TAMSAYILI PROGRAMLAMA

Doğrusal programlama problemlerinde sık sık çözümün tamsayı olması gereken durumlar ile karşılaşılır. Örneğin ele alınan problem masa, sandalye, otomobil vb. üretimlerinin optimum düzeyini bulmak ise bu durumda 1.72 masa, 5.6 sandalye veya 103.6 otomobil gibi çözümler anlamlı olmayacaktır. Bazı durumlarda bu değerleri en yakın tamsayıya yuvarlamak düşünülse de çözümü optimalden uzaklaştırabilir. Hatta uygun olmayan çözümler verebilir.

Tamsayılı programlama problemlerinin türleri

1. Tüm değişkenlerin tamsayı olması gerektiği problemlere «*saf tamsayılı programlama problemi*» denir.

$$Zmax = 11\,x_1 + 4\,x_2$$

$$7\,x_1 + 6\,x_2 \leq 84$$
$$4\,x_1 + 2\,x_2 \leq 32$$
$$x_1,\, x_2 \geq 0 \text{ ve } x_1,\, x_2 \text{ tamsayı}$$

2. Sadece bazı değişkenlerin tamsayı olması diğerlerinin ise reel sayı olması gerektiği problemlere «*karma tamsayılı programlama problemi*» denir.

$$Zmax = 11\,x_1 + 4\,x_2$$

$$7\,x_1 + 6\,x_2 \leq 84$$
$$4\,x_1 + 2\,x_2 \leq 32$$
$$x_1,\, x_2 \geq 0 \text{ ve } x_1 \text{ tamsayı}$$

3. Tüm değişkenlerin «0» ya da «1» e eşit olmasının istendiği bir tamsayılı probramlama problemine «*0-1 tamsayılı programlama problemi*» denir.

$$Z_{max} = 11\,x_1 + 4\,x_2$$

$$7\,x_1 + 6\,x_2 \leq 84$$
$$4\,x_1 + 2\,x_2 \leq 32$$
$$x_1,\, x_2 = 0 \text{ ya da } 1$$

ÇÖZÜM YÖNTEMLERİ

DAL SINIR YÖNTEMİ

Dal - sınır yöntemi, temelde tüm olurlu çözüm seçeneklerini belirlemeye yönelik bir tekniktir. Ancak optimal çözüme götürmeyen bazı çözüm seçenekleri önceden elimine edilmektedir. Bu nedenle gerekli değerlendirmelerin sayısı, genellikle çözüm alanını küçük alt setlere böler. Bu alt setlere " dallandırma noktaları" adı verilir..

Her alt set, daha fazla araştırma gerekip gerekmediği belirlenmek üzere değerlendirilir. Değerlendirme, amaç fonksiyon değerlerini sınırlarla karşılaştırarak gerçekleştirilir. Maliyet minimizasyonu sorunlarında alt setin olurlu çözümleri için amaç fonksiyon değerlerine bir alt sınır bulunur

Algoritmayı bir örnek problem üzerinde inceleyelim. Aşağıdaki doğrusal programlama probleminin grafik çözümünden sonuçların tam sayı çıkmadığı görülmektedir.

Örnek problem:

$$Zmax = 5X_1 + 4X_2$$

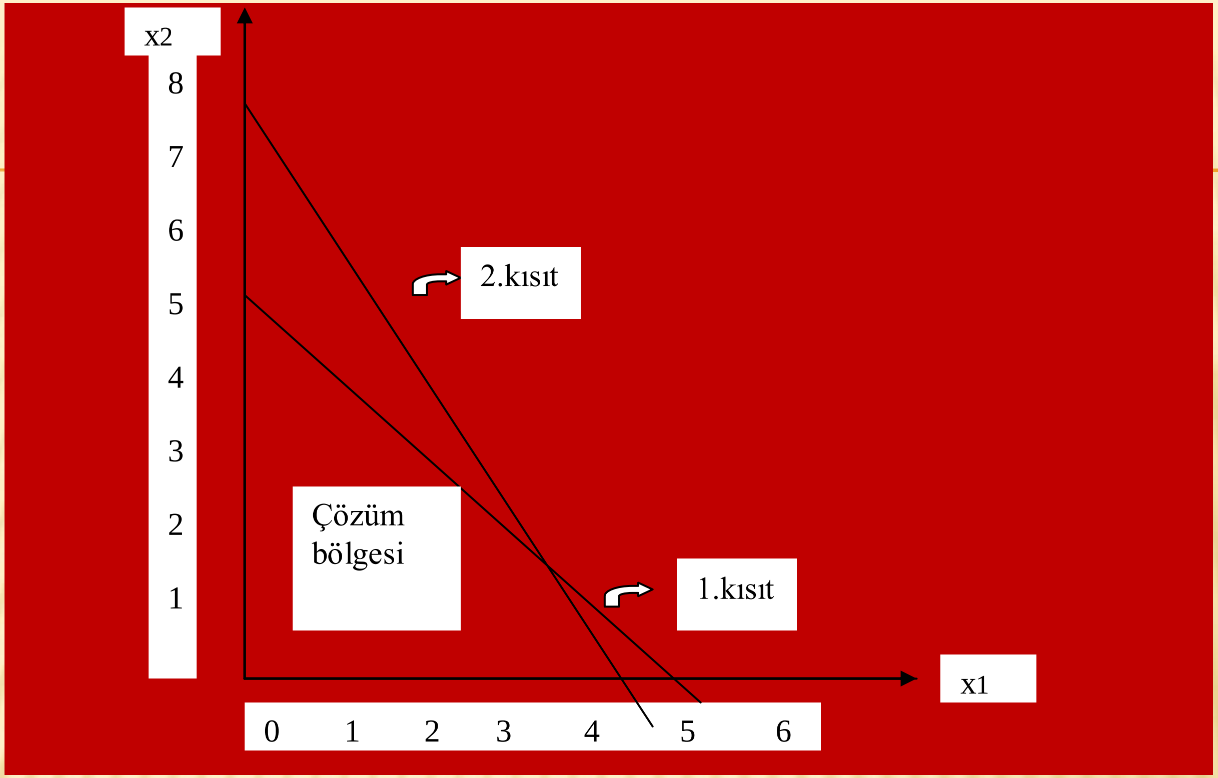
Kısıtlar:

$$X_1 + X_2 <= 5$$
$$10X_1 + 6X_2 <= 45$$

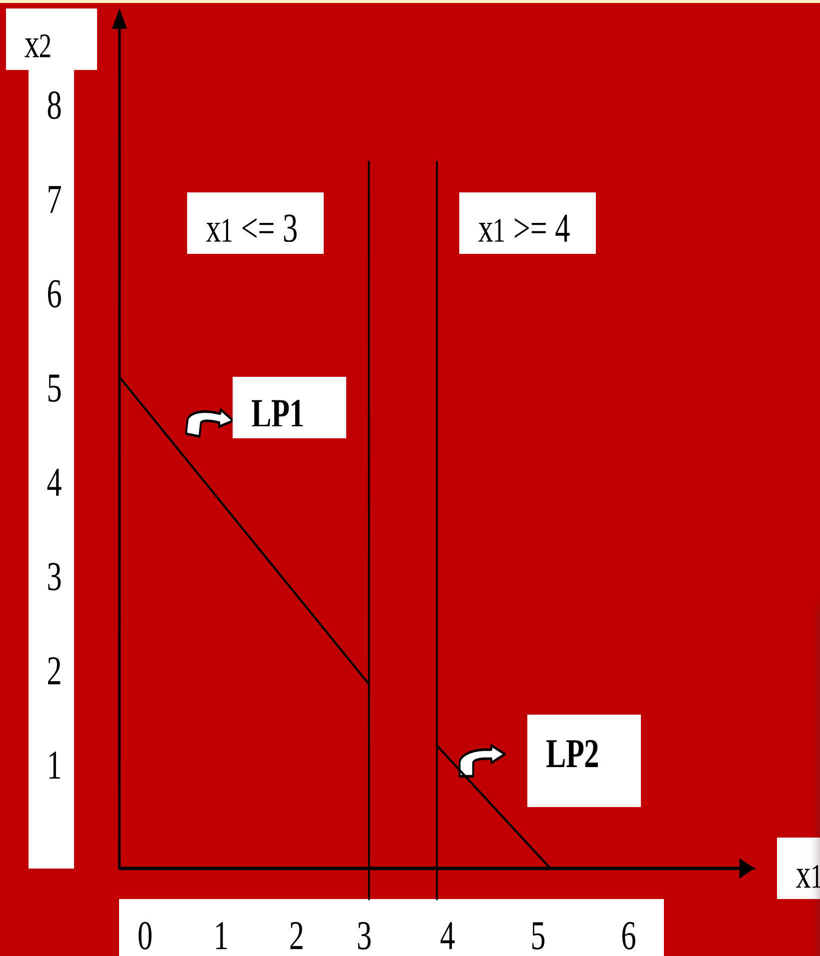
Pozitiflik koşulu:

$$X_1 , X_2 >= 0$$

problemin çözüm uzayı aşağıdaki gibidir



Problemin çözümünde : Z= 23.75, x1 = 3.75, x2 =1.25 çıkmaktadır. Değişkenler tam sayı çıkmadığı için dal-sınır algoritması ile optimum tam sayılı çözümü buluncaya kadar çözüm uzayının düzenlenmesi yapılacaktır. İlk aşamada LP0 çözümünde(DP çözümü) tam sayı değer almayan bir değişken rasgele seçilir. x1 değişkenini seçelim(x1=3.75) LP0 çözüm uzayının 3< x1 <4 bölgesinde tamsayılı değerler olmayacaktır dolayısıyla bu bölge elemine edilebilir.



LP1 uzayı = LP0 uzayı + (x1 < = 3)

LP2 uzayı = LP0 uzayı + (x1 >= 4)

Optimum çözüm ya LP1 uzayında ya da LP2 uzayında olacaktır. Her iki alt problem ayrı ayrı çözülmelidir. Önce LP1 problemini (x1 <=3) kısıtını ekleyerek çözelim.

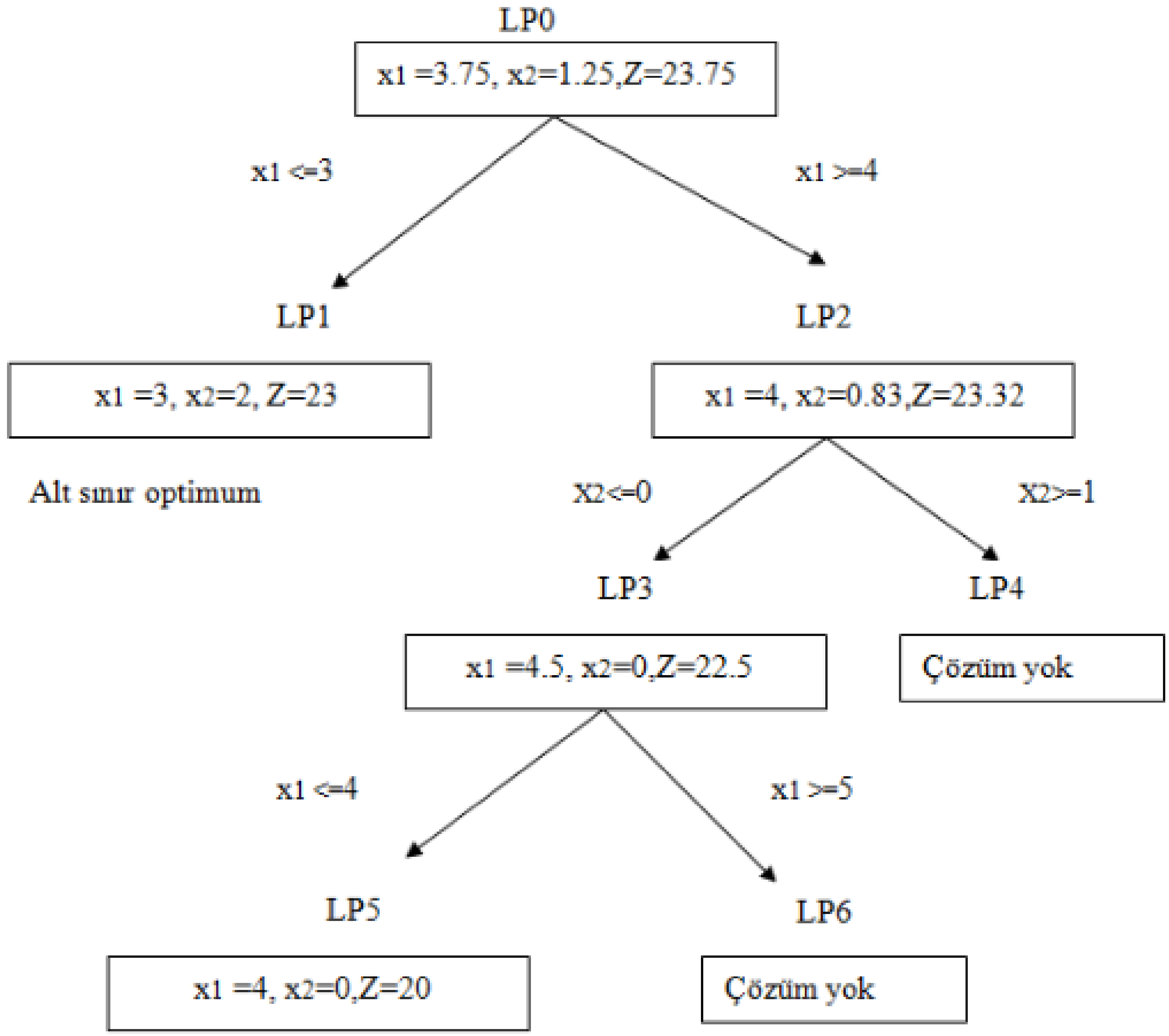
$$\max Z = 5X_1 + 4X_2$$
$$X_1 + X_2 \le 5$$
$$10X_1 + 6X_2 \le 45$$
$$x_1 \le 3$$
$$X_1, X_2 \ge 0$$

problem çözüldüğünde **Z= 23, x1 =3, x2 = 2** çıkmaktadır. O halde LP1 optimum değere ulaşmıştır diyebiliriz. Çözümdeki Z=23.75 değeri LP1 de Z=23 olarak çıktığına göre bu bir alt sınır olarak alınabilir. Bir tamsayılı çözüm elde edildiği için daha fazla ilerletmeye gerek yoktur. LP2 ye ait çözümde de **Z= 23.32 , x1 =4, x2=0.83** çıkmaktadır. x2=0.83 olduğundan yeniden bir dallanma yapılarak ;

x2<= 0 ve x2 >=1 kontrolü yapılabilir.

LP3 uzayı = LP2 uzayı + (x2 <= 0)

= LP0 uzayı +(x1 >=4)+ (x2<=0)



LP5 çözümünde de sonuç tamsayılı çıkmaktadır. Ancak LP1 çözümünde **Z = 23** alt sınır olarak alınırsa(en büyük alt sınır) bu çözümün optimum olmadığı söylenebilir. Burada hangi dalın seçilip önce çözülmesi konusunda kesin bir kural olmayıp seçim tahmini yapılmaktadır.

Örnek 2:

Bir mobilya şirketi masa ve sandalye üretmektedir. Bir masa 1 saat işçilik ve 9 m^2 ağaç, bir sandalye ise 1 saat işçilik 5 m^2 ağaç malzeme gerektirmektedir. Şu anda 6 saat işçilik ve 45 m^2 ağaç malzeme mevcuttur. Her bir masa 8$ ve her bir sandalye $5 kar getirmektedir. Şirketin karını maksimize eden modeli kurunuz ve çözünüz.

Karar değişkenleri

x1 = üretilecek masa sayısı
x2 = üretilecek sandalye sayısı

Amaç fonksiyonu

$Zmax = 8x_1 + 5x_2$

Kısıtlar

x1 + x2 ≤ 6 (işçilik kısıdı)
9x1 + 5x2 ≤ 45 (malzeme kısıdı)
x1, x2 ≥ 0 ve tamsayı

Çözüm

- LPO

	16:34:18		Thursday	December	27	2012		
	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
1	X1	3,7500	8,0000	30,0000	0	basic	5,0000	9,0000
2	X2	2,2500	5,0000	11,2500	0	basic	4,4444	8,0000
	Objective	Function	(Max.) =	41,2500				
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
1	C1	6,0000	<=	6,0000	0	1,2500	5,0000	9,0000
2	C2	45,0000	<=	45,0000	0	0,7500	30,0000	54,0000

X1 = 3,75
X2 = 2,25

Zmax= 41,25

- LP1

X1 + X2 ≤ 6 (işçilik kısıdı)
9x1 + 5X2 ≤ 45 (malzeme kısıdı)
X1 ≥ 4

- LP2

X1 + X2 ≤ 6 (işçilik kısıdı)
9x1 + 5X2 ≤ 45 (malzeme kısıdı)
X1 ≤ 3

- LP1

16:41:14		Thursday	December	27	2012		
Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
1 X1	4,0000	8,0000	32,0000	0	basic	-M	9,0000
2 X2	1,8000	5,0000	9,0000	0	basic	4,4444	M
Objective	Function	(Max.) =	41,0000				
Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
1 C1	5,8000	<=	6,0000	0,2000	0	5,8000	M
2 C2	45,0000	<=	45,0000	0	1,0000	36,0000	46,0000
3 C3	4,0000	>=	4,0000	0	-1,0000	3,7500	5,0000

X1 = 4
X2 = 1,8

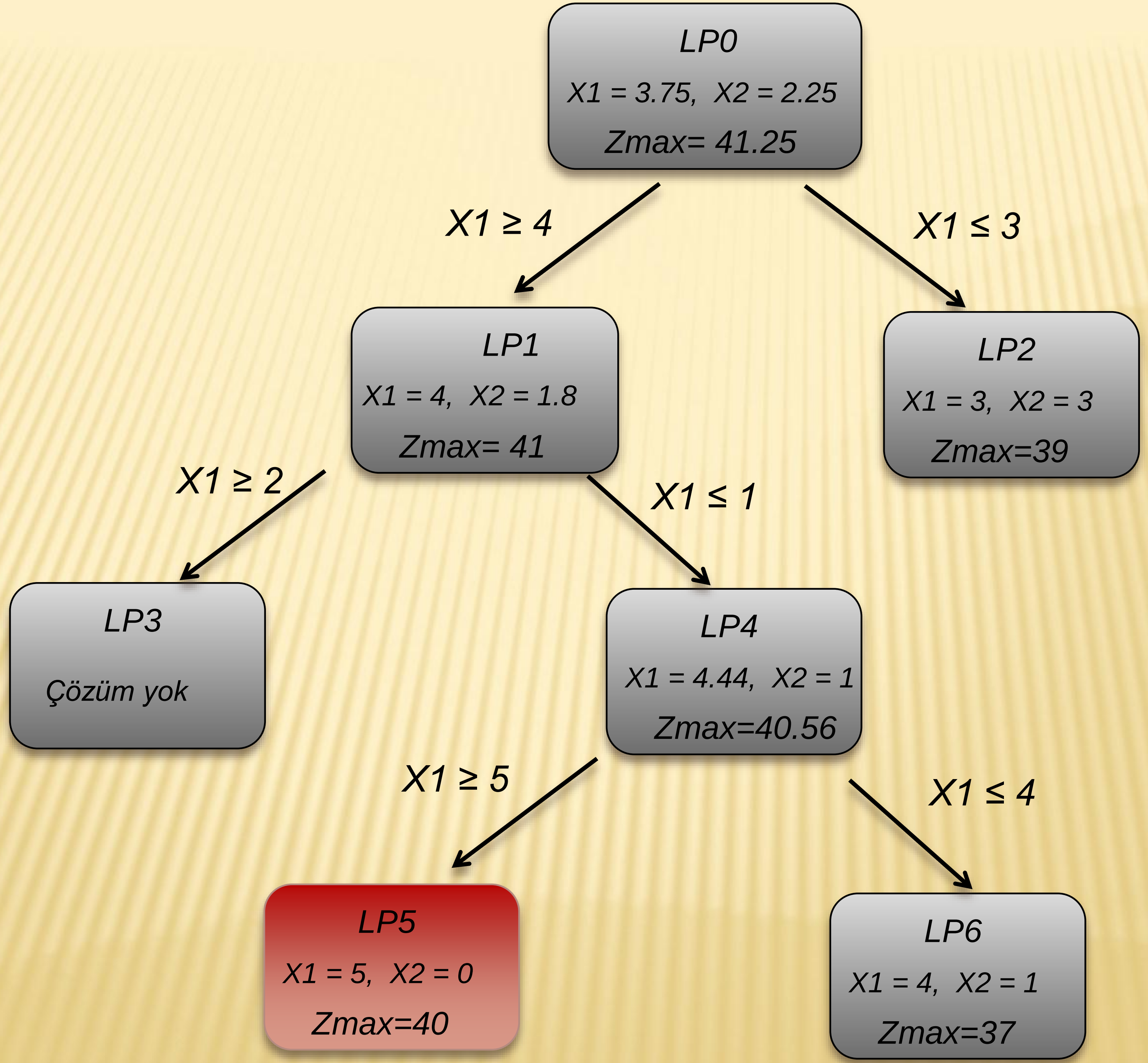
Zmax= 41

- LP2

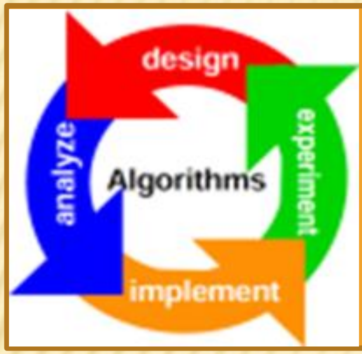
16:53:23		Thursday	December	27	2012		
Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status	Allowable Min. c(j)	Allowable Max. c(j)
1 X1	3,0000	8,0000	24,0000	0	basic	5,0000	M
2 X2	3,0000	5,0000	15,0000	0	basic	0	8,0000
Objective	Function	(Max.) =	39,0000				
Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price	Allowable Min. RHS	Allowable Max. RHS
1 C1	6,0000	<=	6,0000	0	5,0000	3,0000	6,6000
2 C2	42,0000	<=	45,0000	3,0000	0	42,0000	M
3 C3	3,0000	<=	3,0000	0	3,0000	0	3,7500

X1 = 3
X2 = 3

Zmax= 39


LP0
X1 = 3.75, X2 = 2.25
Zmax= 41.25
X1 ≥ 4
X1 ≤ 3
LP1
X1 = 4, X2 = 1.8
Zmax= 41
LP2
X1 = 3, X2 = 3
Zmax=39
X1 ≥ 2
X1 ≤ 1
LP3
Çözüm yok
LP4
X1 = 4.44, X2 = 1
Zmax=40.56
X1 ≥ 5
X1 ≤ 4
LP5
X1 = 5, X2 = 0
Zmax=40
LP6
X1 = 4, X2 = 1
Zmax=37

KESME YÖNTEMİ



DS Algoritmasındaki gibi, Kesme Düzlemi Algoritması da sürekli bir doğrusal programlama probleminin optimum çözümüyle başlar. Ancak, bu yöntemde dallanma ve sınırlamadan çok, kesme adı verilen özel kısıtlar ard arda oluşturularak çözüm uzayının düzenlenmesine gidilir

TAMSAYILI PROGRAMLAMA

- DAL-SINIR YÖNTEMİ

- KESME YÖNTEMİ

«YENİ KISITLAR EKLEYEREK ÇÖZÜM UZAYINI TAMSAYILARDAN OLUŞTURMAK»

Tamsayı programlama kesme tekniğini bir simpleks metodla gösterecek olursak.

$Z_{max}=3X_1+5X_2$

KISITLAR

$X_1+4X_2 \leq 9$

$2X_1+3X_2 \leq 11$

$Z_{max}=3X_1+5X_2 + 0S1+0S2$

BAŞLANGIÇ SİMPLEKS TABLO						
AMAÇ SÜTUNU	AMAÇ SATIRI cj		3	5	0	0
	TEMEL DEĞİŞKENLER	SABİTLER	X1	X2	S1	S2
0	S1	9	1	**4**	1	0
0	S2	11	2	3	0	1
	Zj		0	0	0	0
	Cj-Zj		3	5	0	0

En küçük (sabit/x2)

En büyük (Cj-Zj)

SON SİMPLEKS TABLO						
AMAÇ SÜTUNU	AMAÇ SATIRI cj		3	5	0	0
	TEMEL DEĞİŞKENLER	SABİTLER	X1	X2	S1	S2
5	X2	7/5	0	1	2/5	-1/5
3	X1	17/5	1	0	-3/5	4/5
	Zj	86/5	3	5	1/5	7/5
	Cj-Zj		0	0	-1/5	-7/5

Bu durumda yeni kısıt ekleyerek modeli tekrar çözmek gerekir.

Yeni kısıt eklemek için optimum çözümdeki tamsayı olmayan herhangi bir değişken seçilebilir.
Örneğin; X2 değişkenini seçelim ve bu değişkenin olduğu satırı yeniden yazmaya çalışalım. Tam sayı olmayan sayıları ;

(tamsayı) + (1 den küçük pozitif kesri) olarak yazalım.

Örneğin 4/3 ------ 1 + 1 / 3

5/4------ 1 + 1 / 4

2/3------ 0 + 2 / 3

-2/3 ------ -1 + 1 / 3

Şimdi X2 değişkeninin olduğu satırı (tamsayı) + (1 den küçük pozitif kesir) olarak yazalım.

X2 satırındaki katsayılar: (7/5, 0, 1, 2/5, -1/5,)

(1+ 2/5) = (1+0) X2+ (0+ 2/5)S1 + (-1 + 4/5) S2

Tamsayılı katsayıları sağ tarafa alarak yeniden yazalım.

2/5 S1 + 4/5 S2 = 2/5 +(1- 1 x2 + 1S2)

Tam sayılı kısmı herhangi bir tamsayı olarak düşünüp eşitlikten çıkarırsak;

2/5 S1 + 4/5 S2 > = 2/5 şeklinde yazabiliriz.

Probleme yapay(artificial) değişken eklememek için her iki tarafı – 1ile çarparak eşitliğin yönünü değiştirip bir slack(boş) değişken eklersek kısıt aşağıdaki şekli alacaktır.

-2/5 S1 -4/5 S2 + S3 = -2/5

Bu kısıtı son tabloya ekleyelim

SİMPLEKS TABLO							
AMAÇ SÜTUNU	AMAÇ SATIRI cj		3	5	0	0	0
	TEMEL DEĞİŞKENLER	SABİTLER	X1	X2	S1	S2	S3
5	x2	1	0	1	0	-1	1
3	x1	4	1	0	0	2	-3/2
0	S1	1	0	0	1	2	-5/2
	Zj	17	3	5	0	1	1/2
	Cj-Zj		0	0	0	-1	-1/2

Bu tablodaki sonuca bakarsak;

- **X1=4 , x2=1 , Z=17**
- **Tamsayılı sonuç elde edilmiştir literasyona devam edilmez.**
- **Bu örnekte tek kısıtla optimum sonuca ulaşılmıştır her zaman böyle olmayabilir.**
- **Bu örnekte eklenen ilk kısıtta tamsayı elde edilmiştir. Eğer eklenen ilk kısıtta tamsayı elde edilmeseydi yeniden bir kısıt eklenerek tamsayılı sonuç alınıncaya kadar işlemler tekrarlanırdı.**