucunda elde edilen simpleks tablosunda yapay değişkenlerin bulunduğu sütunlar **silinir**.
 - Gerçek problemin amaç fonksiyonu, amaç fonksiyonu satırına yerleştirilerek simpleks metodu uygulanır.

Yöneylem Araştırması II

42

[ÖRNEK: İki Safha Metodu



DOĞRUSAL MODEL

$$\begin{array}{ll} \text{MIN } Z = a_2 + a_3 & \rightarrow \text{ - MAKS } Z = -a_2 \quad -a_3 \\ \text{S.T.} & \end{array}$$

$$\begin{array}{rcl} \frac{1}{2} X_1 + \frac{1}{4} X_2 & \leq & 4 \\ X_1 + 3 X_2 & \geq & 20 \\ X_1 + X_2 & = & 10 \\ X_1, X_2 & \geq & 0 \end{array}$$

STANDART YAPI

$$\begin{array}{rclcl}
 -Z & & & +a_2 & +a_3 & =0 \\
 & \frac{1}{2} X_1 + \frac{1}{4} X_2 & +S_1 & & & =4 \\
 & X_1 + 3 X_2 & & -e_2 & +a_2 & =20 \\
 & X_1 + & X_2 & & & +a_3 =10 \\
 & X_1, & X_2, S_1, & e_2, & a_2, & a_3 \geq 0
 \end{array}$$

[ÖRNEK: İki Safha Metodu



STANDART YAPI

$$\begin{array}{rclcl}
-Z & & & +a_2 & +a_3 & =0 \\
\frac{1}{2} X_1 + \frac{1}{4} X_2 + S_1 & & & & & =4 \\
X_1 + 3 X_2 & -e_2 & +a_2 & & & =20 \\
X_1 + X_2 & & & & +a_3 & =10 \\
X_1, X_2, S_1, e_2, a_2, a_3 & & & & & \geq 0
\end{array}$$

İte. No	TD	Değ. No	Katsayılar							STD
			Z	X ₁	X ₂	S ₁	e ₂	a ₂	a ₃	
0	Z	0	-1	0	0	0	0	1	1	0
	S ₁	1	0	1/2	1/4	1	0	0	0	4
	a ₂	2	0	1	3	0	-1	1	0	20
	a ₃	3	0	1	1	0	0	0	1	10

[ÖRNEK: İki Safha Metodu]



■ Birinci Safha

İte. No	TD	Değ. No	Katsayılar							STD
			Z	X ₁	X ₂	S ₁	e ₂	a ₂	a ₃	
0	Z	0	-1	-2	-4	0	1	0	0	-30
	S ₁	1	0	1/2	1/4	1	0	0	0	4
	a ₂	2	0	1	3	0	-1	1	0	20
	a ₃	3	0	1	1	0	0	0	1	10
1	Z	0	-1	-2/3	0	0	-1/3	4/3	0	-10/3
	S ₁	1	0	5/12	0	1	1/12	-1/12	0	7/3
	X ₂	2	0	1/3	1	0	-1/3	1/3	0	20/3
	a ₃	3	0	2/3	0	0	1/3	-1/3	1	10/3
2	Z	0	-1	0	0	0	0	1	1	0
	S ₁	1	0	0	0	1	-1/8	1/8	-5/8	1/4
	X ₂	2	0	0	1	0	-1/2	1/2	-1/2	5
	X ₁	3	0	1	0	0	1/2	-1/2	3/2	5

Yöneylem Araştırması II

45

[ÖRNEK: İki Safha Metodu]



■ İkinci Safha

$$\text{MIN } Z = 2X_1 + 3X_2 \rightarrow \boxed{-\text{MAKS } Z = -2X_1 - 3X_2}$$

AMAÇ FONKSİYONU STANDART YAPISI

$$-Z + 2X_1 + 3X_2 = 0$$

İte. No	TD	Değ. No	Katsayılar					STD
			Z	X ₁	X ₂	S ₁	e ₂	
0	Z	0	-1	2	3	0	0	0
	S ₁	1	0	0	0	1	-1/8	1/4
	X ₂	2	0	0	1	0	-1/2	5
	X ₁	3	0	1	0	0	1/2	5
0	Z	0	-1	0	0	0	1/2	-25
	S ₁	1	0	0	0	1	-1/8	1/4
	X ₂	2	0	0	1	0	-1/2	5
	X ₁	3	0	1	0	0	1/2	5

x(-3)

x(-2)

OPTİMAL ÇÖZÜM
 $Z = (-1) \times -25 = 25$,
 $X_1 = 5$,
 $X_2 = 5$

Yöneylem Araştırması II

46

[Geçersiz Çözüm]



- Problemin çözümünde **Büyük-M Metodu** kullanılıyorsa ve optimal çözüm tablosunda yapay değişkenlerden **en az** bir tanesi sıfırdan farklı bir değer almış ise elde edilen çözüm uygun bir çözüm değildir (yani problemin optimal çözümü yoktur).
- Problemin çözümünde **İki Safha Metodu** kullanılıyorsa ve birinci safhanın optimal çözüm tablosunda yapay değişkenlerden **en az** bir tanesi sıfırdan farklı bir değer almışsa, elde edilen optimal çözüm uygun bir çözüm değildir (yani problemin optimal çözümü yoktur).

[Dual Simpleks Metodu]



- Primal simpleks metodu optimal olmayan uygun bir çözümle algoritmaya başlarken, dual simpleks metodu optimal ancak uygun olmayan bir çözümle algoritmaya başlar. Öyleki primal simpleks tablosunun sağ taraf değerleri dual simpleks tablosunun amaç fonksiyonu değerlerini oluşturur.
- Dual simpleks metodu, amaç fonksiyonu satırındaki tüm katsayıların negatif olmasını engeller (dual uygunluk). Aynı zamanda tüm sağ taraf değerlerinin de negatif olmamasını (primal uygunluk) sağladığı noktada optimal çözüm elde edilmiş olur. Bu teknik her adımda dual uygunluğu sağlaması sebebiyle dual simpleks olarak adlandırılır.

[Dual Simpleks Metodu]



- Adım 1: Amaç fonksiyonu satırındaki tüm katsayıların negatif olmaması koşulu ile eğer tüm sağ taraf değerleri negatif değil ise **optimal çözüme ulaşılmıştır**. Eğer en az bir sağ taraf değeri negatif ise ikinci adıma geçilir.
- Adım 2: **En negatif sağ** taraf değerine sahip değişken **çıkan değişken** olarak seçilir ve o değişkene ait satır **anahtar satırdır**. Giren değişkeni belirlemek için ise anahtar satırdaki negatif katsayıları sahip olan her X_j değişkeni için aşağıdaki oran testi uygulanır ve minimum orana sahip değişken **giren değişken** olarak seçilir. Elemanter satır işlemleriyle bir sonraki dual simpleks tablosu bulunur.

$$\frac{X_j \text{ Değişkeninin amaç fonksiyonu satırındaki katsayısı}}{X_j \text{ Değişkeninin anahtar satırındaki katsayısı}}$$

- Adım 3: Eğer anahtar satırda hiçbir değişkenin katsayısı negatif değilse modelin çözümü yoktur. Böyle bir durum yok ise birinci adıma geri dönlür.

[Dual Simpleks Metodu]



- Dual simpleks metodunun kullanım alanları
 - DP modeline yeni bir kısıt eklenmesi durumunda yeni optimal çözümün bulunmasında.
 - DP modelinin sağ taraf değerlerinin değişmesi durumunda yeni optimal çözümün bulunmasında.
 - Normal minimizasyon problemlerinin çözümünde.
- Bu ders kapsamında sadece yeni kısıt eklenmesine örnek verilecektir.

[Dual Simpleks Metodu (Yeni Kısıt Ek.)]



- Modele yeni bir kısıt eklendiğinde aşağıdaki üç durumdan biri ile karşılaşılır:
 - Mevcut optimal çözümün yeni kısıtı sağlaması.
 - Mevcut optimal çözümün yeni kısıtı sağlamaması ancak doğrusal programlama modelinin hala uygun bir çözüme sahip olması.
 - Modele yeni kısıt eklemenin doğrusal programlama modelinde çözümsüzlüğe yol açması.

Yöneylem Araştırması II

51

[Dual Simpleks Metodu (Yeni Kısıt Ek.)]



$$\begin{aligned}
 \text{MaksZ} &= 60 X_1 + 30 X_2 + 20 X_3 \\
 \text{S.T.} \quad &8 X_1 + 6 X_2 + X_3 \leq 48 \\
 &4 X_1 + 2 X_2 + 1.5 X_3 \leq 20 \\
 &2 X_1 + 1.5 X_2 + 0.5 X_3 \leq 8 \\
 &\quad \quad \quad X_2 \leq 5 \\
 &X_1, X_2, X_3 \geq 0
 \end{aligned}$$

OPTİMAL ÇÖZÜM TABLOSU

TD	D. No	Katsayılar								STD
		Z	X ₁	X ₂	X ₃	S ₁	S ₂	S ₃	S ₄	
Z	0	1	0	5	0	0	10	10	0	280
S ₁	1	0	0	-2	0	1	2	-8	0	24
X ₃	2	0	0	-2	1	0	2	-4	0	8
X ₁	3	0	1	5/4	0	0	-1/2	3/2	0	2
S ₄	4	0	0	1	0	0	0	0	1	5

Yöneylem Araştırması II

52

[Dual Simpleks Metodu (Yeni Kısıt Ek.)]



- $X_1 + X_2 + X_3 \leq 11 \rightarrow X_1 + X_2 + X_3 = 2 + 0 + 8 = 10 < 11 \Rightarrow$
Optimal Çözüm Değişmez. (Durum 1)

OPTIMAL ÇÖZÜM TABLOSU

TD	D. No	Katsayılar								STD
		Z	X_1	X_2	X_3	S_1	S_2	S_3	S_4	
Z	0	1	0	5	0	0	10	10	0	280
S_1	1	0	0	-2	0	1	2	-8	0	24
X_3	2	0	0	-2	1	0	2	-4	0	8
X_1	3	0	1	5/4	0	0	-1/2	3/2	0	2
S_4	4	0	0	1	0	0	0	0	1	5

Yöneylem Araştırması II

53

[Dual Simpleks Metodu (Yeni Kısıt Ek.)]



- $X_2 \geq 1 \rightarrow X_2 = 0 \not\geq 1 \Rightarrow$ Optimal Çözüm Değişir.
- $X_2 \geq 1 \rightarrow X_2 - e_5 = 1 \rightarrow -X_2 + e_5 = -1$ Bu kısıt optimal tabloya eklenirse (Temel değişkeni e_5):

TD	D. No	Katsayılar										STD
		Z	X_1	X_2	X_3	S_1	S_2	S_3	S_4	e_5		
Z	0	1	0	5	0	0	10	10	0	0	280	
S_1	1	0	0	-2	0	1	2	-8	0	0	24	
X_3	2	0	0	-2	1	0	2	-4	0	0	8	
X_1	3	0	1	5/4	0	0	-1/2	3/2	0	0	2	
S_4	4	0	0	1	0	0	0	0	1	0	5	
e_5	5	0	0	-1	0	0	0	0	0	1	-1	
Z	0	1	0	0	0	0	10	10	0	5	275	
S_1	1	0	0	0	0	1	2	-8	0	-2	26	
X_3	2	0	0	0	1	0	2	-4	0	-2	10	
X_1	3	0	1	0	0	0	-1/2	3/2	0	5/4	3/4	
S_4	4	0	0	0	0	0	0	0	1	1	4	
X_2	5	0	0	1	0	0	0	0	0	-1	1	

- Optimal çözüm değişmiştir (Durum 2).

Yöneylem Araştırması II

54

[Dual Simpleks Metodu (Yeni Kısıt Ek.)]



- $X_1 + X_2 \geq 12 \rightarrow 2+0=2 \neq 12 \Rightarrow$ Optimal Çözüm Değişir.
- $X_1 + X_2 \geq 12 \rightarrow X_1 + X_2 - e_5 = 12 \rightarrow -X_1 - X_2 + e_5 = -12$ Bu kısıt optimal tabloya eklenirse (Temel değişkeni e_5):

TD	D. No	Katsayılar									STD
		Z	X ₁	X ₂	X ₃	S _f	S ₂	S ₃	S ₄	e ₅	
Z	0	1	0	5	0	0	10	10	0	0	280
S _f	1	0	0	-2	0	1	2	-8	0	0	24
X ₃	2	0	0	-2	1	0	2	-4	0	0	8
X ₁	3	0	1	5/4	0	0	-1/2	3/2	0	0	2
S ₄	4	0	0	1	0	0	0	0	1	0	5
e ₅	5	0	-1	-1	0	0	0	0	0	1	-12

Yöneylem Araştırması II

55

[Dual Simpleks Metodu (Yeni Kısıt Ek.)]



TD	D. No	Katsayılar										STD
		Z	X ₁	X ₂	X ₃	S _f	S ₂	S ₃	S ₄	e ₅		
Z	0	1	0	5	0	0	10	10	0	0	280	
S _f	1	0	0	-2	0	1	2	-8	0	0	24	
X ₃	2	0	0	-2	1	0	2	-4	0	0	8	
X ₁	3	0	1	5/4	0	0	-1/2	3/2	0	0	2	
S ₄	4	0	0	1	0	0	0	0	1	0	5	
e ₅	5	0	0	1/4	0	0	-1/2	3/2	0	1	-10	
Z	0	1	0	10	0	0	0	40	0	20	80	
S _f	1	0	0	-1	0	1	0	-2	0	4	-16	
X ₃	2	0	0	-1	1	0	0	2	0	4	-32	
X ₁	3	0	1	1	0	0	0	0	0	-1	12	
S ₄	4	0	0	1	0	0	0	0	1	0	5	
S ₂	5	0	0	-1/2	0	0	1	-3	0	-2	20	
Z	0	1	0	0	10	0	0	60	0	60	-240	
S _f	1	0	0	0	-1	1	0	-4	0	0	16	
X ₂	2	0	0	1	-1	0	0	-2	0	-4	32	
X ₁	3	0	1	0	1	0	0	2	0	3	-20	
S ₄	4	0	0	0	1	0	0	2	1	4	-27	
S ₂	5	0	0	0	-1/2	0	1	-4	0	-4	36	

- Çözumsuz (Durum 3)

Yöneylem Araştırması II

56

BÖLÜM III: Tamsayı Programlama

- Tamsayı programlamaya giriş
- Tam sayılı programlama problemlerinin modellenmesi
- Tamsayı programlamada çözüm yaklaşımları
- Dal ve sınır algoritması
- Gomory kesme düzlemi algoritması

Tamsayı Programlamaya Giriş



- **Tamsayı doğrusal programlama problemi** (ya da yaygın olarak bilinen ismiyle tamsayı programlama problemi), karar değişkenlerinin tamamının ya da bir kısmının **tamsayı değerler almaya zorlandığı** bir doğrusal programlama problemidir.
- Genellikle çözümü doğrusal programlama problemlerine göre **daha zordur**.
- Karar değişkenlerinin tanımlanmasına göre tamsayı programlama problemleri **üç** grupta değerlendirilir. Bunlar;
 - **Tümü tamsayı** programlama problemi
 - **Karma tamsayı** programlama problemi
 - **0-1 tamsayı** programlama problemi

[Tamsayıli Programlamaya Giriş]



■ Tümü tamsayıli programlama problemi

- Tüm karar deęişkenlerinin tamsayı olma zorunluluęu olan problemlerdir.
- Bu tip problemlerin modellenmesinde negatif olmama koşuluna ilave olarak tüm deęişkenler için **tamsayı olma koşulu** ilave edilir.

$$\begin{aligned} \text{Maks } z &= 3x_1 + 2x_2 \\ x_1 + 2x_2 &\leq 6 \\ x_1, \quad x_2 &\geq 0 \\ x_1 \text{ ve } x_2 &\text{ tamsayı} \end{aligned}$$

[Tamsayıli Programlamaya Giriş]



■ Karma Tamsayıli Programlama Problemi

- Karar deęişkenlerinden **bazılarının tamsayı** olma zorunluluęu olan problemlerdir.
- Bu tip problemlerin modellenmesinde tamsayı deęer alma zorunluluęu olan deęişkenler için negatif olmama koşuluna ilave olarak **tamsayı olma koşulu** ilave edilir. Dięer deęişkenler için ise sadece negatif olmama koşulu kullanılır.

$$\begin{aligned} \text{Maks } z &= 3x_1 + 2x_2 \\ x_1 + 2x_2 &\leq 6 \\ x_1, \quad x_2 &\geq 0 \\ x_1 &\text{ tamsayı} \end{aligned}$$

[Tamsayılı Programlamaya Giriş]



■ 0-1 Tamsayılı Programlama Problemi

- Herhangi bir faaliyet için sadece **iki karar alternatifinin bulunduđu** ve bunlardan birisine karar verilmesi gerektiđi durumlarda 0-1 tamsayılı programlama modeli kullanılır.
- 0-1 tamsayılı programlama modelinde karar deđişkenleri **ya "0" ya da "1"** deđerini alabilir ve bu deđişkenler **0-1 deđişken** olarak adlandırılır.
- 0-1 tamsayılı programlama modelindeki deđişkenler 0 veya 1 deđerini aldıđından karar deđişkenleri için **negatif olmama koşulunun kullanılmasına gerek yoktur**.

$$\begin{aligned} \text{Maks } z &= 3x_1 + 2x_2 \\ x_1 + 2x_2 &\leq 3 \\ x_1, x_2 &= 0 \text{ veya } 1 \end{aligned}$$

[Tamsayılı Programlamaya Giriş]



■ Gevşetilmiş Tamsayılı Programlama

- Bir tamsayılı programlama modelinde deđişkenler üzerindeki bütün 0-1 ve/veya **tamsayı olma koşullarını** göz ardı ederek elde edilen modele **Gevşetilmiş Tamsayılı Programlama Modeli** adı verilir.
- Gevşetilmiş tamsayılı programlama modeli tamsayılı programlama modeline göre **daha az sınırlandırılmıştır**.
- Gevşetilmiş tamsayılı programlama modelinin **uygun çözüm bölgesi**, aynı zamanda tamsayılı programlama modelinin uygun çözüm bölgesini de **kapsamaktadır**.
- Tamsayılı programlama probleminin uygun çözüm bölgesi **konveks küme olmak zorunda deđildir**.
- $Z_{(TP)}$ tamsayılı programlama modeli optimal çözüm deđeri ve $Z_{(DP)}$ gevşetilmiş tamsayılı programlama modeli optimal çözüm deđeri olmak üzere aşağıdaki durum her zaman sağlanır:

$$Z_{(TP)} \leq Z_{(DP)}$$

[Tamsayılı Programlamaya Giriş]

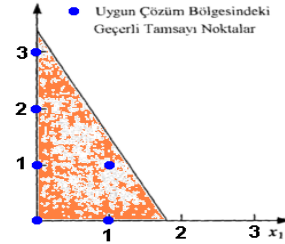


■ Gevşetilmiş Tamsayılı Programlama

- Aşağıda tamsayı olma koşulu ilave edilmeden verilen bir gevşetilmiş tamsayılı programlama modeli (doğrusal programlama modeli) verilmiş ve uygun çözüm bölgesi gösterilmiştir.

$$\begin{aligned} \text{Maks } z &= 21x_1 + 11x_2 \\ 7x_1 + 4x_2 &\leq 13 \\ x_1, \quad x_2 &\geq 0 \end{aligned}$$

OPTİMAL ÇÖZÜM:
 $x_1=1.85, x_2=0, Z_{(DP)}=39$



- Aynı modele tamsayı olma koşulu ilave edildiğinde, uygun çözümleri veren noktalar kümesi $s = \{(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1)\}$ şeklinde bulunur. Her noktaya karşılık gelen amaç fonksiyonu değeri hesaplanarak optimal çözüm (en yüksek Z değerini veren nokta), $x_1=0, x_2=3$ ve $Z_{(TP)}=33$ olarak bulunur. Bu durumda;

$$Z_{(TP)} = 33 \leq 39 = Z_{(DP)}$$

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Sermaye Bütçeleme Problemleri

- Belirli bir miktar para, işgücü, enerji, mekan vb. Kaynakların, bir dönem veya belirli sayıda dönemler için; hisse senedi, tahvil, yatırım, araştırma-geliştirme tekliflerinin ya da belirli projelerin değerlendirilmesi ve seçimi gibi benzeri faaliyetlere paylaştırılmasına ilişkin karar problemleri, genel olarak **Sermaye Bütçeleme Problemi** olarak adlandırılır.

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sermaye Bütçeleme Problemleri

Yatırım alternatiflerinin değerlendirilmesi

- Bir firmanın yatırım yapabileceği 4 alternatifi vardır, 1nci yatırımdan elde edilecek net şimdiki değer 16 birim, 2nci yatırımdan 22, 3ncü yatırımdan 12, 4ncü yatırımdan ise 8 birimdir. Yatırımlar için şu an yapılması gereken birim harcamalar ise sırası ile 5, 7, 4 ve 3 birimdir. Firmanın yatırımlar için ayırdığı toplam para miktarı 14 birim olduğuna göre yatırımlardan elde edilecek net şimdiki değeri **maksimize edecek** tamsayılı programlama modelini kurunuz.

1. KARAR DEĞİŞKENLERİ

$$x_j \ (j=1,2,3,4) = \begin{cases} 1, j \text{ yatırımı yapılırsa} \\ 0, j \text{ yatırımı yapılmazsa} \end{cases}$$

2. AMAÇ FONKSİYONU

$$\text{Maks } z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$

3. KISITLAR

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sermaye Bütçeleme Problemleri

Yatırım alternatiflerinin değerlendirilmesi

- Firmanın yapabileceği yatırım alternatiflerinin değerlendirilerek, karını maksimum yapacak **0-1** tamsayılı doğrusal programlama modeli aşağıdaki gibi kurulur.

$$\begin{aligned} \text{Maks } z = & 16x_1 + 22x_2 + 12x_3 + 8x_4 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_j = 0 \text{ veya } 1 \quad (j=1, 2, 3, 4) \end{aligned}$$

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sermaye Bütçeleme Problemleri

- Yatırım alternatiflerinin değerlendirilmesi

Durum

Kısıt

Firma en fazla iki yatırım yapmak istemektedir.

$$x_1 + x_2 + x_3 + x_4 \leq 2$$

Firma dördüncü yatırımı yaptığında ikinci yatırımı da yapmayı tercih etmektedir

$$x_4 \leq x_2 \quad \text{veya} \quad x_4 - x_2 \leq 0$$

Firma birinci ve üçüncü yatırımları aynı zamanda yapamamaktadır

$$x_1 + x_3 \leq 1$$

$$\text{Maks } z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$

$$\text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$x_1 + x_2 + x_3 + x_4 \leq 2$$

$$-x_2 + x_4 \leq 0$$

$$x_1 + x_3 \leq 1$$

$$x_j = 0 \text{ veya } 1 \quad (j=1, 2, 3, 4)$$

Yöneylem Araştırması II

67

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri

- Herhangi bir faaliyeti gerçekleştirebilmek için, bu faaliyetin seviyesine bağlı olmaksızın bir maliyete katlanılmasını gerektiren problemlere **Sabit Maliyet Problemi** denir.

Yöneylem Araştırması II

68

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Üretim Planlama)

- Bir tekstil firması 3 tip giyecek üretmektedir; **Gömlek**, **Şort** ve **Pantolon**. Her tip giyeceğin üretimi uygun tip tezgahlar gerektirmektedir. Gömlek üretimi için gerekli tezgahlar haftalığı 200 birim, şort üretimi için gerekli tezgahlar haftalığı 150 birim ve pantolon üretimi için gerekli tezgahlar haftalığı 100 birim karşılığında kiralanmaktadır. Her tip ürünün 1 adetinin gerektirdiği kumaş ve işçilik miktarları ile her tip ürünün değişken birim maliyeti ve birim satış fiyatı aşağıdaki tabloda verilmiştir. Her hafta 150 saat işçilik ve 160 metrekare kumaş mevcut olduğuna göre firmanın haftalık karını maksimum yapacak tamsayılı programlama modelini kurunuz.

	İşçilik (saat)	Kumaş (m ²)	Satış Fiyatı	Değişken Maliyet
Gömlek	3	4	12	6
Şort	2	3	8	4
Pantolon	6	4	15	8

Yöneylem Araştırması II

69

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Üretim Planlama)

- Karar değişkenleri:
 - X1** :bir haftada üretilecek gömlek miktarı,
 - X2** :bir haftada üretilecek şort miktarı,
 - X3** :bir haftada üretilecek pantolon miktarı
 - Ayrıca tezgahların kiralanması, ürünlerin üretim miktarından **bağımsız** olmakta ve sadece ilgili ürünün üretilip üretilmemesi kararına bakmaktadır. Öyleyse modelde kullanılmak üzere aşağıdaki değişkenlerin de tanımlanması gerekir.

$$y_1 = \begin{cases} 1, & \text{gömlek üretilirse} \\ 0, & \text{gömlek üretilmezse} \end{cases}$$

$$y_2 = \begin{cases} 1, & \text{şort üretilirse} \\ 0, & \text{şort üretilmezse} \end{cases}$$

$$y_3 = \begin{cases} 1, & \text{pantolon üretilirse} \\ 0, & \text{pantolon üretilmezse} \end{cases}$$

Yöneylem Araştırması II

70

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Sabit Maliyet Problemleri (Üretim Planlama)

○ Amaç fonksiyonu:

- Firmanın amacı daha önce belirtildiği gibi haftalık satış karını maksimum yapmaktır. Firmanın haftalık karı; ürünlerin satışından elde edilecek haftalık gelirden toplam haftalık maliyetlerin (haftalık değişken maliyet ile haftalık tezgah kiralama maliyeti toplamının) çıkarılması ile bulunur. Öyleyse;
- Haftalık kar = $(12-6)X_1 + (8-4)X_2 + (15-8)X_3 - (200y_1 + 150y_2 + 100y_3)$
- $\text{Maks } z = 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$

	İşçilik (saat)	Kumaş (m ²)	Satış Fiyatı	Değişken Maliyet
Gömlek	3	4	12	6
Şort	2	3	8	4
Pantolon	6	4	15	8

Yöneylem Araştırması II

71

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Sabit Maliyet Problemleri (Üretim Planlama)

○ Kısıtlar:

- $3x_1 + 2x_2 + 6x_3 \leq 150$ (işçilik kısıtı)
- $4x_1 + 3x_2 + 4x_3 \leq 160$ (kumaş kısıtı)

- $x_j > 0$ olduğunda $y_j = 1$ olmasını sağlayacak ilave kısıtlar (M_j çok büyük bir sayı olmak üzere)

- $x_1 \leq M_1 y_1$
- $x_2 \leq M_2 y_2$
- $x_3 \leq M_3 y_3$

	İşçilik (saat)	Kumaş (m ²)	Satış Fiyatı	Değişken Maliyet
Gömlek	3	4	12	6
Şort	2	3	8	4
Pantolon	6	4	15	8

Yöneylem Araştırması II

72

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Üretim Planlama)

- Problemin tamsayı programlama modeli

$$\begin{aligned}
 \text{Maks } z &= 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3 \\
 \text{s.T. } 3x_1 + 2x_2 + 6x_3 &\leq 150 \\
 4x_1 + 3x_2 + 4x_3 &\leq 160 \\
 x_1 &\leq M_1y_1 \\
 x_2 &\leq M_2y_2 \\
 x_3 &\leq M_3y_3 \\
 x_j &\geq 0 \text{ ve tamsayı } (j=1, 2, 3) \\
 y_j &= 0 \text{ veya } 1 \quad (j=1, 2, 3)
 \end{aligned}$$

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Depo Yeri seçimi)

- Bir firma, satış politikası olarak, ürünlerini belirli yerlerdeki depolar kanalıyla müşterilerine teslim ederek pazarlamak istemektedir. Açılabilir depo yerleri, bunların kapasiteleri, bu yerlerden müşterilere birim taşıma maliyetleri, müşterilerin talepleri ve belirlenen yerlerde depo açma masrafları aşağıdaki tabloda verilmiştir. Firma yöneticisi, talebi karşılamak üzere, hangi yerlerde depo açması gerektiğini bilmek istemektedir.

Depolar	Birim Taşıma Maliyetleri				Depo Kapasiteleri	Depo Açma Masrafı
	Müşteriler					
	M ₁	M ₂	M ₃	M ₄		
D ₁	4	5	4	3	140	7500
D ₂	1	2	3	1	150	6200
D ₃	2	2	2	3	170	7100
Talep	50	70	40	50		

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Depo Yeri seçimi)

Karar değişkenleri:

- Firmanın karar vermesi gereken iki husus vardır. Birinci husus belirlenen yerlere depo açılıp açılmayacağı, ikincisi ise her bir depodan her bir müşteriye ne kadar ürün gönderileceğidir.

x_{ij} : i -nci depodan j -nci müşteriye gönderilecek mal miktarı ($i=1, 2, 3$)
($j=1, 2, 3, 4$)

$$y_i \ (i=1,2,3) = \begin{cases} 1, & \text{inci depo açılırsa} \\ 0, & \text{inci depo açılmazsa} \end{cases}$$

Depolar	Birim Taşıma Maliyetleri				Depo Kapasiteleri	Depo Açma Masrafı
	Müşteriler					
	M ₁	M ₂	M ₃	M ₄		
D ₁	4	5	4	3	140	7500
D ₂	1	2	3	1	150	6200
D ₃	2	2	2	3	170	7100
Talep	50	70	40	50		

Yöneylem Araştırması II

75

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Depo Yeri seçimi)

Amaç fonksiyonu:

- Firmanın amacı minimum maliyetle dağıtımın gerçekleştirilmesi olduğuna göre, toplam taşıma maliyeti (her bir depodan her bir müşteriye gönderilen ürünlerin taşıma maliyetleri toplamı) ile depo açma maliyetleri toplamını minimum yapan amaç fonksiyonu aşağıda verilmiştir.

$$\text{Min } z = \sum_{i=1}^3 \sum_{j=1}^4 c_{ij} x_{ij} + 7500y_1 + 6200y_2 + 7100y_3$$

	Birim Taşıma Maliyetleri				Depo Kapasiteleri	Depo Açma Masrafı
	Müşteriler					
Depolar	M ₁	M ₂	M ₃	M ₄		
D ₁	4	5	4	3	140	7500
D ₂	1	2	3	1	150	6200
D ₃	2	2	2	3	170	7100
Talep	50	70	40	50		

Yöneylem Araştırması II

76

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Depo Yeri seçimi)

Kısıtlar

- Arz kısıtları depo kapasitesinin yanı sıra, ilgili depodan herhangi bir müşteriye bir ürün gönderilecekse mutlaka o deponun açılmasını sağlar.

Talep kısıtları

$$\begin{aligned}x_{11} + x_{21} + x_{31} &\geq 50 \\x_{12} + x_{22} + x_{32} &\geq 70 \\x_{13} + x_{23} + x_{33} &\geq 40 \\x_{14} + x_{24} + x_{34} &\geq 50\end{aligned}$$

Arz kısıtları

$$\begin{aligned}x_{11} + x_{12} + x_{13} + x_{14} &\leq 140y_1 \\x_{21} + x_{22} + x_{23} + x_{24} &\leq 150y_2 \\x_{31} + x_{32} + x_{33} + x_{34} &\leq 170y_3\end{aligned}$$

Depolar	Birim Taşıma Maliyetleri				Depo Kapasiteleri	Depo Açma Masrafı
	Müşteriler					
	M ₁	M ₂	M ₃	M ₄		
D ₁	4	5	4	3	140	7500
D ₂	1	2	3	1	150	6200
D ₃	2	2	2	3	170	7100
Talep	50	70	40	50		

Yöneylem Araştırması II

77

Tam Sayılı Programlama Problemlerinin Modellenmesi



Sabit Maliyet Problemleri (Depo Yeri seçimi)

Problemin tamsayı programlama modeli

- Oluşturulan modele negatif olmama ve tamsayı olma koşulları da eklendiğinde modelin genel yapısı aşağıda verilmiştir.

$$\text{Min } z = \sum_{i=1}^3 \sum_{j=1}^4 c_{ij} x_{ij} + 7500y_1 + 6200y_2 + 7100y_3$$

$$\begin{aligned}\text{S.T. } &x_{11} + x_{21} + x_{31} \geq 50 \\&x_{12} + x_{22} + x_{32} \geq 70 \\&x_{13} + x_{23} + x_{33} \geq 40 \\&x_{14} + x_{24} + x_{34} \geq 50 \\&x_{11} + x_{12} + x_{13} + x_{14} \leq 140y_1 \\&x_{21} + x_{22} + x_{23} + x_{24} \leq 150y_2 \\&x_{31} + x_{32} + x_{33} + x_{34} \leq 170y_3 \\&x_{ij} \geq 0 \text{ ve tamsayı } (i=1, 2, 3) (j=1, 2, 3, 4) \\&y_i = 0 \text{ veya } 1 \quad (i=1, 2, 3)\end{aligned}$$

Yöneylem Araştırması II

78

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Küme kaplama problemleri

- Verilen bir kümenin her elemanı başka bir kümenin kabul edilebilir bir elemanı tarafından kaplanıyor ise, bu tarz problemlere **Küme Kaplama Problemi** denir.

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Küme kaplama problemleri (Yer Seçimi Prob.)

- Asayışı sağlamak amacıyla bir bölgede bulunan 6 ilçeden bazılarında jandarma karakolları kurulacaktır. Her ilçeye en fazla 15 dakika mesafede en az bir jandarma karakolu olacak (yani olaylara en fazla 15 dakikada müdahaleyi sağlayacak) şekilde en az sayıda jandarma karakolu kurulması istenmektedir. Bir yerleşim biriminden diğerine araçla gitmek için gerekli ortalama süreler aşağıdaki tabloda verilmiştir. Buna göre kaç jandarma karakolu kurulması gerektiğini ve bu karakolların nerelere kurulacağını belirlemede kullanılacak tamsayılı programlama modelini kurunuz.

	1	2	3	4	5	6
1	0	10	20	30	30	20
2	10	0	25	35	20	10
3	20	25	0	15	30	20
4	30	35	15	0	15	25
5	30	20	30	15	0	14
6	20	10	20	25	14	0

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Küme kaplama problemleri (Yer Seçimi Prob.)

○ Karar değişkenleri:

- Bu problemde hangi ilçelere jandarma karakolu kurulacağına karar verilecektir. Bu durumda karar değişkenleri 0-1 değişken olarak tanımlanmalıdır.

$$x_j \ (j=1,2,3,4,5,6) = \begin{cases} 1, j \text{ ilçesine jandarma karakolu kurulursa} \\ 0, j \text{ ilçesine jandarma karakolu kurulmazsa} \end{cases}$$

	1	2	3	4	5	6
1	0	10	20	30	30	20
2	10	0	25	35	20	10
3	20	25	0	15	30	20
4	30	35	15	0	15	25
5	30	20	30	15	0	14
6	20	10	20	25	14	0

Yöneylem Araştırması II

81

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Küme kaplama problemleri (Yer Seçimi Prob.)

○ Amaç fonksiyonu:

- Bu modelin amacı kurulacak karakol sayısını en aza indirmek olduğuna göre amaç fonksiyonu;

$$\text{Min } z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

	1	2	3	4	5	6
1	0	10	20	30	30	20
2	10	0	25	35	20	10
3	20	25	0	15	30	20
4	30	35	15	0	15	25
5	30	20	30	15	0	14
6	20	10	20	25	14	0

Yöneylem Araştırması II

82

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Küme kaplama problemleri (Yer Seçimi Prob.)

- Tamsayı programlama modeli:
 - Oluşturulan modele 0-1 olma koşulları da eklendiğinde modelin genel yapısı aşağıda verilmiştir.

$$\begin{aligned}
 \text{Min } z &= x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
 x_1 + x_2 &\geq 1 \\
 x_1 + x_2 + x_6 &\geq 1 \\
 x_3 + x_4 &\geq 1 \\
 x_3 + x_4 + x_5 &\geq 1 \\
 x_4 + x_5 + x_6 &\geq 1 \\
 x_2 + x_5 + x_6 &\geq 1 \\
 x_j &= 0 \text{ veya } 1 \quad (j=1, 2, \dots, 6)
 \end{aligned}$$

Yöneylem Araştırması II

83

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Özel kısıtlar (Ya -Ya da)

- İki kısıt aşağıdaki şekilde verilmiş olsun:

$$(1) \quad f(x_1, x_2, \dots, x_n) \leq 0$$

$$(2) \quad g(x_1, x_2, \dots, x_n) \leq 0$$

- (1) ve (2) numaralı kısıtlardan en az bir tanesinin sağlanması “Ya - Ya da” şeklinde kısıtlar olarak adlandırılır. Formülasyona (3) ve (4) şeklinde iki kısıtın ilave edilmesi (1) ve (2) numaralı kısıtların en az birinin sağlanmasını temin eder.

$$(3) \quad f(x_1, x_2, \dots, x_n) \leq My$$

$$(4) \quad g(x_1, x_2, \dots, x_n) \leq M(1-y)$$

Yöneylem Araştırması II

84

Tam Sayılı Programlama Problemlerinin Modellenmesi



■ Özel kısıtlar (Araba üretimi problemi)

- Bir otomobil firması **spor**, **sedan** ve **lüks** olmak üzere 3 tip otomobil üretmeyi düşünmektedir. Herhangi bir tip otomobilin üretiminin ekonomik olarak uygun olması için en az **1000** adet üretilmesi gerekmektedir. Fabrikanın elinde kaynak olarak **6.000** ton çelik ve **60.000** saat işçilik mevcuttur. Her tip otomobil için gerekli kaynak miktarları ve elde edilecek karlar aşağıdaki tabloda verilmiştir. Fabrikanın toplam karını maksimum yapacak tamsayılı programlama modelini kurunuz.

	Araba Tipleri		
	Spor	Sedan	Lüks
Gerekli çelik (ton)	1,5	3	5
Gerekli işçilik (saat)	30	25	40
Birim Kar	2000	3000	4000

Yöneylem Araştırması II

85

Tam Sayılı Programlama Problemlerinin Modellenmesi



■ Özel kısıtlar (Araba üretimi problemi)

- Karar değişkenleri:
 - Firma her tip arabadan ne kadar üretmesi gerektiğine karar vereceğine göre karar değişkenleri:

$$x_j = \text{üretilcek } j\text{nci tip araba sayısı } (j=1, 2, 3)$$

	Araba Tipleri		
	Spor	Sedan	Lüks
Gerekli çelik (ton)	1,5	3	5
Gerekli işçilik (saat)	30	25	40
Birim Kar	2000	3000	4000

Yöneylem Araştırması II

86

Tam Sayılı Programlama Problemlerinin Modellenmesi



Özel kısıtlar (Araba üretimi problemi)

○ Amaç fonksiyonu:

- Firma karını maksimum yapmak istediğinden amaç fonksiyonu :

$$\text{Maks } z = 2000x_1 + 3000x_2 + 4000x_3$$

	Araba Tipleri		
	Spor	Sedan	Lüks
Gerekli çelik (ton)	1,5	3	5
Gerekli işçilik (saat)	30	25	40
Birim Kar	2000	3000	4000

Yöneylem Araştırması II

87

Tam Sayılı Programlama Problemlerinin Modellenmesi



Özel kısıtlar (Araba üretimi problemi)

○ Kısıtlar:

$$1.5x_1 + 3x_2 + 5x_3 \leq 6000 \quad (\text{ÇELİK KISITI})$$

$$30x_1 + 25x_2 + 40x_3 \leq 60000 \quad (\text{İŞÇİLİK KISITI})$$

- Herhangi bir tip otomobilin üretiminin ekonomik olarak uygun olabilmesi için, eğer üretim kararı verilirse, en az 1.000 adet üretilmesi gerektiğine göre. $x_j \geq 1000$ veya $x_j \leq 0$ olmalıdır. bu durum modele, y_j 0-1 değişken ve M pozitif ve çok büyük bir sayı olmak üzere,

$$(1) \quad x_j \leq My_j$$

$$(2) \quad 1000 - x_j \leq M(1 - y_j)$$

	Araba Tipleri		
	Spor	Sedan	Lüks
Gerekli çelik (ton)	1,5	3	5
Gerekli işçilik (saat)	30	25	40
Birim Kar	2000	3000	4000

Yöneylem Araştırması II

88

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Özel kısıtlar (Araba üretimi problemi)

○ Tamsayı programlama modeli:

- Üretilecek araba miktarının kesirli olması bir anlam ifade etmeyeceğinden, modele x_j karar değişkenleri için negatif olmama ve tamsayı olma koşulları da eklendiğinde modelin genel yapısı:

$$\begin{array}{llll}
 \text{Maks } z = & 2000x_1 + & 3000x_2 + & 4000x_3 \\
 \text{S.T.} & 1.5x_1 + & 3x_2 + & 5x_3 \leq 6000 \\
 & 30x_1 + & 25x_2 + & 40x_3 \leq 60000 \\
 & x_1 & & \leq M_1y_1 \\
 & 1000 - x_1 & & \leq M_1(1 - y_1) \\
 & & x_2 & \leq M_2y_2 \\
 & & 1000 - x_2 & \leq M_2(1 - y_2) \\
 & & & x_3 \leq M_3y_3 \\
 & & & 1000 - x_3 \leq M_3(1 - y_3) \\
 & & & x_1, x_2, x_3 \geq 0 \text{ ve tamsayı} \\
 & & & y_1, y_2, y_3 = 0 \text{ veya } 1
 \end{array}$$

Yöneylem Araştırması II

89

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Özel kısıtlar (m adet kısıttan k adetinin etkili olması)

- m adet kısıt aşağıdaki şekilde verilmiş olsun:

$$\begin{array}{l}
 f_1(x_1, x_2, \dots, x_n) \leq d_1 \\
 f_2(x_1, x_2, \dots, x_n) \leq d_2 \\
 \vdots \\
 f_m(x_1, x_2, \dots, x_n) \leq d_m
 \end{array}$$

- Bu m adet kısıttan k adetinin geçerli olmasını sağlamak için ($k < m$ olmak şartı ile); y_i 0-1 değişken ve M çok büyük pozitif sayı olmak üzere modelde aşağıdaki değişiklikleri yapmak gerekir.

$$\begin{array}{l}
 f_1(x_1, x_2, \dots, x_n) \leq d_1 + My_1 \\
 f_2(x_1, x_2, \dots, x_n) \leq d_2 + My_2 \\
 \vdots \\
 f_m(x_1, x_2, \dots, x_n) \leq d_m + My_m \\
 \sum_{i=1}^m y_i = m - k
 \end{array}$$

- Bu yapı “Ya - Ya da” kısıt yapısının genelleştirilmiş halidir. $m=2$ ve $k=1$ alınırsa “Ya - Ya da” kısıt tipi elde edilir.

Yöneylem Araştırması II

90

Tam Sayılı Programlama Problemlerinin Modellenmesi



■ Özel kısıtlar (Atölye Problemi)

- 3 tip parça yapımının söz konusu olduğu bir atölyede 5 tezgah kullanılabilecektir. Tezgahların günlük kullanılabilir zamanları ve parçaların birim üretimleri için gerekli tezgah zamanları (birim/dakika) aşağıdaki tabloda verilmiştir.

Üretilcek parça	Tezgah				
	1	2	3	4	5
1	6	5	6	5	8
2	8	8	5	6	4
3	5	7	5	8	4
Günlük kapasite (saat)	12	14	15	12	13

Yöneylem Araştırması II

91

Tam Sayılı Programlama Problemlerinin Modellenmesi



■ Özel kısıtlar (Atölye Problemi)

- M pozitif ve çok büyük bir sayı. X_j =j-nci parçadan üretilcek miktar ve Y_i 0-1 değişken olmak üzere, tezgahların kullanılmasıyla ilgili tüm kısıtlar aşağıda verilmiştir.

$$\begin{aligned}
 6x_1 + 8x_2 + 5x_3 &\leq 12*60 + My_1 \\
 5x_1 + 8x_2 + 7x_3 &\leq 14*60 + My_2 \\
 6x_1 + 5x_2 + 5x_3 &\leq 15*60 + My_3 \\
 5x_1 + 6x_2 + 8x_3 &\leq 12*60 + My_4 \\
 8x_1 + 4x_2 + 4x_3 &\leq 13*60 + My_5
 \end{aligned}$$

Üretilcek parça	Tezgah				
	1	2	3	4	5
1	6	5	6	5	8
2	8	8	5	6	4
3	5	7	5	8	4
Günlük kapasite (saat)	12	14	15	12	13

Yöneylem Araştırması II

92

Tam Sayılı Programlama Problemlerinin Modellenmesi



■ Özel kısıtlar (Atölye Problemi)

- Eğer yedek parçalar 1 tezgahta üretilmek istenirse;

$$\sum_{i=1}^m y_i = m - k$$

- Formülünü kullanarak:

$$y_1 + y_2 + y_3 + y_4 + y_5 = 4$$

- kısıtı modele ilave edilmelidir. Bu kısıt ile 4 adet y değişkeni "1" değerini, 1 adet y değişkeni "0" değerini alır. $y_i=1$ değerini alan kısıtların sağ taraf değeri oldukça büyük bir değere dönüşür ve bu kısıtlar bağlayıcı olma özelliklerini yitirir. Sonuç olarak sadece $y_i=0$ değerini alan kısıtın bağlayıcı olma özelliği devam eder.

$$\begin{aligned} 6x_1 + 8x_2 + 5x_3 &\leq 12*60 + My_1 \\ 5x_1 + 8x_2 + 7x_3 &\leq 14*60 + My_2 \\ 6x_1 + 5x_2 + 5x_3 &\leq 15*60 + My_3 \\ 5x_1 + 6x_2 + 8x_3 &\leq 12*60 + My_4 \\ 8x_1 + 4x_2 + 4x_3 &\leq 13*60 + My_5 \end{aligned}$$

Yöneylem Araştırması II

93

Tam Sayılı Programlama Problemlerinin Modellenmesi



■ Özel kısıtlar (Atölye Problemi)

- Eğer yedek parçalar en çok 3 tezgahta üretilmek istenirse, bu durumda modele aşağıdaki kısıt eklenmelidir.

$$y_1 + y_2 + y_3 + y_4 + y_5 \geq 2$$

- Eğer yedek parçalar 1 veya 2nci tezgahlarda üretilmek istenirse ("ya-ya da" tipi kısıt) modele sadece aşağıdaki iki kısıt eklenmelidir.

$$\begin{aligned} 6x_1 + 8x_2 + 5x_3 &\leq 12*60 + My \\ 5x_1 + 8x_2 + 7x_3 &\leq 14*60 + M(1-y) \end{aligned}$$

$$\begin{aligned} 6x_1 + 8x_2 + 5x_3 &\leq 12*60 + My_1 \\ 5x_1 + 8x_2 + 7x_3 &\leq 14*60 + My_2 \\ 6x_1 + 5x_2 + 5x_3 &\leq 15*60 + My_3 \\ 5x_1 + 6x_2 + 8x_3 &\leq 12*60 + My_4 \\ 8x_1 + 4x_2 + 4x_3 &\leq 13*60 + My_5 \end{aligned}$$

Yöneylem Araştırması II

94

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Özel kısıtlar (Kısıtın n Adet Farklı Değer Alması)

- Aşağıdaki gösterim, bir kısıtın sağ taraf değerinin alabileceği n farklı değeri ifade etmektedir.

$$f(x_1, x_2, \dots, x_n) = b_1, \text{ veya } b_2, \dots, \text{ veya } b_N$$

- Sağ taraf değerlerinden sadece bir tanesinin kullanılması durumu, modele eklenecek aşağıdaki kısıtlar ile sağlanabilir.

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \sum_{i=1}^N b_i y_i \\ \sum_{i=1}^N y_i &= 1 \\ y_i &= 0 \text{ veya } 1 \quad (i=1, 2, \dots, N) \end{aligned}$$

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ Özel kısıtlar (Kısıtın n Adet Farklı Değer Alması)

- Örneğin, yatırım alternatiflerinin değerlendirilmesi probleminde, firmanın yatırımlar için ayırabileceği kaynak miktarının 14, 18 ya da 23 olacağını ve kaynak miktarındaki bu farklılaşmanın firmanın yürüttüğü diğer yatırımlardan gelebilecek kaynakları dikkate alması sonucu olduğunu varsayalım. Bu durumda sağ taraf değerlerinin sadece 1 tanesinin kullanılması için modele aşağıdaki kısıtların ilave edilmesi gerekir.

$$\begin{aligned} 5x_1 + 7x_2 + 4x_3 + 3x_4 &\leq 14y_1 + 18y_2 + 23y_3 \\ y_1 + y_2 + y_3 &= 1 \\ y_1, y_2, y_3 &= 0 \text{ veya } 1 \end{aligned}$$

[Tam Sayılı Programlama Problemlerinin Modellenmesi]



■ ÖZET

Durum

m adet kısıttan k adedinin geçerli olması (her kısıtın sağ taraf değerine My_i ilave edildikten sonra);

n adet değişkenden k adedinin geçerli olması;

n adet değişkenden en fazla k adedinin geçerli olması;

n adet değişkenden en fazla k adedinden vazgeçilecek olması;

i -nci değişkenin seçilmesinin j -nci değişkene bağlı olması;

j -nci ve i -nci değişkenlerin birlikte seçilmesi veya reddi;

Kısıt

$$\sum_{i=1}^m y_i = m - k$$

$$\sum_{j=1}^n x_j = k$$

$$\sum_{j=1}^n x_j \leq k$$

$$\sum_{j=1}^n x_j \geq n - k$$

$$x_i \geq x_j$$

$$x_j = x_i$$

[Tamsayıli Programlamada Çözüm Yaklaşımları]



- Tamsayıli programlama modelini çözmek için, doğrusal programlama modelinin çözümünde olduğu gibi genel bir teknik yoktur. Belirli bir tür problemde hareketle geliştirilen bir dizi çözüm yaklaşımı mevcuttur. Bu nedenle, bunların birinin diğerine üstünlüğü problemin yapısına göre değişmektedir. Geliştirilen yaklaşımlar aşağıda verilen ana başlıklar altında toplanabilir.

- Grafik metot
- Yuvarlatma
- Sayımlama
- Kesme düzlemi algoritmaları
- Sezgisel algoritmalar
- Hibrid algoritmalar

- Tamsayıli programlama problemlerinin çözümü için en yaygın olarak kullanılan metotlardan biri **Dal ve Sınır Algoritması**dır.

[Dal-Sınır Algoritması]



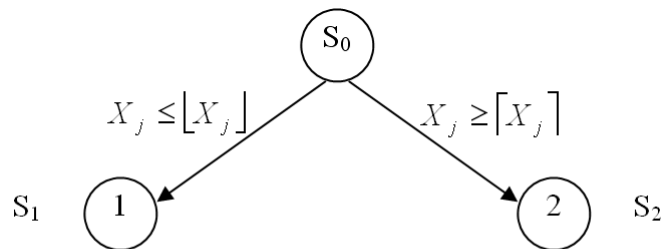
- **Tanım:** Bütün olası uygun çözümleri analiz eden çok etkili bir sayımlama yöntemidir.
- Orijinal problem alt problemlere ayrılır ve alt problemler teker teker çözülür. Bir alt problem için çözüm bulunduğu haldе karara varılamıyorsa (yani düğüm budanamıyorsa) bu alt problem tekrardan yeni iki alt probleme ayrılır. Bu yöntemle tüm alternatifler incelenerek optimal çözüme ulaşılır.
- Dal sınır tekniğinin **4 kuralı** vardır:
 1. Dallanma kuralı
 2. Düğüm kuralları (sınır kuralı)
 3. Budama kuralları
 4. Bitiş kuralları

[Dal-Sınır Algoritması]



1 Dallanma Kuralları

- S_0 , gevşetilmiş modelin optimal çözümü olsun.



S_1 ve S_2 alt kümelerimiz

[Dal-Sınır Algoritması]



1 Dallanma Kuralları

$$\begin{array}{lcl}
 \min / \max Z = & Cx & \\
 S_1 & Ax \leq b & \\
 \text{Alt Problemi :} & X_j \geq 0 & \\
 & X_j \leq \lfloor X_j \rfloor &
 \end{array} \left. \vphantom{\begin{array}{l} \min / \max Z = \\ S_1 \\ \text{Alt Problemi :} \end{array}} \right\} S_0$$

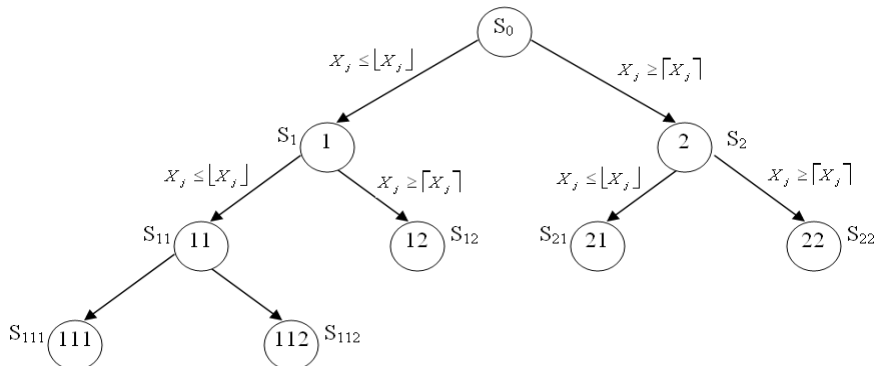
$$\begin{array}{lcl}
 S_2 & S_0 & \\
 \text{Alt Problemi :} & X_j \geq \lceil X_j \rceil &
 \end{array}$$

[Dal-Sınır Algoritması]



1 Dallanma Kuralları

- S_1 alt problemi çözülür. Bu çözümde elde edilen düğüm budanamıyorsa, S_1 kendi içerisinde dallandırılır.



[Dal-Sınır Algoritması]



2 Sınır Kuralı (Düğüm Kuralı)

- Z_m : Maksimizasyon problemi için alt sınır, minimizasyon problemi için üst sınır değeri
- Z_i : i düğümünün amaç fonksiyonu değeri

[Dal-Sınır Algoritması]



3 Budama Kuralı

- Çözümün tamsayı olması halinde,
- Düğümün amaç fonksiyonu değeri yani Z_i , Z_m 'den daha iyi olmaması halinde, yani:
 - Max $\rightarrow Z_i \leq Z_m$ durumunda i düğümü budanır.
 - Min $\rightarrow Z_i \geq Z_m$ durumunda
- Uygun çözüm olmaması halinde

4 Bitiş Kuralı

- Dalların hepsi budandığında işlem durur.

[Dal-Sınır Algoritması]



■ Arama Stratejileri

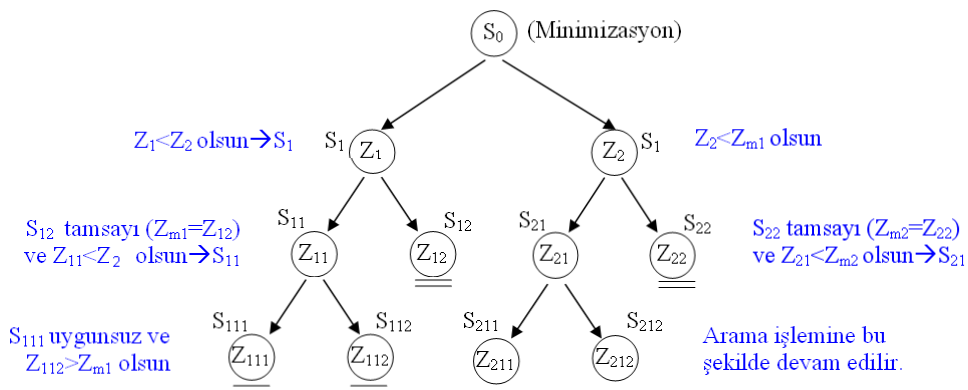
- Uygun çözümün aranmasında 2 strateji vardır.
 1. Öncelikle genişlik (en iyi çözüm)
 2. Öncelikle derinlik (en yeni çözüm)

[Dal-Sınır Algoritması]



1 Öncelikle Genişlik (En İyi Çözüm)

- S_0 dan sonra oluşturulmuş alt problemler içerisinde en iyi çözüme sahip olanından arama işlemine devam edilir.

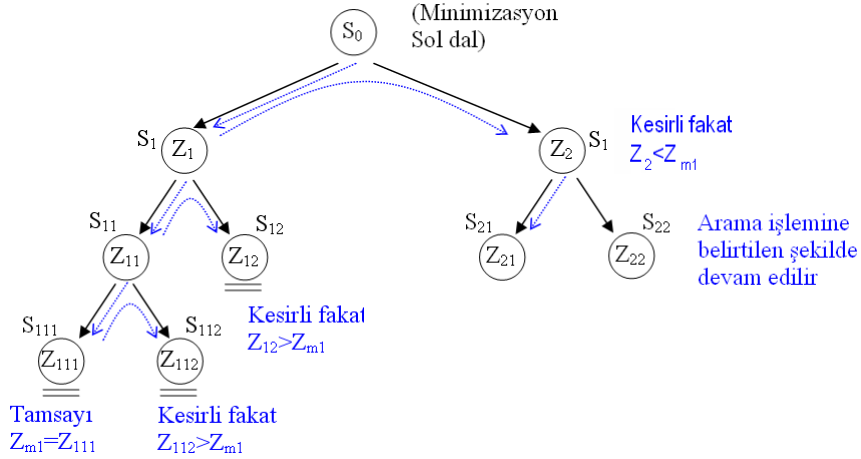


[Dal-Sınır Algoritması]



2 Öncelikle Derinlik (En Yeni Çözüm)

- Keyfi olarak sağ veya sol daldan birisi seçilir, arama işlemine bu daldan devam edilir.



Yöneylem Araştırması II

107

[Dal-Sınır Algoritması]



■ Stratejilerin Karşılaştırılması

- Öncelikle Genişlik stratejisi **daha basit ve işlemleri daha azdır**. Ancak bilgisayar hafızasında **çok yer tutar**.
- Öncelikle Derinlik stratejisinde işlemler daha çoktur, fakat bilgisayar hafızasında **daha az yer tutar**.

Belirtilen nedenlerle ticari yazılım paketlerinde Öncelikle Derinlik stratejisi kullanılır.

Yöneylem Araştırması II

108

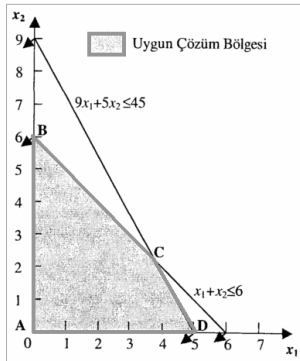
[Dal-Sınır Algoritması (Örnek)]



$$\begin{aligned}\text{Maks } z &= 8x_1 + 5x_2 \\ x_1 + x_2 &\leq 6 \\ 9x_1 + 5x_2 &\leq 45 \\ x_1, x_2 &\geq 0 \text{ ve tamsayı}\end{aligned}$$

- Modelde sadece iki karar değişkeni olduğu için grafik çözüm kullanılabilir.
- Dal ve sınır metodu ile çözümün ilk adımı gevşetilmiş tamsayılı programlama problemini (S_0) çözmektir.

[Dal-Sınır Algoritması (Örnek)]



	x_1	x_2	z
A	0	0	0
B	0	6	30
C	15/4	9/4	165/4
D	5	0	40

GEVŞETİLMİŞ MODEL İÇİN
OPTİMAL ÇÖZÜM:

$$\begin{aligned}x_1^* &= 15/4 \\ x_2^* &= 9/4 \\ z_{(S_0)}^* &= 165/4\end{aligned}$$

- Tamsayı olma koşulu göz ardı edilerek bulunan gevşetilmiş tamsayılı programlama probleminin optimal amaç fonksiyonu değeri, tümü tamsayılı programlama probleminin amaç fonksiyonu için **üst sınır** oluşturur.

$$-\infty \leq z_{(TP)} \leq 165/4$$

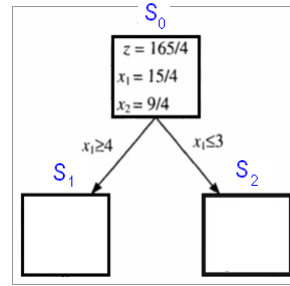
[Dal-Sınır Algoritması (Örnek)]



- Sonraki adımlarda ise modele ilave kısıtlar ekleyerek karar değişkenlerinin tamsayı değer almaları sağlanır.

$$\begin{array}{l}
 S_1 \quad \text{Maks } z = 8x_1 + 5x_2 \\
 x_1 + x_2 \leq 6 \\
 9x_1 + 5x_2 \leq 45 \\
 x_1 \geq 4 \\
 x_1, x_2 \geq 0
 \end{array}$$

$$\begin{array}{l}
 S_2 \quad \text{Maks } z = 8x_1 + 5x_2 \\
 x_1 + x_2 \leq 6 \\
 9x_1 + 5x_2 \leq 45 \\
 x_1 \leq 3 \\
 x_1, x_2 \geq 0
 \end{array}$$



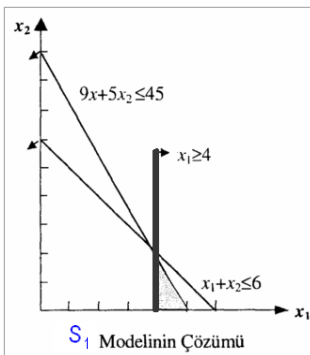
Yöneylem Araştırması II

111

[Dal-Sınır Algoritması (Örnek)]



- S_1 Çözümü



- $x_1=4, x_2=9/5, Z(S_1)=41$
- $x_2 \geq 2$ ve $x_2 \leq 1$ kısıtları eklenerek S_{11} ve S_{12} olmak üzere iki dala ayrılır.

$$\begin{array}{l}
 S_{11} \quad \text{Maks } z = 8x_1 + 5x_2 \\
 x_1 + x_2 \leq 6 \\
 9x_1 + 5x_2 \leq 45 \\
 x_1 \geq 4 \\
 x_2 \geq 2 \\
 x_1, x_2 \geq 0
 \end{array}$$

$$\begin{array}{l}
 S_{12} \quad \text{Maks } z = 8x_1 + 5x_2 \\
 x_1 + x_2 \leq 6 \\
 9x_1 + 5x_2 \leq 45 \\
 x_1 \geq 4 \\
 x_2 \leq 1 \\
 x_1, x_2 \geq 0
 \end{array}$$

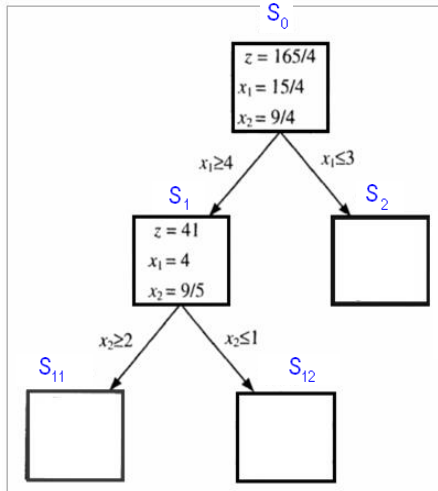
Yöneylem Araştırması II

112

[Dal-Sınır Algoritması (Örnek)]



■ S_1 Çözümü



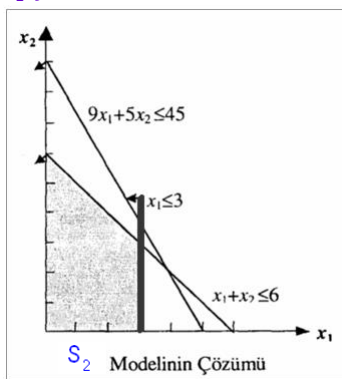
Yöneylem Araştırması II

113

[Dal-Sınır Algoritması (Örnek)]

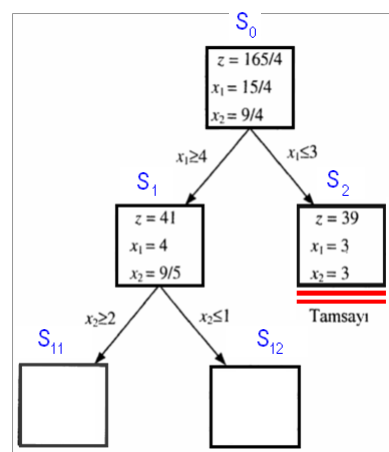


■ S_2 Çözümü



- $x_1=3, x_2=3, Z(S_2)=39$
- Bütün karar değişkenleri **tamsayı değer** aldığından **budandır**.
- $Z(S_2)=39 \geq -\infty = \text{Alt sınır}$ olduğundan alt sınır değeri güncellenir.

$$39 \leq z_{(TP)} \leq 165/4$$



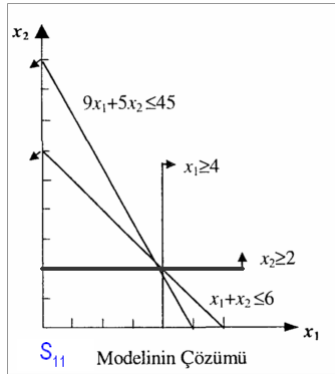
Yöneylem Araştırması II

114

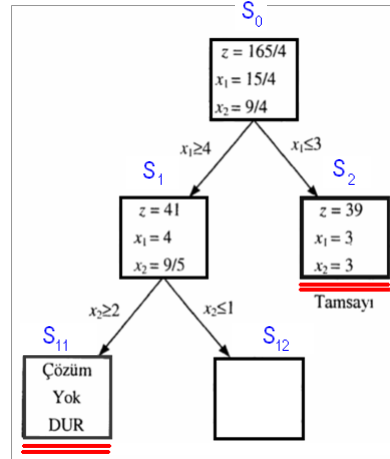
[Dal-Sınır Algoritması (Örnek)]



■ S_{11} Çözümü



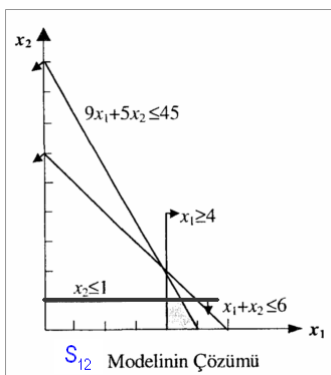
■ Çözüm Yok, budanır.



[Dal-Sınır Algoritması (Örnek)]



■ S_{12} Çözümü



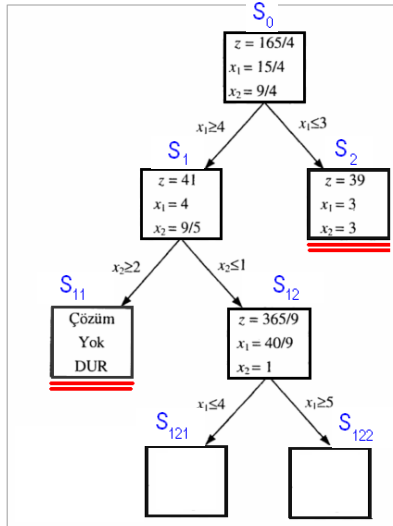
- $x_1=40/9, x_2=1, Z(S_{12})=365/9=40.55$
- $x_1 \leq 4$ ve $x_1 \geq 5$ kısıtları eklenerek S_{121} ve S_{122} olmak üzere iki dala ayrılır.

$$\begin{array}{l}
 S_{121} \quad \text{Maks } z = 8x_1 + 5x_2 \\
 \quad x_1 + x_2 \leq 6 \\
 \quad 9x_1 + 5x_2 \leq 45 \\
 \quad x_1 \geq 4 \\
 \quad x_2 \leq 1 \\
 \quad x_1 \leq 4 \\
 \quad x_1, x_2 \geq 0 \\
 \\
 S_{122} \quad \text{Maks } z = 8x_1 + 5x_2 \\
 \quad x_1 + x_2 \leq 6 \\
 \quad 9x_1 + 5x_2 \leq 45 \\
 \quad x_1 \geq 4 \\
 \quad x_2 \leq 1 \\
 \quad x_1 \geq 5 \\
 \quad x_1, x_2 \geq 0
 \end{array}$$

[Dal-Sınır Algoritması (Örnek)]



■ S_{12} Çözümü



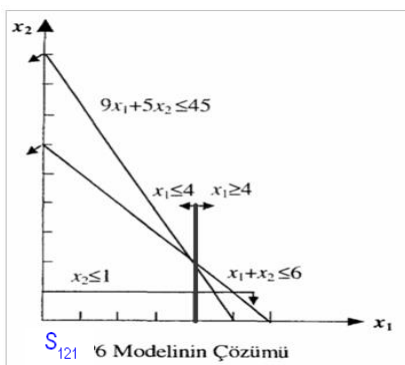
Yöneylem Araştırması II

117

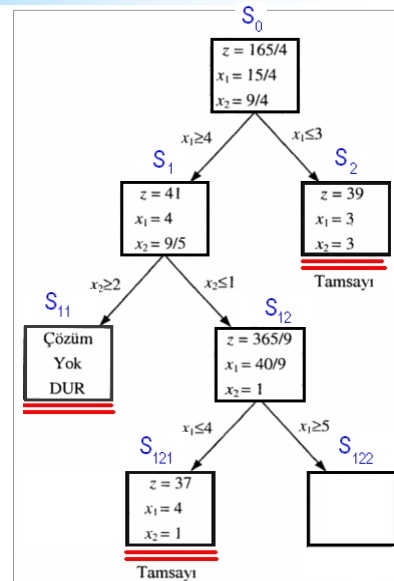
[Dal-Sınır Algoritması (Örnek)]



■ S_{121} Çözümü



- $x_1=4, x_2=1, Z(S_{121})=37$
- Bütün karar değişkenleri tamsayı değer aldığından **budandır**.
 $Z(S_{121})=37 \leq 39 = \text{Alt sınır}$ olduğundan **alt sınır değeri değişmez**.



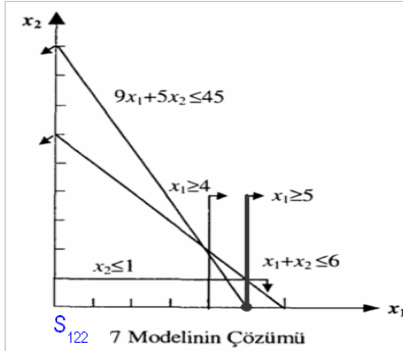
Yöneylem Araştırması II

118

[Dal-Sınır Algoritması (Örnek)]

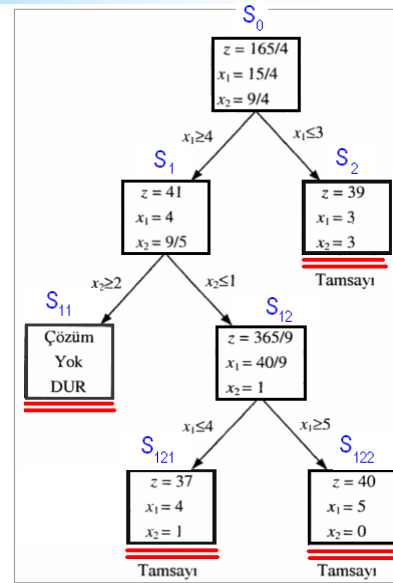


■ S_{122} Çözümü



- $x_1=5, x_2=0, Z(S_{122})=40$
- Bütün karar değişkenleri tamsayı değer aldığından bu dal budanır.
 $Z(S_{122})=40 \geq 39 = \text{Alt sınır}$
olduğundan alt sınır değeri güncellenir.

$$40 \leq z_{(TP)} \leq 165/4$$



Yöneylem Araştırması II

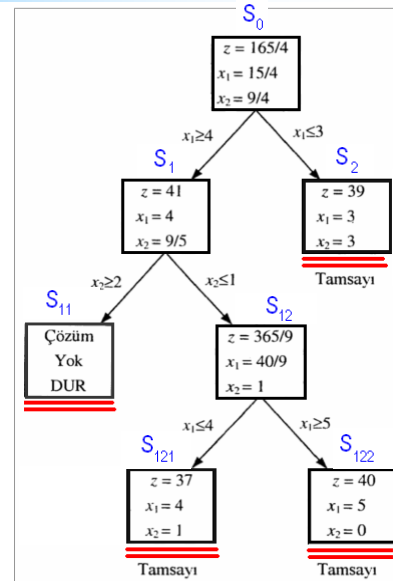
119

[Dal-Sınır Algoritması (Örnek)]



- Tüm dallarda işlemler tamamlandığından (tüm düğümler budandığından), alt sınır değerini oluşturan S_{122} modelinin optimal çözümü aynı zamanda tamsayılı programlama modelinin de optimal çözümüdür.

$$\begin{aligned} x_1 &= 5, \\ x_2 &= 0, \\ Z(TP) &= 40 \end{aligned}$$



Yöneylem Araştırması II

120

Dal-Sınır Algoritması (Örnek 2)



$$\begin{aligned}
 \max Z &= 2x_1 + 8x_2 \\
 \text{s.t. } 2x_1 - 6x_2 &\leq 3 && \leftarrow S_1 \\
 -x_1 + 4x_2 &\leq 5 && \leftarrow S_2 \\
 2x_1 + 2x_2 &\leq 13 && \leftarrow S_3 \\
 x_1, x_2 &\geq 0 \text{ ve tamsayı}
 \end{aligned}$$

S_0 Alt problemi (Gevşetilmiş model)

T.D.	x_1	x_2	S_1	S_2	S_3	b_i
Z	-2	-8	0	0	0	0
S_1	2	-6	1	0	0	3
S_2	-1	4	0	1	0	5
S_3	2	2	0	0	1	13
Z	-4	0	0	2	0	10
S_1	1/2	0	1	3/2	0	21/2
x_2	-1/4	1	0	1/4	0	5/4
S_3	5/2	0	0	-1/2	1	21/2
Z	0	0	0	6/5	8/5	134/5
S_1	0	0	1	8/5	-1/5	42/5
x_2	0	1	0	1/5	1/10	23/10
x_1	1	0	0	-1/5	2/5	21/5

-

$5/4 = 1,25 < \min$

$13/2 = 6,5$

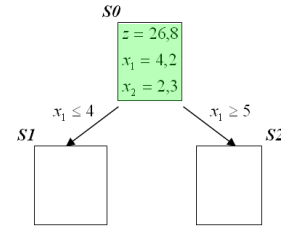
$(21/2)/(1/2) = 21$

-

$(21/2)/(5/2) = 4,2 < \min$

OPTIMAL

S_0 alt problemi



121

Dal-Sınır Algoritması (Örnek 2)

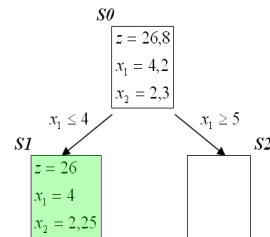


S_1 Alt Problemi: S_0 $x_1 \leq 4$

$x_1 + S_4 = 4$ (S_0 tablosundan $x_1 = 21/5 + 1/5S_2 - 2/5S_3$ yerine yazılırsa)

$1/5S_2 - 2/5S_3 + S_4 = -1/5$ (Eklenecek Satır)

T.D.	x_1	x_2	S_1	S_2	S_3	S_4	b_i
Z	0	0	0	6/5	8/5	0	134/5
S_1	0	0	1	8/5	-1/5	0	42/5
x_2	0	1	0	1/5	1/10	0	23/10
x_1	1	0	0	-1/5	2/5	0	21/5
S_4	0	0	0	1/5	-2/5	1	-1/5
Z	0	0	0	2	0	4	26
S_1	0	0	1	3/2	0	-1/2	17/2
x_2	0	1	0	1/4	0	1/4	9/4
x_1	1	0	0	0	0	1	4
S_3	0	0	0	-1/2	1	-5/2	1/2



Yöneylem Araştırması II

122

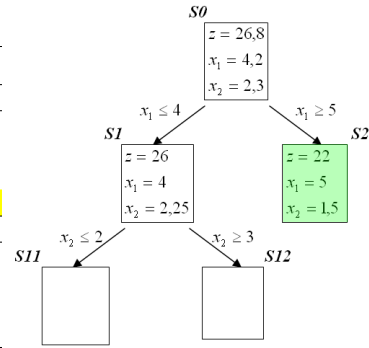
Dal-Sınır Algoritması (Örnek 2)



S_2 S_0
Alt Problemi: $x_1 \geq 5$

$-x_1 + S_4 = -5$ (S_0 tablosundan $x_1 = 21/5 + 1/5 S_2 - 2/5 S_3$ yerine yazılırsa)
 $-1/5 S_2 + 2/5 S_3 + S_4 = -4/5$ (Eklenen Satır)

T.D.	x_1	x_2	S_1	S_2	S_3	S_4	b_i
Z	0	0	0	6/5	8/5	0	134/5
S_1	0	0	1	8/5	-1/5	0	42/5
x_2	0	1	0	1/5	1/10	0	23/10
x_1	1	0	0	-1/5	2/5	0	21/5
S_4	0	0	0	-1/5	2/5	1	-4/5
Z	0	0	0	0	4	6	22
S_1	0	0	1	0	3	8	2
x_2	0	1	0	0	1/2	1	15/10
x_1	1	0	0	0	0	-1	5
S_2	0	0	0	1	-2	-5	4



Yöneylem Araştırması II

123

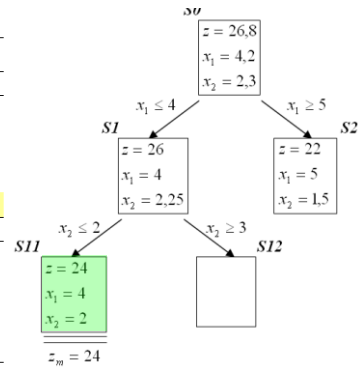
Dal-Sınır Algoritması (Örnek 2)



S_{11} S_1
Alt Problemi: $x_2 \leq 2$

$x_2 + S_5 = 2$ (S_1 tablosundan $x_2 = 9/4 - 1/4 S_2 - 1/4 S_4$ yerine yazılırsa)
 $-1/4 S_2 - 1/4 S_4 + S_5 = -1/4$ (Eklenen Satır)

T.D.	x_1	x_2	S_1	S_2	S_3	S_4	S_5	b_i
Z	0	0	0	2	0	4	0	26
S_1	0	0	1	3/2	0	-1/2	0	17/2
x_2	0	1	0	1/4	0	1/4	0	9/4
x_1	1	0	0	0	0	1	0	4
S_3	0	0	0	-1/2	1	-5/2	0	1/2
S_5	0	0	0	-1/4	0	-1/4	1	-1/4
Z	0	0	0	0	0	2	8	24
S_1	0	0	1	0	0	-2	6	7
x_2	0	1	0	0	0	0	1	2
x_1	1	0	0	0	0	1	0	4
S_3	0	0	0	0	1	-2	-2	1
S_2	0	0	0	1	0	1	-4	1



Yöneylem Araştırması II

124

[Dal-Sınır Algoritması (Örnek 2)]

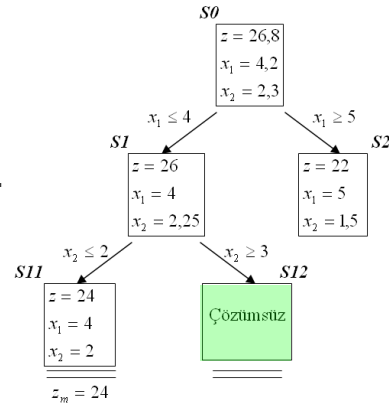


S_{12} S_1
Alt Problemi: $x_2 \geq 3$

$-x_2 + S_5 = -3$ (S_1 tablosundan $x_2 = 9/4 - 1/4S_2 - 1/4S_4$ yerine yazılırsa)

$1/4S_2 - 1/4S_4 + S_5 = -3/4$ (Eklenecek Satır)

- Yeni kısıtın sol tarafında negatif katsayı yok.
Dolayısıyla ÇÖZÜMSÜZ.



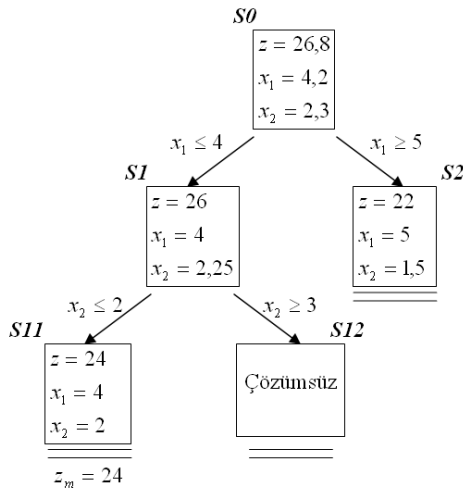
Yöneylem Araştırması II

125

[Dal-Sınır Algoritması (Örnek 2)]



$z_m = 24 > z_2 = 22$ olduğundan S_2 düğümü budanır.



Optimal ÇÖZÜM

$z = 24$
 $x_1 = 4$
 $x_2 = 2$

Yöneylem Araştırması II

126

[Dal-Sınır Algoritması]



■ Özet

- **Adım 1:** Verilen model gevşetilerek çözülür. Bu çözüm sonucunda üç durum ile karşılaşılabilir.
 - **Sınırsız veya geçersiz çözüm.** Bu durumda **durulur**. Çünkü, gevşetilmiş tamsayılı programlama probleminin çözümü yoksa, tamsayılı programlama probleminin de çözümü yoktur.
 - **Bulunan çözümde değişkenlerin tamsayı değer alması.** Bu durumda da **durulur**. Çünkü, gevşetilmiş tamsayılı programlama probleminde tamsayılı bir çözüm elde edilmiş ise, bu çözüm tamsayılı programlama probleminin de çözümüdür. Yani ilk adımda optimal çözüm elde edilmiş olur.
 - **Bulunan çözümde tamsayı değer alması istenen değişkenlerden en az bir tanesinin tamsayı olmaması.** Bu durumda amaç fonksiyonunun alt ve üst sınırları belirlenerek, **Adım 2** 'e geçilir.

$$Z_{(üst)} = Z_{(DP)}, \quad Z_{(alt)} = -\infty$$

[Dal-Sınır Algoritması]



■ Özet

- **Adım 2:** Yapı, tamsayı değer almamış bir değişkene göre dallara ayrılarak alt modeller (düşümler) elde edilir. Alt model çözümsüzse veya bulunan optimal çözümde amaç fonksiyonu değeri hedef değerden kötüyse bu dal **budanır**. Aksi durumda;
 - Değişkenler tamsayı değer almışsa **Adım 3** 'e,
 - Değişkenler tamsayı değer almamışsa **Adım 4** 'e geçilir.

[Dal-Sınır Algoritması]



■ Özet

- **Adım 3:** Alt sınır güncellenerek düğüm budanır. Yeni alt sınır üst sınıra eşit ise **Adım 5** 'e gidilir, değilse **Adım 4** 'e geçilir.
- **Adım 4:** İşlem dışı olmamış düğüm varsa **Adım 2** 'ye dönülür, yoksa **Adım 5** 'e geçilir.
- **Adım 5:** Algoritma sona erer. Son hedef değere karşı gelen çözüm **optimal çözümdür**.

[Gomory Kesme Düzlemi Metodu]



- Kesme düzlemi algoritmaları **ardışık iterasyonlarla** uygun çözüm bölgelerini **daraltarak** optimal tamsayılı çözümü bulan algoritmalarlardır.
- Kesme düzlemi uygun çözüm bölgesinin **kesirli kısmını kesip atar**. Yani, tamsayılı optimal çözüm uygun çözüm bölgesi içerisinde kalır. Kesme yapılan satıra **Kaynak Satır** denir.
- **Yöntem:** Problem **önce** gevşetilmiş tamsayılı programlama modeli olarak bilinen tekniklerle çözülür. Tamsayılı çözüm elde edilirse çözüm, hem gevşetilmiş tamsayılı programlama modelinin çözümü hem de asıl tamsayılı problemin çözümü olur. Çünkü, orijinal problemin **tek farkı** tamsayılı olmasıdır.

[Gomory Kesme Düzlemi Metodu]



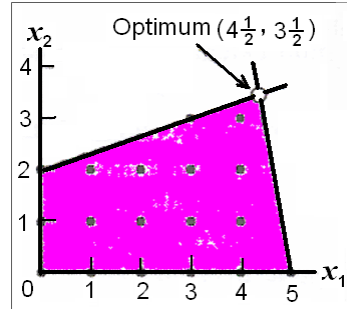
■ Gevşetilmiş modelin optimum çözümü

$$\begin{aligned} \text{maks. } z &= 7x_1 + 10x_2 \\ -x_1 + 3x_2 &\leq 6 \\ 7x_1 + x_2 &\leq 35 \\ x_1, x_2 &\geq 0 \text{ ve tamsayı} \end{aligned}$$

Optimal Çözüm

$$(x_1, x_2) = (4\frac{1}{2}, 3\frac{1}{2}),$$

$$z = 66\frac{1}{2}$$



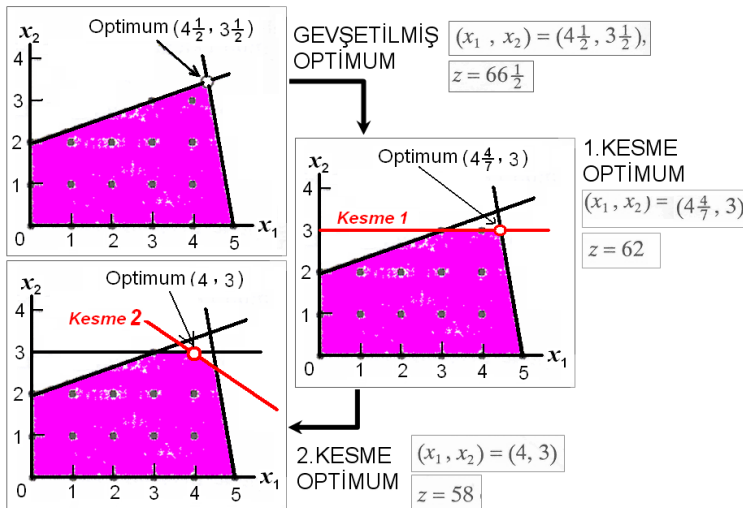
Yöneylem Araştırması II

131

[Gomory Kesme Düzlemi Metodu]



■ Kesme Düzlemlerinin Eklenmesi



Yöneylem Araştırması II

132

[Gomory Kesme Düzlemi Metodu]



$$\left. \begin{array}{l} \max / \min Z \quad \sum cx \\ \text{s.t.} \quad Ax \leq b \\ x \geq 0 \text{ ve} \end{array} \right\} \begin{array}{l} \text{Gevşetilmiş T.D.P.} \\ \text{(Tamsayılı problem)} \end{array}$$

tamsayı

α_{ij} : i . kısıtın j . değişkeni katsayısı

(örneğin $3x_1$ için $\alpha_{ij} = 3$)

$$\alpha_{ij} = \lfloor \alpha_{ij} \rfloor + F_{ij}$$

$$\beta_i = \lfloor \beta_i \rfloor + F_i$$

$$0 \leq F_{ij} < 1$$

$$0 \leq F_i < 1$$

Örnek:

Katsayı		$\lfloor \alpha_{ij} \rfloor$	F_{ij}
$\frac{7}{3}$	$= 2 + 0.33$	$= 2$	$+\frac{1}{3}$
$-\frac{7}{3}$	$= -3 + 0.67$	$= -3$	$+\frac{2}{3}$
$-\frac{11}{2}$	$= -6 + 0.50$	$= -6$	$+\frac{1}{2}$

Yöneylem Araştırması II

133

[Gomory Kesme Düzlemi Metodu]



- Kesirli kesmede yapılan işlem:

$$\sum_{j=1}^n F_{ij} S_j \geq F_i$$

α_{ij} 'nin kesirli kısmı
 b_i 'nin kesirli kısmı

$$-\bar{S}_i + \sum_{j=1}^n F_{ij} S_j = F_i$$

← Kesme denklemi

(ilave edilecek satır)

- İşlemlere **Dual Simpleks** ile devam edilmesi gerektiğinden sağ tarafın **negatif** olması gerekir. Bu amaçla eşitliğin her iki tarafını **(-1)** ile çarparsak:

$$\bar{S}_i - \sum_{j=1}^n F_{ij} S_j = -F_i$$

Yöneylem Araştırması II

134

[Gomory Kesme Düzlemi Metodu]



$$\overline{S}_i - \sum_{j=1}^n F_{ij} S_j = -F_i$$

	x_1	x_2	$\dots$	x_m	S_1	S_2	$\dots$	S_n	$\overline{S}_i$	b_i
z	0	0	$\dots$	0	C_1	C_2	$\dots$	C_n	0	β_0
x_1	1	0	$\dots$	0						β_1
x_2	0	1	$\dots$	0						β_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$					$\vdots$	$\vdots$
x_m	0	0	$\dots$	1						β_m
$\overline{S}_i$	0	0	$\dots$	0	$-F_{i1}$	$-F_{i2}$	$\dots$	$-F_{in}$	1	$-F_i$

Yöneylem Araştırması II

135

[Gomory Kesme Düzlemi Metodu]



■ Örnek 1

$$\begin{aligned}
 \max Z &= 2x_1 + 8x_2 \\
 \text{s.t. } 2x_1 - 6x_2 &\leq 3 \\
 -x_1 + 4x_2 &\leq 5 \\
 2x_1 + 2x_2 &\leq 13 \\
 x_1, x_2 &\geq 0 \text{ ve tamsayı}
 \end{aligned}$$

Gevşetilmiş modelin Primal Simpleks yöntemiyle bulunan optimal çözüm tablosu

	x_1	x_2	S_1	S_2	S_3	b_i	
z	0	0	0	6/5	8/5	134/5	$\leftarrow 26 \quad 4/5 \leftarrow 0.8$
S_1	0	0	1	8/5	-1/5	42/5	$\leftarrow 8 \quad 2/5 \leftarrow 0.4$
x_2	0	1	0	1/5	1/10	23/10	$\leftarrow 2 \quad 3/10 \leftarrow 0.3$
x_1	1	0	0	-1/5	2/5	21/5	$\leftarrow 21/5 \quad \leftarrow 0.07$

Kesirlik kısmı büyük olan kaynak satır seçilir. Öyleyse kaynak satır z satırıdır:

$$z + \frac{6}{5}S_2 + \frac{8}{5}S_3 = \frac{134}{5}$$

Yöneylem

136

[Gomory Kesme Düzlemi Metodu]



Kaynak satır: z

$$z + \frac{6}{5}S_2 + \frac{8}{5}S_3 = \frac{134}{5}$$

$$z + (1 + \frac{1}{5})S_2 + (1 + \frac{3}{5})S_3 = (26 + \frac{4}{5})$$

$$\boxed{\frac{1}{5}S_2 + \frac{3}{5}S_3 \geq \frac{4}{5}} \leftarrow \text{I Kesme}$$

Her iki taraf (-1) ile çarpılırsa;

$$-\frac{1}{5}S_2 - \frac{3}{5}S_3 \leq -\frac{4}{5}$$

$$\boxed{-\frac{1}{5}S_2 - \frac{3}{5}S_3 + \overline{S_4} = -\frac{4}{5}} \leftarrow \text{Eklenecek kısıt}$$

Yöneylem Araştırması II

137

[Gomory Kesme Düzlemi Metodu]



$$\boxed{-\frac{1}{5}S_2 - \frac{3}{5}S_3 + \overline{S_4} = -\frac{4}{5}} \leftarrow \text{Eklenecek kısıt}$$

	x_1	x_2	S_1	S_2	S_3	$\overline{S_1}$	b_i	
z	0	0	0	6/5	8/5	0	134/5	
S_1	0	0	1	8/5	-1/5	0	42/5	
x_2	0	1	0	1/5	1/10	0	23/10	
x_1	1	0	0	-1/5	2/5	0	21/5	
$\overline{S_1}$	0	0	0	-1/5	-3/5	1	-4/5	
z	0	0	0	2/3	0	8/3	74/3	$\leftarrow 24 \quad 2/3$
S_1	0	0	1	5/3	0	-1/3	26/3	$\leftarrow 8 \quad 2/3$
x_2	0	1	0	1/6	0	1/6	13/6	$\leftarrow 2 \quad 1/6$
x_1	1	0	0	-1/3	0	2/3	11/3	$\leftarrow 3 \quad 2/3$
S_3	0	0	0	1/3	1	-5/3	4/3	$\leftarrow 1 \quad 1/3$

$$\text{kaynak satır } x_1 \quad x_1 \quad -\frac{1}{3}S_2 \quad +\frac{2}{3}\overline{S_1} \quad =\frac{11}{3}$$

Yöneylem Araştırması II

138

[Gomory Kesme Düzlemi Metodu]



Kaynak satır: x_1

$$x_1 - \frac{1}{3}S_2 + \frac{2}{3}\overline{S}_1 = \frac{11}{3}$$

$$x_1 + (-1 + \frac{2}{3})S_2 + (0 + \frac{2}{3})\overline{S}_1 = (3 + \frac{2}{3})$$

$$\boxed{\frac{2}{3}S_2 + \frac{2}{3}\overline{S}_1 \geq \frac{2}{3}}$$

← 2. Kesme

Her iki taraf (-1) ile çarpılırsa;

$$-\frac{2}{3}S_2 - \frac{2}{3}\overline{S}_1 \leq -\frac{2}{3}$$

$$\boxed{-\frac{2}{3}S_2 - \frac{2}{3}\overline{S}_1 + \overline{S}_2 = -\frac{2}{3}} \leftarrow \text{Eklencek kısıt}$$

Yöneylem Araştırması II

139

[Gomory Kesme Düzlemi Metodu]



$$\boxed{-\frac{2}{3}S_2 - \frac{2}{3}\overline{S}_1 + \overline{S}_2 = -\frac{2}{3}} \leftarrow \text{Eklencek kısıt}$$

	x_1	x_2	S_1	S_2	S_3	$\overline{S}_1$	$\overline{S}_2$	b_i
z	0	0	0	2/3	0	8/3	0	74/3
S_1	0	0	1	5/3	0	-1/3	0	26/3
x_2	0	1	0	1/6	0	1/6	0	13/6
x_1	1	0	0	-1/3	0	2/3	0	11/3
S_3	0	0	0	1/3	1	-5/3	0	4/3
$\overline{S}_2$	0	0	0	-2/3	0	-2/3	1	-2/3
z	0	0	0	0	0	2	1	24
S_1	0	0	1	0	0	-2	5/2	7
x_2	0	1	0	0	0	0	1/4	2
x_1	1	0	0	0	0	1	-1/2	4
S_3	0	0	0	0	1	-2	-1/2	1
S_2	0	0	0	1	0	1	-3/2	1

Tamsayılı
Optimal Çözüm

$$z = 24$$

$$x_1 = 4$$

$$x_2 = 2$$

Yöneylem Araştırması II

140

[Gomory Kesme Düzlemi Metodu]



■ Kesmelerin x_1 ve x_2 cinsinden ifadesi

$$\frac{1}{5}S_2 + \frac{3}{5}S_3 \geq \frac{4}{5} \Rightarrow 1S_2 + 3S_3 \geq 4 \quad \leftarrow \text{I Kesme}$$

$$\frac{2}{3}S_2 + \frac{2}{3}\bar{S}_1 \geq \frac{2}{3} \Rightarrow 2S_2 + 2\bar{S}_1 \geq 2 \Rightarrow S_2 + \bar{S}_1 \geq 1 \quad \leftarrow \text{2. Kesme}$$

Modelin standart yapısından

$$\begin{array}{rclcl} z & -2x_1 & -8x_2 & & = 0 \\ & 2x_1 & -6x_2 & +S_1 & = 3 \Rightarrow S_1 = 3 - 2x_1 + 6x_2 \quad \dots\dots\dots(1) \\ & -x_1 & +4x_2 & +S_2 & = 5 \Rightarrow S_2 = 5 + x_1 - 4x_2 \quad \dots\dots\dots(2) \\ & 2x_1 & +2x_2 & +S_3 & = 13 \Rightarrow S_3 = 13 - 2x_1 - 2x_2 \quad \dots\dots\dots(3) \\ & x_1, & x_2, & S_1, & S_2, & S_3 & \geq 0 \end{array}$$

Birinci kesmenin standart yapısından

$$\begin{aligned} -\frac{1}{5}S_2 - \frac{3}{5}S_3 + \bar{S}_1 &= -\frac{4}{5} \Rightarrow \bar{S}_1 = -\frac{4}{5} + \frac{1}{5}S_2 + \frac{3}{5}S_3 \\ \Rightarrow \bar{S}_1 &= -\frac{4}{5} + \frac{1}{5}(5 + x_1 - 4x_2) + \frac{3}{5}(13 - 2x_1 - 2x_2) \Rightarrow \bar{S}_1 = 8 - x_1 - 2x_2 \quad \dots\dots\dots(4) \end{aligned}$$

[Gomory Kesme Düzlemi Metodu]



■ Kesmelerin x_1 ve x_2 cinsinden ifadesi

$$\begin{aligned} S_1 &= 3 - x_1 + 6x_2 \quad \dots\dots\dots(1) \\ S_2 &= 5 + x_1 - 4x_2 \quad \dots\dots\dots(2) \\ S_3 &= 13 - 2x_1 - 2x_2 \quad \dots\dots\dots(3) \\ \bar{S}_1 &= 8 - x_1 - 2x_2 \quad \dots\dots\dots(4) \end{aligned}$$

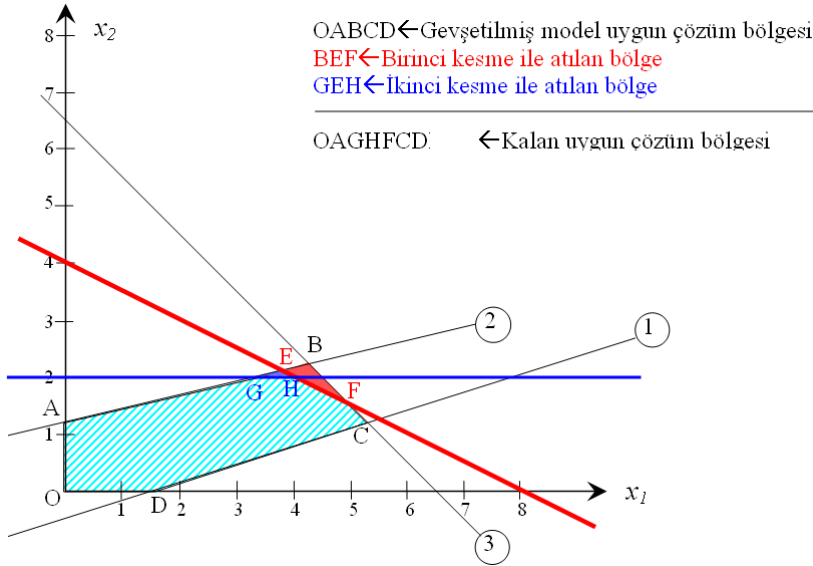
I Kesme

$$S_2 + 3S_3 \geq 4 \Rightarrow (5 + x_1 - 4x_2) + 3(13 - 2x_1 - 2x_2) \geq 4 \Rightarrow x_1 + 2x_2 \leq 8$$

2. Kesme

$$S_2 + \bar{S}_1 \geq 1 \Rightarrow (5 + x_1 - 4x_2) + (8 - x_1 - 2x_2) \geq 1 \Rightarrow x_2 \leq 2$$

[Gomory Kesme Düzlemi Metodu]



Yöneylem Araştırması II

143

[Gomory Kesme Düzlemi (Kesirli Kesme Düzlemi) Metodu]



■ Karma Tam Sayılı Programlama

- Karma tam sayılı programlamada kesme düzlemleri **sadece tam sayı olması istenen** karar değişkeni satırları için oluşturulur.
- S_k oluşturulan kesme düzlemine ilave edilen gevşek değişken olmak üzere k satırı için kesme düzlemi aşağıdaki şekilde oluşturulur.

$$S_k - \left\{ \sum_{j \in J^+} \alpha_{kj} x_j + \frac{f_k}{f_k - 1} \sum_{j \in J^-} \alpha_{kj} x_j \right\} = f_k \text{ (Karışık Kesme)}$$

- Bunlar dikkate alındığında şu kesme denklemi elde edilir;

$$S_k = -f_k + \sum_{j=1}^n \lambda_j x_j \text{ (m kesme)}$$

$$\lambda_j = \begin{cases} \alpha_{kj} & \alpha_{kj} \geq 0 \text{ ve } x_j \text{ tamsayılı değilse} \\ \frac{f_k - 1}{f_k} \alpha_{kj} & \alpha_{kj} < 0 \text{ ve } x_j \text{ tamsayılı değilse} \\ f_{ki} & f_{kj} \leq f_k \text{ ve } x_j \text{ tamsayılı ise} \\ \frac{f_k}{1 - f_k} (1 - f_{kj}) & f_{kj} > f_k \text{ ve } x_j \text{ tamsayılı ise} \end{cases}$$

- Burada;
 - λ_j : kesme düzlemi katsayıları
 - α_{kj} : x_j değişkeninin kaynak satırı (yani k satırı) katsayısının kesirli kısmı
 - f_k : kaynak satırı sağ taraf değerinin kesirli kısmı

Yöneylem Araştırması II

144

[Gomory Kesme Düzlemi Metodu]



■ Karma Tam Sayılı Programlama (Örnek 1)

$$\begin{aligned}
 \text{Maks } Z &= 7x_1 + 9x_2 \\
 \text{S.T.} \quad &-x_1 + 3x_2 \leq 6 \\
 &7x_1 + x_2 \leq 35 \\
 &x_1 \geq 0 \quad \text{ve tam sayı} \\
 &x_2 \geq 0
 \end{aligned}$$

Gevşetilmiş modelin optimal simpleks tablosu
(x_3 ve x_4 gevşek değ.)

T.D.	Değ. No	Değişkenler					STD
		z	x_1	x_2	x_3	x_4	
z	0	1	0	0	28/11	15/11	63
x_2	1	0	0	1	7/22	1/22	7/2
x_1	2	0	1	0	-1/22	3/22	9/2

Yöneylem Araştırması II

145

[Gomory Kesme Düzlemi Metodu]



■ Karma Tam Sayılı Programlama (Örnek 1)

T.D.	Değ. No	Değişkenler					STD
		z	x_1	x_2	x_3	x_4	
z	0	1	0	0	28/11	15/11	63
x_2	1	0	0	1	7/22	1/22	7/2
x_1	2	0	1	0	-1/22	3/22	9/2

Problemde x_1 temel değişkeninin tam sayılı olması istendiğinden;

$$x_1 - \frac{1}{22}x_3 + \frac{3}{22}x_4 = \frac{9}{2} = 4 + \frac{1}{2}$$

Burada;

$$f_l = f_1 = \frac{1}{2}, \quad k_{lj} = k_{13} = -\frac{1}{22} < 0, \quad k_{lj} = k_{14} = \frac{3}{22} > 0$$

Yeni kesme düzlemi;

$$x_5 - \left[\frac{3}{22}x_4 + \left(\frac{1/2}{1/2 - 1} \right) \left(-\frac{1}{22} \right) x_3 \right] = -\frac{1}{2} \quad \text{veya} \quad x_5 - \frac{1}{22}x_3 - \frac{3}{22}x_4 = -\frac{1}{2}$$

Yöneylem Araştırması II

146

[Gomory Kesme Düzlemi Metodu]



■ Karma Tam Sayılı Programlama (Örnek 1)

Oluşturulan kesme düzlemi tabloya eklenerek dual simpleks metodu uygulanırsa;

T.D.	Değ. No	Değişkenler						STD
		z	x ₁	x ₂	x ₃	x ₄	x ₅	
z	0	1	0	0	28/11	15/11	0	63
x ₂	1	0	0	1	7/22	1/22	0	7/2
x ₁	2	0	1	0	-1/22	3/22	0	9/2
x ₅	3	0	0	0	-1/22	-3/22	1	-1/2
Z	0	1	0	0	23/11	0	10	58
x ₂	1	0	0	1	10/33	0	1/3	10/3
x ₁	2	0	1	0	-1/11	0	1	4
x ₄	3	0	0	0	1/3	1	-22/3	11/3

Tam sayılı optimal çözüm;

$$x_1=4$$

$$x_2=10/3$$

$$Z=58$$

Yöneylem Araştırması II

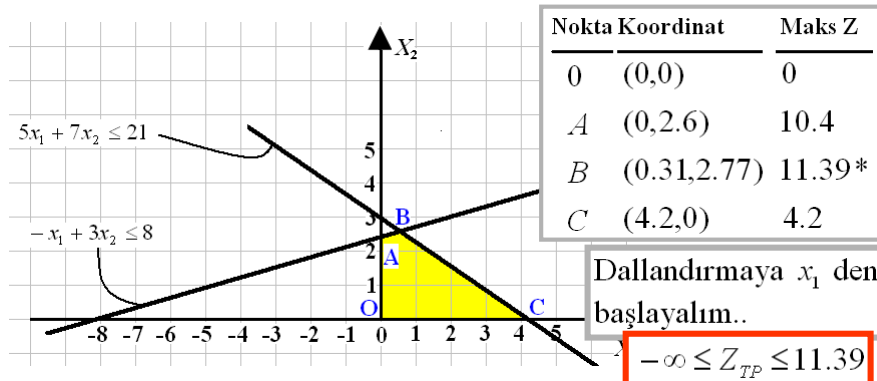
147

[Dal-sınır algoritması (Tekrar)]



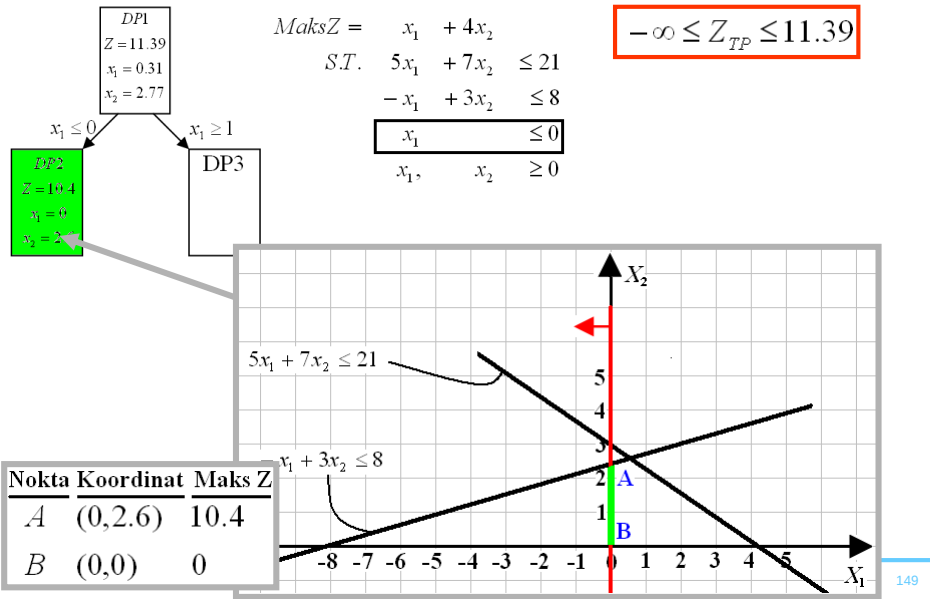
$$\begin{aligned}
 \text{Maks } Z &= x_1 + 4x_2 \\
 \text{S.T. } 5x_1 + 7x_2 &\leq 21 \\
 -x_1 + 3x_2 &\leq 8 \\
 x_1, x_2 &\geq 0 \quad \text{ve tam sayı}
 \end{aligned}$$

$$\begin{aligned}
 x_1 = 0 &\Rightarrow x_2 = 3 \\
 x_2 = 0 &\Rightarrow x_1 = 4.2 \\
 x_1 = 0 &\Rightarrow x_2 = 2.6 \\
 x_2 = 0 &\Rightarrow x_1 = -8
 \end{aligned}$$

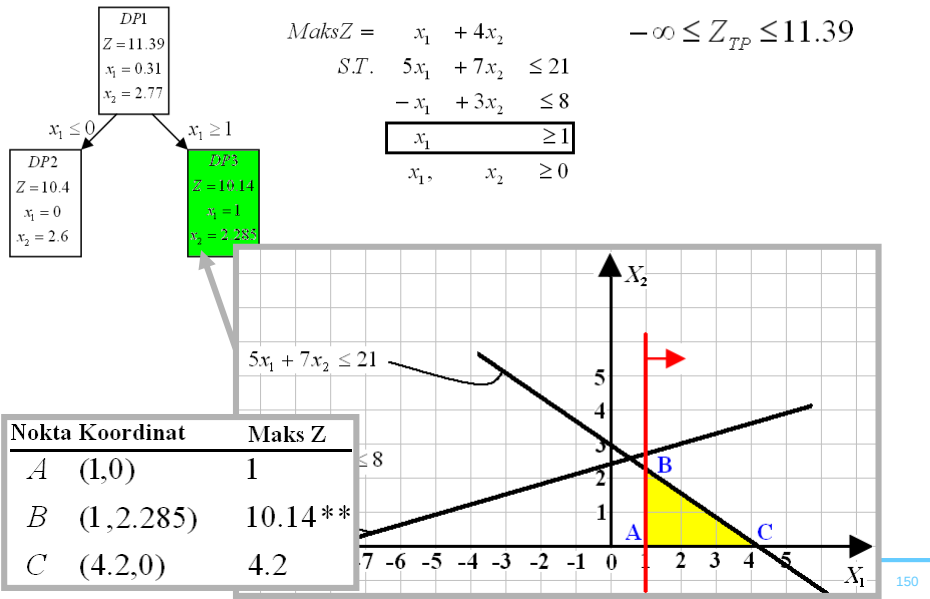


148

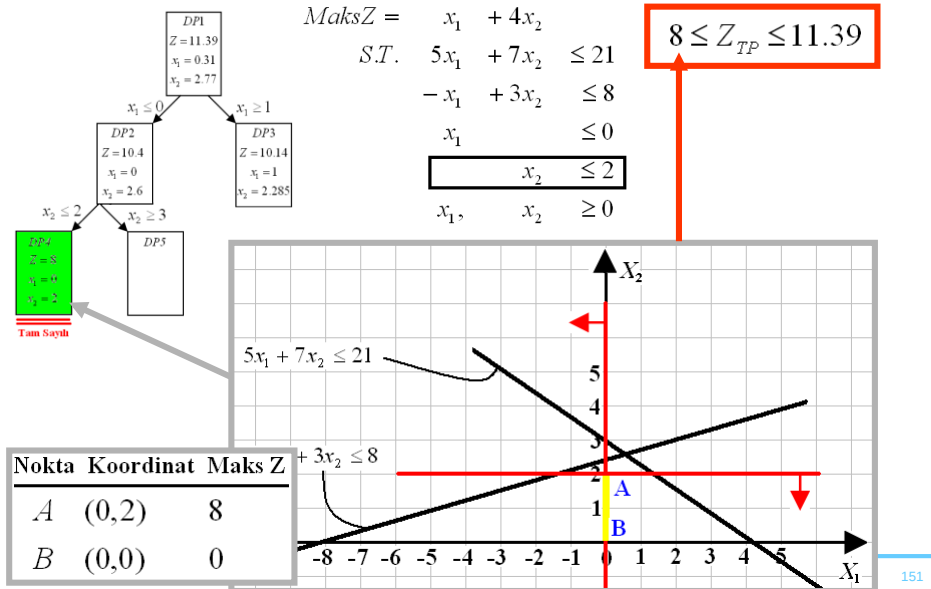
[Dal-sınır algoritması (Tekrar)]



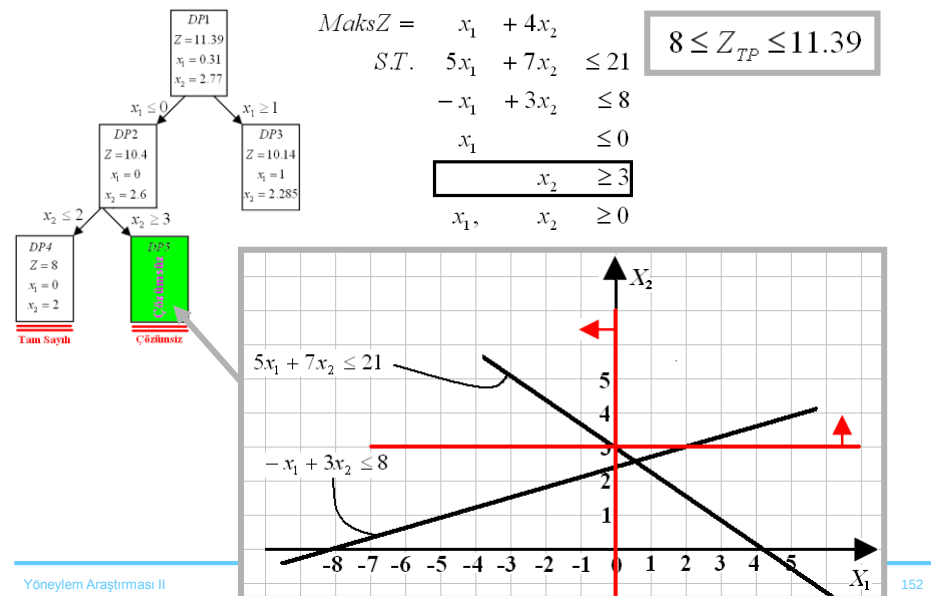
[Dal-sınır algoritması (Tekrar)]



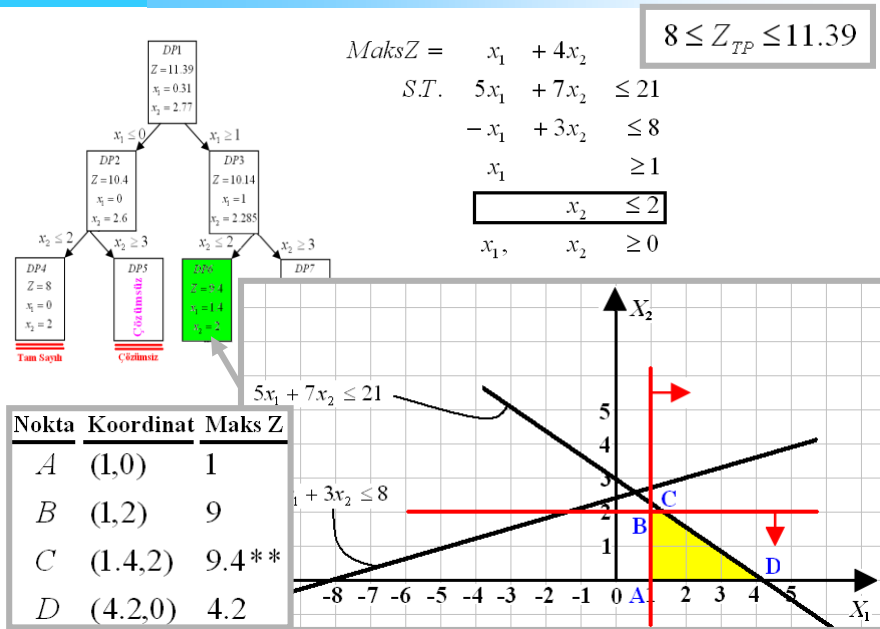
[Dal-sınır algoritması (Tekrar)]



[Dal-sınır algoritması (Tekrar)]

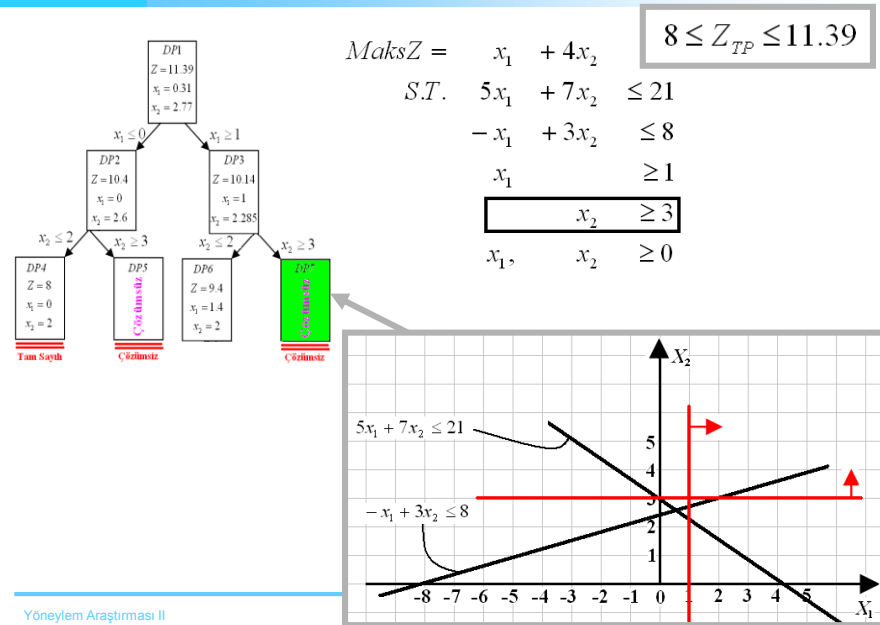


Dal-sınır algoritması (Tekrar)



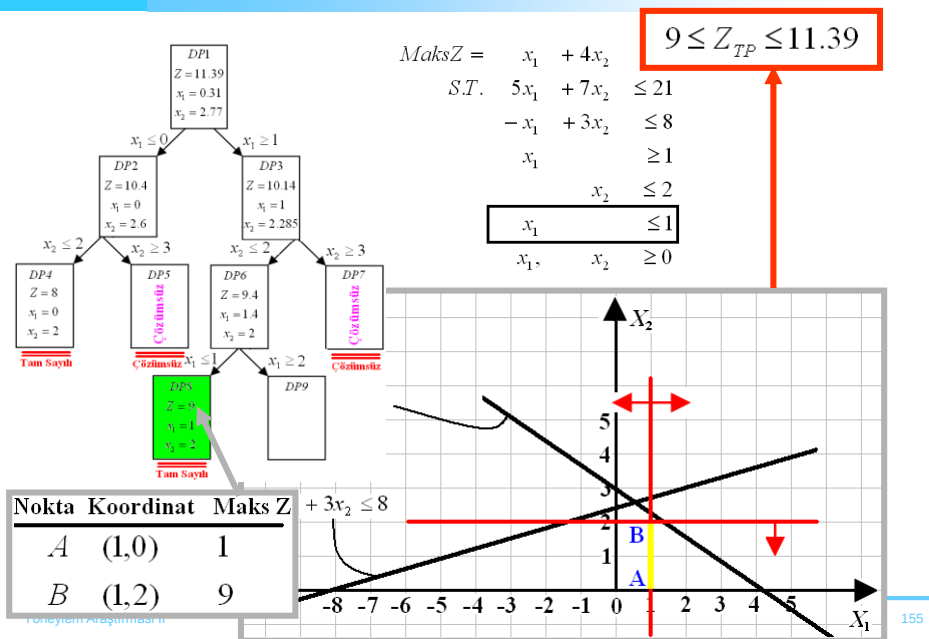
153

Dal-sınır algoritması (Tekrar)

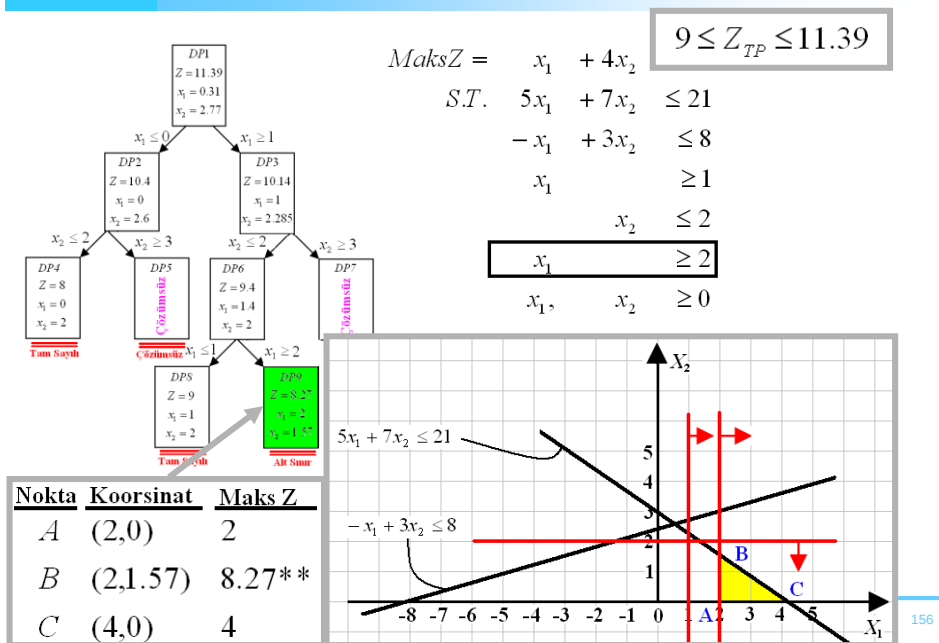


154

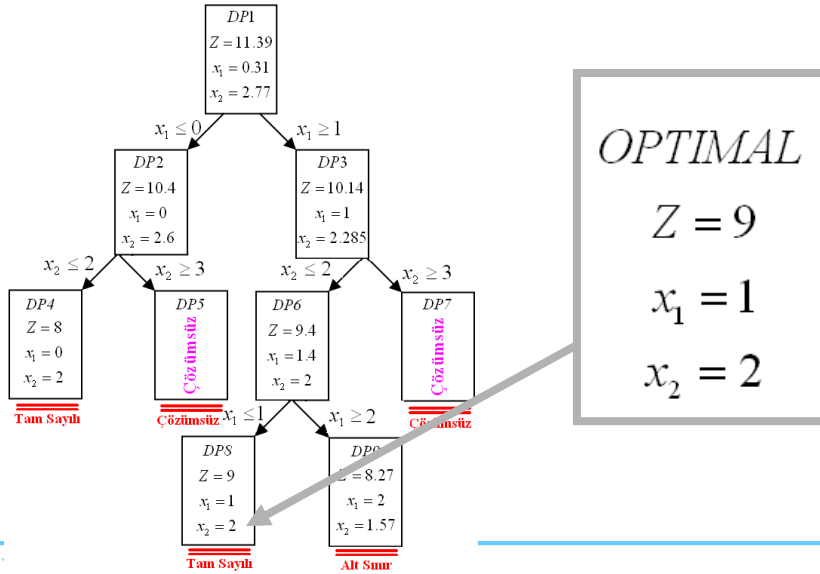
[Dal-sınır algoritması (Tekrar)]



[Dal-sınır algoritması (Tekrar)]



[Dal-sınır algoritması (Tekrar)]



157

[Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)]



$$\begin{aligned}
 \max Z = & 7x_1 + 10x_2 \\
 \text{s.t.} \quad & -x_1 + 3x_2 \leq 6 \\
 & 7x_1 + x_2 \leq 35 \\
 & x_1, x_2 \geq 0 \quad \text{ve tamsayı}
 \end{aligned}$$

Gevşetilmiş modelin Primal Simpleks yöntemiyle bulunan optimal çözüm tablosu

	x ₁	x ₂	S ₁	S ₂	b _i	
z	0	0	63/22	21/22	133/2	← 66 1/2 ← 0.50
x ₂	0	1	7/22	1/22	7/2	← 3 1/2 ← 0.50
x ₁	1	0	-1/22	3/22	9/2	← 4 1/2 ← 0.50

Tüm kesirli kısımlar birbirine eşit olduğundan keyfi olarak x₂ satırını kaynak satır seçelim.

$$x_2 + \frac{7}{22}S_1 + \frac{1}{22}S_2 = \frac{7}{2}$$

[Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)]



$$\begin{array}{rclcl}
 x_2 & + \frac{7}{22}S_1 & + \frac{1}{22}S_2 & = & \frac{7}{2} \\
 x_2 & + (0 + \frac{7}{22})S_1 & + (0 + \frac{1}{22})S_2 & = & (0 + \frac{7}{2}) \\
 \frac{7}{22}S_1 & + \frac{1}{22}S_2 & & \geq & \frac{7}{22} & \leftarrow \text{1. kesme} \\
 \hline
 -\frac{7}{22}S_1 & - \frac{1}{22}S_2 & & \leq & -\frac{7}{22} \\
 -\frac{7}{22}S_1 & - \frac{1}{22}S_2 & + \overline{S}_1 & = & -\frac{7}{22} & \leftarrow \text{Eklenecek kısıt}
 \end{array}$$

Yöneylem Araştırması II

159

[Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)]



1. kesme

$$-\frac{7}{22}S_1 - \frac{1}{22}S_2 + \overline{S}_1 = -\frac{7}{22}$$

	x_1	x_2	S_1	S_2	$\overline{S}_1$	b_i
z	0	0	63/22	21/22	0	133/2
x_2	0	1	7/22	1/22	0	7/2
x_1	1	0	-1/22	3/22	0	9/2
$\overline{S}_1$	0	0	-7/22	-1/22	1	-7/22
z	0	0	0	12/22	9	62
x_2	0	1	0	0	1	3
x_1	1	0	0	1/7	-1/7	32/7
S_1	0	0	1	1/7	-22/7	11/7

$\leftarrow -$
 $\leftarrow -$
 $\leftarrow 4 \quad 4/7 \leftarrow 0.57$
 $\leftarrow 1 \quad 4/7 \leftarrow 0.57$

Tüm kesirli kısımlar birbirine eşit olduğundan keyfi olarak x_1 satırını kaynak satır seçelim.

$$x_1 + \frac{1}{7}S_2 - \frac{1}{7}\overline{S}_1 = \frac{32}{7}$$

Yöneylem Araştırması II

160

[Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)]



Kaynak satır: x_1

$$\begin{array}{rclcl}
 x_1 & + \frac{1}{7} S_2 & - \frac{1}{7} \overline{S}_1 & = & \frac{32}{7} \\
 x_1 & + (0 + \frac{1}{7}) S_2 & + (-1 + \frac{6}{7}) \overline{S}_1 & = & (4 + \frac{4}{7}) \\
 & \frac{1}{7} S_2 & + \frac{6}{7} \overline{S}_1 & \geq & \frac{4}{7} & \leftarrow 2. \text{ kesme} \\
 \hline
 & - \frac{1}{7} S_2 & - \frac{6}{7} \overline{S}_1 & \leq & - \frac{4}{7} \\
 & - \frac{1}{7} S_2 & - \frac{6}{7} \overline{S}_1 & + \overline{S}_2 & = - \frac{4}{7} & \leftarrow \text{Eklenecek kısıt}
 \end{array}$$

Yöneylem Araştırması II

161

[Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)]



2. kesme

$$- \frac{1}{7} S_2 - \frac{6}{7} \overline{S}_1 + \overline{S}_2 = - \frac{4}{7}$$

	x_1	x_2	S_1	S_2	$\overline{S}_1$	$\overline{S}_2$	b_i
z	0	0	0	1	9	0	133/2
x_2	0	1	0	0	1	0	7/2
x_1	1	0	0	1/7	-1/7	0	9/2
S_1	0	0	1	1/7	-22/7	0	11/7
$\overline{S}_2$	0	0	0	-1/7	-6/7	1	-4/7
z	0	0	0	0	3	7	58
x_2	0	1	0	0	1	0	3
x_1	1	0	0	0	-1	1	4
S_1	0	0	1	0	-4	1	1
S_2	0	0	0	1	6	-7	4

Optimal çözüm

$$z = 58$$

$$x_1 = 4$$

$$x_2 = 3$$

Yöneylem Araştırması II

162

Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)



Birinci Kesme $\frac{7}{22}S_1 + \frac{1}{22}S_2 \geq \frac{7}{22} \rightarrow \frac{7}{22}(6 + x_1 - 3x_2) + \frac{1}{22}(35 - 7x_1 - x_2) \geq \frac{7}{22} \rightarrow x_2 \leq \frac{70}{22}$

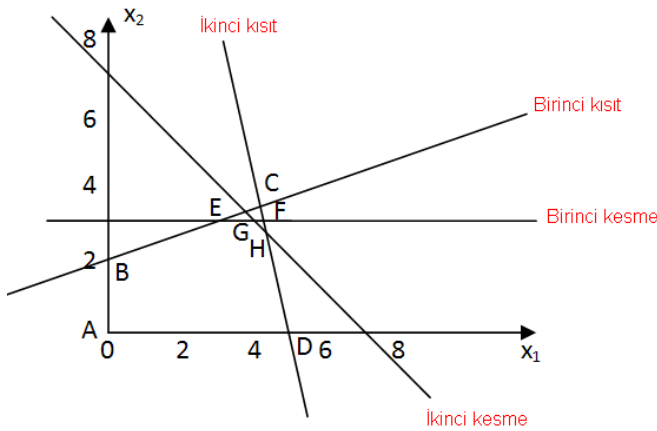
Birinci kısıt $-x_1 + 3x_2 + S_1 = 6 \rightarrow S_1 = 6 + x_1 - 3x_2$

İkinci kısıt $7x_1 + x_2 + S_2 = 35 \rightarrow S_2 = 35 - 7x_1 - x_2$

Birinci Kesme $x_2 \leq \frac{70}{22} \rightarrow x_2 + \bar{S}_1 = \frac{70}{22} \rightarrow \bar{S}_1 = \frac{70}{22} - x_2$

İkinci Kesme $\frac{1}{7}S_2 + \frac{6}{7}\bar{S}_1 \geq \frac{4}{7} \rightarrow \frac{1}{7}(35 - 7x_1 - x_2) + \frac{6}{7}\left(\frac{70}{22} - x_2\right) \geq \frac{4}{7} \rightarrow x_1 + x_2 \leq \frac{551}{77}$

Gomory Kesme Düzlemi metodu (Tümü Tamsayı Programlama - Tekrar)



Gevşetilmiş Problemin Uygun çözüm bölgesi: ABCD

Birinci kesmeyle atılan: CEF

İkinci kesmeyle atılan: FGH

Tamsayı optimal nokta: G(4, 3)

[Gomory Kesme Düzlemi metodu (Karma Tamsayılı Programlama - Tekrar)]



$$\begin{aligned} \text{Maks } Z &= 2x_1 + 2x_2 \\ \text{S.T. } 5x_1 + 3x_2 &\leq 8 \\ 2x_1 + 4x_2 &\leq 8 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \text{ ve tam sayı} \end{aligned}$$

Gevşetilmiş modelin optimal simpleks tablosu
(x_3 ve x_4 gevşek değ.)

T.D.	Değ. No	Değişkenler					STD
		z	x_1	x_2	x_3	x_4	
z	0	1	0	0	8/7	2/7	32/7
x_1	1	0	1	0	5/7	-3/14	4/7
x_2	2	0	0	1	-1/7	5/14	12/7

Yöneylem Araştırması II

165

[Gomory Kesme Düzlemi metodu (Karma Tamsayılı Programlama - Tekrar)]



T.D.	Değ. No	Değişkenler					STD
		z	x_1	x_2	x_3	x_4	
z	0	1	0	0	8/7	2/7	32/7
x_1	1	0	1	0	5/7	-3/14	4/7
x_2	2	0	0	1	-1/7	5/14	12/7

Problemde x_2 temel değişkeninin tam sayılı olması istendiğinden;

$$x_2 - \frac{1}{7}x_3 + \frac{5}{14}x_4 = \frac{12}{7} = 1 + \frac{5}{7}$$

Burada;

$$f_1 = f_2 = \frac{5}{7}, \quad k_{ij} = k_{23} = -\frac{1}{7} < 0, \quad k_{ij} = k_{24} = \frac{5}{14} > 0$$

Yeni kesme düzlemi;

$$x_5 - \left[\frac{5}{14}x_4 + \left(\frac{5/7}{5/7-1} \right) \left(-\frac{1}{7} \right) x_3 \right] = -\frac{5}{7} \quad \text{veya} \quad x_5 - \frac{5}{14}x_3 - \frac{5}{14}x_4 = -\frac{5}{7}$$

Yöneylem Araştırması II

166

[Gomory Kesme Düzlemi metodu (Karma Tamsayılı Programlama - Tekrar)]



Oluşturulan kesme düzlemi tabloya eklenerek dual simpleks metodu uygulanırsa;

T.D.	Değ. No	Değişkenler						STD
		z	x_1	x_2	x_3	x_4	x_5	
z	0	1	0	0	8/7	2/7	0	32/7
x_1	1	0	1	0	5/7	-3/14	0	4/7
x_2	2	0	0	1	-1/7	5/14	0	12/7
x_5	3	0	0	0	-5/14	-5/14	1	-5/7
z	0	1	0	0	6/7	0	4/5	4
x_1	1	0	1	0	13/14	0	-3/5	1
x_2	2	0	0	1	-1/2	0	1	1
x_4	3	0	0	0	1	1	-14/5	2

Tam sayılı optimal çözüm;

$X_1=1$ (Tasadüfen tam sayı çıktı)

$X_2=1$

$Z=4$

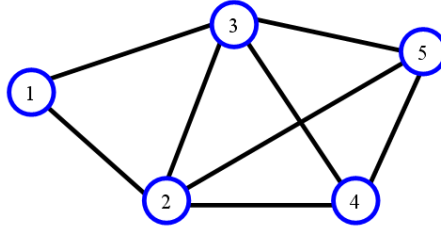
[BÖLÜM IV: Şebeke Modelleri]

- Şebeke kavramları
- En az yayılan ağaç problemleri
- En kısa yol problemleri
- En büyük akış problemleri

[Şebeke Kavramları]



- **Şebeke (Network)**
- Sonlu sayıdaki düğümler kümesiyle, bunlarla bağlantılı oklar (veya dallar) kümesinin oluşturduğu yapı şeklinde tanımlanabilir ve (N,A) şeklinde gösterilir.
 - Burada N düğümler kümesini,
 - A ise düğümleri birleştiren oklar (veya dallar) kümesini ifade eder.



- Verilen şebekede
 - $N=\{1, 2, 3, 4, 5\}$
 - $A=\{(1,3), (1,2), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)\}$ şeklinde olacaktır.

[Şebeke Kavramları]



- **Yönlü Şebeke (Directed Network)**
- Bir şebekede dallar yönlü ise **Yönlü Şebeke** değilse **Yönsüz Şebeke** adını alır.

■ Dal / ok

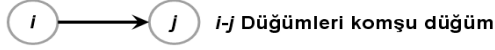
- i ve j düğümlerini bağlayan çizgiye **Çizgi** veya **Dal** denir. Bu çizgi veya dalda yön varsa **Ok** adını alır.

[Şebeke Kavramları]



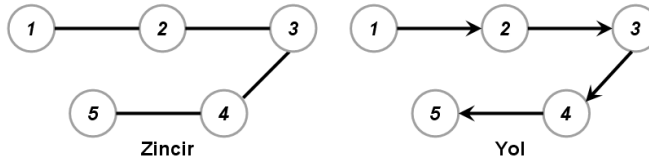
■ Bitişik (Komşu) Dğümler

- İki düğüm arasında doğrudan dal veya ok varsa o düğümlere **Komşu Dğümler** (veya Bitişik Dğümler) denir.



■ Zincir/yol

- *i* ve *j* düğümlerini birbirine bağlayan dal veya oklar dizisine **Zincir** denir. Zincirdeki okların *j* düğümüne yönelme şartı yoktur. Bir zincirdeki oklar aynı yönde yönelmiş ise **Yol** adını alır.



- Her yol bir zincirdir (tersi doğru değildir).

Yöneylem Araştırması II

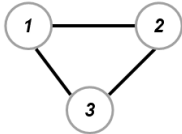
171

[Şebeke Kavramları]



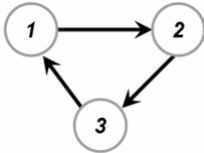
■ Çevrim (Cycle)

- Kapalı zincire **Çevrim** denir.



■ Döngü (Kapalı Yol - Loop)

- Yolun başlangıcı ile sonu aynı ise **Döngü (Kapalı Yol)** adını alır.



- Her döngü bir çevrim oluşturur. Tersisi doğru değildir.

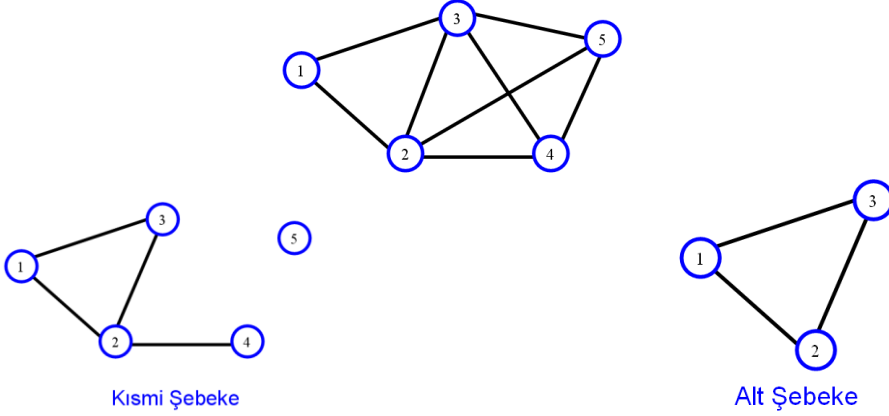
Yöneylem Araştırması II

172

[Şebeke Kavramları]



- **Kısmi Şebeke (Partial Network)**
- Şebekenin bazı dalları/okları ihmal edilerek elde edilen şebekeye **Kısmi Şebeke**, şebekenin bazı düğümleri ihmal edilerek elde edilen şebekeye **Alt Şebeke** adı verilir.



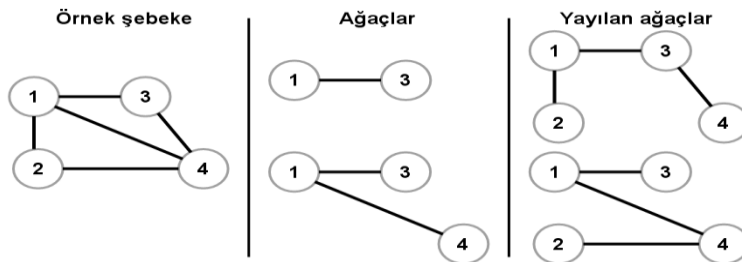
Yöneylem Araştırması II

173

[Şebeke Kavramları]



- **Ağaç (Tree)**
 - Her düğüm çiftini bağlayan bir zincirin olduğu şebekeye **Bağlantılı Şebeke** denir.
 - Şebekenin düğümlerinin sadece belli bir alt setinden oluşabilen ve çevrim ihtiva etmeyen şebekeye **Ağaç** denir. n düğümlü bir ağaçta $(n-1)$ adet dal bulunur.
 - Şebekenin bütün düğümlerinin ağaçta bulunma zorunluluğu **yoktur**.
 - Eğer ağaç şebekenin bütün düğümlerini içeriyorsa **Yayılan Ağaç** adını alır.



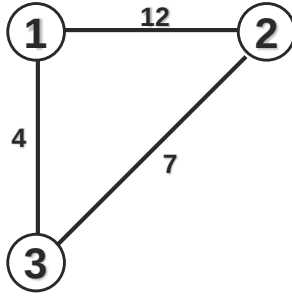
Yöneylem Araştırması II

174

[En Az Yayılan Ağaç Problemleri]



- Şebekenin yayılan ağaçlarından **en küçük uzunluğa** sahip olanına **En Az Yayılan Ağaç** (Minimum Spanning Tree) adı verilir.



Yayılan ağaçlar

1. (1,2) ve (2,3) dalları
2. (1,2) ve (1,3) dalları
3. (1,3) ve (2,3) dalları

En az yayılan ağaç

(1,3) ve (2,3) dalları

[En Az Yayılan Ağaç Problemleri]



- En az yayılan ağaç problemleri günlük hayatta çeşitli şekillerde ortaya çıkar;
 - Elektrik hattı problemleri,
 - Boru hattı problemleri,
 - Telefon hattı problemleri,
 - Digital bilgisayar sistemlerinin kurulması.
- En az yayılan ağaç problemleri sadece yönsüz şebekelere uygulanır.
- Bu bölüm kapsamında çözüm algoritmalarından biri olan **Prim Algoritması** incelenecektir.

[En Az Yayılan Ağaç Problemleri]



Prim Algoritması

- $N=\{1,2,\dots,n\}$ şebekenin düğümlerinin kümesi, c_k algoritmanın k iterasyonunda bağlanan düğümlerin kümesi ve $\hat{c}_k$ henüz bağlanmamış olan düğümlerin kümesi olsun.
 - Adım 0: $c_0=\{\}$, $\hat{c}_0=n$ olarak alınır.
 - Adım 1: $\hat{c}_0$ deki herhangi bir i düğümünden başlanır. $c_1=\{i\}$ ve $\hat{c}_1=\hat{c}_0 - \{i\}$ alınır.
 - Adım k : Bağlanmış düğümlerden birine **en yakın** bağlanmamış düğüm belirlenir. İki veya daha fazla düğüm en yakın ise **rasgele** seçilir. Bu yeni düğüm bağlanmış düğümler kümesine eklenir. Yani c_{k-1} deki herhangi bir düğüme en yakın olan $\hat{c}_{k-1}$ kümesindeki bir j^* düğümü seçilir. j^* , c_{k-1} 'e eklenir ve $\hat{c}_{k-1}$ kümesinden çıkarılır:

$$c_k=c_{k-1}+\{j^*\} \quad \text{ve} \quad \hat{c}_k=\hat{c}_{k-1}-\{j^*\}$$
 bağlanmamış düğümler kümesi $\hat{c}_k$, boş ise durulur, aksi taktirde $k=k+1$ alınarak işlem tekrarlanır.

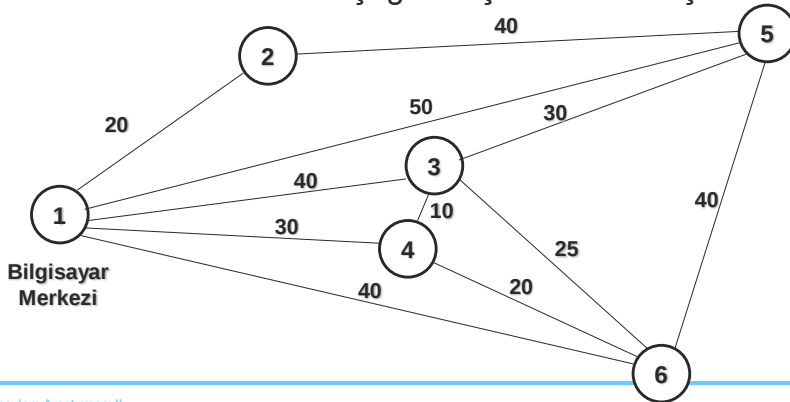
Yöneylem Araştırması II

177

[En Az Yayılan Ağaç Problemleri]



- Örnek (bilgisayar hattı problemi)
- Bilgisayar merkezinden 5 kullanıcı binasına bilgisayar bağlantısı kurulacaktır. Maliyeti azaltmak için çekilecek hatların mümkün olduğu kadar kısa olması istenmektedir. Muhtemel bağlantı alternatifleri ile metre cinsinden binalar arasındaki mesafeler aşağıdaki şekilde verilmiştir.



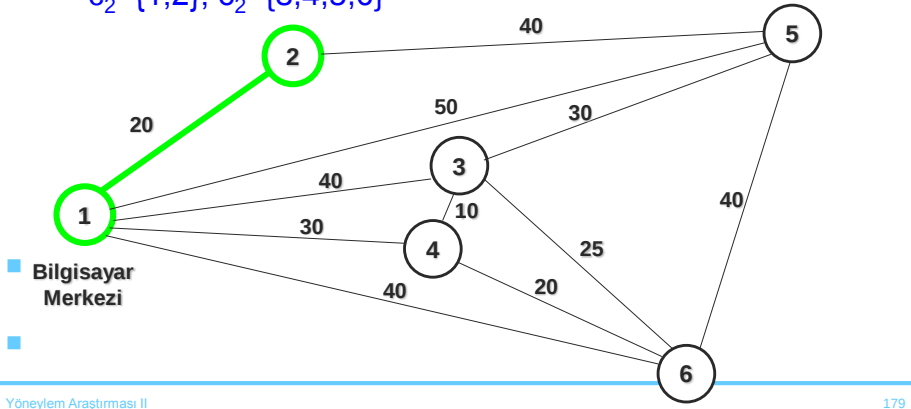
Yöneylem Araştırması II

178

[En Az Yayılan Ağaç Problemleri]



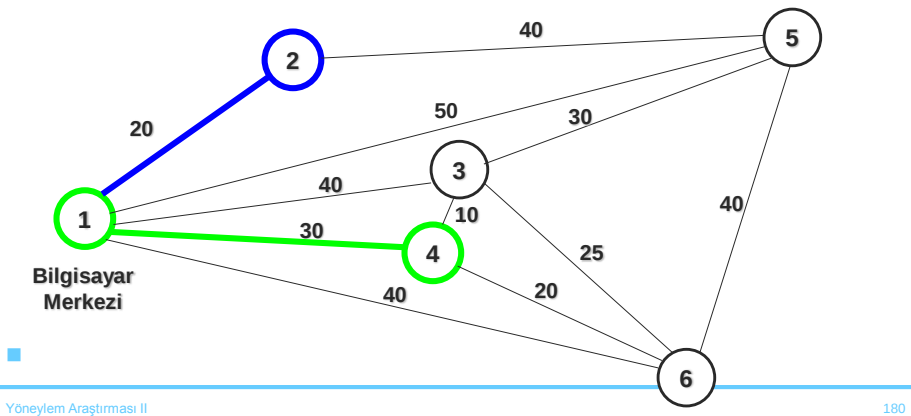
- Örnek (bilgisayar hattı problemi)
- Adım 1: Algoritmaya 1 düğümünden başlayalım.
 $c_1=\{1\}$, $\hat{c}_1=\{2,3,4,5,6\}$
- Adım 2: 1 düğümüne en yakın bağlanmamış düğüm 20 mt uzaklıktaki 2 düğümünü ise;
 $c_2=\{1,2\}$, $\hat{c}_2=\{3,4,5,6\}$



[En Az Yayılan Ağaç Problemleri]



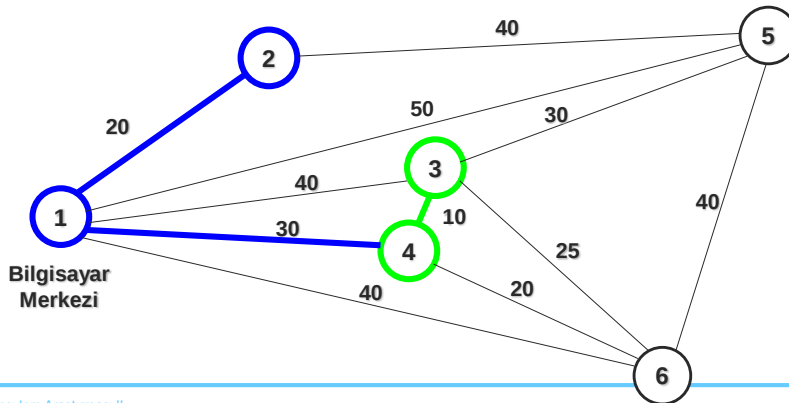
- Örnek (bilgisayar hattı problemi)
- Adım 3: 1 ve 2 düğümlerinden birine en yakın bağlanmamış düğüm 1 düğümüne 30 mt uzaklıktaki 4 düğümünü ise;
 $c_3=\{1,2,4\}$, $\hat{c}_3=\{3,5,6\}$



[En Az Yayılan Ağaç Problemleri]



- Örnek (bilgisayar hattı problemi)
- Adım 4: 1, 2 ve 4 düğümlerinden birine en yakın bağlanmamış düğüm 4 düğümüne 10 mt uzaklıktaki 3 düğümü ise;
 $c_4 = \{1, 2, 4, 3\}$, $c_4 = \{5, 6\}$



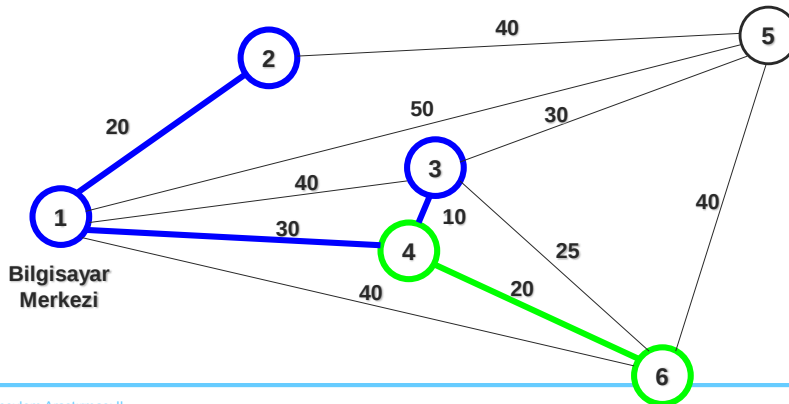
Yöneylem Araştırması II

181

[En Az Yayılan Ağaç Problemleri]



- Örnek (bilgisayar hattı problemi)
- Adım 5: 1, 2, 4 ve 3 düğümlerinden birine en yakın bağlanmamış düğüm 4 düğümüne 20 mt uzaklıktaki 6 düğümü ise;
 $c_5 = \{1, 2, 4, 3, 6\}$, $c_5 = \{5\}$



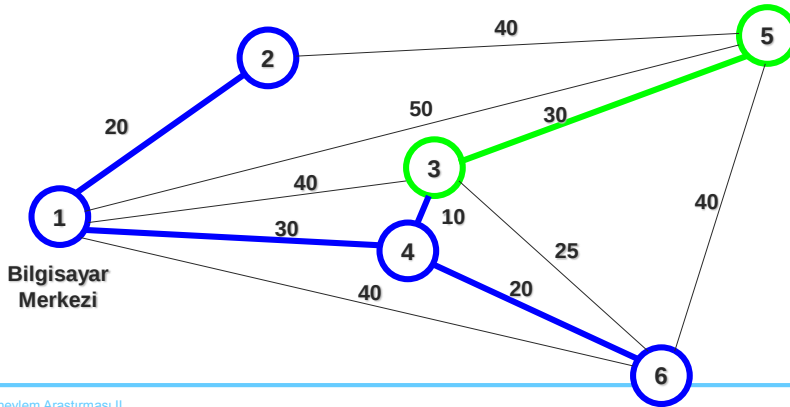
Yöneylem Araştırması II

182

[En Az Yayılan Ağaç Problemleri]



- Örnek (bilgisayar hattı problemi)
- Adım 6: 1, 2, 4, 3 ve 6 düğümlerinden birine en yakın bağlanmamış düğüm 3 düğümüne 30 mt uzaklıktaki 5 düğümü ise ((3,6) dalı çevrim oluşturduğundan dolayı **incelenmez**);
 $c_6 = \{1, 2, 4, 3, 6, 5\}$, $c_6 = \{ \}$



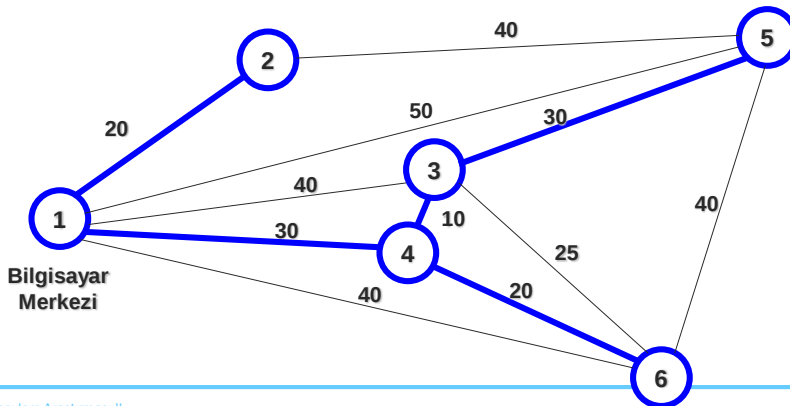
Yöneylem Araştırması II

183

[En Az Yayılan Ağaç Problemleri]



- Örnek (bilgisayar hattı problemi)
- $c_6 = \{ \}$ olduğu için en az yayılan ağaç bulunmuş olur.
- Ağacın **uzunluğu** ağacı oluşturan dalların uzunlukları toplamına eşit olup **110** metredir.



Yöneylem Araştırması II

184

[En Kısa Yol Problemleri]



- Bu problemlerde her hangi bir düğümden diğer düğüme olan en **Kısa Mesafenin Bulunması** amaçlanır.
- En kısa yol problemlerinin çözümünde kullanılan **iki** algoritma aşağıdaki şekildedir.
 1. Dijkstra algoritması
 2. Floyd algoritması

[En Kısa Yol Problemleri]



- **Dijkstra algoritması**
- Kaynak bir düğüm ile şebekedeki diğer bir düğüm arasındaki en kısa yolun belirlenmesi amacıyla tasarlanmıştır.
- **Avantajı**, döngülü veya döngüsüz şebekelere uygulanabilmesi, **dezavantajı** ise negatif uzunluklu ok bulunduran şebekelere uygulanamamasıdır.
- Başlangıç ve ulaşılması istenen düğüm **sabitletiğinde** işlem biter.
- Başlangıç belli fakat ulaşılması istenen düğüm belli değilse **bütün düğümler sabitletiğinde** işlem biter.

[En Kısa Yol Problemleri]



■ Dijkstra algoritması

NOTASYON

- N : Şebekedeki Tüm Döğümler Kümesi,
 S : Kalıcı Etikete Sahip Döğümler Kümesi,
 $\tilde{S}$: Geçici Etikete Sahip Döğümler Kümesi,
 Γ_k : k Döğümüne Direkt Bağlı Döğümler Kümesi,
 L_j : j Döğümünün Başlangıç Döğümüne Olan Uzaklığı,
 L_{kj} : Komşu Döğümler Olan k ve j Döğümleri Arasındaki Uzaklık,

ALGORİTMA

Adım 1: s başlangıç döğümünü göstermek üzere aşağıdaki işlemleri yap;

$$S = \{s\}, \tilde{S} = N - \{s\}, L_j = \begin{cases} 0 & j = s \text{ ise} \\ L_{sj} & j \in \Gamma_s \text{ ise} \\ \infty & j \notin \Gamma_s \text{ ise} \end{cases}$$

Adım 2: $L_k = \min_{j \in \tilde{S}} \{L_j\}$ olacak şekilde k 'yı tespit et. Eğer $\Gamma_k \subseteq S$ ise dur.

Değilse $S = S \cup \{k\}, \tilde{S} = \tilde{S} - \{k\}$ al.

Adım 3: k -ıncı çevrimi yap. Yani $j \in \Gamma_k \cap \tilde{S}$ olmak üzere tüm j döğümleri için aşağıdaki işlemleri yap ve Adım 2'ye git.

$$L_j = \min \{L_j, L_k + L_{kj}\}$$

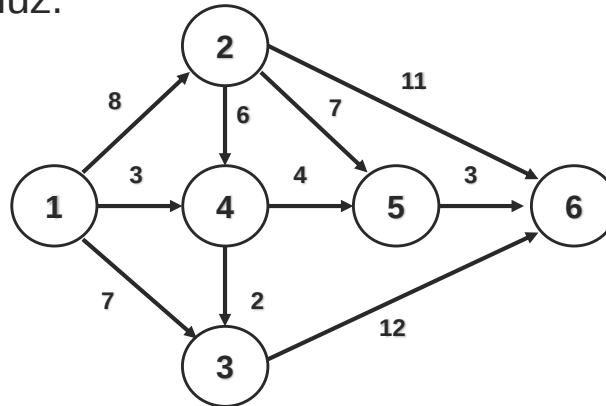
Yöneylem Araştırması II

187

[En Kısa Yol Problemleri]



- Dijkstra algoritması (Örnek)
- 1 ile 6 arasındaki minimum yolu ve uzunluğu bulunuz.



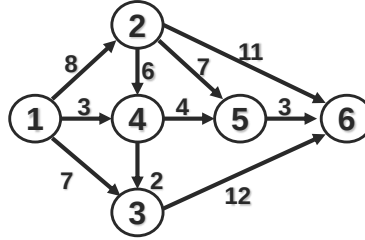
Yöneylem Araştırması II

188

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



Adım1 : $s=1$, $S=\{1\}$, $\tilde{S}=\{2,3,4,5,6\}$, $L_s=L_1=0$, $\Gamma_1=\{2,3,4\}$ olmak üzere;

$$L_2 = L_{12} = 8$$

$$L_3 = L_{13} = 7$$

$$L_4 = L_{14} = 3$$

$$L_5 = L_6 = \infty$$

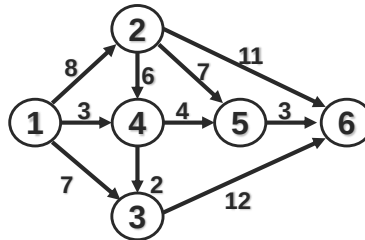
Yöneylem Araştırması II

189

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



Adım 2: $L_k = \min\{L_2, L_3, L_4, L_5, L_6\} = \{8, 7, 3, \infty, \infty\} = 3 \Rightarrow k=4$.

$\Gamma_k = \Gamma_4 = \{3, 5\} \notin S$ olduğundan $S = \{1, 4\}$, $\tilde{S} = \{2, 3, 5, 6\}$ alınır.

Adım 3: $k=4$ çevrimi

$$L_3 = \min\{L_3, L_4 + L_{43}\} = \min\{7, 3 + 2\} = 5$$

$$L_5 = \min\{L_5, L_4 + L_{45}\} = \min\{\infty, 3 + 4\} = 7$$

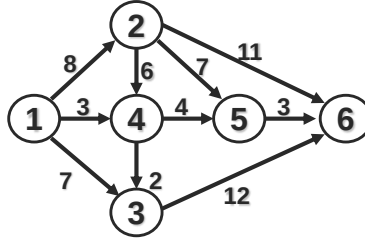
Yöneylem Araştırması II

190

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



Adım 2: $L_k = \min\{L_2, L_3, L_5, L_6\} = \{8, 5, 7, \infty\} = 5 \Rightarrow k = 3$. $\Gamma_k = \Gamma_3 = \{6\} \notin S$
 olduğundan $S = \{1, 4, 3\}$, $\tilde{S} = \{2, 5, 6\}$ alınır.

Adım 3: $k = 3$ çevrimi

$$L_6 = \min\{L_6, L_3 + L_{36}\} = \min\{\infty, 5 + 12\} = 17$$

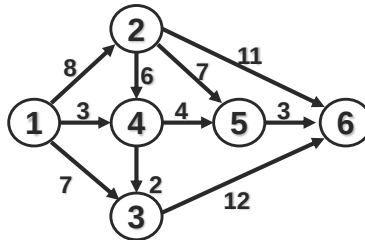
Yöneylem Araştırması II

191

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



Adım 2: $L_k = \min\{L_2, L_5, L_6\} = \{8, 7, 17\} = 7 \Rightarrow k = 5$. $\Gamma_k = \Gamma_5 = \{6\} \notin S$
 olduğundan $S = \{1, 4, 3, 5\}$, $\tilde{S} = \{2, 6\}$ alınır.

Adım 3: $k = 5$ çevrimi

$$L_6 = \min\{L_6, L_5 + L_{56}\} = \min\{17, 7 + 3\} = 10$$

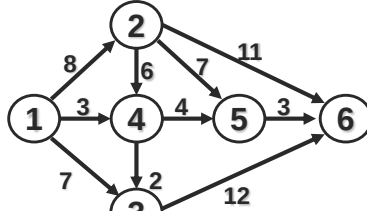
Yöneylem Araştırması II

192

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



Adım 2: $L_k = \min\{L_2, L_6\} = \{8, 10\} = 8 \Rightarrow k = 2$. $\Gamma_k = \Gamma_2 = \{4, 5, 6\}$

$4 \in S \Rightarrow$ Elenir | $5 \in S \Rightarrow$ Elenir | $6 \notin S \Rightarrow$ Hesaplanır

$S = \{1, 4, 3, 5, 2\}$, $\tilde{S} = \{6\}$ alınır.

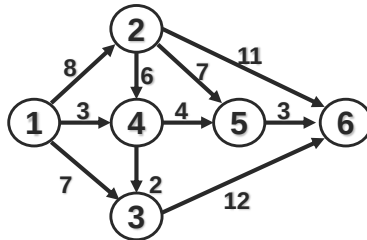
Adım 3: $k = 2$ çevrimi

$L_6 = \min\{L_6, L_2 + L_{26}\} = \min\{10, 8 + 11\} = 10$

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



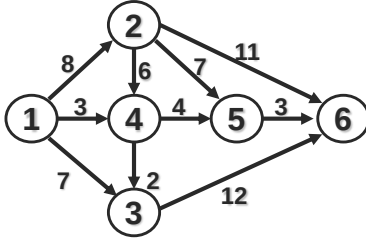
Adım 2: $L_k = \min\{L_6\} = \{10\} = 10 \Rightarrow k = 6$, $S = \{1, 4, 3, 5, 2, 6\}$, $\tilde{S} = \{\}$. Tüm Düşümler sabitlendiğinden işlem burada biter.

- 1 - 6 düğümleri arasındaki en kısa yolun uzunluğu 10 birimdir.

[En Kısa Yol Problemleri]



■ Dijkstra algoritması (Örnek)



* EN KISA YOLUN BULUNMASI

$$L_6 = 10 \begin{cases} \neq L_2 + L_{26} = 8 + 11 = 19 \\ \neq L_3 + L_{36} = 5 + 12 = 17 \\ = L_5 + L_{56} = 7 + 3 = 10 \end{cases} *$$

$$L_5 = 7 \begin{cases} \neq L_2 + L_{25} = 8 + 7 = 15 \\ = L_4 + L_{45} = 3 + 4 = 7 \end{cases} *$$

$$L_4 = 3 \begin{cases} = L_1 + L_{14} = 0 + 3 = 3 \end{cases} *$$

* EN KISA YOL (UZUNLUĞU 10 BİRİM)

1 → 4 → 5 → 6

Yöneylem Araştırması II

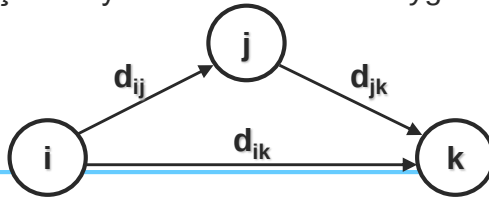
195

[En Kısa Yol Problemleri]



■ Floyd Algoritması

- Şebekedeki herhangi iki düğüm arasındaki en kısa yolu belirler. Bu nedenle Dijkstra algoritmasına göre daha geneldir.
- Algoritma n düğümlü şebekeyi $n \times n$ boyutlu kare matris olarak gösterir.
- Matrisin (i,j) elemanı i -nci düğümden j -nci düğümüne olan d_{ij} uzaklığını verir. i doğrudan j ye bağlıysa d_{ij} sonlu, bağlı değilse sonsuzdur.
- Aşağıdaki şebekede eğer $d_{ij} + d_{jk} < d_{ik}$ ise i den k ya gitmek için j den geçmek optimaldir. Floyd algoritması bu işlemi bütün şebekeye sistematik olarak uygular.



Yöneylem Araştırması II

196

[En Kısa Yol Problemleri]



- Floyd Algoritması
- **Adım 0:** Başlangıç “uzaklık matrisi” D_0 ve “düğüm sırası matrisi” S_0 ’ı oluştur. Her iki matrisde de diagonal elemanları bloke edildiklerini göstermek için (-) ile işaretle. $k=1$ al ve **Adım 1**’e git.

$$D_0 =$$

	1	2	...	j	...	n
1	-	d_{12}	...	d_{1j}	...	d_{1n}
2	d_{21}	-	...	d_{2j}	...	d_{2n}
...	...	...	...	...	...	...
i	d_{i1}	d_{i2}	...	d_{ij}	...	d_{in}
...	...	...	...	...	...	...
n	d_{n1}	d_{n2}	...	d_{nj}	...	-

$$S_0 =$$

	1	2	...	j	...	n
1	-	2	...	j	...	n
2	1	-	...	j	...	n
...	...	...	...	...	...	...
i	1	2	...	j	...	n
...	...	...	...	...	...	...
n	1	2	...	j	...	-

Yöneylem Araştırması II

197

[En Kısa Yol Problemleri]



- Floyd Algoritması
- **Adım k:** k satırı ve k sütununu anahtar satır ve anahtar sütun olarak belirle. Tüm i ve j ler için D_{k-1} deki her bir d_{ij} elemanı için eğer $d_{ik} + d_{kj} < d_{ij}$ ise;
 - D_{k-1} deki d_{ij} ‘yi $d_{ik} + d_{kj}$ ile değiştirerek D_k ‘yi oluştur.
 - S_{k-1} de S_{ij} yi k ile değiştirerek S_k ‘yi oluştur. $k=k+1$ olarak belirle ve **Adım k** ‘yi tekrarla.

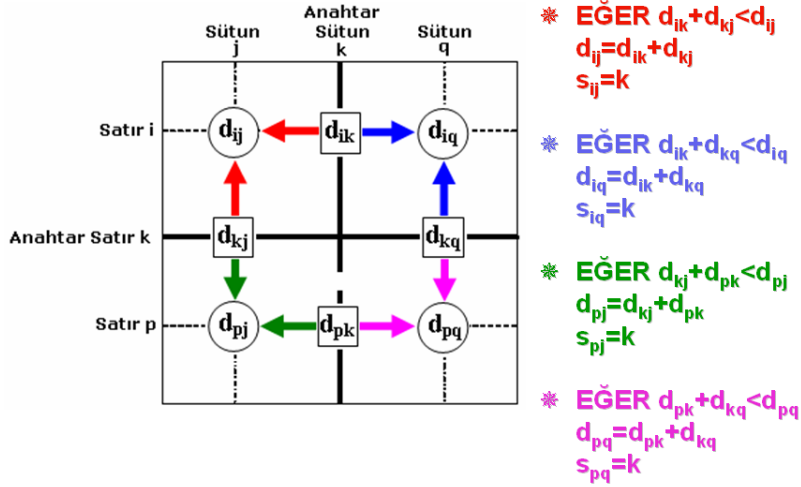
Yöneylem Araştırması II

198

[En Kısa Yol Problemleri]



■ Floyd Algoritması



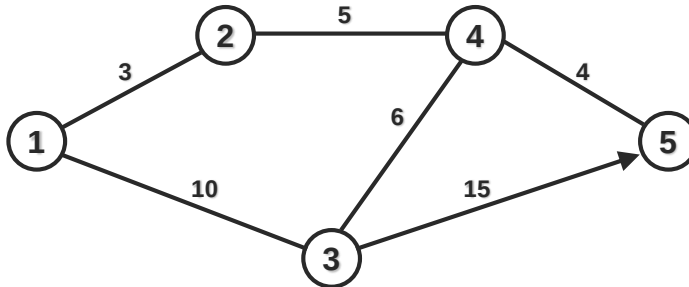
Yöneylem Araştırması II

199

[En Kısa Yol Problemleri]



- Floyd Algoritması (Örnek)
- Aşağıdaki şebekede dal ve okların üzerindeki rakamlar iki düğüm arasındaki mesafeyi **km** cinsinden ifade etmektedir. **Bütün düğümler** arasındaki en kısa yolu bulunuz.



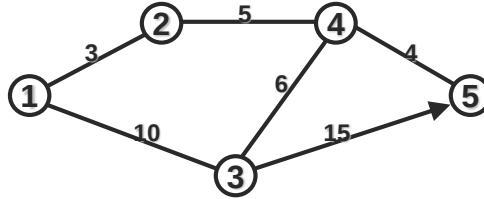
Yöneylem Araştırması II

200

[En Kısa Yol Problemleri]



- Floyd Algoritması (Örnek)
- Adım 0: D_0 matrisi $d_{53}=\infty$ hariç simetriktr. $k=1$ alınır.



$$D_0 =$$

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

$$S_0 =$$

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Yöneylem Araştırması II

201

[En Kısa Yol Problemleri]



- Floyd Algoritması (Örnek)
- Adım 1:
 - $k=1$ olduğundan 1nci satır anahtar satır, 1nci sütun anahtar sütundur.
 - d_{23} ve d_{32} geliştirilebilecek elemanlardır. D_1 ve S_1 matrislerini D_0 ve S_0 dan elde etmek için;
 - $d_{23}=d_{21}+d_{13}=3+10=13$ ve $S_{23}=k=1$
 - $d_{32}=d_{31}+d_{12}=10+3=13$ ve $S_{32}=k=1$

$$D_0 =$$

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

$$S_0 =$$

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Yöneylem Araştırması II

202

[En Kısa Yol Problemleri]



■ Floyd Algoritması (Örnek)

■ Adım 2:

- $k=2$ olduğundan 2nci satır **anahtar satır**, 2nci sütun **anahtar sütundur**.
- d_{14} ve d_{41} geliştirilebilecek elemanlardır. D_2 ve S_2 matrislerini D_1 ve S_1 den elde etmek için;
 - $d_{14}=d_{12}+d_{24}=3+5=8$ ve $S_{14}=k=2$
 - $d_{41}=d_{42}+d_{21}=5+3=8$ ve $S_{41}=k=2$

$k=3$ alınır.

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

$D_1 =$

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

$S_1 =$

Yöneylem Araştırması II

203

[En Kısa Yol Problemleri]



■ Floyd Algoritması (Örnek)

■ Adım 3:

- $k=3$ olduğundan 3ncü satır **anahtar satır**, 3ncü sütun **anahtar sütundur**.
- d_{15} ve d_{25} geliştirilebilecek elemanlardır. D_3 ve S_3 matrislerini D_2 ve S_2 den elde etmek için;
 - $d_{15}=d_{13}+d_{35}=10+15=25$ ve $S_{15}=k=3$
 - $d_{25}=d_{23}+d_{35}=13+15=28$ ve $S_{25}=k=3$

$k=4$ alınır.

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

$D_2 =$

	1	2	3	4	5
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

$S_2 =$

Yöneylem Araştırması II

204

[En Kısa Yol Problemleri]



■ Floyd Algoritması (Örnek)

■ Adım 4:

- $k=4$ olduğundan 4ncü satır **anahtar satır**, 4ncü sütun **anahtar sütundur**.
- $d_{15}, d_{23}, d_{25}, d_{32}, d_{35}, d_{51}, d_{52}$ ve d_{53} geliştirilebilecek elemanlardır.
- $k=5$ alınır.

	1	2	3	4	5
1	-	3	10	8	25
2	3	-	13	5	28
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

 $D_3 =$

	1	2	3	4	5
1	-	2	3	2	3
2	1	-	1	4	3
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

 $S_3 =$

Yöneylem Araştırması II

205

[En Kısa Yol Problemleri]



■ Floyd Algoritması (Örnek)

■ Adım 5:

- $k=5$ olduğundan 5ncü satır **anahtar satır**, 5ncü sütun **anahtar sütundur**.
- Geliştirilebilecek eleman yok.

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

 $D_4 =$

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

 $S_4 =$

Yöneylem Araştırması II

206

[En Kısa Yol Problemleri]



- Floyd Algoritması (Örnek)
- Sonuç Matrisleri:
- D_5 ve S_5 matrisleri şebekedeki her hangi iki düğüm arasındaki en kısa yolu belirlemek için gereken bilgiyi içerir.
- 1 düğümünden 5 düğümüne en kısa uzaklık $d_{15}=12$,
- 1 düğümünden 5 düğümüne en kısa yol;
 $1 \rightarrow 5$ ($S_{15}=4$) $\Rightarrow$
 $1 \rightarrow 4 \rightarrow 5$ ($S_{14}=2$) $\Rightarrow$
 $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ ($S_{12}=2, S_{24}=4, S_{45}=5$)

$$D_5 =$$

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

$$S_5 =$$

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

Yöneylem Araştırması II

207

[Maksimum Akış Problemleri]



- Bir çok sistem, her birinin kendileri üzerinden taşınan malzeme miktarını kısıtlayan **kapasiteye** sahip dallar içeren bir şebeke olarak modellenebilir.
- Bu durumlarda genellikle bir başlama noktasından (**kaynak**) bir bitiş noktasına (**batak**) maksimum miktarda akışın ulaştırılması istenir. Bu tip problemler **Maksimum Akış Problemleri** olarak adlandırılır.
- Problemin **DP** modeli verildikten sonra, çözümünde sıkça kullanılan **Düğüm Etiketleme Algoritması** açıklanacaktır.

Yöneylem Araştırması II

208

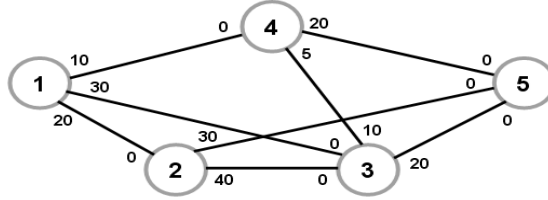
[Maksimum Akış Problemleri]



DP modeli

x_{ij} : i düğümünden j düğüme akış miktarı,

c_{ij} : i düğümünden j düğüme akış kapasitesi ((i,j) dalında i düğüme yakın yazılan kapasite)

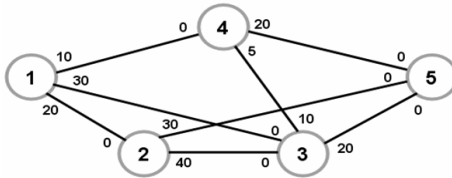


$$\begin{aligned} \text{Maks } Z &= \sum_k x_{1k} \\ \text{S.T. } x_{ij} &\leq c_{ij} && \text{Her } (i,j) \text{ dalı için} \\ \sum_i x_{ik} - \sum_j x_{kj} &= 0 && k = 2, \dots, N-1 \\ \sum_k x_{1k} &= \sum_k x_{kN} \\ x_{ij} &\geq 0 && \text{Her } (i,j) \text{ dalı için} \end{aligned}$$

Yöneylem Araştırması II

209

[Maksimum Akış Problemleri]



$$\max Z = x_{12} + x_{13} + x_{14}$$

S.T.

$$\begin{aligned} x_{12} &\leq 20 && \{(1,2) \text{ bağlantısı kapasite kısıtı}\} \\ x_{13} &\leq 30 && \{(1,3) \text{ "}\} \\ x_{14} &\leq 10 && \{(1,4) \text{ "}\} \\ x_{21} &\leq 0 && \{(2,1) \text{ "}\} \\ x_{23} &\leq 40 && \{(2,3) \text{ "}\} \\ x_{25} &\leq 30 && \{(2,5) \text{ "}\} \\ x_{31} &\leq 0 && \{(3,1) \text{ "}\} \\ x_{32} &\leq 0 && \{(3,2) \text{ "}\} \end{aligned}$$

$$x_{12} + x_{32} + x_{52} - x_{21} - x_{23} - x_{25} = 0$$

$$x_{13} + x_{23} + x_{43} + x_{53} - x_{31} - x_{32} - x_{34} - x_{35} = 0$$

$$x_{14} + x_{34} + x_{54} - x_{41} - x_{43} - x_{45} = 0$$

$$x_{12} + x_{13} + x_{14} - x_{25} - x_{35} - x_{45} = 0$$

$$x_{ij} \geq 0$$

2 düğümü için akış koruma kısıtı

3 düğümü için akış koruma kısıtı

4 düğümü için akış koruma kısıtı

1 düğümünden çıkan 5 düğüme giren akışlar toplamına eşit olsun

Negatif olmama koşulları

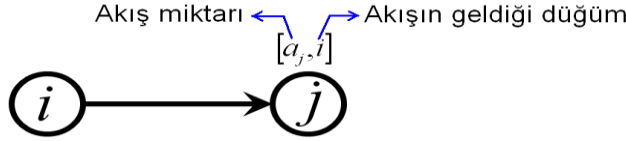
Optimal Çözüm

OBJECTIVE FUNC. VAL.	
1)	60.00000
VAR.	VALUE
X12	20.000000
X13	30.000000
X14	10.000000
X21	0.000000
X23	0.000000
X25	20.000000
X31	0.000000
X32	0.000000
X34	10.000000
X35	20.000000
X41	0.000000
X43	0.000000
X45	20.000000
X52	0.000000
X53	0.000000
X54	0.000000

[Maksimum Akış Problemleri]



Etiket yapısı



Algoritmanın Adımları

- Adım 1** $a_1 = \infty$ olmak üzere 1nci düğümün (kaynak) etiketini $[\infty, -]$ olarak ata. $i = 1$ al ve **Adım 2**'ye git.
- Adım 2** $S_i = \left\{ \begin{array}{l} i \text{ düğümünden direkt ulaşılabilen, etiketlenmemiş,} \\ \text{pozitif kapasiteli düğümlerin kümesi} \end{array} \right\}$ olarak belirle. $S_i = \{ \}$ ise **Adım 4**'e, değilse **Adım 3**'e git.
- Adım 3** C_{ij} , (i, j) arkının kapasitesi olmak üzere $a_k = C_{ik} = \max_{j \in S_i} \{C_{ij}\}$ formülasyonu ile belirlenen k düğümünü $[a_k, i]$ olarak etiketle. Son düğüm etiketlenmişse, yani çıkış yolu bulunmuşsa **Adım 5**'e, aksi halde $i = k$ al ve **Adım 2**'ye git.

Yöneylem Araştırması II

211

[Maksimum Akış Problemleri]



Algoritmanın Adımları

- Adım 4** $i = 1$ ise başka bir çıkış olası değildir, **Adım 6**'ya git. Aksi halde $r = \{i \text{ düğümünde bir önce etiketlenen düğüm}\}$ olmak üzere i düğümünü r 'ye komşu olan düğümler kümesinden çıkar, $i = r$ al ve **Adım 2**'ye git.
- Adım 5** $N_p = \{1, k_1, k_2, \dots, n\}$, 1. düğümden n . düğüme p . çıkış yolunun düğümleri olarak tanımla. Daha sonra yol boyunca maksimum akışı $F_p = \min \{a_1, a_{k_1}, a_{k_2}, \dots, a_n\}$ olarak tanımla. Çıkış yolu boyunca her dalın kalan kapasitesini akış yönünde F_p kadar azalt, akışın tersi yönünde ise F_p kadar artır. Etiketleri sil, $i = 1$ al ve **Adım 2**'ye git.
- Adım 6** m adet çıkış yolu belirlenmişse şebekenin maksimum akışı $F = F_1 + F_2 + \dots + F_m$ olarak hesapla. Şebekedeki dalların başlangıç ve son kalan kapasiteleri arasındaki farktan dalların optimum akış miktarlarını hesapla.

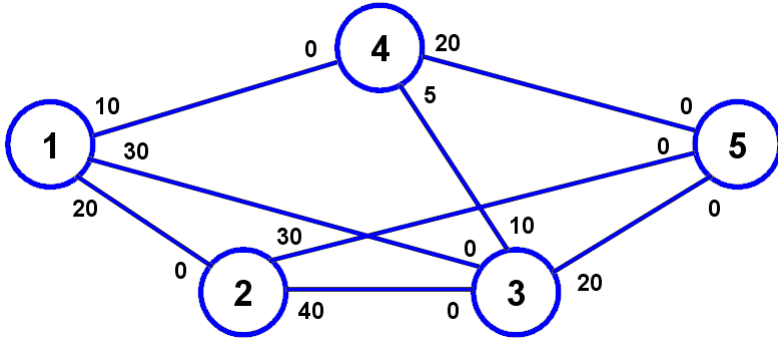
Yöneylem Araştırması II

212

[Maksimum Akış Problemleri]



- Örnek
- Aşağıdaki şebekenin 1 düğümünden 5 düğümüne olan maksimum akış miktarını belirleyin.



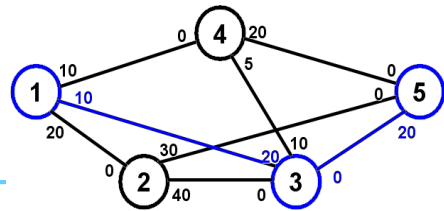
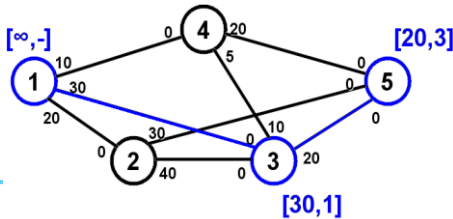
Yöneylem Araştırması II

213

[Maksimum Akış Problemleri (İterasyon 1)]



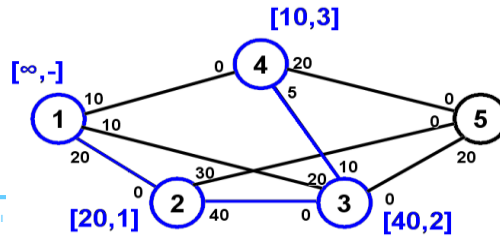
- Adım 1** $a_1 = \infty$ ata, 1. düğümü $[\infty, -]$ ile etiketle, $i = 1$ al.
- Adım 2** $S_1 = \{2, 3, 4\} \neq \emptyset$
- Adım 3** $a_k = C_{1k} = \max_{j \in S_1} \{C_{12}, C_{13}, C_{14}\} = \max\{20, 30, 10\} = 30 = C_{13} \Rightarrow k = 3$ ve $a_k = a_3 = C_{13} = 30$. 3. düğümü $[30, 1]$ ile etiketle, $i = k = 3$ al ve **Adım 2**'ye git
- Adım 2** $S_3 = \{4, 5\} \neq \emptyset$ (1 düğümü etiketli, 2. düğümüne giden kapasite sıfır)
- Adım 3** $a_k = C_{3k} = \max_{j \in S_3} \{C_{34}, C_{35}\} = \max\{10, 20\} = 20 = C_{35} \Rightarrow k = 5$ ve $a_k = a_5 = C_{35} = 20$. 5. düğümü $[20, 3]$ ile etiketle, $k = 5$ düğümü çıkış düğümü olduğundan **Adım 5**'ye git
- Adım 5** $N_1 = \{1, 3, 5\}$, $F_1 = \min\{a_1, a_3, a_5\} = \min\{\infty, 30, 20\} = 20$. Çıkış yolu boyunca kalan kapasiteler
- $(C_{13}, C_{31}) = (30 - 20, 0 + 20) = (10, 20)$ ve
 - $(C_{35}, C_{53}) = (20 - 20, 0 + 20) = (0, 20)$



[Maksimum Akış Problemleri (İterasyon 2)]



- Adım 1** $a_1 = \infty$ ata, 1. düğümü $[\infty, -]$ ile etiketle, $i = 1$ al.
- Adım 2** $S_1 = \{2, 3, 4\} \neq \{\}$
- Adım 3** $a_k = C_{1k} = \max_{j \in S_1} \{C_{12}, C_{13}, C_{14}\} = \max\{20, 10, 10\} = 20 = C_{12} \Rightarrow k = 2$ ve
 $a_k = a_2 = C_{12} = 20$. 2. düğümü $[20, 1]$ ile etiketle, $i = k = 2$ al ve
Adım 2 'ye git
- Adım 2** $S_2 = \{3, 5\} \neq \{\}$ (1 düğümü etiketli)
- Adım 3** $a_k = C_{2k} = \max_{j \in S_2} \{C_{23}, C_{25}\} = \max\{40, 30\} = 40 = C_{23} \Rightarrow k = 3$ ve
 $a_k = a_3 = C_{23} = 40$. 3. düğümü $[40, 2]$ ile etiketle, $i = k = 3$ al ve
Adım 2 'ye git
- Adım 2** $S_3 = \{4\} \neq \{\}$ (1 ve 2 düğümleri etiketli, 5 düğümüne giden kap. sıfır)
- Adım 3** $a_k = C_{3k} = \max_{j \in S_3} \{C_{34}\} = \max\{10\} = 10 = C_{34} \Rightarrow k = 4$ ve
 $a_k = a_4 = C_{34} = 10$. 4. düğümü $[10, 3]$ ile etiketle, $i = k = 4$ al ve
Adım 2 'ye git



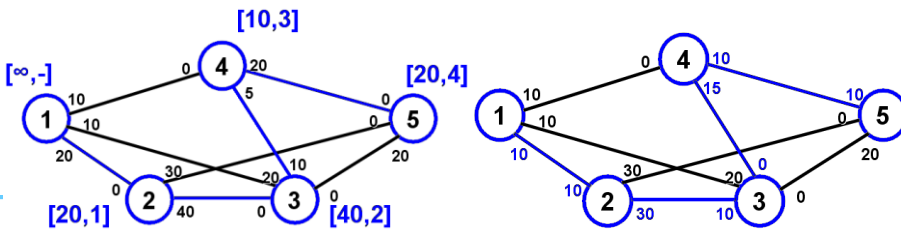
Yöneylem Araştırması

215

[Maksimum Akış Problemleri (İterasyon 2)]



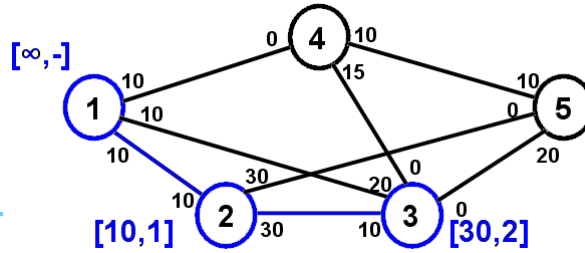
- Adım 2** $S_4 = \{5\} \neq \{\}$ (1 ve 3 düğümleri etiketli)
- Adım 3** $a_k = C_{4k} = \max_{j \in S_4} \{C_{45}\} = \max\{20\} = 20 = C_{45} \Rightarrow k = 5$ ve
 $a_k = a_5 = C_{45} = 20$. 5. düğümü $[20, 4]$ ile etiketle, $k = 5$ düğümü çıkış düğümü olduğundan **Adım 5** 'ye git
- Adım 5** $N_2 = \{1, 2, 3, 4, 5\}, F_2 = \min\{a_1, a_2, a_3, a_4, a_5\} = \min\{\infty, 20, 40, 10, 20\} = 10$.
Çıkış yolu boyunca kalan kapasiteler
- $(C_{12}, C_{21}) = (20 - 10, 0 + 10) = (10, 10)$,
 - $(C_{23}, C_{32}) = (40 - 10, 0 + 10) = (30, 10)$,
 - $(C_{34}, C_{43}) = (10 - 10, 5 + 10) = (0, 15)$ ve
 - $(C_{45}, C_{54}) = (20 - 10, 0 + 10) = (10, 10)$



[Maksimum Akış Problemleri (İterasyon 3)]



- Adım 1** $a_1 = \infty$ ata, 1. düğümü $[\infty, -]$ ile etiketle, $i = 1$ al.
- Adım 2** $S_1 = \{2, 3, 4\} \neq \{\}$
- Adım 3** $a_k = C_{1k} = \max_{j \in S_1} \{C_{12}, C_{13}, C_{14}\} = \max\{10, 10, 10\} = 10 = C_{12} \Rightarrow k = 2$ (Keyfi seçim yapıldı) ve $a_k = a_2 = C_{12} = 10$. 2. düğümü $[10, 1]$ ile etiketle, $i = k = 2$ al ve **Adım 2** 'ye git
- Adım 2** $S_2 = \{3, 5\} \neq \{\}$ (1 düğümü etiketli)
- Adım 3** $a_k = C_{2k} = \max_{j \in S_2} \{C_{23}, C_{25}\} = \max\{30, 30\} = 30 = C_{23} \Rightarrow k = 3$ (Keyfi seçim yapıldı) ve $a_k = a_3 = C_{23} = 30$. 3. düğümü $[30, 2]$ ile etiketle, $i = k = 3$ al ve **Adım 2** 'ye git
- Adım 2** $S_3 = \{\}$ (1 ve 2 düğümleri etiketli, 4 ve 5 düğümlerine giden kapasiteler sıfır). Geriye dönüş için **Adım 4** 'e git.



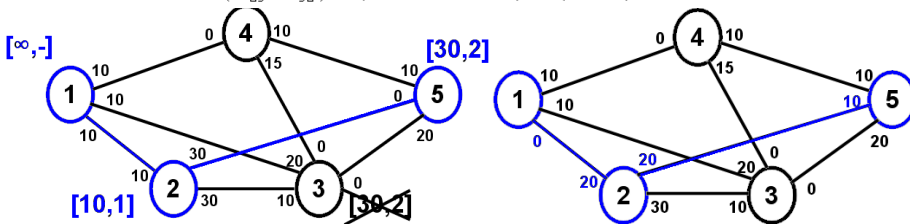
Yöneylem Araştırması II

217

[Maksimum Akış Problemleri (İterasyon 3)]



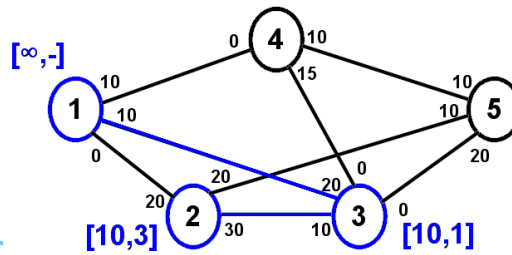
- Adım 4** 3. düğümdeki $[30, 2]$ etiketi hemen önceki düğümü $r = 2$ olarak verecektir. 3. düğümü göz ardı etmek için çarpı koyarak çıkar.
- $i = k = 2$ al ve **Adım 2** 'ye git.
- Adım 2** $S_2 = \{5\} \neq \{\}$ (1 ve 3 düğümleri etiketli)
- Adım 3** $a_k = C_{2k} = \max_{j \in S_2} \{C_{25}\} = \max\{30\} = 30 = C_{25} \Rightarrow k = 5$ ve $a_k = a_5 = C_{25} = 30$. 5. düğümü $[30, 2]$ ile etiketle, $k = 5$ düğümü çıkış düğümü olduğundan **Adım 5** 'ye git
- Adım 5** $N_3 = \{1, 2, 5\}$, $F_3 = \min\{a_1, a_2, a_5\} = \min\{\infty, 10, 30\} = 10$. Çıkış yolu boyunca kalan kapasiteler
- $(C_{12}, C_{21}) = (10 - 10, 10 + 10) = (0, 20)$ ve
 - $(C_{25}, C_{52}) = (30 - 10, 0 + 10) = (20, 10)$



[Maksimum Akış Problemleri (İterasyon 4)]



- Adım 1** $a_1 = \infty$ ata, 1. düğümü $[\infty, -]$ ile etiketle, $i = 1$ al.
- Adım 2** $S_1 = \{3, 4\} \neq \{\}$ (2. düğüme giden kapasite sıfır)
- Adım 3** $a_k = C_{1k} = \max_{j \in S_1} \{C_{13}, C_{14}\} = \max\{10, 10\} = 10 = C_{13} \Rightarrow k = 3$ (Kayfi seçim yapıldı) ve $a_k = a_3 = C_{13} = 10$. 3. düğümü $[10, 1]$ ile etiketle, $i = k = 3$ al ve **Adım 2** 'ye git
- Adım 2** $S_3 = \{2\} \neq \{\}$ (1 düğümü etiketli, 4 ve 5. düğümlere giden kap. sıfır)
- Adım 3** $a_k = C_{3k} = \max_{j \in S_3} \{C_{32}\} = \max\{10\} = 10 = C_{32} \Rightarrow k = 2$ ve $a_k = a_2 = C_{32} = 10$. 2. düğümü $[10, 3]$ ile etiketle, $i = k = 2$ al ve **Adım 2** 'ye git



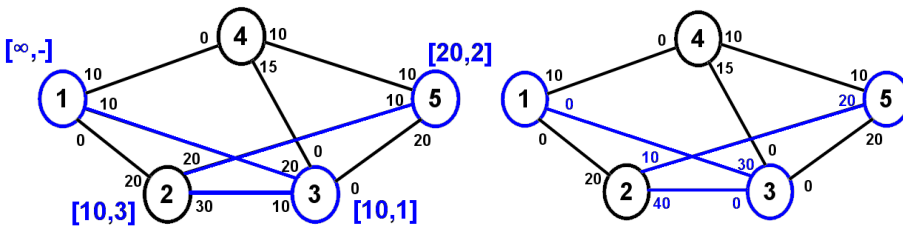
Yöneylem Araştırması II

219

[Maksimum Akış Problemleri (İterasyon 4)]



- Adım 2** $S_2 = \{5\} \neq \{\}$ (1 ve 3 düğümleri etiketli)
- Adım 3** $a_k = C_{2k} = \max_{j \in S_2} \{C_{25}\} = \max\{20\} = 20 = C_{25} \Rightarrow k = 5$ ve $a_k = a_5 = C_{25} = 20$. 5. düğümü $[20, 2]$ ile etiketle, $k = 5$ düğümü çıkış düğümü olduğundan **Adım 5** 'ye git
- Adım 5** $N_4 = \{1, 3, 2, 5\}$, $F_4 = \min\{a_1, a_3, a_2, a_5\} = \min\{\infty, 10, 10, 20\} = 10$. Çıkış yolu boyunca kalan kapasiteler
- $(C_{13}, C_{31}) = (10 - 10, 20 + 10) = (0, 30)$,
 - $(C_{32}, C_{23}) = (10 - 10, 30 + 10) = (0, 40)$ ve
 - $(C_{25}, C_{52}) = (20 - 10, 10 + 10) = (10, 20)$



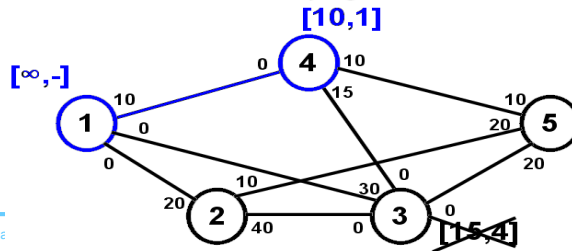
Yöneylem Araştırması II

220

[Maksimum Akış Problemleri (İterasyon 5)]



- Adım 1** $a_1 = \infty$ ata, 1. düğümü $[\infty, -]$ ile etiketle, $i = 1$ al.
- Adım 2** $S_1 = \{4\} \neq \{\}$ (2 ve 3 düğümlerine giden kapasite sıfır)
- Adım 3** $a_k = C_{1k} = \max_{j \in S_1} \{C_{14}\} = \max_{j \in S_1} \{10\} = 10 = C_{14} \Rightarrow k = 4$ ve $a_k = a_4 = C_{14} = 10$. 4. düğümü $[10, 1]$ ile etiketle, $i = k = 4$ al ve **Adım 2**'ye git
- Adım 2** $S_4 = \{3, 5\} \neq \{\}$ (1 düğümü etiketli)
- Adım 3** $a_k = C_{4k} = \max_{j \in S_4} \{C_{43}, C_{45}\} = \max_{j \in S_4} \{15, 10\} = 15 = C_{43} \Rightarrow k = 3$ ve $a_k = a_3 = C_{43} = 15$. 3. düğümü $[15, 4]$ ile etiketle, $i = k = 3$ al ve **Adım 2**'ye git
- Adım 2** $S_3 = \{\}$ (1 ve 4 düğümleri etiketli, 2 ve 5 düğümlerine giden kapasiteler sıfır). Geriye dönüş için **Adım 4**'e git.
- Adım 4** 3. düğümdeki $[15, 4]$ etiketi hemen önceki düğümü $r = 4$ olarak verecektir. 3. düğümü göz ardı etmek için çarpı koyarak çıkar. $i = k = 4$ al ve **Adım 2**'ye git.



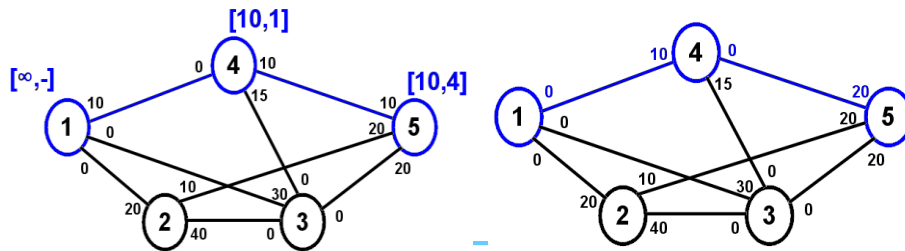
Yöneylem Araştırma

221

[Maksimum Akış Problemleri (İterasyon 5)]



- Adım 2** $S_4 = \{5\} \neq \{\}$ (1 ve 3 düğümleri etiketli)
- Adım 3** $a_k = C_{4k} = \max_{j \in S_4} \{C_{45}\} = \max_{j \in S_4} \{10\} = 10 = C_{45} \Rightarrow k = 5$ ve $a_k = a_5 = C_{45} = 10$. 5. düğümü $[10, 4]$ ile etiketle, $k = 5$ düğümü çıkış düğümü olduğundan **Adım 5**'ye git
- Adım 5** $N_5 = \{1, 4, 5\}$, $F_5 = \min\{a_1, a_4, a_5\} = \min\{\infty, 10, 10\} = 10$. Çıkış yolu boyunca kalan kapasiteler
- $(C_{14}, C_{41}) = (10 - 10, 0 + 10) = (0, 10)$ ve
 - $(C_{45}, C_{54}) = (10 - 10, 10 + 10) = (0, 20)$



Yöneylem Araştırması II

222

[Maksimum Akış Problemleri]



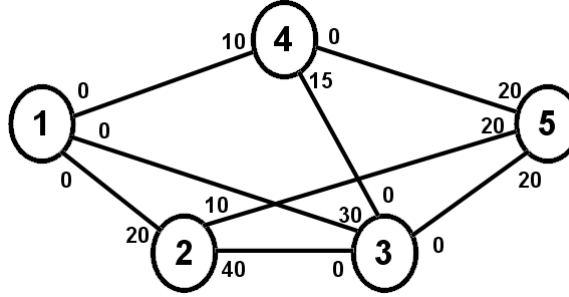
İterasyon 6

1. düğümden çıkan tüm bağlantıların kalan kapasiteleri sıfır olduğundan başka olası bir çıkış yoktur. **Adım 6**'ya gidilerek çözüm elde edilir.

Adım 6 : Şebekedeki maksimum akış

$$F = F_1 + F_2 + \dots + F_5 = 20 + 10 + 10 + 10 + 10 = 60 \text{ olur.}$$

dallardaki akışlar aşağıdaki tabloda verilmiştir.



Yöneylem Araştırması II

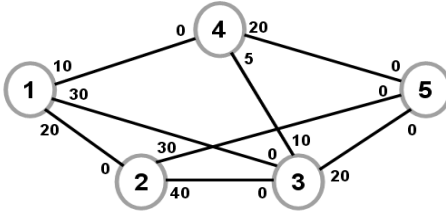
223

[Maksimum Akış Problemleri]

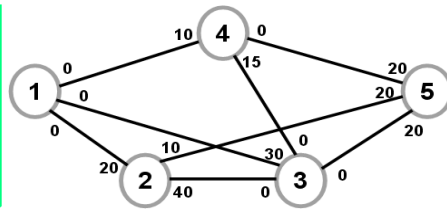


Dallardaki Akışlar

Şebekenin orijinal hali



Şebekenin son hali



Bağlantı	Akış Farkları	Akış Miktarı	Yön
(1,2)	$(20,0) - (0,20) = (20,-20)$	20	1→2
(1,3)	$(30,0) - (0,30) = (30,-30)$	30	1→3
(1,4)	$(10,0) - (0,10) = (10,-10)$	10	1→4
(2,3)	$(40,0) - (40,0) = (0,0)$	0	-
(2,5)	$(30,0) - (10,20) = (20,-20)$	20	2→5
(3,4)	$(10,5) - (0,15) = (10,-10)$	10	3→4
(3,5)	$(20,0) - (0,20) = (20,-20)$	20	3→5
(4,5)	$(20,0) - (0,20) = (20,-20)$	20	4→5

Yöneylem Araştırması II

224

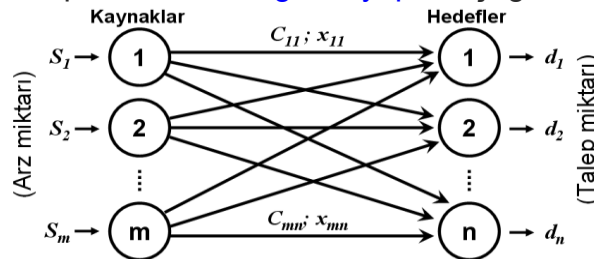
BÖLÜM V: Ulaştırma Modelleri

- Genel Kavramlar ve Tanım
- Ulaştırma Problemleri
- Başlangıç Çözümünün Elde Edilmesi
- Ulaştırma Simpleks Metodu
- Ulaştırma Problemlerinde Duyarlılık Analizi
- Atama Problemleri

[Genel Kavramlar ve Tanım]



- Ulaştırma problemleri, birden fazla **kaynak**tan birden fazla **hedefe** yapılan **ulaştırma planı**nın belirlenmesi ile ilgilidir.
- Modelin amacı, **toplam ulaştırma maliyetini minimum yapacak** şekilde, her kaynaktan her hedefe gönderilmesi gereken **miktarı** belirlemektir.
- Problemin **verileri** şunları kapsar:
 - Her kaynaktaki **arz miktarı** (kapasite) ve her hedefteki **talep miktarı**.
 - Her kaynaktan her hedefe **bir birim malı ulaştırma maliyeti**.
- Ulaştırma problemlerinin **genel yapısı** aşağıda verilmiştir.



[Ulaştırma Problemleri]



- Örnek (Akçelik A.Ş. Örneği)
- Buz dolabı imal eden Akçelik A.Ş.'in 5 şehirde satış merkezi ve 3 şehirde de fabrikaları bulunmaktadır. Satış merkezlerinin buzdolabı siparişleri ve fabrikalarının üretim kapasiteleri aşağıdaki şekildedir.

	Samsun	Ankara	İstanbul	İzmir	Mersin	Toplam
	(1)	(2)	(3)	(4)	(5)	
Talep	45	52	38	62	163	360

	Bursa	İzmit	Balıkesir	Toplam
	(a)	(b)	(c)	
Kapasite	120	110	130	360

Yöneylem Araştırması II

227

[Ulaştırma Problemleri]



- Örnek (Akçelik A.Ş. Örneği)
- Her bir fabrikadan her bir satış merkezine gönderilen bir buzdolabının ulaştırma maliyetleri tabloda gösterildiği gibidir.

Satış M. \ Fab.	1	2	3	4	5
a	37	23	28	19	35
b	36	25	20	26	29
c	26	19	17	27	33

- Hangi fabrikadan hangi satış merkezine ne miktar buzdolabı gönderelim ki toplam ulaştırma maliyeti en küçük olsun.

Yöneylem Araştırması II

228

[Ulaştırma Problemleri]



- Örnek (Akçelik A.Ş. Örneği - DP Modeli)
- X_{ij} : i fabrikasından ($i=a, b, c$) j satış merkezine gönderilecek buzdolabı sayısı

Ulaştırma modeli tablosu

Satış M. Fab.	1	2	3	4	5	Üretim
a	37 X_{a1}	23 X_{a2}	28 X_{a3}	19 X_{a4}	35 X_{a5}	120
b	36 X_{b1}	25 X_{b2}	20 X_{b3}	26 X_{b4}	29 X_{b5}	110
c	26 X_{c1}	19 X_{c2}	17 X_{c3}	27 X_{c4}	33 X_{c5}	130
Talep	45	52	38	62	163	360

Yöneylem Araştırması II

229

[Ulaştırma Problemleri]



- Örnek (Akçelik A.Ş. Örneği - DP Modeli)

Satış M. Fab.	1	2	3	4	5	Üretim
a	37 X_{a1}	23 X_{a2}	28 X_{a3}	19 X_{a4}	35 X_{a5}	120
b	36 X_{b1}	25 X_{b2}	20 X_{b3}	26 X_{b4}	29 X_{b5}	110
c	26 X_{c1}	19 X_{c2}	17 X_{c3}	27 X_{c4}	33 X_{c5}	130
Talep	45	52	38	62	163	360

- Modelin Amaç Fonksiyonu

$$\begin{aligned} \text{Min } Z = & 37X_{a1} + 23X_{a2} + 28X_{a3} + 19X_{a4} + 35X_{a5} + \\ & 36X_{b1} + 25X_{b2} + 20X_{b3} + 26X_{b4} + 29X_{b5} + \\ & 26X_{c1} + 19X_{c2} + 17X_{c3} + 27X_{c4} + 33X_{c5} \end{aligned}$$

Yöneylem Araştırması II

230

[Ulaştırma Problemleri]



■ Örnek (Akçelik A.Ş. Örneği - DP Modeli)

Satış M.	1	2	3	4	5	Üretim
Fab.						
a	³⁷ X_{a1}	²³ X_{a2}	²⁸ X_{a3}	¹⁹ X_{a4}	³⁵ X_{a5}	120
b	³⁶ X_{b1}	²⁵ X_{b2}	²⁰ X_{b3}	²⁶ X_{b4}	²⁹ X_{b5}	110
c	²⁶ X_{c1}	¹⁹ X_{c2}	¹⁷ X_{c3}	²⁷ X_{c4}	³³ X_{c5}	130
Talep	45	52	38	62	163	360

■ Modelin Talep Kısıtları

$$X_{a1} + X_{b1} + X_{c1} \geq 45$$

$$X_{a2} + X_{b2} + X_{c2} \geq 52$$

$$X_{a3} + X_{b3} + X_{c3} \geq 38$$

$$X_{a4} + X_{b4} + X_{c4} \geq 62$$

$$X_{a5} + X_{b5} + X_{c5} \geq 163$$

[Ulaştırma Problemleri]



■ Örnek (Akçelik A.Ş. Örneği - DP Modeli)

Satış M.	1	2	3	4	5	Üretim
Fab.						
a	³⁷ X_{a1}	²³ X_{a2}	²⁸ X_{a3}	¹⁹ X_{a4}	³⁵ X_{a5}	120
b	³⁶ X_{b1}	²⁵ X_{b2}	²⁰ X_{b3}	²⁶ X_{b4}	²⁹ X_{b5}	110
c	²⁶ X_{c1}	¹⁹ X_{c2}	¹⁷ X_{c3}	²⁷ X_{c4}	³³ X_{c5}	130
Talep	45	52	38	62	163	360

■ Modelin Kapasite Kısıtları

$$X_{a1} + X_{a2} + X_{a3} + X_{a4} + X_{a5} \leq 120$$

$$X_{b1} + X_{b2} + X_{b3} + X_{b4} + X_{b5} \leq 110$$

$$X_{c1} + X_{c2} + X_{c3} + X_{c4} + X_{c5} \leq 130$$

[Ulaştırma Problemleri]



■ Örnek (Akçelik A.Ş. Örneği - DP Modeli)

Satış M.	1	2	3	4	5	Üretim
Fab.						
a	³⁷ X_{a1}	²³ X_{a2}	²⁸ X_{a3}	¹⁹ X_{a4}	³⁵ X_{a5}	120
b	³⁶ X_{b1}	²⁵ X_{b2}	²⁰ X_{b3}	²⁶ X_{b4}	²⁹ X_{b5}	110
c	²⁶ X_{c1}	¹⁹ X_{c2}	¹⁷ X_{c3}	²⁷ X_{c4}	³³ X_{c5}	130
Talep	45	52	38	62	163	360

■ Modelin Negatif Olmama Kısıtları

$$X_{ij} \geq 0 \quad (i=a,b,c; j=1,2,3,4,5)$$

[Ulaştırma Problemleri]



■ Örnek (Akçelik A.Ş. Örneği - DP Modeli)

$$\text{Min } Z = 37X_{a1} + 23X_{a2} + 28X_{a3} + 19X_{a4} + 35X_{a5} + \\ 36X_{b1} + 25X_{b2} + 20X_{b3} + 26X_{b4} + 29X_{b5} + \\ 26X_{c1} + 19X_{c2} + 17X_{c3} + 27X_{c4} + 33X_{c5}$$

$$\text{S.T. } \begin{aligned} X_{a1} + X_{b1} + X_{c1} &\geq 45 \\ X_{a2} + X_{b2} + X_{c2} &\geq 52 \\ X_{a3} + X_{b3} + X_{c3} &\geq 38 \\ X_{a4} + X_{b4} + X_{c4} &\geq 62 \\ X_{a5} + X_{b5} + X_{c5} &\geq 163 \\ X_{a1} + X_{a2} + X_{a3} + X_{a4} + X_{a5} &\leq 120 \\ X_{b1} + X_{b2} + X_{b3} + X_{b4} + X_{b5} &\leq 110 \\ X_{c1} + X_{c2} + X_{c3} + X_{c4} + X_{c5} &\leq 130 \\ X_{ij} &\geq 0 \quad (i=a,b,c; j=1,2,3,4,5) \end{aligned}$$

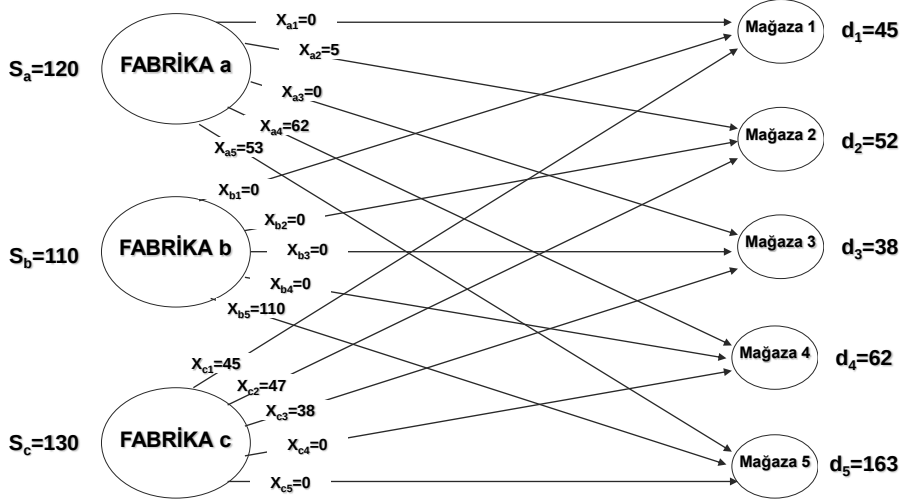
■ Optimal çözümü

$$Z=9047, X_{a2}=5, X_{a4}=62, X_{a5}=53, X_{b5}=110, X_{c1}=45, X_{c2}=47, X_{c3}=38 \\ X_{a1}=X_{a3}=X_{b1}=X_{b2}=X_{b3}=X_{b4}=X_{c4}=X_{c5}=0$$

[Ulaştırma Problemleri]



■ Örnek (Akçelik A.Ş. Örneği - DP Modeli)



Yöneylem Araştırması II

235

[Ulaştırma Problemleri]



- m adet kaynak n adet hedeften oluşan ulaştırma problemlerinin Doğrusal Programlama Modelini aşağıdaki şekilde genelleştirmek mümkündür.

S_i : i kaynağının arz miktarı (kapasitesi). ($i = 1, 2, \dots, m$)

d_j : j hedefinin talep miktarı. ($j = 1, 2, \dots, n$)

c_{ij} : i kaynağından j hedefine gönderilen malların birim ulaştırma maliyeti.

x_{ij} : i kaynağından j hedefine gönderilen mal miktarı.

$$\text{Min } z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq S_i \quad (i = 1, 2, \dots, m) \text{ (arz kısıtları)} \\ & \sum_{i=1}^m x_{ij} \geq d_j \quad (j = 1, 2, \dots, n) \text{ (talep kısıtları)} \\ & x_{ij} \geq 0 \quad (i = 1, 2, \dots, m, j = 1, 2, \dots, n) \end{aligned}$$

Yöneylem Araştırması II

236

[Ulaştırma Problemleri]



■ Dengeli Ulaştırma Problemi

- Toplam arzın toplam talebe **eşit olduğu** (yani $\sum_{i=1}^m S_i = \sum_{j=1}^n d_j$) ulaştırma problemidir.
- Dengeli ulaştırma probleminde bütün kısıtlar **bağlayıcı olmalıdır**, yani arz ve **talep kısıtları eşitlik şeklinde yazılır**. Bu problemlerin **genel yapısı** aşağıdaki gibidir.

$$\begin{aligned}
 \text{Min } z &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\
 \text{s.t. } \quad &\sum_{j=1}^n x_{ij} = S_i \quad (i = 1, 2, \dots, m) \text{ (arz kısıtları)} \\
 &\sum_{i=1}^m x_{ij} = d_j \quad (j = 1, 2, \dots, n) \text{ (talep kısıtları)} \\
 &x_{ij} \geq 0 \quad (i = 1, 2, \dots, m; j = 1, 2, \dots, n)
 \end{aligned}$$

Yöneylem Araştırması II

237

[Ulaştırma Problemleri]



- **Ulaştırma tablosu**
- Ulaştırma problemlerinin **gösteriminde** ve **çözümünde** aşağıda verilen ulaştırma tablosu kullanılır.
- Ulaştırma tablosu çözümün elde edilmesinde ve işlemlerin takibinde **kolaylık sağlar**.

Kaynak 1	c_{11}	c_{12}	$\dots$	c_{1n}	s_1
	x_{11}	x_{12}	$\dots$	x_{1n}	
Kaynak 2	c_{21}	c_{22}	$\dots$	c_{2n}	s_2
	x_{21}	x_{22}	$\dots$	x_{2n}	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Kaynak m	c_{m1}	c_{m2}	$\dots$	c_{mn}	s_m
	x_{m1}	x_{m2}	$\dots$	x_{mn}	
Talep (d_j)	d_1	d_2	$\dots$	d_n	

Yöneylem Araştırması II

238

[Ulaştırma Problemleri]



- Ulaştırma Problemlerinin Dengelenmesi
- Ulaştırma problemlerinin dengelenmesinde **iki durum** söz konusudur;
- 1. **Toplam arz > toplam talep ise**
 - a) **Sahte** (hayali) bir hedef (talep noktası) oluşturulur.
 - b) Sahte **hedefin talebi arz fazlasına eşittir**.
 - c) Sahte hedefe gönderilecek ürünler gerçekte gönderilmeyeceği için, her bir kaynaktan sahte hedefe olan **birim ulaştırma maliyeti sıfır** alınır.
 - d) Sahte hedefe gönderilen ürünler, **kullanılmayan kaynak miktarını** (arz fazlasını) **gösterir**.
- 2. **Toplam arz < toplam talep ise**
 - a) **Sahte** (hayali) bir kaynak (arz noktası) oluşturulur.
 - b) Sahte **kaynağın kapasitesi talep fazlasına eşittir**.
 - c) Sahte kaynaktan talep noktalarına olan **birim ulaştırma maliyeti**, karşılanmayan her bir birim talep için **belirlenen cezaya eşittir** (ceza saptanmayan durumlarda bu değer **sıfır** alınabilir).
 - d) Sahte kaynaktan gönderilen ürünler **karşılanmayan talep miktarını** (talep fazlasını) **gösterir**.

[Ulaştırma Problemleri]



- Ulaştırma Problemlerinin Dengelenmesi (Toplam arz > Toplam talep)
- Örneğin, **Akçelik A.Ş.** probleminde mağaza-1'e ait talep miktarının **45** adet buzdolabı değil de **30** buzdolabı olduğunu varsayalım. Buna göre;
Toplam arz = 360 > Toplam talep = 345
 olacağından, **15** adet buzdolabı arz fazlası vardır. O halde **sahte bir hedef** (mağaza-6) oluşturulur ve bütün fabrikalardan bu mağazaya birim ulaştırma maliyeti **sıfır** olarak belirlenir.
- ($C_{i6}=0, i=1, 2, 3$). bu modelin **optimal ulaştırma tablosu**:

Satış M. Fab.	1	2	3	4	5	6	Üretim
a	37	23	28	19	35	0	120
b	36	25	20	26	29	0	110
c	26	19	17	27	33	0	130
Talep	30	52	38	62	163	15	360

- Gerçekte **mağaza-6 olmadığından**, fabrika a 'ın **15** adet buzdolabı kapasitesi **kullanılmayacaktır**.

[Ulaştırma Problemleri]



- Ulaştırma Problemlerinin Dengelenmesi (Toplam arz < Toplam talep)
- Örneğin, Akçelik A.Ş. probleminde Fabrika a'ya ait arz miktarının 120 adet değil de 100 adet olduğunu ve mağazaların talebinin karşılanamaması durumunda ceza maliyetlerinin sırası ile 16, 17, 18, 19, 20 olacağını varsayalım. bu durumda;
 $\text{Toplam arz} = 340 < \text{Toplam talep} = 360$
 olacağından 20 adet talep fazlası vardır. Sahte bir kaynak (fabrika d) oluşturulur ve her bir mağazanın ceza maliyeti bu fabrikadan o mağazaya birim taşıma maliyeti olarak belirlenir
- ($C_{d1}=16, C_{d2}=17, C_{d3}=18, C_{d4}=19, C_{d5}=20$) bu modelin optimal taşıma tablosu:

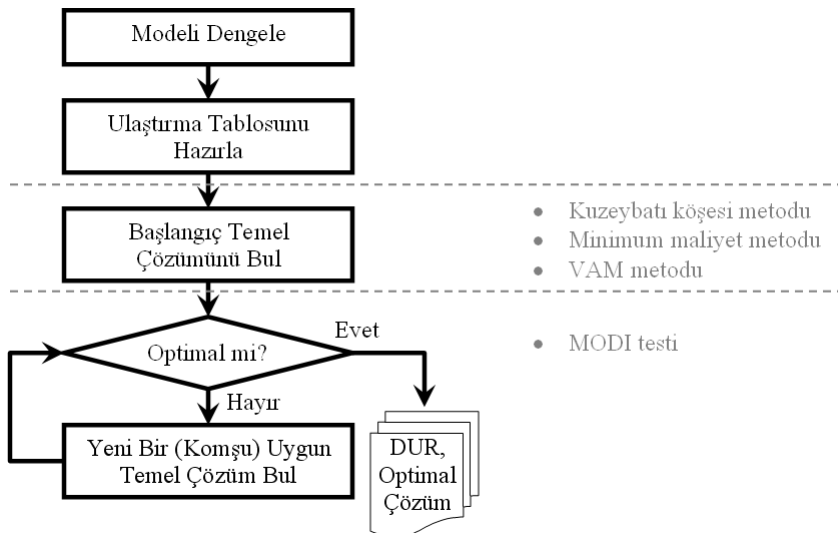
Satış M. Fab.	1	2	3	4	5	Üretim
a	37	23	28	19	35	100
b	36	25	20	26	29	110
c	26	19	17	27	33	130
d	16	17	18	19	20	20
Talep	45	52	38	62	163	360

- Fabrika-d olmadığından, Magaza 5'in 20 adet talebi karşılanamayacaktır.

Yöneylem Araştırması II

241

[Ulaştırma Problemlerinin Çözümü]



Yöneylem Araştırması II

242

[Başlangıç Çözümünün Elde Edilmesi]

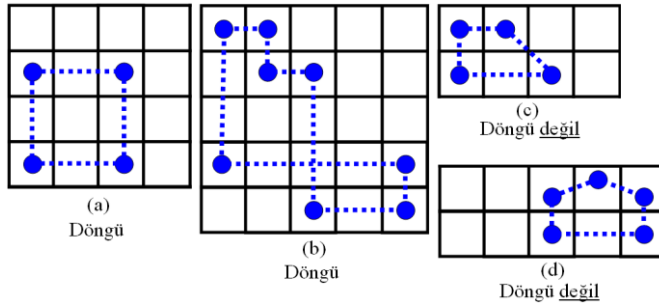


- m adet kaynak ve n adet hedeften oluşan bir ulaştırma problemi $m+n$ eşitlik kısıtı ihtiva eder.
- Eğer herhangi bir çözüm kümesi (x_{ij} 'lerin değerleri), dengeli bir ulaştırma probleminin kısıtlarının **birisi dışında** diğerlerinin tamamını **sağlıyor ise** (sağladığı biliniyor ise), o zaman bu çözüm kümesi otomatik olarak bu **diğer kısıtı da sağlar**. Ancak, $m+n-1$ değişkenden oluşan her kombinasyon uygun bir temel çözüm olması için aşağıdaki **teoremi sağlaması gerekir**.
- Teorem: m adet kaynak ve n adet hedeften oluşan dengeli bir ulaştırma probleminde; $m+n-1$ değişkenden oluşan bir değişkenler kümesi **sadece ve sadece** bu değişkenlere karşılık gelen hücreler **bir döngü içermiyorsa** uygun bir temel çözüm oluşturur.

[Başlangıç Çözümünün Elde Edilmesi]



- **Döngü:** En az dört farklı hücreden oluşan sıralı bir dizi aşağıdaki **3 özelliği sağlıyor ise** döngü olarak adlandırılır.
 - Ardı ardına gelen herhangi 2 hücre **aynı satırda veya aynı sütunda** yer almalıdır.
 - Ardı ardına gelen herhangi 3 hücre aynı satırda veya aynı sütunda yer **almamalıdır**.
 - Dizideki en **son** hücrenin, dizinin **ilk** hücresiyle **ortak bir satır veya sütunu** olmalıdır.



[Başlangıç Çözümünün Elde Edilmesi]



- Dengeli ulaştırma problemlerinin başlangıç uygun temel çözümünün bulunmasında **3 farklı metot kullanılır**. Bunlar;
 - Kuzeybatı köşesi metodu
 - Minimum maliyet metodu
 - Vogel (VAM) metodu

[Başlangıç Çözümünün Elde Edilmesi]



- **1.Kuzeybatı köşesi metodu**
 1. En kuzeybatıdaki hücreyi (i, j hücresi) bul.
 2. X_{ij} 'ye alabileceği en büyük değeri ata ($x_{ij} = \min(s_i, d_j)$).
 - a) $X_{ij} = s_i$ ise; i satırını iptal et ve d_j yerine ($d_j - X_{ij}$) değerini yaz.
 - b) $X_{ij} = d_j$ ise; j sütununu iptal et ve s_i yerine ($s_i - X_{ij}$) değerini yaz.
 - c) $X_{ij} = s_i = d_j$ ise:
 - ya i satırını ya da j sütununu iptal et (ikisini birden iptal etme!)
 - i satırı iptal edilirse. $d_j = 0$ ve
 - j sütunu iptal edilirse, $s_i = 0$ olarak belirle.
 3. İptal edilmeyen hücrelerden en kuzeybatıdaki hücreyi bul ve 2nci adımdaki aynı işlemleri tekrarla.
 4. En sonunda tek bir hücre kalır ($s_i = d_j$). Bu hücreye s_i veya d_j değerini yaz ve hem satır hem de sütunu iptal et. Bu aşamada uygun bir temel çözüm elde edilmiştir.

[Başlangıç Çözümünün Elde Edilmesi]



■ 1. Kuzeybatı köşesi metodu (Örnek)

Depo	D1	D2	D3	Kap.
Fab.				
F1	20 12	8 8	32	20 8 0
F2	15	25 10	9 5	15 5 0
F3	6	12	10 25	25 0
Talep	12	18	30	60

Yukarıdaki tabloda çözüm incelendiğinde, $m+n-1$ adet ($3+3-1=5$ adet) temel değişken elde edildiği ve bu temel değişkenlerin bir döngü oluşturmadığı görülür. O halde teoreme göre Kuzeybatı Köşesi Metodu her zaman uygun bir başlangıç temel çözümü vermektedir.

Toplam dağıtım maliyeti			
FAB.	DEPO	MİK.	MALİYET
F1	D1	12	$20 \times 12 = 240$
F1	D2	8	$8 \times 8 = 64$
F2	D2	10	$25 \times 10 = 250$
F2	D3	5	$9 \times 5 = 45$
F3	D3	25	$10 \times 25 = 250$
			849

[Başlangıç Çözümünün Elde Edilmesi]



■ 2. Minimum maliyet metodu

1. Ulaştırma maliyeti en az olan hücreyi (i, j hücrelerini) bul.
2. X_{ij} 'ye alabileceği en büyük değeri ata ($X_{ij} = \min(s_i, d_j)$). Kuzeybatı Köşesi Metodunda olduğu gibi, i satırını veya j sütununu iptal et, iptal edilmeyen satır veya sütundaki arz veya talebi x_{ij} kadar azalt.
3. İptal edilmeyen hücrelerden maliyeti en az olanı bul ve 2nci adımdaki aynı işlemleri tekrarla.
4. Tek bir hücre kalınca ($s_i = d_j$) bu hücreye s_i veya d_j değerini yaz ve hem satırı hem de sütunu iptal et. Bu aşamada uygun bir temel çözüm elde edilmiştir.

Başlangıç Çözümünün Elde Edilmesi



2. Minimum maliyet metodu (Örnek)

Depo	D1	D2	D3	Kap.
Fab.				
F1	20	8	32	20 2 0
F2	15	25	9	15 0
F3	6	12	10	25 1 0
Talep	12	18	30	60

Yukarıdaki tabloda çözüm incelendiğinde, $m+n-1$ adet ($3+3-1=5$ adet) temel değişken elde edildiği ve bu temel değişkenlerin bir döngü oluşturmadığı görülür. O halde teoreme göre minimum maliyet metodu da her zaman uygun bir başlangıç temel çözümü vermektedir.

Toplam dağıtım maliyeti

FAB.	DEPO	MİK.	MALİYET
F1	D2	18	$8 \times 18 = 144$
F1	D3	2	$32 \times 2 = 64$
F2	D3	15	$9 \times 15 = 135$
F3	D1	12	$6 \times 12 = 72$
F3	D3	13	$10 \times 13 = 130$
			545

Başlangıç Çözümünün Elde Edilmesi



3. Vogel Yaklaşım Metodu (VAM)

- Her satır ve sütun için birer ceza hesapla. Bu ceza, satır veya sütundaki en küçük iki maliyetin arasındaki farka eşittir.
- En yüksek cezası olan satır veya sütunu bul.
- Bu satır veya sütundaki maliyeti en düşük olan hücreyi (i, j hücresi) bul.
- X_{ij} 'ye alabileceği en büyük değeri ata ($X_{ij} = \min(s_i, d_j)$). Kuzeybatı Köşesi Metodunda olduğu gibi i satırını veya j sütununu iptal et. İptal edilmeyen satır veya sütundaki arz veya talebi X_{ij} kadar azalt.
- İptal edilmeyen hücrelere göre cezaları yeniden hesapla ve 1-4 ncü adımlardaki işlemleri tekrarla.
- Tek bir hücre kalınca ($s_i = d_j$) bu hücreye s_i veya d_j değerini yaz ve hem satır hem de sütunu iptal et.

Başlangıç Çözümünün Elde Edilmesi



3. Vogel Yaklaşım Metodu (Örnek)

Depo	D1	D2	D3	Kap.	Fark
Fab.					
F1	20	8	32	20 2	12
F2	15	25	9	15	6
F3	6	12	10	25	4
Talep	12	18 0	30		
Fark	9	4	1		

En büyük farka sahip satırın min. Maliyetli hücresi

En büyük fark

Yöneylem Araştırması II

251

Başlangıç Çözümünün Elde Edilmesi



3. Vogel Yaklaşım Metodu (Örnek)

Depo	D1	D3	Kap.	Fark
Fab.				
F1	20	32	2 0	12
F2	15	9	15	6
F3	6	10	25	4
Talep	12 10	30		
Fark	9	1		

En büyük farka sahip satırın min. Maliyetli hücresi

En büyük fark

Yöneylem Araştırması II

252

Başlangıç Çözümünün Elde Edilmesi



3. Vogel Yaklaşım Metodu (Örnek)

Depo	D1	D3	Kap.	Fark
Fab.				
F2	15	9	15	6
F3	6	10	25 15	4
Talep	10 0	30		
Fark	9	1		

En büyük fark

En büyük farka sahip sütunun min. Maliyetli hücresi

Yöneylem Araştırması II

253

Başlangıç Çözümünün Elde Edilmesi



3. Vogel Yaklaşım Metodu (Örnek)

Depo	D3	Kap.	Fark
Fab.			
F2	9	15 0	-
F3	10	15	-
Talep	30 15		
Fark	1		

En büyük fark

En büyük farka sahip sütunun min. Maliyetli hücresi

Yöneylem Araştırması II

254

[Başlangıç Çözümünün Elde Edilmesi]



■ 3. Vogel Yaklaşım Metodu (Örnek)

Depo	D3	Kap.	Fark
Fab.			
F3	10	15 0	-
Talep	15 0		
Fark	-		

Tek bir hücre kaldığı için uygun şekilde atama yapılır

Yöneylem Araştırması II

255

[Başlangıç Çözümünün Elde Edilmesi]



■ 3. Vogel Yaklaşım Metodu (Örnek)

Depo	D1	D2	D3	Kap.
Fab.				
F1	20	8	32	20
F2	15	25	9	15
F3	6	12	10	25
Talep	12	18	30	60

* Yukarıdaki tabloda çözüm incelendiğinde, $m+n-1$ adet ($3+3-1=5$ adet) temel değişken elde edildiği ve bu temel değişkenlerin bir döngü oluşturmadığı görülür. O halde teoreme göre VAM metodu da her zaman uygun bir başlangıç temel çözümü vermektedir.

Toplam dağıtım maliyeti

FAB.	DEPO	MİK.	MALİYET
F1	D1	2	$20 \times 2 = 40$
F1	D2	18	$8 \times 18 = 144$
F2	D3	15	$9 \times 15 = 135$
F3	D1	10	$6 \times 10 = 60$
F3	D3	15	$10 \times 15 = 150$

529

[Ulaştırma Simpleks Metodu]



- **MODI Metodu (Optimallik Testi)**
- Aşağıda adımları verilen **MODI metodu**, yapılan dağıtımın **optimal olup olmadığını** belirleyen bir algoritmadır.
 1. Problem dengelenmemiş ise **dengele**.
 2. Başlangıç **uygun temel çözümünü** bul.
 3. **Tüm temel değişkenler** ($X_{ij} > 0$) için $u_i = 0$ ve $u_i + v_j = c_{ij}$ denklemlerini kullanarak $u_1, u_2, \dots, u_m$ ve $v_1, v_2, \dots, v_n$ değerlerini hesapla.
 4. **Tüm temel olmayan değişkenler** ($X_{ij} = 0$) için ise $k_{ij} = c_{ij} - u_i - v_j$ veya $k_{ij} = u_i + v_j - c_{ij}$ değerini hesapla.
 - a) Tüm temel olmayan değişkenler için $k_{ij} = c_{ij} - u_i - v_j \geq 0$ veya $k_{ij} = u_i + v_j - c_{ij} \leq 0$ ise dur (**mevcut çözüm optimaldir**).
 - b) Değilse, $k_{ij} = c_{ij} - u_i - v_j$ değeri **en negatif** olan değişkeni veya $k_{ij} = u_i + v_j - c_{ij}$ değeri **en pozitif** olan değişkeni **giren değişken** olarak seç ve **yeni temel çözümü oluştur**.
 5. Yeni temel çözüme göre optimal çözümü bulana kadar **3** ve **4** üncü adımları tekrarla.

[Ulaştırma Simpleks Metodu]



- **MODI Metodu (Yeni Uygun Temel Çözümün Bulunması)**
- 1. **Giren değişkeni** belirle.
- 2. Giren değişken (boş hücre) ile bazı temel değişkenlerden (dolu hücreler) oluşan **döngüyü belirle** (bu durumda olan tek bir döngü vardır).
- 3. Giren değişkenden (boş hücre) başlayarak döngüdeki bütün hücreleri sırasıyla “+ , - , + , - , ...” olarak **işaretle**.
- 4. “-” İşaretli hücrelerden değeri **en küçük olan hücreyi** bul. Bu hücreye karşılık gelen değişken **çıkan değişkendir**. Çıkan değişkenin değerine θ denirse, aşağıdaki işlemlerden sonra **yeni bir** uygun temel çözüm elde edilmiş olur.
 - Her “-” işaretli hücrenin değerini θ kadar **azalt**.
 - Her “+” işaretli hücrenin değerini θ kadar **artır**.
 - Döngü dışındaki hücrelerin değerini **değiştirme**.

[Ulaştırma Simpleks Metodu]



- MODI Metodu (Örnek)
- Minimum maliyet yöntemiyle yapılan atama üzerinde MODI metodu uygulanacaktır. Problem dengelidir.

		$V_1=28$	$V_2=8$	$V_3=32$		* Temel değişkenlerden
	Depo	D1	D2	D3	Kap.	$U_i + V_j = C_{ij}$
	Fab.					$U_1 + V_2 = 8 \Rightarrow 0 + V_2 = 8 \Rightarrow V_2 = 8$
$U_1=0$	F1	20	8	32	20	$U_1 + V_3 = 32 \Rightarrow 0 + V_3 = 32 \Rightarrow V_3 = 32$
			18	2		$U_2 + V_3 = 9 \Rightarrow U_2 + 32 = 9 \Rightarrow U_2 = -23$
$U_2=-23$	F2	15	25	9	15	$U_3 + V_3 = 10 \Rightarrow U_3 + 32 = 10 \Rightarrow U_3 = -22$
			-40	15		$U_3 + V_1 = 6 \Rightarrow -22 + V_1 = 6 \Rightarrow V_1 = 28$
$U_3=-22$	F3	6	12	10	25	* Temel olmayan değ.
		12		13		$k_{ij} = U_i + V_j - C_{ij}$
	Talep	12	18	30	60	$k_{11} = U_1 + V_1 - C_{11} = 0 + 28 - 20 = 8$
						$k_{21} = U_2 + V_1 - C_{21} = -23 + 28 - 15 = -10$
						$k_{22} = U_2 + V_2 - C_{22} = -23 + 8 - 25 = -40$
						$k_{32} = U_3 + V_2 - C_{32} = -22 + 8 - 12 = -26$

Optimal değil, bu hücreye atama yapılması gerekiyor

Yöneyis

259

[Ulaştırma Simpleks Metodu]



- MODI Metodu (Örnek)

		$V_1=28$	$V_2=8$	$V_3=32$	
	Depo	D1	D2	D3	Kap.
	Fab.				
$U_1=0$	F1	20	8	32	20
		+	18	-	2
$U_2=-23$	F2	15	25	9	15
		-	-40	15	
$U_3=-22$	F3	6	12	10	25
		-	-26	+	13
	Talep	12	18	30	60

Negatif işaretli en küçük değere sahip hücre, Değeri: 2

Yöneyem Araştırması II

260

[Ulaştırma Simpleks Metodu]



■ MODI Metodu (Örnek)

		$V_1=20$	$V_2=8$	$V_3=24$		* Temel değişkenlerden
	Depo	D1	D2	D3	Kap.	$U_i + V_j = C_{ij}$
	Fab.					$U_1 + V_1 = 20 \Rightarrow 0 + V_1 = 20 \Rightarrow V_1 = 20$
$U_1=0$	F1	20	8	32	20	$U_1 + V_2 = 8 \Rightarrow 0 + V_2 = 8 \Rightarrow V_2 = 8$
		2	18	-8		$U_3 + V_1 = 6 \Rightarrow U_3 + 20 = 6 \Rightarrow U_3 = -14$
$U_2=-15$	F2	15	25	9	15	$U_3 + V_3 = 10 \Rightarrow (-14) + V_3 = 10 \Rightarrow V_3 = 24$
		-10	-32	15		$U_2 + V_1 = 6 \Rightarrow -15 + V_1 = 6 \Rightarrow V_1 = 21$
$U_3=-14$	F3	6	12	10	25	
		10	-18	15		
	Talep	12	18	30	60	

Bütün
 $k_{ij} \leq 0$
 olduğundan
optimumdur

*** Toplam dağıtım maliyeti**

FAB.	DEPO	MİK.	MALİYET
F1	D1	2	20X2= 40
F1	D2	18	8X18=144
F2	D3	15	9X15=135
F3	D1	10	6X10= 60
F3	D3	15	10X15=150
			529

VAM yöntemindeki sonuçla aynı

[Ulaştırma Problemlerinde Duyarlılık Analizi]



- Ulaştırma problemlerinin **optimal ulaştırma tablosu üzerinde** yapılan duyarlılık analizlerinin **üç** hususu üzerinde durulacaktır. Bunlar;
 1. **Temel olmayan** bir değişkenin **amaç fonksiyonu katsayısının** değiştirilmesi,
 2. **Temel** bir değişkenin **amaç fonksiyonu katsayısının** değiştirilmesi.
 3. Tek bir **arz** ve tek bir **talebin** Δ kadar **artırılması**.

Ulaştırma Problemlerinde Duyarlılık Analizi



- Temel Olmayan Bir Değişkenin Amaç Fonksiyonu Katsayısının Değiştirilmesi ($C_{13}=32 \rightarrow C_{13}=32+\Delta$)

		$V_1=20$	$V_2=8$	$V_3=24$		* Temel değişkenlerden
	Depo	D1	D2	D3	Kap.	
	Fab.					
$U_1=0$	F1	20	8	$32+\Delta$	20	$U_1+V_1=20 \Rightarrow 0+V_1=20 \Rightarrow V_1=20$ $U_1+V_2=8 \Rightarrow 0+V_2=8 \Rightarrow V_2=8$ $U_3+V_1=6 \Rightarrow U_3+20=6 \Rightarrow U_3=-14$ $U_3+V_2=10 \Rightarrow (-14)+V_2=10 \Rightarrow V_2=24$ $U_2+V_3=9 \Rightarrow U_2+(24)=9 \Rightarrow U_2=-15$
$U_2=-15$	F2	15	25	9	15	
$U_3=-14$	F3	6	12	10	25	
	Talep	12	18	30	60	

* Optimum kalabilmesi için

$$-8-\Delta \leq 0 \Rightarrow -8 \leq \Delta \Rightarrow -8 \leq \Delta \leq +\infty \Rightarrow 24 \leq C_{13} \leq +\infty$$

* Öyleyse;

- u_i ve v_j değerleri değişmez, sadece k_{ij} değeri değişir.
- $k_{ij} \leq 0$ olduğu müddetçe temel çözüm optimal kalır.

263

Ulaştırma Problemlerinde Duyarlılık Analizi



- Temel Bir Değişkenin Amaç Fonksiyonu Katsayısının Değiştirilmesi ($C_{11}=20 \rightarrow C_{11}=20+\Delta$)

		$V_1=20+\Delta$	$V_2=8$	$V_3=24+\Delta$		* Temel değişkenlerden
	Depo	D1	D2	D3	Kap.	
	Fab.					
$U_1=0$	F1	$20+\Delta$	8	32	20	$U_1+V_1=20+\Delta \Rightarrow 0+V_1=20+\Delta \Rightarrow V_1=20+\Delta$ $U_1+V_2=8 \Rightarrow 0+V_2=8 \Rightarrow V_2=8$ $U_3+V_1=6 \Rightarrow U_3+(20+\Delta)=6 \Rightarrow U_3=-14-\Delta$ $U_3+V_2=10 \Rightarrow (-14-\Delta)+V_2=10 \Rightarrow V_2=24+\Delta$ $U_2+V_3=9 \Rightarrow U_2+(24+\Delta)=9 \Rightarrow U_2=-15-\Delta$
$U_2=-15-\Delta$	F2	15	25	9	15	
$U_3=-14-\Delta$	F3	6	12	10	25	
	Talep	12	18	30	60	

* Optimum kalabilmesi için

$$\left. \begin{array}{l} -8+\Delta \leq 0 \Rightarrow \Delta \leq 8 \\ -32-\Delta \leq 0 \Rightarrow \Delta \geq -32 \\ -18-\Delta \leq 0 \Rightarrow \Delta \geq -18 \end{array} \right\} \Rightarrow -18 \leq \Delta \leq 8 \Rightarrow 2 \leq C_{11} \leq 28$$

* Öyleyse u_i , v_j ve k_{ij} değerleri bulunarak $k_{ij} \leq 0$ kriterini sağlayan aralık belirlenir.

264

Ulaştırma Problemlerinde Duyarlılık Analizi



- **Arz Ve Talebin Δ Kadar Artırılması**
- Herhangi bir kaynağın **arz** miktarı ile herhangi bir hedefin **talep** miktarı Δ kadar artırıldığını varsayalım. $-u_i$ ve $-v_j$ değerleri amaç fonksiyonu minimizasyon olan bir ulaştırma probleminde kısıtlara ait **Gölge Fiyatları** ifade eder. Bu değişim sonundaki yeni z değeri aşağıdaki formülün kullanılması ile **bulunur**.

$$\text{Yeni } z \text{ değeri} = \text{eski } z \text{ değeri} + \Delta u_i + \Delta v_j$$
- **Karar değişkenlerinin** yeni değerleri ise aşağıdaki şekilde bulunur.
 - X_{ij} optimal çözümdeki **temel** bir değişken ise, X_{ij} 'nin değeri Δ kadar artırılır.
 - X_{ij} optimal çözümdeki **temel olmayan** bir değişken ise, X_{ij} ve temel değişkenlerin bazılarında oluşan **döngü** bulunur. Döngüdeki i satırında "+" işaretli hücre bulunur ve bu hücrenin değeri Δ kadar **artırılır**. Daha sonra döngü boyunca sırayla döngüdeki hücrelerin değerleri Δ kadar bir **azaltılıp** bir **artırılır**.

Yöneylem Araştırması II

265

Ulaştırma Problemlerinde Duyarlılık Analizi



- **Arz Ve Talebin Δ Kadar Artırılması (1.Durum: X_{ij} Temel Bir Değişken ise)**
- $\Delta=2$ için ; $S_1=20 \rightarrow S_1=20+\Delta=22$ ve $d_1=12 \rightarrow d_1=12+\Delta=14$

Depo	D1	D2	D3	Kap.				
Fab.								
F1	20	8	32	20				
F2	15	25	9	15				
F3	6	12	10	25				
Talep	12	18	30	60				

$V_1=20 \quad V_2=8 \quad V_3=24$

Depo	D1	D2	D3	Kap.				
Fab.								
F1	20	8	32	22				
F2	15	25	9	15				
F3	6	12	10	25				
Talep	14	18	30	60				

$U_1=0 \quad U_2=-15 \quad U_3=-14$

* Bu durumda **karar değişkeni** ve **amaç fonksiyonu değeri** aşağıdaki şekilde **değişir**;

$$X_{11} = 2 + \Delta = 4$$

$$Z = 529 + \Delta(U_1) + \Delta(V_1) = 529 + 2(0) + 2(20) = 569$$

Yöneylem Araştırması II

266

Ulaştırma Problemlerinde Duyarlılık Analizi



- Arz Ve Talebin Δ Kadar Artırılması (1.Durum: X_{ij} Temel Olmayan Bir Değişken ise)
- $\Delta=2$ için ; $S_1=20 \rightarrow S_1=20+\Delta=22$ ve $d_3=30 \rightarrow d_3=30+\Delta=32$

Depo	D1	D2	D3	Kap.
Fab.				
F1	20	8	32	20
F2	15	25	9	15
F3	6	12	10	25
Talep	12	18	30	60

→

Depo	D1	D2	D3	Kap.
Fab.				
F1	20	8	32	22
F2	15	25	9	15
F3	6	12	10	25
Talep	12	18	32	60

$V_1=20$ $V_2=8$ $V_3=24$
 $U_1=0$ $U_2=15$ $U_3=14$

* Bu durumda karar değişkeni ve amaç fonksiyonu değeri aşağıdaki şekilde değişir;

$$X_{11} = 2 + \Delta = 4; X_{31} = 10 - \Delta = 8; X_{33} = 15 + \Delta = 17$$

Yöneyim Araştırması II

$$Z = 529 + \Delta(U_1) + \Delta(V_3) = 529 + 2(0) + 2(24) = 577$$

267

Atama Problemleri



- m adet işin veya işçinin n adet makinalara atanması **Atama Problemi** olarak adlandırılır.
- **Amaç** bir makinalara bir iş olmak üzere, işleri makinalara **en az maliyetle** atamaktır.
- Bu problemin formülasyonu, **ulaştırma problemlerinin bir özel hali** olarak düşünülebilir. Burada işler kaynakları ve makinalarda hedefleri temsil ederler. Her kaynaktaki arz miktarı ve her hedefteki talep miktarı 1 ve i işini j makinasına **atama maliyeti** ise c_{ij} 'dir.
- n adet iş ve n adet makinalardan oluşan dengeli bir atama problemlerinin **doğrusal programlama modeli**:

$$\begin{aligned}
 \text{Min } z &= \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
 \text{s.t. } &\sum_{j=1}^n x_{ij} = 1 \quad (i = 1, 2, \dots, n) \\
 &\sum_{i=1}^n x_{ij} = 1 \quad (j = 1, 2, \dots, n) \\
 &x_{ij} = 0 \text{ veya } 1 \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, n)
 \end{aligned}$$

$$x_{ij} = \begin{cases} 0, & \text{eğer } i\text{nci iş } j\text{nci makinalara atanmamışsa} \\ 1, & \text{eğer } i\text{nci iş } j\text{nci makinalara atanmışsa} \end{cases}$$

Yöneyim

268

[Atama Problemleri



- **Atölye problemi**
- Bir imalat atölyesinde **4 türlü iş** ve bunların **4'ünü** de yapabilecek kalifiyede **4 teknisyen** bulunmaktadır. Teknisyenlerin bu işleri ne kadar sürede yaptıkları **saat cinsinden** aşağıdaki tabloda verilmiştir. Hangi iş hangi teknisyene atanmalı ki toplam **harcanan süre minimum** olsun.

Tek. İşler	X	Y	Z	K
1	8	20	17	11
2	13	28	4	26
3	38	19	18	15
4	19	26	24	10

Yöneylem Araştırması II

269

[Atama Problemleri



- Atölye Problemi - Tamsayılı Programlama Modeli

Tek. İşler	X	Y	Z	K
1	8	20	17	11
2	13	28	4	26
3	38	19	18	15
4	19	26	24	10

❄ Karar değişkenleri:

$$x_{ij} = \begin{cases} 0, & \text{eğer } i\text{-nci iş } j\text{-nci teknisyene atanmamışsa} \\ 1, & \text{eğer } i\text{-nci iş } j\text{-nci teknisyene atanmışsa} \end{cases}$$

$$\begin{array}{rcccccc}
MinZ & = & 8x_{11} & + 20x_{12} & + 17x_{13} & + 11x_{14} & + \dots + \\
& & 19x_{41} & + 26x_{42} & + 24x_{43} & + 10x_{44} & \\
S.T. & & x_{11} & + x_{12} & + x_{13} & + x_{14} & = 1 \\
& & \vdots & \vdots & \vdots & \vdots & \vdots \\
& & x_{41} & + x_{42} & + x_{43} & + x_{44} & = 1 \\
& & x_{11} & & x_{21} & & x_{31} & & x_{41} & = 1 \\
& & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\
& & x_{14} & & & + x_{24} & & + x_{34} & & + x_{44} & = 1 \\
& & x_{ij} & = & 0 \vee 1 & (i = 1..4; & j = 1..4)
\end{array}$$

Yöneylem Araştırması II

270

[Atama Problemleri]



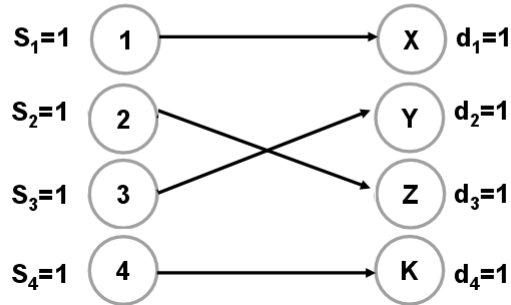
■ Atölye Problemi - Tamsayı Programlama Modeli

Tek. İşler	X	Y	Z	K
1	8	20	17	11
2	13	28	4	26
3	38	19	18	15
4	19	26	24	10

※ Problemin optimal çözümü

$$Z^* = 41$$

$$x_{11} = 1, \quad x_{23} = 1, \quad x_{32} = 1, \quad x_{44} = 1$$



Yöneylem Araştırması II

271

[Atama Problemleri (Macar Metodu)]



- Macar metodunun **algoritması** aşağıda verilmiştir.
- 1. Birim işlem maliyetlerinden oluşan $n \times n$ 'lik maliyet matrisini oluştur. Her bir **satırın en küçük maliyet** değerine sahip elemanını bul. Satırlardaki bütün maliyet değerlerinden en küçük maliyet değerini **çıkarmak** suretiyle yeni bir matris oluştur. Bu yeni matrisde her bir **sütunun en küçük maliyet** değerine sahip elemanını bul. Sütunlardaki bütün maliyet değerlerinden en küçük maliyet değerini **çıkarmak** suretiyle **azaltılmış maliyet** (veya **fırsat maliyeti**) **matrisini** oluştur.

Yöneylem Araştırması II

272

[Atama Problemleri (Macar Metodu)]



2. Azaltılmış maliyet matrisindeki **bütün sıfırları** kaplayacak şekilde, mümkün olan en az sayıda **doğru (yatay ve/veya dikey)** çiz. Doğruların sayısı satır veya sütunların sayısına (n 'e) eşitse optimal atamalar elde edilebilir. Doğruların sayısı $< n$ ise **3**ncü adıma git.
3. Azaltılmış maliyet matrisindeki herhangi bir doğru ile **kapatılmamış en küçük maliyet değerine** sahip elemanı bul. Bu elemanın değerini, herhangi bir doğru ile **kapatılmamış** olan bütün elemanların maliyet değerlerinden **çıkart** ve **iki doğrunun kesiştiği** yerlerdeki elemanların maliyet değerlerine **ekle**. Üzerinden tek çizgi geçen elemanların maliyet değerlerinde bir değişiklik yapma ve **2**nci adıma git.

Yöneylem Araştırması II

273

[Atama Problemleri (Macar Metodu)]



■ Örnek: Atölye Problemi

Tek.	X	Y	Z	K	Satır Minimumu
İşler					
1	8	20	17	11	8
2	13	28	4	26	4
3	38	19	18	15	15
4	19	26	24	10	10

Her satırın **minimum** maliyet değeri satırın **tüm** elemanlarından çıkartılır.

Tek.	X	Y	Z	K
İşler				
1	0	12	9	3
2	9	24	0	22
3	23	4	3	0
4	9	16	14	0
Sütun Minimumu	0	4	0	0

Her sütunun **minimum** maliyet değeri **sütunun** **tüm** elemanlarından çıkartılır.

Yöneylem Araştırması II

274

[Atama Problemleri (Macar Metodu)]



■ Örnek: Atölye Problemi

	Tek.	X	Y	Z	K
İşler					
1		0	8	9	3
2		9	20	0	22
3		23	0	3	0
4		9	12	14	0

Azaltılmış maliyet matrisindeki tüm sıfırları kaplayacak şekilde **en az sayıda doğru** çizilir.

Azaltılmış maliyet matrisi

- Bu işlem 4 doğru ile mümkün olabilmektedir. *Doğru sayısı*= $n=4$ olduğundan, problemin **optimal çözümü bulunmuştur**.

■ Atama işlemi

- **Sadece bir sıfır** bulunan satır veya sütunlar bulunarak o hücrelere **atama yapılır**. Atama yapılan satır ve sütunlar **kapatılır**.
- Kalan satır ve sütunlar dikkate alınarak **sıfır bulunan** hücrelere atama yapılır.

Yöneylem Araştırması II

275

[Atama Problemleri (Macar Metodu)]



■ Örnek: Atölye Problemi

	Tek.	X	Y	Z	K
İşler					
1					
2					
3					
4					0

Satır veya sütunda tek sıfır var ise $X_{11}=1$

Satır veya sütunda tek sıfır var ise $X_{23}=1$

Sütunda tek sıfır var ise $X_{32}=1$

Satır veya sütunda tek sıfır var ise $X_{44}=1$

Azaltılmış maliyet matrisi

■ $Z=8(1)+4(1)+19(1)+10(1)=41$

■ Yani:

- 1nci iş X teknisyenine,
- 2nci iş Z teknisyenine,
- 3ncü iş Y teknisyenine,
- 4ncü iş K teknisyenine

atanırsa tüm işler **minimum** süre olan **41 saatte** bitirilecektir.

Yöneylem Araştırması II

276

[Atama Problemleri (Macar Metodu)]



- **Örnek: Servis problemi**
- Bir buzdolabı firması yurt çapındaki servis hizmetlerini ellerindeki 5 değişik şehirdeki servis istasyonları aracılığı ile gerçekleştirmektedir. Servis istasyonları her hafta **ancak bir bölgeye** gidebilme durumundadır. Bu durumda her hafta **ancak 5 bölgeye** hizmet götürülebilmektedir. Aşağıdaki tabloda 5 servis istasyonunun gitmesi gerekli diğer 5 bölgeye olan **uzaklıkları km olarak** verilmiştir.

Bölge Servis	A	B	C	D	E
I	110	220	80	180	230
II	90	70	130	60	180
III	130	190	150	120	160
IV	210	240	170	300	260
V	140	100	120	110	150

Buzdolabı firması, 5 servis istasyonlarının gitmek zorunda bulunduğu bölgelere **en az yolu** katederek gitmesi için **nasıl bir program** yapmalıdır.

Yöneylem Araştırması II

277

[Atama Problemleri (Macar Metodu)]



- **Örnek: Servis problemi**

Bölge Servis	A	B	C	D	E	Min
I	110	220	80	180	230	80
II	90	70	130	60	180	60
III	130	190	150	120	160	120
IV	210	240	170	300	260	170
V	140	100	120	110	150	100

Her satırın **minimum** maliyet değeri satırın **tüm** elemanlarından çıkartılır.

Bölge Servis	A	B	C	D	E
I	30	140	0	100	150
II	30	10	70	0	120
III	10	70	30	0	40
IV	40	70	0	130	90
V	40	0	20	10	50
Min	10	0	0	0	40

Her sütunun **minimum** maliyet değeri **sütunun tüm** elemanlarından çıkartılır.

Yöneylem Araştırması II

278

[Atama Problemleri (Macar Metodu)]



■ Örnek: Servis problemi

Servis \ Bölge	A	B	C	D	E
I	20	140	0	100	110
II	20	10	70	0	80
III	0	70	50	0	0
IV	30	70	0	130	50
V	30	0	20	10	10

Azaltılmış maliyet matrisi

Azaltılmış maliyet matrisindeki tüm sıfırları kaplayacak şekilde **en az sayıda** doğru çizilir.

- Bu işlem 4 doğru ile mümkün olabilmektedir.
- $Doğru\ sayısı = 4 < n = 5$ olduğundan, bu matristeki herhangi bir doğru ile kapatılmamış **en küçük** maliyet değerine sahip eleman seçilir; $C_{11} = 20$.

Yöneylem Araştırması II

279

[Atama Problemleri (Macar Metodu)]



■ Örnek: Servis problemi

Servis \ Bölge	A	B	C	D	E
I	20	140	0	100	110
II	20	10	70	0	80
III	0	70	50	0	0
IV	30	70	0	130	50
V	30	0	20	10	10

Azaltılmış maliyet matrisi

Bu elemanın değeri,
 • herhangi bir doğru ile **kapatılmamış** olan bütün elemanlardan **çıkarılır** ve
 • iki doğrunun **kesiştği** yerlerdeki elemanlara **eklenir**.

Servis \ Bölge	A	B	C	D	E
I	0	120	0	80	90
II	20	10	90	0	80
III	0	70	50	0	0
IV	10	50	0	110	30
V	30	0	40	10	10

Yöneylem Araştırması II

280

[Atama Problemleri (Macar Metodu)]



■ Örnek: Servis problemi

Bölge Servis	A	B	C	D	E
I	0	120	0	80	90
II	20	10	90	0	80
III	0	70	50	0	0
IV	10	50	0	110	30
V	30	0	40	10	10

Azaltılmış maliyet matrisindeki tüm sıfırları kaplayacak şekilde **en az sayıda doğru** çizilir.

- Bu işlem 5 doğru ile mümkün olabilmektedir. *Doğru sayısı* = $n=5$ olduğundan, problemin **optimal çözümü bulunmuştur**.
- **Atama işlemi**
 - **Sadece bir sıfırı** bulunan satır veya sütunlar bulunarak o hücrelere **atama yapılır**. Atama yapılan satır ve sütunlar **kapatılır**.
 - Kalan satır ve sütunlar dikkate alınarak **sıfır bulunan** hücrelere atama yapılır.

Yöneylem Araştırması II

281

[Atama Problemleri (Macar Metodu)]



■ Örnek: Servis problemi

Bölge Servis	A	B	C	D	E
I					
II					
III					
IV					
V		0			

- Satırda tek sıfır var ise $X_{24}=1$
- Satırda tek sıfır var ise $X_{43}=1$
- Satırda tek sıfır var ise $X_{11}=1$
- Satır veya sütundatek sıfır var ise $X_{35}=1$
- Satır veya sütundatek sıfır var ise $X_{52}=1$

■ $Z=60(1)+170(1)+110(1)+160(1) +100(1)=600$

■ Yani:

- 2nci servise D bölgesi,
- 4ncü servise C bölgesi,
- 1nci servise A bölgesi,
- 3ncü servise E bölgesi ve
- 5ncü servise B bölgesi

atanırsa **katedilecek minimum mesafe 600 km** olacaktır.

Yöneylem Araştırması II

282

[Atama Problemleri (Macar Metodu)]



- Bazı durumlarda atama probleminin amaç fonksiyonu **maksimizasyon** olabilir. Bu durumda maliyet matrisinin tüm elemanları **-1** ile çarpılmak suretiyle **Macar Metodu** kullanılabilir.
- Macar metodunun uygulanabilmesi için **arz** miktarı ile **talep** miktarının birbirine **eşit** olması gerekmektedir. Bu durumda maliyet matrisi **$n \times n$** 'lik bir matris oluşturur.
- Eğer atama problemi **dengeli değil** ise, aynen ulaştırma problemlerinde olduğu gibi **hayali arz** veya **talep** noktası açmak suretiyle **problem dengelenmelidir**.

Yöneylem Araştırması II

283

[Atama Problemleri (Macar Metodu)]



- **Örnek Uygulama 1: Atölye Problemi-Modelin Dengelenmesi**
- **1, 2 ve 4** işlerini sırasıyla **10, 15 ve 6** saatte yapacak ve tecrübesi olmaması nedeniyle **3**-işini yapamayan yeni bir **I** işçisinin varlığı durumunda mevcut **4** işçiden birisinin **yerine** bu işçinin alınmasının maliyeti azaltıp azaltmayacağını hesaplayın.
- Bu durumda Macar Metodunu kullanabilmek maksadıyla modeli **dengeli hale getirebilmek** için bir **hayali iş** eklenir ve atama maliyeti (amaç fonksiyonu katsayısı) **sıfır** alınır.

Tek. İşler	X	Y	Z	K	L	Min
1	8	20	17	11	10	8
2	13	28	4	26	15	4
3	38	19	18	15	M	15
4	19	26	24	10	6	6
H	0	0	0	0	0	0

Tek. İşler	X	Y	Z	K	L	Min
1	0	12	9	3	2	
2	9	24	0	22	11	
3	23	4	3	0	M	
4	13	20	18	4	0	
H	0	0	0	0	0	
Min	0	0	0	0	0	

Yöneylem Araştırması II

284

[Atama Problemleri (Macar Metodu)]



■ Örnek Uygulama 1: Atölye Problemi-Modelin Dengelenmesi

Tek. İşler	X	Y	Z	K	L
1	0	12	8	3	2
2	9	24	0	22	11
3	23	4	3	0	M
4	13	20	18	4	0
5	0	0	0	0	0

Azaltılmış Maliyet Matrisi

Azaltılmış maliyet matrisindeki tüm sıfırları kaplayacak şekilde en az sayıda doğru çizilir.

- Bu işlem 5 doğru ile mümkün olabilmektedir. *Doğru sayısı* = $n=5$ olduğundan, problemin optimal çözümü bulunmuştur.

$$X_{11}=1, X_{23}=1, X_{34}=1, X_{45}=1, X_{52}=1$$

$$Z=8(1)+4(1)+15(1)+6(1)+0(1)=33$$

Toplam süre azalmıştır.

[Atama Problemleri (Macar Metodu)]



- Örnek Uygulama 2: Atölye Problemi-Modelin Dengelenmesi
- Firma yeni bir iş alsın ve mevcut 4 işçi bu işi sırasıyla 20, 10, 20 ve 50 saatte yapabilirse, bu yeni işin maliyet minimizasyonu açısından diğer işlere göre öncelik elde edip etmeyeceğini macar metodu ile değerlendirin.
- Bu durumda Macar Metodunu kullanabilmek maksadıyla modeli dengeli hale getirmek için bir hayali işçi eklenir ve atama maliyeti (amaç fonksiyonu katsayısı) sıfır alınır.

Tek. İşler	X	Y	Z	K	H	Min
1	8	20	17	11	0	0
2	13	28	4	26	0	0
3	38	19	18	15	0	0
4	19	26	24	10	0	0
5	20	10	20	50	0	0

Tek. İşler	X	Y	Z	K	H
1	8	20	17	11	0
2	13	28	4	26	0
3	38	19	18	15	0
4	19	26	24	10	0
5	20	10	20	50	0
Min	8	10	4	10	0

[Atama Problemleri (Macar Metodu)]



- Örnek Uygulama 1: Atölye Problemi-Modelin Dengelenmesi

Tek. İşler	X	Y	Z	K	H
1	0	10	13	1	0
2	5	18	0	16	0
3	30	9	14	5	0
4	11	16	20	0	0
5	12	0	16	40	0

Azaltılmış Maliyet Matrisi

Azaltılmış maliyet matrisindeki tüm sıfırları kaplayacak şekilde en az sayıda doğru çizilir.

- Bu işlem 5 doğru ile mümkün olabilmektedir. *Doğru sayısı* = $n=5$ olduğundan, problemin optimal çözümü bulunmuştur.

$$X_{11}=1, X_{23}=1, X_{35}=1, X_{44}=1, X_{52}=1$$

$$Z=8(1)+4(1)+0(1)+10(1)+10(1)=32$$

Toplam süre azalmıştır.

[BÖLÜM VI: Proje Yönetimi]

- Proje yönetimine giriş
- CPM (Kritik yol metodu)
 - Şebeke çizimi
 - Şebeke hesaplamaları
 - Kritik yolun belirlenmesi
- Kritik yolun Doğrusal Programlama ile bulunması
- Zaman çizelgelemesi
- PERT metodu

[Proje Yönetimine Giriş]



- Görevin belirli bir sürede tamamlanabilmesi için belirli bir sıraya göre icra edilmesi gereken ve birbiriyle bağlantılı olan faaliyetlerin bütününe **Proje** denir.
- **Faaliyet**, tamamlanabilmesi için belirli bir süre ve kaynak gerektiren işlerdir.
- Görevin zamanında, bütçe sınırları içerisinde ve beklenen özelliklere uygun olarak yapılmasının sağlanması işlemi **Proje Yönetimi** olarak adlandırılır.
- Proje yönetiminde karşılaşılan **bazı sorular**;
 - Projenin tamamlanması için **gerekli süre** ne kadardır?
 - Her faaliyetin **başlangıç ve bitiş tarihleri** nelerdir?
 - Projenin zamanında tamamlanabilmesi için, hangi faaliyetler **kritiktir** ve planlanan süre içerisinde tamamen bitirilmelidir?
 - Hangi faaliyetler **kritik değildir** ve projenin süresini geciktirmeyecek şekilde bu faaliyetlerin süreleri ne kadar uzatılabilir?

Yöneylem Araştırması II

289

[Proje Yönetimine Giriş]



- Proje yönetiminde kullanılan **analiz teknikleri**;
 - **CPM** (Critical Path Method)-**Kritik Yol Metodu**
 - **PERT** (Program Evaluation and Review Technique)-**Proje Değerlendirme ve Gözden Geçirme Tekniği**
- CPM ve PERT şebeke modelleri ile proje yönetiminin **üç temel safhası**;
 1. **Planlama**: Projenin alt safhalara bölünmesi, faaliyet süreleri ve öncelik ilişkilerinin tespiti, proje şebekesinin oluşturulması, şebeke hesaplamaları ile proje süresi, kritik faaliyetler ve faaliyetlerin bolluklarının belirlenmesi.
 2. **Çizelgeleme**: Her faaliyetin başlangıç ve bitiş zamanları ile projedeki diğer faaliyetlerle ilişkisini gösteren bir zaman çizelgesinin oluşturulması.
 3. **Kontrol**: Şebeke ve zaman çizelgelerini kullanarak proje ilerlemesinin kontrolü, şebeke veya çizelgenin güncellenmesi.

Yöneylem Araştırması II

290

[CPM (Kritik Yol Metodu)]



- Faaliyet sürelerinin **kesin** olarak bilindiği durumlarda Kritik Yol Metodu (CPM) kullanılabilir.
- Kritik Yol Metodunun uygulanmasında izlenen adımlar aşağıdaki gibi özetlenebilir;
 1. Proje için gerçekleştirilmesi gerekli **faaliyetlerin bir listesi** hazırlanır.
 2. Her bir faaliyetin hangi faaliyet veya faaliyetler tamamlandıktan sonra başlaması gerektiği (yani **önceliği**) belirlenir.
 3. Her faaliyetin **süresi** tahmin edilir.
 4. Faaliyetler arasında belirlenen ilişkileri uygun olarak temsil eden **şebeke** çizilir.
 5. Şebekenin **hesaplaması** yapılır. Her düğümün **En Erken** ve **En Geç Ortaya Çıkma Zamanları** hesaplanarak proje süresi ve faaliyet bollukları belirlenir.
 6. Bir tablo hazırlanarak her faaliyetin süresi, en erken ve en geç başlama ve tamamlanma zamanları ve bollukları yazılır.
 7. **Kritik Yol üzerindeki faaliyetler** belirlenir.
 8. En son aşamada projenin başlayacağı tarihe göre bir **zaman çizelgesi** hazırlanır.

[CPM (Şebeke Çizimi)]

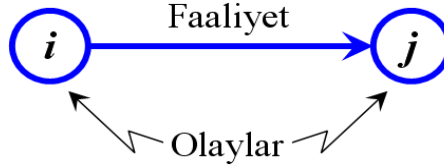


- Her proje **Proje Şebekesi** olarak adlandırılan grafikte gösterilir. Şebekeler **iki** türlü çizilebilir;
 - **Faaliyetlerin oklarla gösterildiği şebekeler** (AOA-Activity on Arc)
 - **Faaliyetlerin düğümlerle gösterildiği şebekeler** (AOA-Activity on Node)
 Burada birinci yaklaşım kullanılacaktır.
- Proje şebekesinde **faaliyetler oklarla gösterilir**.
- Bu şekilde hazırlanan şebekelerde her bir düğüm **olay** olarak adlandırılır ve **faaliyetlerin başlangıç ve bitiş zamanlarını gösterir**.

[CPM (Şebeke Çizimi)]



- Oklar **düz**, **kırık** veya **eğri** bir çizgi ile gösterilebilir. **Olay faaliyet ilişkisi** aşağıdaki şekil üzerinde gösterilmiştir.



- **Faaliyetler** kullandıkları kaynaklar bakımından **üç sınıfta** toplanabilir;
 1. Belli bir zaman süresinde gerçekleştirilen ve işgücü, araç, gereç ve **gider gerektiren** normal faaliyetler (montaj, beton dökümü vs.),
 2. Tamamlanması belli bir süre gerektiren ancak işgücü, malzeme, para gibi **kaynak gerektirmeyen** faaliyetler (sipariş bekleme vs),
 3. Sadece faaliyetler arasındaki ilişkileri belirtmek için kullanılan, zaman ve kaynak gerektirmeyen **Boş (Kukla) Faaliyetler**.

Yöneylem Araştırması II

293

[CPM (Şebeke Çizimi)]



- Faaliyetler arasında **eksik veya fazla ilişki kurulmamalıdır**.
- **Öncelik ilişkisinin** doğru olarak sağlanmasına dikkat edilmelidir.
- Bir olayın tamamlanması ancak o olaya yönlendirilen faaliyetlerin **tamamının** tamamlanmasıyla gerçekleşir.
- **Boş (Kukla) Faaliyetler** kesik oklarla gösterilir ve süreleri **sıfırdır**.



Yöneylem Araştırması II

294

[CPM (Şebeke Çizimi)]

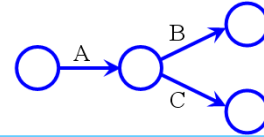


- Bir şebekede faaliyetler (işlemler) arasındaki ilişkiler aşağıdaki biçimde ifade edilirler.

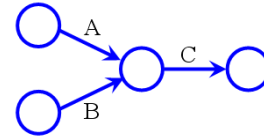
- A faaliyeti tamamlandıktan sonra B faaliyeti başlıyorsa;



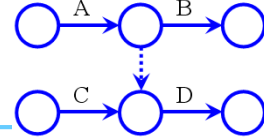
- A faaliyeti tamamlandıktan sonra B ve C faaliyeti başlıyorsa;



- A ve B faaliyeti tamamlandıktan sonra C faaliyeti başlıyorsa;



- A ve C faaliyeti tamamlandıktan sonra B ve D faaliyeti başlıyorsa;



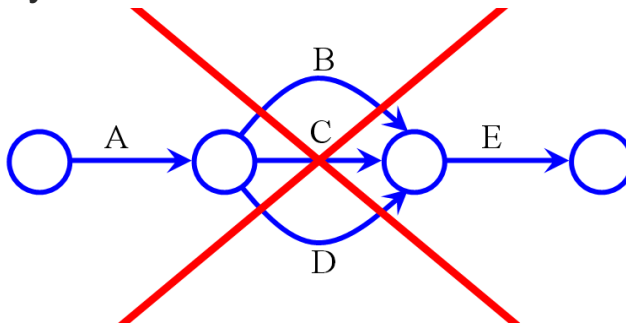
Yöneylem Araştırması II

295

[CPM (Şebeke Çizimi)]



- Şebeke çiziminde dikkat edilmesi gereken hususlar
- İki düğüm ile birden fazla faaliyet gösterilemez. Bu yolla her faaliyetin bolluğu ayrı ayrı incelenebilir.



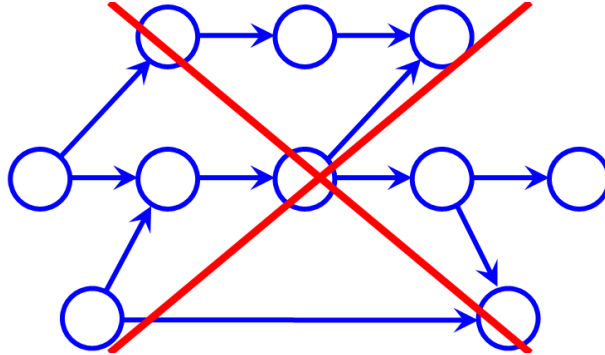
Yöneylem Araştırması II

296

[CPM (Şebeke Çizimi)]



- Şebeke çiziminde dikkat edilmesi gereken hususlar
- Projenin normalde yalnız bir başlangıç ve yalnız bir bitiş düğümü **olmalıdır**. Böylece başlangıç ve bitiş düğümleri arasında **Kritik Yol veya Yollar** bulunabilir.



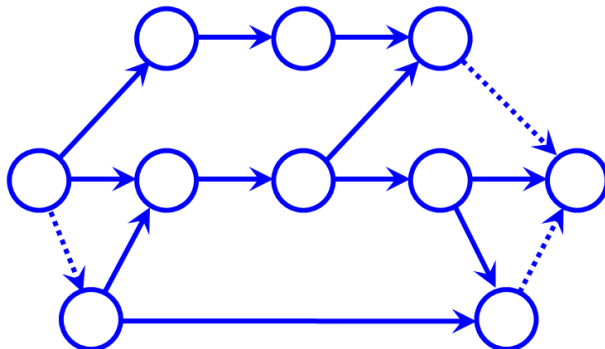
Yöneylem Araştırması II

299

[CPM (Şebeke Çizimi)]



- Şebeke çiziminde dikkat edilmesi gereken hususlar
- Projenin normalde yalnız bir başlangıç ve yalnız bir bitiş düğümü **olmasını sağlamak için** aşağıdaki gibi **Kukla (Boş) Faaliyetler** kullanılır.



Yöneylem Araştırması II

300

[CPM (Şebeke Çizimi-ÖRNEK)]



- Bir otomatik depolama sistemi kurulacağını varsayarak ve bu proje ile ilgili faaliyetleri aşağıdaki tabloda olduğu gibi varsayalım.

Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A, B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D, F, G	Bilgisayar bağlantılarını kurma
I	D, F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I, J	Sistemi test etme

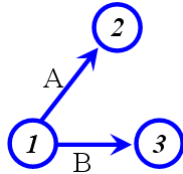
Yöneylem Araştırması II

301

[CPM (Şebeke Çizimi-ÖRNEK)]



Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A, B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D, F, G	Bilgisayar bağlantılarını kurma
I	D, F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I, J	Sistemi test etme



- Şebekeyi çizmeye **başlangıç düğümü (olay)** ile başlanır.
- Bu düğümden başlayan faaliyetler, başlaması için herhangi bir diğer faaliyetin tamamlanmasına ihtiyaç duymayan faaliyetlerdir.
- Tablodaki faaliyet tanımlarına ve önceliklerine bakıldığında bu faaliyetlerin **A** ve **B** faaliyetleri olduğu görülür.

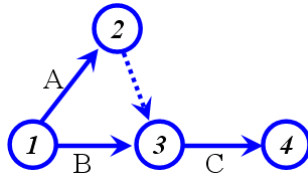
Yöneylem Araştırması II

302

CPM (Şebeke Çizimi-ÖRNEK)



Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A,B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D,F,G	Bilgisayar bağlantılarını kurma
I	D,F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I,J	Sistemi test etme



- C faaliyetinin başlaması ancak ve ancak A ve B faaliyetlerinin bitmesi ile mümkün olmaktadır.
- 2 ve 3 düğümleri arasında boş bir faaliyet kullanarak, C faaliyeti 3 ve 4 düğümleri arasına aşağıdaki gibi çizilir.

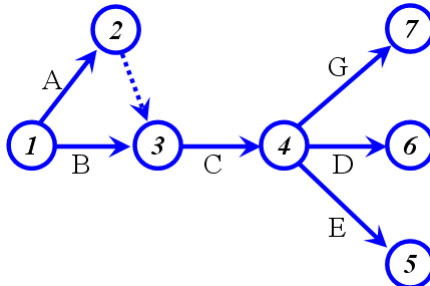
Yöneylem Araştırması II

303

CPM (Şebeke Çizimi-ÖRNEK)



Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A,B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D,F,G	Bilgisayar bağlantılarını kurma
I	D,F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I,J	Sistemi test etme



- Sıradaki D, E ve G faaliyetlerinin hepsi C faaliyetinin tamamlanmasından sonra başlayacaktır;

Yöneylem Araştırması II

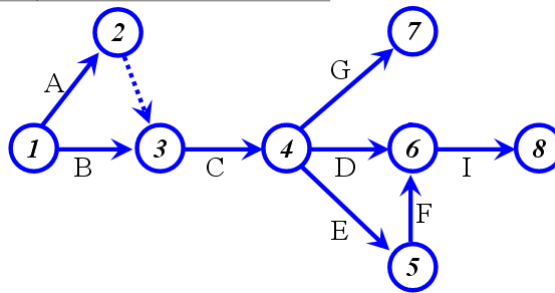
304

CPM (Şebeke Çizimi-ÖRNEK)



Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A,B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D,F,G	Bilgisayar bağlantılarını kurma
I	D,F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I,J	Sistemi test etme

- F faaliyeti E faaliyeti tamamlandıktan sonra,
- I faaliyeti de D ve F faaliyetleri tamamlandıktan sonra başlayacaktır.



Yöneylem Araştırması II

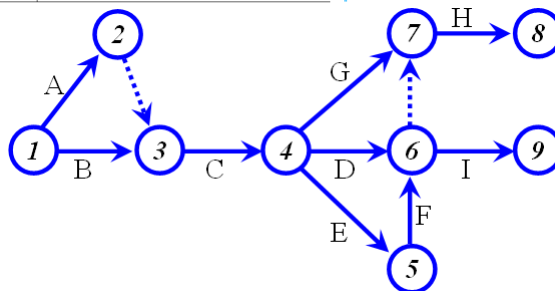
305

CPM (Şebeke Çizimi-ÖRNEK)



Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A,B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D,F,G	Bilgisayar bağlantılarını kurma
I	D,F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I,J	Sistemi test etme

- H faaliyetinin başlaması ancak ve ancak D, F ve G faaliyetlerinin tamamlanması ile mümkün olmaktadır.



Yöneylem Araştırması II

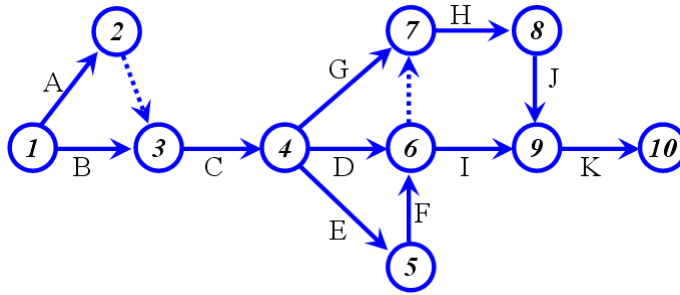
306

CPM (Şebeke Çizimi-ÖRNEK)



Faaliyet	Öncelik	Faaliyetlerin tanımı
A	-	Malzeme ihtiyaçlarının tespiti
B	-	İhale için teklif mektuplarını alma
C	A,B	Satıcıyı seçme
D	C	Sistemin ısmarlanması
E	C	Deponun şeklini tasarlama
F	E	Depoyu kurma
G	C	Bilgisayar bağlantılarını tasarlama
H	D,F,G	Bilgisayar bağlantılarını kurma
I	D,F	Bilgisayar sistemini kurma
J	H	Sistem operatörlerinin eğitimi
K	I,J	Sistemi test etme

- J ve K faaliyetleri de ilave edilirse proje şebekesi aşağıdaki gibi olacaktır.



Yöneylem Araştırması II

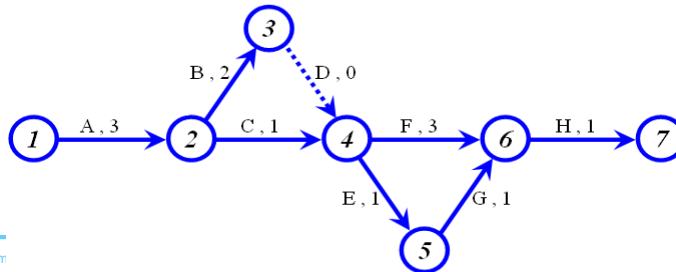
307

CPM (Şebeke Hesaplamaları)



- Her hangi bir projenin şebekesi oluşturulduktan sonra bu şebeke üzerinde yapılması gereken **bazı hesaplamalar** vardır. Bunu gösterebilmek için aşağıdaki şekilde bir proje olduğunu varsayalım.
- Bir ev inşaatı projesine ait faaliyetlerin sürelerini ve aralarındaki öncelik ilişkisini gösteren faaliyet listesi ve proje şebekesi aşağıda verilmiştir.

Faaliyet	Öncelik	Faaliyetlerin Tanımı	Süre (Ay)
A	-	Hazırlık (tasarım, para...)	3
B	A	Temel inşaatı	2
C	A	Malzeme tedarik	1
E	B,C	Boya seçimi	1
F	B,C	Evin inşaatı	3
G	E	Döşeme seçimi	1
H	F,G	İşin tamamlanması	1



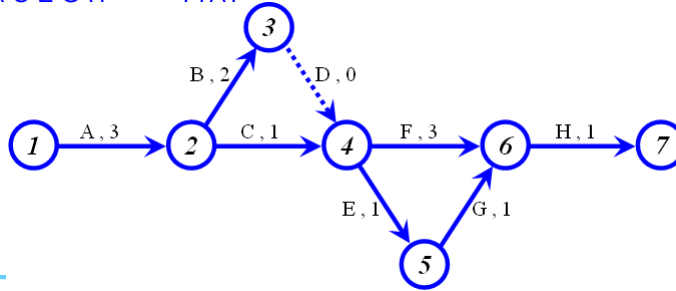
Yöneylem Araştırma

308

[CPM (Şebeke Hesaplamaları)]



- Şebekede, projenin ilk düğümünden son düğümüne kadar uzanan **birden fazla yol vardır**. Örneğin ev inşaatı projesinde, 1 düğümünden 2 düğümüne giden bir yol (**A faaliyeti**) olmakla birlikte, 2 düğümünde yani **A faaliyeti** tamamlanınca iki faaliyet birden başlamakta (**B ve C faaliyetleri**) ve bu noktadan itibaren iki yol oluşmaktadır.
- Buna göre bu projede **dört yol** (faaliyetler zinciri) bulunmaktadır ve bu yollar ile bu yollar üzerindeki faaliyetlerin toplam süreleri şöyledir;
 1. A-B-D-F-H : 9AY
 2. A-B-D-E-G-H : 8 AY
 3. A - C - F - H : 8 AY
 4. A-C-E-G-H : 7AY



Yöneylem Araştırması II

309

[CPM (Şebeke Hesaplamaları)]



- Projenin tamamlanması demek, projedeki bütün faaliyetlerin tamamen bitmiş olması demektir.
- O halde **projenin süresi** direkt olarak **en uzun** yolun süresine eşit olacaktır.
- Süresi en uzun olan bu yola Kritik Yol (Yörünge), bu yol üzerindeki faaliyetlere **Kritik Faaliyet** adı verilir.
- Burada kritik kelimesi, projenin belirlenen zamanda tamamlanabilmesi için zamanında tamamlanması, dolayısıyla hassasiyetle üzerinde durulması gereken faaliyetleri tanımlamaktadır. Bir şebekede **birden fazla** kritik yol bulunabilir. Örnekte

$$\{Projenin\ Süresi\} = \{Kritik\ Yolun\ Süresi\} = 9\ Ay$$
- Kritik yolların sürelerine ilişkin olarak kritik yol ile ilgili yapılan bu tanım her zaman yeterli olmamaktadır. Kritik yolun kesin olarak belirlenebilmesi için aşağıdaki hesaplamaların yapılması gerekir.
 1. **Olay zamanları**
 - En Erken Ortaya Çıkma Zamanı
 - En Geç Ortaya Çıkma Zamanı
 2. **Faaliyetlerin bollukları**
 - Toplam Bolluk
 - Serbest Bolluk

Yöneylem Araştırması II

310

[CPM (Şebeke Hesaplamaları)]



1. En Erken Ortaya Çıkma Zamanı, ET_j (Earliest Occurrence)

- j olayının “en erken ortaya çıkma zamanı” ET_j , bu olayın gerçekleşebileceği en erken zamandır.
- $ET_0=0$ alınır ve başlangıç olayından başlanarak şebekedeki tüm düğümler için ileriye doğru aşağıdaki formül kullanılarak hesaplanır.

t_{ij} : i ve j olayları arasında yer alan

faaliyetin süresi

$$ET_j = \max_i \{ET_i + t_{ij}\}$$

i : Sırayla j olayında sona eren bütün faaliyetlerin başlangıç olay numarası

- j olayının ET_j değerini hesaplamak için, j olayına yönelen tüm (i, j) faaliyetlerinin başlangıç olaylarının ET_i değerini öncelikle hesaplamak gerekir.

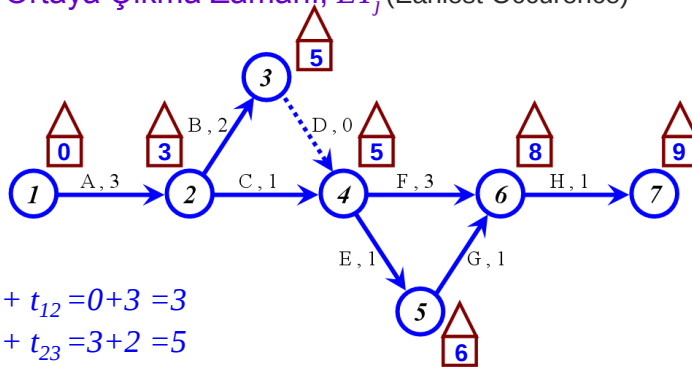
Yöneylem Araştırması II

311

[CPM (Şebeke Hesaplamaları)]



1. En Erken Ortaya Çıkma Zamanı, ET_j (Earliest Occurrence)



- $ET_1 = 0$
- $ET_2 = ET_1 + t_{12} = 0 + 3 = 3$
- $ET_3 = ET_2 + t_{23} = 3 + 2 = 5$
- $ET_4 = \max \{ET_2 + t_{24}, ET_3 + t_{34}\} = \max\{3+1, 5+0\} = 5$
- $ET_5 = ET_4 + t_{45} = 5 + 1 = 6$
- $ET_6 = \max \{ET_4 + t_{46}, ET_5 + t_{56}\} = \max\{5+3, 6+1\} = 8$
- $ET_7 = ET_6 + t_{67} = 8 + 1 = 9$ (Proje süresi)

Yöneylem Araştırması II

312

[CPM (Şebeke Hesaplamaları)]



2. En Geç Ortaya Çıkma Zamanı, LT_i (Latest Occurrence)

- i olayının “En Geç Ortaya Çıkma Zamanı” LT_i , proje süresinin geciktirmeyecek şekilde bu olayın gerçekleşebileceği en geç zamandır.
- $LT_n = ET_n$ alınır ve şebekenin son olayından başlanarak şebekedeki tüm düğümler için geriye doğru aşağıdaki formül kullanılarak hesaplanır.

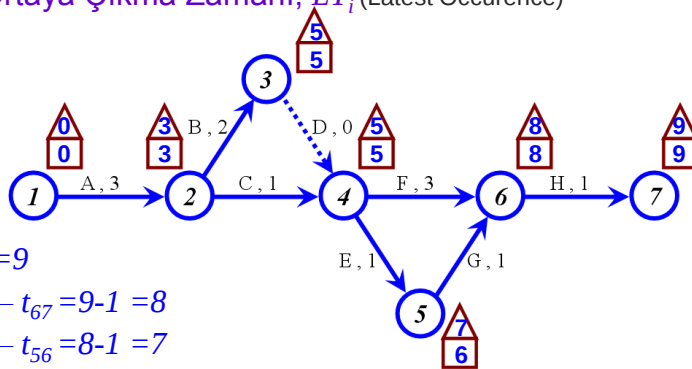
$$LT_i = \min_j \{LT_j - t_{ij}\}$$

- i olayının LT_i değerini hesaplamak için, i olayından çıkan tüm (i, j) faaliyetlerinin bitiş olaylarının LT_j değerini öncelikle hesaplamak gerekir.

[CPM (Şebeke Hesaplamaları)]



2. En Geç Ortaya Çıkma Zamanı, LT_i (Latest Occurrence)



- $LT_7 = ET_7 = 9$
- $LT_6 = LT_7 - t_{67} = 9 - 1 = 8$
- $LT_5 = LT_6 - t_{56} = 8 - 1 = 7$
- $LT_4 = \min \{LT_6 - t_{46}, LT_5 - t_{45}\} = \min \{8 - 3, 7 - 1\} = 5$
- $LT_3 = LT_4 - t_{34} = 5 - 0 = 5$
- $LT_2 = \min \{LT_4 - t_{24}, LT_3 - t_{23}\} = \min \{5 - 1, 5 - 2\} = 3$
- $LT_1 = LT_2 - t_{12} = 3 - 3 = 0$

[CPM (Şebeke Hesaplamaları)]



Faaliyetlerin Bollukları

- Kritik **olmayan** faaliyetler bolluklu faaliyetlerdir.
- Bolluklar faaliyetin **kapasite fazlalığından** kaynaklanır.
- Faaliyetlerin bolluğu aşağıdaki **iki kavramı** kapsar;
 1. Toplam bolluk (TB)
 2. Serbest bolluk (SB)

[CPM (Şebeke Hesaplamaları)]



1. Toplam Bolluk (Total float, TB)

- Bollukların **en büyüğü** toplam bolluktur.
- **Toplam bolluk**, bir projenin tamamlanma zamanını **geciktirmeden**, (diğer hiçbir faaliyetin gecikmeyeceği varsayılarak) bir faaliyetin başlama veya bitişinin gecikebileceği süredir .
- Bir başka ifade ile, bir projenin tamamlanma zamanını geciktirmeden bir faaliyetin **uzatılabileceği** süredir.
- Toplam bolluğun sıfır olduğu faaliyetler **kritik** olurlar.
- Toplam bolluğu en büyük faaliyetlerden oluşan yol başlangıç ve bitiş düğümleri arasındaki **en kısa yoldur**.

[CPM (Şebeke Hesaplamaları)]



1. Toplam Bolluk (Total float, TB)

- Toplam bolluk, j olayının en geç ortaya çıkma zamanından, i olayının en erken ortaya çıkma zamanı ve faaliyet süresi (t_{ij}) çıkartılarak hesaplanır;

$$TB_{ij} = LT_j - ET_i - t_{ij} = LT_j - EC_{ij} = LS_{ij} - ET_i, (j > i)$$

- Burada LS_{ij} , (i,j) faaliyeti için en geç başlama (latest start) ve EC_{ij} en erken tamamlanma (earliest completion) zamanları olup aşağıdaki formüllerle hesaplanır:

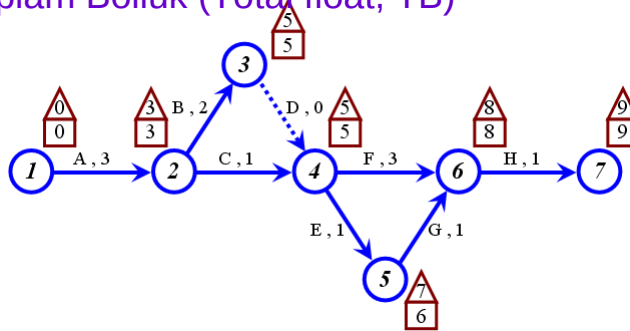
$$LS_{ij} = LT_j - t_{ij}$$

$$EC_{ij} = ET_i + t_{ij}$$

[CPM (Şebeke Hesaplamaları)]



1. Toplam Bolluk (Total float, TB)



$$\begin{aligned}
 TB_{12} &= LT_2 - ET_1 - t_{12} = 3 - 0 - 3 = 0 & TB_{46} &= LT_6 - ET_4 - t_{46} = 8 - 5 - 3 = 0 \\
 TB_{23} &= LT_3 - ET_2 - t_{23} = 5 - 3 - 2 = 0 & TB_{56} &= LT_6 - ET_5 - t_{56} = 8 - 6 - 1 = 1 \\
 TB_{24} &= LT_4 - ET_2 - t_{24} = 5 - 3 - 1 = 1 & TB_{67} &= LT_7 - ET_6 - t_{67} = 9 - 8 - 1 = 0 \\
 TB_{45} &= LT_5 - ET_4 - t_{45} = 7 - 5 - 1 = 1
 \end{aligned}$$

CPM (Şebeke Hesaplamaları)



2. Serbest Bolluk (Free Float, SB)

- İzleyen herhangi bir faaliyetin en erken başlama zamanını **geciktirmeden** bir faaliyetin başlama veya bitişinin (veya süresinin) gecikebileceği süredir.
- Faaliyetin en erken tamamlanma zamanından, faaliyetin en erken başlama zamanı ve faaliyetin süresini (t_{ij}) çıkartılarak hesaplanır:

$$SB_{ij} = ET_j - ET_i - t_{ij} \quad , (j > i)$$

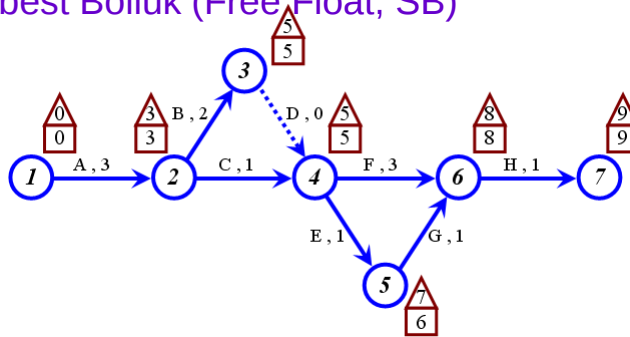
Yöneylem Araştırması II

319

CPM (Şebeke Hesaplamaları)



2. Serbest Bolluk (Free Float, SB)



$$SB_{12} = ET_2 - ET_1 - t_{12} = 3 - 0 - 3 = 0$$

$$SB_{23} = ET_3 - ET_2 - t_{23} = 5 - 3 - 2 = 0$$

$$SB_{24} = ET_4 - ET_2 - t_{24} = 5 - 3 - 1 = 1$$

$$SB_{45} = ET_5 - ET_4 - t_{45} = 6 - 5 - 1 = 0$$

$$SB_{46} = ET_6 - ET_4 - t_{46} = 8 - 5 - 3 = 0$$

$$SB_{56} = ET_6 - ET_5 - t_{56} = 8 - 6 - 1 = 1$$

$$SB_{67} = ET_7 - ET_6 - t_{67} = 9 - 8 - 1 = 0$$

Yöneylem Araştırması II

320

[CPM (Şebeke Hesaplamaları)]



- Her zaman $SB_{ij} \leq TB_{ij}$ dir.
- Eğer kritik olmayan bir (i,j) faaliyeti için $SB_{ij} = TB_{ij}$ ise faaliyet, proje süresinde bir gecikmeye meydan vermeksizin (ET_i, LT_j) aralığında herhangi bir zamana planlanabilir.
- $SB_{ij} < TB_{ij}$ ise, proje süresinde bir gecikmeye meydan vermeksizin faaliyetin başlaması serbest bolluktan daha fazla **geciktirilemez**. Serbest bolluktan daha büyük (toplam bolluktan küçük) bir gecikme j düğümünden başlayan bütün faaliyetlerin başlama zamanlarında **eşit sürede gecikmeye neden olur**.

Yöneylem Araştırması II

321

[CPM (Kritik Yolun Belirlenmesi)]



Kritik Yolun Özellikleri

- Kritik yolun süresi diğer yolların süresinden **fazladır**. Bu koşul kritik yol için **gerekli** ancak her zaman **yeterli değildir**. (Süresi diğer yollardan uzun olan **sadece** bir yol varsa, o zaman bu koşul yeterlidir, yani o yol kritiktir.)
- Kritik yol üzerindeki bütün olaylar (yani düğümler) için $LT_i = ET_i$ 'dir. Bu koşul kritik yol için **gerekli** ancak her zaman **yeterli değildir**.
- Kritik yol üzerindeki bütün faaliyetlerin **Toplam Bolluğu sıfırdır**. Bu koşul kritik yol için **hem gerekli hem de yeterlidir**.
- Özetle, bir (i,j) faaliyeti aşağıdaki üç şartı sağlıyorsa **kritik yol üzerindedir** yada **kritik faaliyet** denir;

$$ET_i = LT_i$$

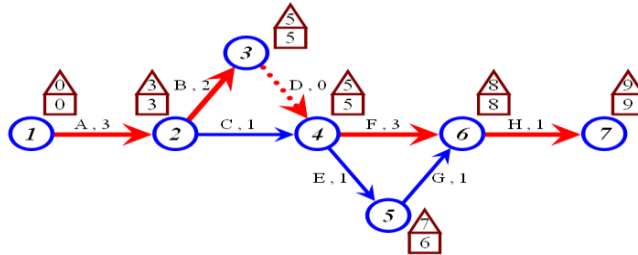
$$ET_j = LT_j$$

$$ET_j - ET_i = LT_j - LT_i = t_{ij}$$

Yöneylem Araştırması II

322

[CPM (Kritik Yolun Belirlenmesi)]



Faaliyet (i, j)	Süre (t _{ij})	En Erken		En Geç		Bolluklar	
		Başlama (ET _i)	Bitiş (EC _{ij})	Başlama (LS _{ij})	Bitiş (LT _j)	Toplam (TB _{ij})	Serbest (SB _{ij})
(1, 2)	3	0	3	0	3	0*	0
(2, 3)	2	3	5	3	5	0*	0
(2, 4)	1	3	4	4	5	1	1
(3, 4)	0	5	5	5	5	0*	0
(4, 5)	1	5	6	6	7	1	0
(4, 6)	3	5	8	5	8	0*	0
(5, 6)	1	6	7	7	8	1	1
(6, 7)	1	8	9	8	9	0*	0

■ * Kritik faaliyet

■ Kritik yol: 1-2-3-4-6-7 (veya A-B-D-F-H)

Yöneylem Araştırması II

323

[Kritik Yolun Doğrusal Programlama ile Bulunması]



■ Karar Değişkenleri

x_j : j olayının gerçekleşme zamanı

■ Amaç Fonksiyonu

- l düğümü başlangıç, s düğümü bitiş düğümü olsun. Amaç, projenin tamamlanması için gerekli **süreyi minimize etmek** olduğundan;

$$\min Z = x_s - x_l$$

■ Kısıtlar

- Projedeki her bir (i, j) faaliyeti için; i olayı j olayından **önce** gerçekleşmeli ve bu faaliyet **tamamlanmalıdır**. Öyleyse;

$$x_j \geq x_i + t_{ij} \quad (\text{Tüm } (i, j) \text{ faaliyetleri için})$$

■ Negatif Olmama Kısıtları

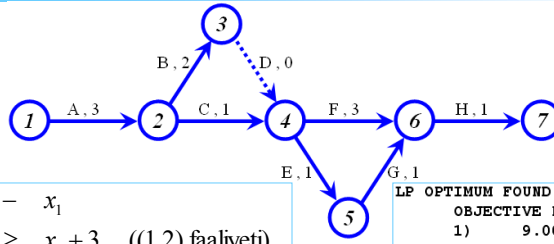
$$x_j \geq 0$$

- Kritik yol üzerindeki faaliyetlerle ilgili kısıtların gölge fiyatları **-1**'dir.

Yöneylem Araştırması II

324

Kritik Yolun Doğrusal Programlama ile Bulunması



$$\begin{aligned} \min Z &= x_7 - x_1 \\ \text{s.t.} \quad x_2 &\geq x_1 + 3 \quad ((1,2) \text{ faaliyeti}) \\ x_3 &\geq x_2 + 2 \quad ((2,3) \text{ faaliyeti}) \\ x_4 &\geq x_2 + 1 \quad ((2,4) \text{ faaliyeti}) \\ x_4 &\geq x_3 \quad ((3,4) \text{ faaliyeti}) \\ x_5 &\geq x_4 + 1 \quad ((4,5) \text{ faaliyeti}) \\ x_6 &\geq x_4 + 3 \quad ((4,6) \text{ faaliyeti}) \\ x_6 &\geq x_5 + 1 \quad ((5,6) \text{ faaliyeti}) \\ x_7 &\geq x_6 + 1 \quad ((6,7) \text{ faaliyeti}) \\ x_j &\geq 0 \quad (j = 1, 2, \dots, 7) \end{aligned}$$

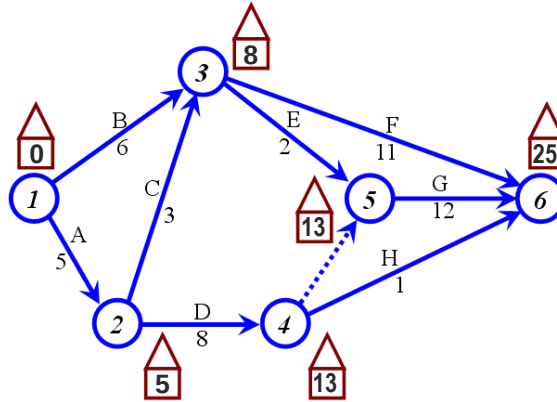
LP OPTIMUM FOUND AT STEP 1			
OBJECTIVE FUNCTION VALUE		1	
1)		9.000000	
VARIABLE	VALUE	REDUCED COST	
X7	9.000000	0.000000	
X1	0.000000	0.000000	
X2	3.000000	0.000000	
X3	5.000000	0.000000	
X4	5.000000	0.000000	
X5	7.000000	0.000000	
X6	8.000000	0.000000	
ROW	SLACK OR SURPLUS	DUAL PRICES	
2)	0.000000	-1.000000	
3)	0.000000	-1.000000	
4)	1.000000	0.000000	
5)	0.000000	-1.000000	
6)	1.000000	0.000000	
7)	0.000000	-1.000000	
8)	0.000000	0.000000	
9)	0.000000	-1.000000	

Yöneylem Araştırması II

Kritik yol : (1,2) - (2,3) - (3,4) - (4,6) - (6,7)

325

Zaman Çizelgelemesi

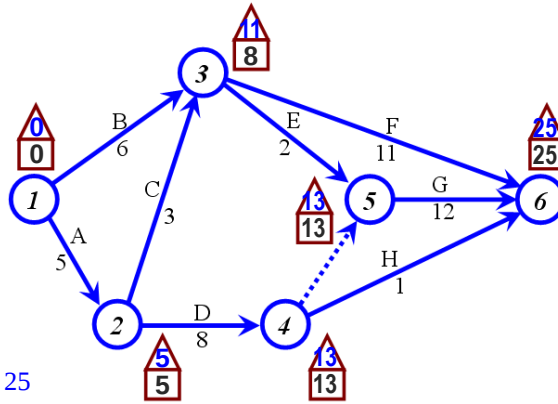


- $ET_1 = 0$
- $ET_2 = ET_1 + t_{12} = 0 + 5 = 5$
- $ET_3 = \max\{ET_1 + t_{13}, ET_2 + t_{23}\} = \max\{0 + 6, 5 + 3\} = 8$
- $ET_4 = ET_2 + t_{24} = 5 + 8 = 13$
- $ET_5 = \max\{ET_3 + t_{35}, ET_4 + t_{45}\} = \max\{8 + 2, 13 + 0\} = 13$
- $ET_6 = \max\{ET_3 + t_{36}, ET_4 + t_{46}, ET_5 + t_{56}\} = \max\{8 + 11, 13 + 1, 13 + 12\} = 25$

Yöneylem Araştırması II

326

Zaman Çizelgelemesi

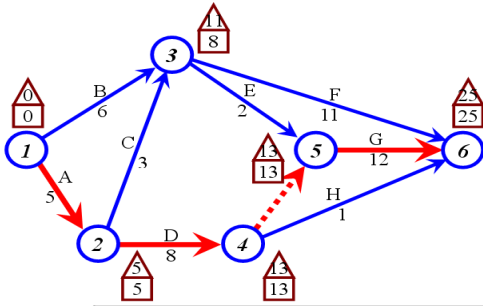


- $LT_6 = ET_6 = 25$
- $LT_5 = LT_6 - t_{56} = 25 - 12 = 13$
- $LT_4 = \min\{LT_5 - t_{45}, LT_6 - t_{46}\} = \min\{13 - 0, 25 - 1\} = 13$
- $LT_3 = \min\{LT_5 - t_{35}, LT_6 - t_{36}\} = \min\{13 - 2, 25 - 11\} = 11$
- $ET_2 = \min\{LT_3 - t_{23}, LT_4 - t_{24}\} = \min\{11 - 3, 13 - 8\} = 5$
- $ET_1 = \min\{LT_2 - t_{12}, LT_3 - t_{13}\} = \min\{5 - 5, 11 - 6\} = 0$

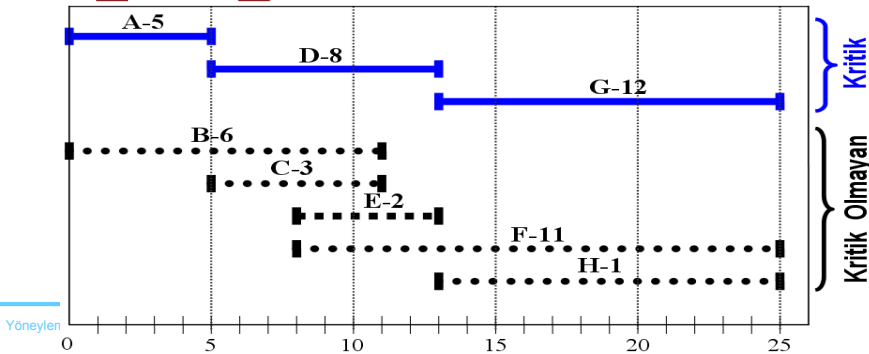
Yöneylem Araştırması II

327

Zaman Çizelgelemesi



- Projenin zaman çizelgesini hazırlamak için öncelikle, **kritik faaliyetler** 25 günlük proje süresi içinde bitirilecek şekilde **bir biri ardı sıra** olduğu gibi **sürekli çizgi** ile çizelgenir.
- **Kritik olmayan faaliyetler** ise tamamlanmaları gereken zaman aralıkları içinde **kesikli çizgi** ile çizelgenir.



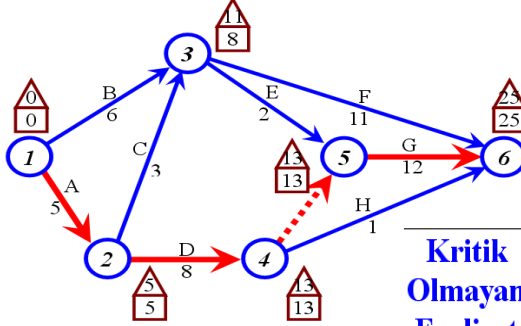
Yöneylem

328

[Zaman Çizelgelemesi]



- Kritik olmayan faaliyetlerin bollukları



$$TB_{ij} = LT_j - ET_i - t_{ij} \quad , (j > i)$$

$$SB_{ij} = ET_j - ET_i - t_{ij} \quad , (j > i)$$

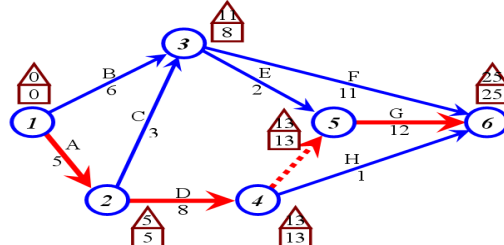
Yöneylem Araştırması II

Kritik Olmayan Faaliyet	Süresi	Toplam Bolluk (TB)	Serbest Bolluk (SB)
B (1, 3)	6	11-0-6=5	8-0-6=2
C (2, 3)	3	11-5-3=3	8-5-3=0
E (3, 5)	2	13-8-2=3	13-8-11=3
F (3, 6)	11	25-8-11=6	25-8-11=6
H (4, 6)	1	25-13-1=11	25-13-1=11

[Zaman Çizelgelemesi]



Kritik Olmayan Faaliyet	Süresi	Toplam Bolluk (TB)	Serbest Bolluk (SB)
B (1, 3)	6	11-0-6=5	8-0-6=2
C (2, 3)	3	11-5-3=3	8-5-3=0
E (3, 5)	2	13-8-2=3	13-8-11=3
F (3, 6)	11	25-8-11=6	25-8-11=6
H (4, 6)	1	25-13-1=11	25-13-1=11



- B ve C faaliyetlerinin serbest bolluğu toplam bolluğundan **küçüktür**. Diğer bütün faaliyetlerin serbest ve toplam bollukları eşittir.
- B faaliyetinin **toplam bolluğu 5** gündür ve en erken **sıfırıncı** gün, en geç **5nci** gün başlayabilir. Bununla birlikte **serbest bolluğu 2 gün** olduğundan, B'in başlangıcı 0 ile 2nci gün arasında herhangi bir zamanda olabilir. Bu yolla, bu faaliyeti izleyen E ve F faaliyetleri üzerinde hiçbir etki yaratmayacaktır. B'in başlaması, 2nci günü ne kadar geçerse, E ve F faaliyetlerinin başlaması da o kadar gecikecektir.
- C faaliyetinin serbest bolluğu ise 0 olduğundan, C'in en erken başlama zamanında başlamaması durumunda meydana gelecek gecikme, E ve F faaliyetlerini de aynı süre kadar geciktirecektir.

Yöneylem Araştırması II

330

[PERT Metodu]



- **Kritik Yol Metodunun** uygulanabilmesi için faaliyet sürelerinin **kesin olarak** bilinmesi gerekmektedir.
- Faaliyetlerin sürelerinin ne olacağı konusunda önemli bir belirsizlik varsa ve bu süre bir olasılık dağılımına sahip rasgele değişkense **PERT** kullanılabilir.
- **PERT** tekniğinde her değişken (faaliyet süresi) için kullanılan üç zaman tahmini şunlardır;
 - **a** = İyimser (optimistic) zaman
 - **b** = Kötümser (pessimistic) zaman
 - **m** = En muhtemel (most likely) zaman
- **PERT** 'de faaliyet sürelerinin **Beta Dağılımı** özelliği gösterebileceği doğrulanmış ve bu sürelerle ilişkin **ortalama** ($\bar{D}$) ve **varyans** (V) değerlerinin aşağıdaki gibi bulunabileceği ifade edilmiştir.

$$\bar{D} = \frac{a + 4m + b}{6}$$

$$V = \left(\frac{b - a}{6} \right)^2$$

Yöneylem Araştırması II

331

[PERT Metodu]



- μ_i : i olayının **en erken ortaya çıkma zamanı** olsun. Şebekedeki tüm faaliyetlerin istatistiksel bağımsız olduğu varsayımıyla;
 - μ_i 'in **ortalaması** ($E\{\mu_i\}$) i olayına giden **en uzun yol** üzerindeki faaliyetlerin beklenen (ortalama) sürelerinin ($\bar{D}$) **toplamına**,
 - μ_i 'in **varyansı** ($Var\{\mu_i\}$) i olayına giden **en uzun yol** üzerindeki faaliyetlerin **varyanslarının** toplamına eşittir.

- **Özetle**, seçilen bir yol için;

$$E\{\mu_i\} = ES_i$$

$$Var\{\mu_i\} = \sum_k V_k$$

Burada k , i olayına giden en uzun yol üzerindeki faaliyetleri göstermektedir.

Yöneylem Araştırması II

332

PERT Metodu



- μ_i , bağımsız random değişkenlerin toplamı olduğundan, **Merkezi Limit Teoremine** göre ortalaması $E\{\mu_i\}$, varyansı $Var\{\mu_i\}$ olan bir **normal dağılım özelliği gösterir**. μ_i en erken ortaya çıkma zamanını temsil ettiğinden, i olayının analist tarafından belirlenen **planlanan bir zamanda** (scheduled time, ST_i) **bitme ihtimali**;

$$P\{\mu_i \leq ST_i\} = P\left\{\frac{\mu_i - E\{\mu_i\}}{\sqrt{Var\{\mu_i\}}} \leq \frac{ST_i - E\{\mu_i\}}{\sqrt{Var\{\mu_i\}}}\right\} = P\{Z \leq K_i\}$$

şeklinde bulunur. Burada Z , ortalaması sıfır, varyansı 1 olan **standart normal dağılımı** ifade etmekte, K_i 'de

$$K_i = \frac{ST_i - E\{\mu_i\}}{\sqrt{Var\{\mu_i\}}}$$

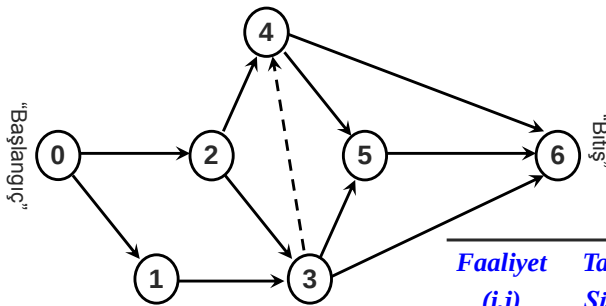
şeklinde belirlenmektedir.

- Burada genel eğilim bir i olayının, LC_i süresini **aşmadan** ortaya çıkma ihtimalini hesaplamaktır. Bu ihtimaller aynı zamanda izleyen i olaylarının (ES_i, LC_i) **süreleri içerisinde** ortaya çıkma şanslarını temsil etmektedir.

Yöneylem Araştırması II

333

PERT Metodu (Örnek)

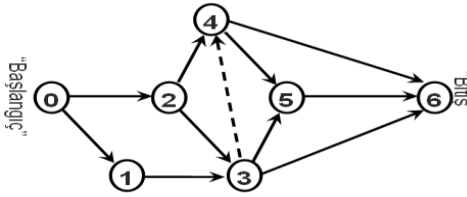


Faaliyet (i,j)	Tahmin Süreler (a,b,m)	Faaliyet (i,j)	Tahmin Süreler (a,b,m)
(0, 1)	(1, 3, 2)	(3, 5)	(1, 7, 2.5)
(0, 2)	(2, 8, 2)	(3, 6)	(1, 3, 2)
(1, 3)	(2, 3, 2)	(4, 5)	(6, 8, 7)
(2, 3)	(11, 11, 1.5)	(4, 6)	(3, 11, 4)
(2, 4)	(0.5, 7.5, 1)	(5, 6)	(4, 8, 6)

Yöneylem Araştırması II

334

[PERT Metodu]



Faaliyet (i,j)	Tahmin Süreler (a,b,m)	Faaliyet (i,j)	Tahmin Süreler (a,b,m)
(0, 1)	(1, 3, 2)	(3, 5)	(1, 7, 2.5)
(0, 2)	(2, 8, 2)	(3, 6)	(1, 3, 2)
(1, 3)	(2, 3, 2)	(4, 5)	(6, 8, 7)
(2, 3)	(11, 11, 1.5)	(4, 6)	(3, 11, 4)
(2, 4)	(0.5, 7.5, 1)	(5, 6)	(4, 8, 6)

■ D_{ij} ve V_{ij} Hesaplamaları

- Örneğin (0, 2) faaliyeti için

$$D_{02} = \frac{a + 4m + b}{6} = \frac{2 + 4(2) + 8}{6} = 3,33 \approx 3$$

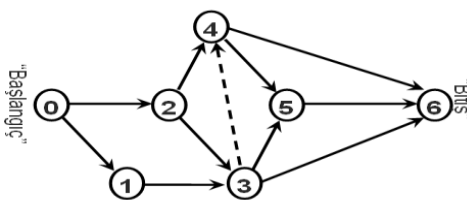
$$V_{02} = \left(\frac{b-a}{6}\right)^2 = \left(\frac{8-2}{6}\right)^2 = 1$$

Faaliyet (i,j)	D_i	V_{ij}	Faaliyet (i,j)	D_i	V_{ij}
(0, 1)	2	0.11	(3, 5)	3	1.00
(0, 2)	3	1.00	(3, 6)	2	0.11
(1, 3)	2	0.11	(4, 5)	7	0.11
(2, 3)	3	2.78	(4, 6)	5	1.78
(2, 4)	2	1.36	(5, 6)	6	0.44

Yöneylem Araştırması II

335

[PERT Metodu]



Faaliyet (i,j)	Tahmin Süreler (a,b,m)	Faaliyet (i,j)	Tahmin Süreler (a,b,m)
(0, 1)	(1, 3, 2)	(3, 5)	(1, 7, 2.5)
(0, 2)	(2, 8, 2)	(3, 6)	(1, 3, 2)
(1, 3)	(2, 3, 2)	(4, 5)	(6, 8, 7)
(2, 3)	(11, 11, 1.5)	(4, 6)	(3, 11, 4)
(2, 4)	(0.5, 7.5, 1)	(5, 6)	(4, 8, 6)

■ Düğüm 0

$$E(\mu_0) = 0, \quad Var(\mu_0) = 0$$

■ Düğüm 1

$$E(\mu_1) = E(\mu_0) + D_{01} = 0 + 2 = 2$$

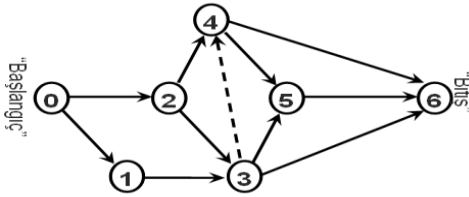
Önceki düğüm : 0 Yol : (0,1)

$$Var(\mu_1) = Var(\mu_0) + V_{01} = 0 + 0.11 = 0.11$$

Yöneylem Araştırması II

336

[PERT Metodu]



Faaliyet (i,j)	Tahmin Süreler (a,b,m)	Faaliyet (i,j)	Tahmin Süreler (a,b,m)
(0, 1)	(1, 3, 2)	(3, 5)	(1, 7, 2.5)
(0, 2)	(2, 8, 2)	(3, 6)	(1, 3, 2)
(1, 3)	(2, 3, 2)	(4, 5)	(6, 8, 7)
(2, 3)	(11, 11, 1.5)	(4, 6)	(3, 11, 4)
(2, 4)	(0.5, 7.5, 1)	(5, 6)	(4, 8, 6)

■ Düğüm 2

$$E(\mu_2) = E(\mu_0) + D_{02} = 0 + 3 = 3$$

$$\text{Önceki düğüm : 0} \quad \text{Yol : (0,2)}$$

$$Var(\mu_2) = Var(\mu_0) + V_{02} = 0 + 1.00 = 1.00$$

■ Düğüm 3

$$E(\mu_3) = \max\{E(\mu_1) + D_{13}, E(\mu_2) + D_{23}\} = \max\{2 + 2, \underline{3 + 3}\} = 6$$

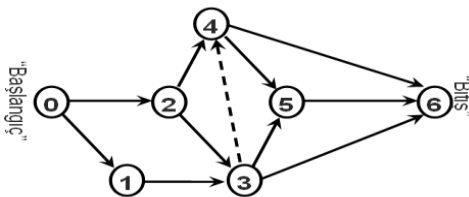
$$\text{Önceki düğüm : 2} \quad \text{Yol : (0,2,3)}$$

$$Var(\mu_3) = Var(\mu_2) + V_{23} = 1.00 + 2.78 = 3.78$$

Yöneylem Araştırması II

337

[PERT Metodu]



Faaliyet (i,j)	Tahmin Süreler (a,b,m)	Faaliyet (i,j)	Tahmin Süreler (a,b,m)
(0, 1)	(1, 3, 2)	(3, 5)	(1, 7, 2.5)
(0, 2)	(2, 8, 2)	(3, 6)	(1, 3, 2)
(1, 3)	(2, 3, 2)	(4, 5)	(6, 8, 7)
(2, 3)	(11, 11, 1.5)	(4, 6)	(3, 11, 4)
(2, 4)	(0.5, 7.5, 1)	(5, 6)	(4, 8, 6)

■ Düğüm 4

$$E(\mu_4) = \max\{E(\mu_2) + D_{24}, E(\mu_3) + D_{34}\} = \max\{3 + 2, \underline{6 + 0}\} = 6$$

$$\text{Önceki düğüm : 3} \quad \text{Yol : (0,2,3,4)}$$

$$Var(\mu_4) = Var(\mu_3) + V_{34} = 3.78 + 0.00 = 3.78$$

■ Düğüm 5

$$E(\mu_5) = \max\{E(\mu_3) + D_{35}, E(\mu_4) + D_{45}\} = \max\{6 + 3, \underline{6 + 7}\} = 13$$

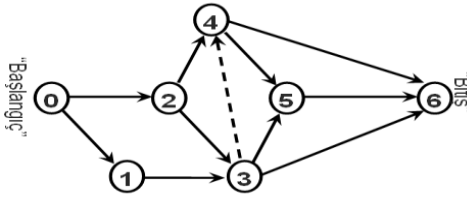
$$\text{Önceki düğüm : 4} \quad \text{Yol : (0,2,3,4,5)}$$

$$Var(\mu_5) = Var(\mu_4) + V_{45} = 3.78 + 0.11 = 3.89$$

Yöneylem Araştırması II

338

[PERT Metodu]



<i>Faaliyet</i> <i>(i,j)</i>	<i>Tahmin</i> <i>Süreler</i> <i>(a,b,m)</i>	<i>Faaliyet</i> <i>(i,j)</i>	<i>Tahmin</i> <i>Süreler</i> <i>(a,b,m)</i>
(0, 1)	(1, 3, 2)	(3, 5)	(1, 7, 2.5)
(0, 2)	(2, 8, 2)	(3, 6)	(1, 3, 2)
(1, 3)	(2, 3, 2)	(4, 5)	(6, 8, 7)
(2, 3)	(11, 11, 1.5)	(4, 6)	(3, 11, 4)
(2, 4)	(0.5, 7.5, 1)	(5, 6)	(4, 8, 6)

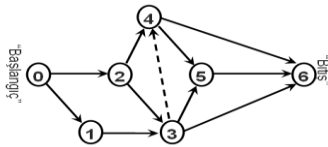
■ Düğüm 6

$$E(\mu_6) = \max\{E(\mu_3) + D_{36}, E(\mu_4) + D_{46}, E(\mu_5) + D_{56}\} = \max\{6 + 2, 6 + 5, 13 + 6\} = 19$$

Önceki düğüm : 5 Yol : (0,2,3,4,5,6)

$$Var(\mu_6) = Var(\mu_5) + V_{56} = 3.89 + 0.44 = 4.33$$

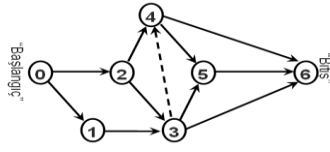
[Pert Metodu]



<i>Faaliyet</i>	<i>D_{ij}</i>	<i>V_{ij}</i>	<i>Faaliyet</i>	<i>D_{ij}</i>	<i>V_{ij}</i>
(0, 1)	2	0.11	(3, 5)	3	1.00
(0, 2)	3	1.00	(3, 6)	2	0.11
(1, 3)	2	0.11	(4, 5)	7	0.11
(2, 3)	3	2.78	(4, 6)	5	1.78
(2, 4)	2	1.36	(5, 6)	6	0.44

<i>Olay (i)</i>	<i>Yol</i>	<i>E(μ_i)</i>	<i>Var(μ_i)</i>
1	(0, 1)	2	0.11
2	(0, 2)	3	1.00
3	(0, 2, 3)	6	3.78
4	(0, 2, 3, 4)	6	3.78
5	(0, 2, 3, 4, 5)	13	3.89
6	(0, 2, 3, 4, 5, 6)	19	4.33

[Pert Metodu]



Olay (i)	Yol	$E(\mu_i)$	$Var(\mu_i)$
1	(0, 1)	2	0.11
2	(0, 2)	3	1.00
3	(0, 2, 3)	6	3.78
4	(0, 2, 3, 4)	6	3.78
5	(0, 2, 3, 4, 5)	13	3.89
6	(0, 2, 3, 4, 5, 6)	19	4.33

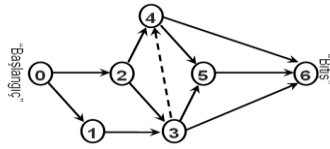
■ Örneğin $i=1$ olayının $ST_1=4$ günde ortaya çıkma ihtimali aşağıdaki gibi hesaplanır.

$$K_1 = \frac{ST_1 - E\{\mu_1\}}{\sqrt{Var\{\mu_1\}}} = \frac{4 - 2}{\sqrt{0.11}} = 6.03$$

$$P(Z \leq K_1) = P(Z \leq 6.03) = 1.000 \text{ (Standart normal dağılım tablosundan okunmuştur)}$$



[Pert Metodu]

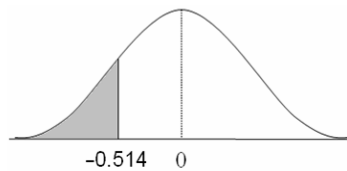


Olay (i)	Yol	$E(\mu_i)$	$Var(\mu_i)$
1	(0, 1)	2	0.11
2	(0, 2)	3	1.00
3	(0, 2, 3)	6	3.78
4	(0, 2, 3, 4)	6	3.78
5	(0, 2, 3, 4, 5)	13	3.89
6	(0, 2, 3, 4, 5, 6)	19	4.33

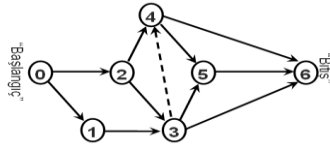
■ Örneğin $i=3$ olayının $ST_3=5$ günde ortaya çıkma ihtimali aşağıdaki gibi hesaplanır.

$$K_3 = \frac{ST_3 - E\{\mu_3\}}{\sqrt{Var\{\mu_3\}}} = \frac{5 - 6}{\sqrt{3.78}} = -0.514$$

$$P(Z \leq K_3) = P(Z \leq -0.514) = 0.304 \text{ (Standart normal dağılım tablosundan okunmuştur)}$$



[PERT Metodu]



Faaliyet	D_{ij}	V_{ij}	Faaliyet	D_{ij}	V_{ij}
(0, 1)	2	0.11	(3, 5)	3	1.00
(0, 2)	3	1.00	(3, 6)	2	0.11
(1, 3)	2	0.11	(4, 5)	7	0.11
(2, 3)	3	2.78	(4, 6)	5	1.78
(2, 4)	2	1.36	(5, 6)	6	0.44

Olay (i)	Yol	$E\{\mu_i\}$	$Var\{\mu_i\}$	ST_i	K_i	$P\{Z \leq K_i\}$
1	(0, 1)	2	0.11	4	6.03	1.000
2	(0, 2)	3	1.00	2	-1.00	0.159
3	(0, 2, 3)	6	3.78	5	-0.514	0.304
4	(0, 2, 3, 4)	6	3.78	6	0	0.500
5	(0, 2, 3, 4, 5)	13	3.89	17	2.028	0.987
6	(0, 2, 3, 4, 5, 6)	19	4.33	20	0.480	0.684*

* Projenin 20 günde tamamlanma olasılığı

- Yol sütunundaki bilgiler başlangıç olayından ilgili olaya **en uzun yolu** temsil eder.

Yöneylem Araştırması II

343

[BÖLÜM VII: Doğrusal Olmayan Optimizasyon]

- Genel Kavramlar ve Tanım