EM302 Yöneylem Araştırması 2

Dr. Özgür Kabak

TP Çözümü

- TP problemlerinin çözümü için başlıca iki yaklaşım vardır
  - ***kesme düzlemleri*** (cutting planes)
  - ***dal sınır*** (branch and bound)

- tüm yaklaşımlar tekrarlı bir şekilde DP'ler çözmeyi içerir.

- TP için çözüm yöntemleri iki sınıfa ayrılabilir:
  - Genel amaçlı: herhangi bir TP’yi çözebilecek yapıda) fakat etkin işlem yapamayan (göreli olarak daha küçük sorunları çözen;
  - Özel amaçlı: Belirli bir tipteki TP sorunu için geliştirilen ve daha etkin işlem yapabilen.

- TP çözüm yöntemleri çözüme gitmeleri açısından sınıflandırması:
  - En iyi (Optimal)
  - Sezgisel (Heuristic)

- Ders kapsamı:
  - Genel amaçlı en iyi çözümü veren yaklaşımlar
  - Sayma, Dal sınır

Doğrusal programlama gevşetmesi (LP relaxation)

- Bir tam sayılı programlama modelinde tamsayı veya 0-1 tam tamsayı değişkenlerin sürekli (normal) karar değişkeni olarak kabul edilmesi ile elde edilen doğrusal programlama modeli

Karma tam sayılı programlama modeli

maks $z = x_1 + x_2 + x_3$

öyle ki $x_1 + 6x_2 + x_3 \leq 8$

$x_1 + 2x_2 + 1{,}5x_3 \leq 2$

$2x_1 + x_2 + 5x_3 \leq 8$

$x_1 \geq 0$, x_2 tamsayı, $x_3 \in \{0,1\}$

Modelin DP gevşetmesi

maks $z = x_1 + x_2 + x_3$

öyle ki $x_1 + 6x_2 + x_3 \leq 8$

$x_1 + 2x_2 + 1{,}5x_3 \leq 2$

$2x_1 + x_2 + 5x_3 \leq 8$

$x_1, x_2 \geq 0$

$0 \leq x_3 \leq 1$

<u>En büyükleme probleminde:</u>

DP gevşetmesinin en iyi çözümü $\geq$ TP modelinin en iyi çözümü

TP ve DP gevşetmesi ilişkisi

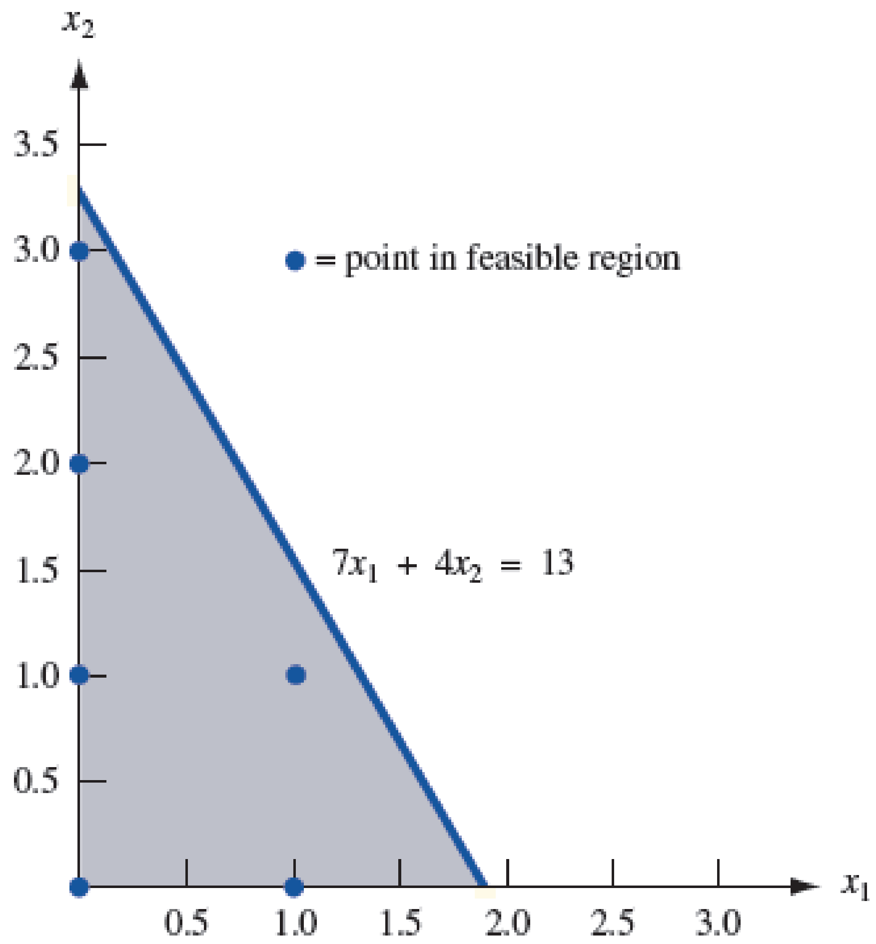
- Eğer DPG’nin en iyi çözümündeki tüm değişkenler tamsayılı değerler alıyorlarsa bu durumda bulunan en iyi çözüm orijinal TP sorununun da en iyi çözümüdür (doğal tamsayılı DP)
- DPG, TP sorununa göre daha az kısıtlı (gevşek) olduğundan aşağıdaki durumlarla karşılaşılabilir:
  - Eğer TP enbüyükleme sorunu ise, DG’nin en iyi amaç değeri TP’ninkine eşit veya daha büyüktür.
  - Eğer TP enküçükleme sorunu ise, DG’nin en iyi amaç değeri TP’ninkine eşit veya daha küçüktür.
  - Eğer DG olurlu değilse (olurlu çözümü yoksa), TP de olurlu değildir.
- Bu durumda DG’nin çözülmesinin bir bilgi vereceği açıktır:
  - en iyi amaç değeri sınırı belirlenir ve şanslıysak en iyi amaç değerini buluruz.
  - Fakat en iyi çözümdeki değişken değerlerini tamsayı değerlere yuvarlamak genel olarak en iyi TP çözümünü vermeyebilir; hatta yeni çözüm olurlu bile olmayabilir.

Örnek

maks $z = 21x_1 + 11x_2$

öyle ki $7x_1 + 4x_2 \leq 13$

$x_1, \; x_2$ tamsayı



Sayma Yöntemi

- DP'den (değişkenlerin sürekli aralıkta değerler alabildiği (>=0)) farklı olarak TP'de (tüm değişkenler tamsayılı) her değişken sadece sonlu sayıda kesikli (tamsayı) değerler alabilir.

- Sayma Çözüm yaklaşımı
  - Tüm olası çözümleri sayma (enumerate)
  - her biri için amaç fonksiyon değerini hesaplama
  - olurlu çözümlerden en iyisini seçme

Sayma Yöntemi - Örnek

Maks $\quad 0.2\, x_1 + 0.3\, x_2 + 0.5\, x_3 + 0.1\, x_4$

Öyle ki $\quad 0.5\, x_1 + 1.0\, x_2 + 1.5\, x_3 + 0.1\, x_4 \leq 3.1$

$0.3\, x_1 + 0.8\, x_2 + 1.5\, x_3 + 0.4\, x_4 \leq 2.5$

$0.2\, x_1 + 0.2\, x_2 + 0.3\, x_3 + 0.1\, x_4 \leq 0.4$

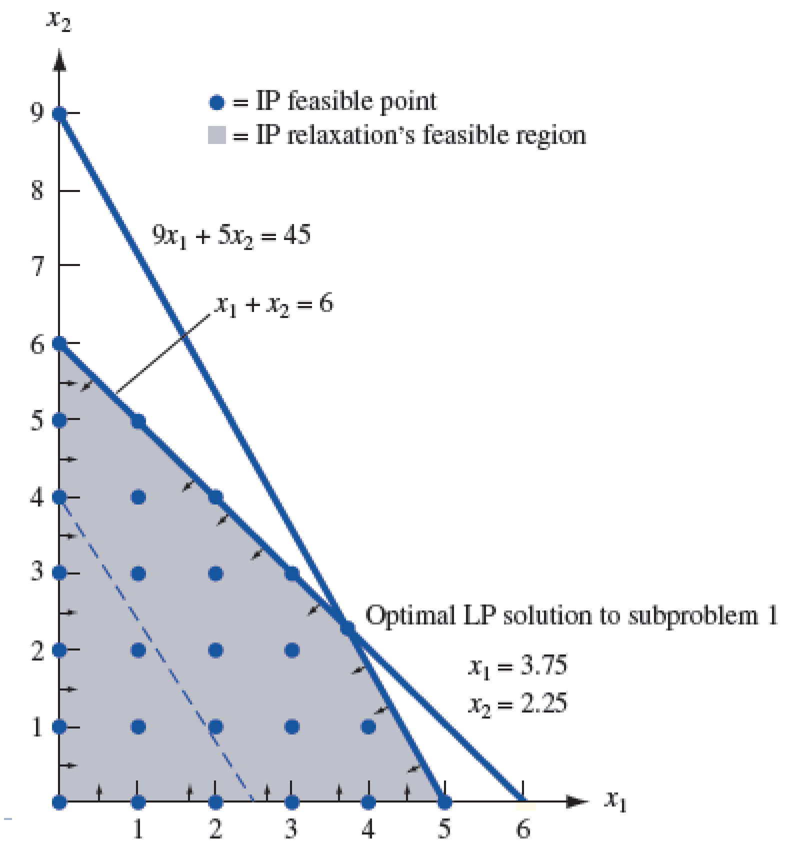
$x_j = 0$ veya $1 \qquad j = 1, \ldots, 4$

Dal Sınır Algoritması

- ***Dal Sınır Algoritması***
  - Sistematik bir şekilde olurlu çözümlerin sayılarak en iyi tamsayılı çözümün bulunması için kullanılan genel amaçlı bir DP tabanlı ağaç arama (LP-based tree search) yöntemi
  - 1960'ların başında Land ve Doig tarafından önerilmiştir.

Dal sınır algoritması Örnek

- Enb z= 8x1+ 5x2
- Öyle ki: x1+ x2 ≤ 6
- 9x1+5x2 ≤45
- x1,x2>0 ve tamsayı



Dal sınır algoritmasında ilerleme

- LIFO (Last in last out): son giren ilk çıkar
  - Son ürettiğin alt problemi önce çöz
  - İzlediğin yoldan geri dön (Backtracking)
- Atlamalı (jumptracking)
  - En iyi z değerine sahip alt problemden dallandır

Dal Sınır Algoritması - Örnek

Maks $0.2\, x_1 + 0.3\, x_2 + 0.5\, x_3 + 0.1\, x_4$

Öyle ki $0.5\, x_1 + 1.0\, x_2 + 1.5\, x_3 + 0.1\, x_4 \leq 3.1$

$0.3\, x_1 + 0.8\, x_2 + 1.5\, x_3 + 0.4\, x_4 \leq 2.5$

$0.2\, x_1 + 0.2\, x_2 + 0.3\, x_3 + 0.1\, x_4 \leq 0.4$

$x_j = 0$ veya $1 \qquad j = 1, \ldots, 4$

LINDO İLE DPG'yi çöz:

Max 0.2 x1 + 0.3 x2 + 0.5 x3 + 0.1 x4

st

0.5 x1 + 1.0 x2 + 1.5 x3 + 0.1 x4 < 3.1

0.3 x1 + 0.8 x2 + 1.5 x3 + 0.4 x4 < 2.5

0.2 x1 + 0.2 x2 + 0.3 x3 + 0.1 x4 < 0.4

x1 <1

x2 <1

x3 <1

x4 <1

end

Karışık TP problemlerinin çözümü

- Bu tarz problemlerin dal sınır algoritması ile çözümünde sadece tamsayı değişkenler üzerinden dallanma yapılır.

- Örnek
- Enb $z = 2x_1 + x_2$
- Öyle ki
  - $5x_1 + 2x_2 \leq 8$
  - $x_1 + x_2 \leq 3$
  - $x_1, x_2 \geq 0$; x_1 tamsayı

Sırt Çantası Problemi

- Sadece bir kısıtı olan herhangi bir TP sırt çantası sorunu olarak isimlendirilir.
- Bir diğer özellik de kısıtın ve amaç fonksiyonunun katsayıları negatif olmayan sayılardır.

Örneğin aşağıdaki model sırt çantası sorunudur:

max $z = 8x_1 + 11x_2 + 6x_3 + 4x_4$

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

$x_j = 0$ or 1 ($j = 1,2,3,4$)

- Geleneksel öykü, bir sırt çantasında (örnekte kapasitesi 14'dür) bazı eşyaları (dört çeşit eşya) taşımak ile ilgilidir. Her eşyanın bir büyüklüğü ve taşıma sonucu elde edilecek bir değeri vardır (eşya 2 için büyüklük 7, değer 11'dir).
- Amaç, sırt çantasına sığabilecek eşyaların toplam değerini enbüyüklemektir.
- Sırt çantası sorunları genellikle kolay çözülürler.

- Enb $z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$
- Öyle ki $a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$
- $x_i = 0$ veya 1 ($i = 1, 2, \ldots, n$)

Sırt çantası problemlerinin çözümü

- Enb $z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$
- Öyle ki $a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$
- $x_i = 0$ veya 1 ($i = 1, 2, \ldots, n$)

- DP gevşetmesini çözmek için
  - $r_i = c_i/a_i$ (i.değişkenin kullandığı her kaynak miktarın başına sağlanan fayda)
  - En büyük r_i oranına sahip değişkenler sırt çantasına eklenir.

Sırt çantası problemlerinin çözümü Örnek

$\max z = 8x_1 + 11x_2 + 6x_3 + 4x_4$

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

$x_j = 0$ or 1 ($j = 1,2,3,4$)

Gezgin Satıcı Problemi için Dal sınır algoritması

- Bir satıcı on adet şehiri birer kez ziyaret edip evine dönecektir. “Satıcının yapacağı bu gezide katedeceği toplam uzaklığı enazlayacak rota (şehir sırası) nasıl olmalıdır?” sorunu Gezgin Satıcı Sorunu (GSS; Traveling Salesperson Problem – TSP) olarak isimlendirilir.

GSS – TP formülasyonu

- N adet şehir olduğunu varsayalım
- $i \neq j$ için c_{ij} = şehir *i*'den şehir *j*'ye uzaklık ve
- c_{ii} = M (gerçek uzaklıklara göre çok büyük bir uzaklık) olsun
- Aynı zamanda x_{ij} aşağıdaki şekilde bir 0-1 değişken olarak tanımlansın:
  - x_{ij} = 1, GSS çözümü şehir i'den şehir j'ye gitmeyi önersin
  - x_{ij} = 0, aksi takdirde

- GSS formülasyonu:

min $\quad \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij}$

öyle ki $\sum_{i=1}^{N} x_{ij} = 1$ her *j* için

$\sum_{j=1}^{N} x_{ij} = 1$ her *i* için

$u_i - u_j + N x_{ij} \leq N - 1$ her $i \neq j$ için; i=2,3,…,N; j=2,3,…,N

Her x_{ij} = 0 veya 1, Her $u_i \geq 0$

Gezgin Satıcı Problemi için Dal Sınır Algoritması

- Problemi döngü engelleyici kısıtlar olmadan atama problemi gibi Macar Yöntemi ile çöz.
- Döngü oluşursa; en kısa döngüyü engelleyecek kısıtlar ile dallandır.

Gezgin Satıcı Problemi için Dal Sınır Algoritması - Örnek

- ATK-Beyaz dört müşterisinin verdiği siparişlerin dağıtımını gerçekleştirecektir. Tabloda depo ve müşteriler arası mesafeler km. olarak verilmiştir.
- Depodan çıkan aracın tüm müşterilerin siparişlerini teslim ettikten sonra tekrar depoya dönme süresini en küçükleyecek rotayı bulunmak istenmektedir.

	Depo	M1	M2	M3	M4
Depo	0	13	22	16	6
M1	13	0	29	20	8
M2	22	29	0	11	30
M3	16	20	11	0	20
M4	6	8	30	20	0

Örnek - Çözüm

- Alt problem 1 için maliyet matrisi

	Depo	M1	M2	M3	M4
Depo	M	13	22	16	6
M1	13	M	29	20	8
M2	22	29	M	11	30
M3	16	20	11	M	20
M4	6	8	30	20	M

- Macar yöntemi uygulanırsa:

	Depo	M1	M2	M3	M4
Depo	M	0	11	10	0
M1	0	M	16	12	0
M2	6	11	M	0	19
M3	5	7	0	M	14
M4	0	0	24	19	M

Alt problem 1
Z=49
$x_{04}=x_{01}=x_{10}=x_{23}=x_{32}=1$

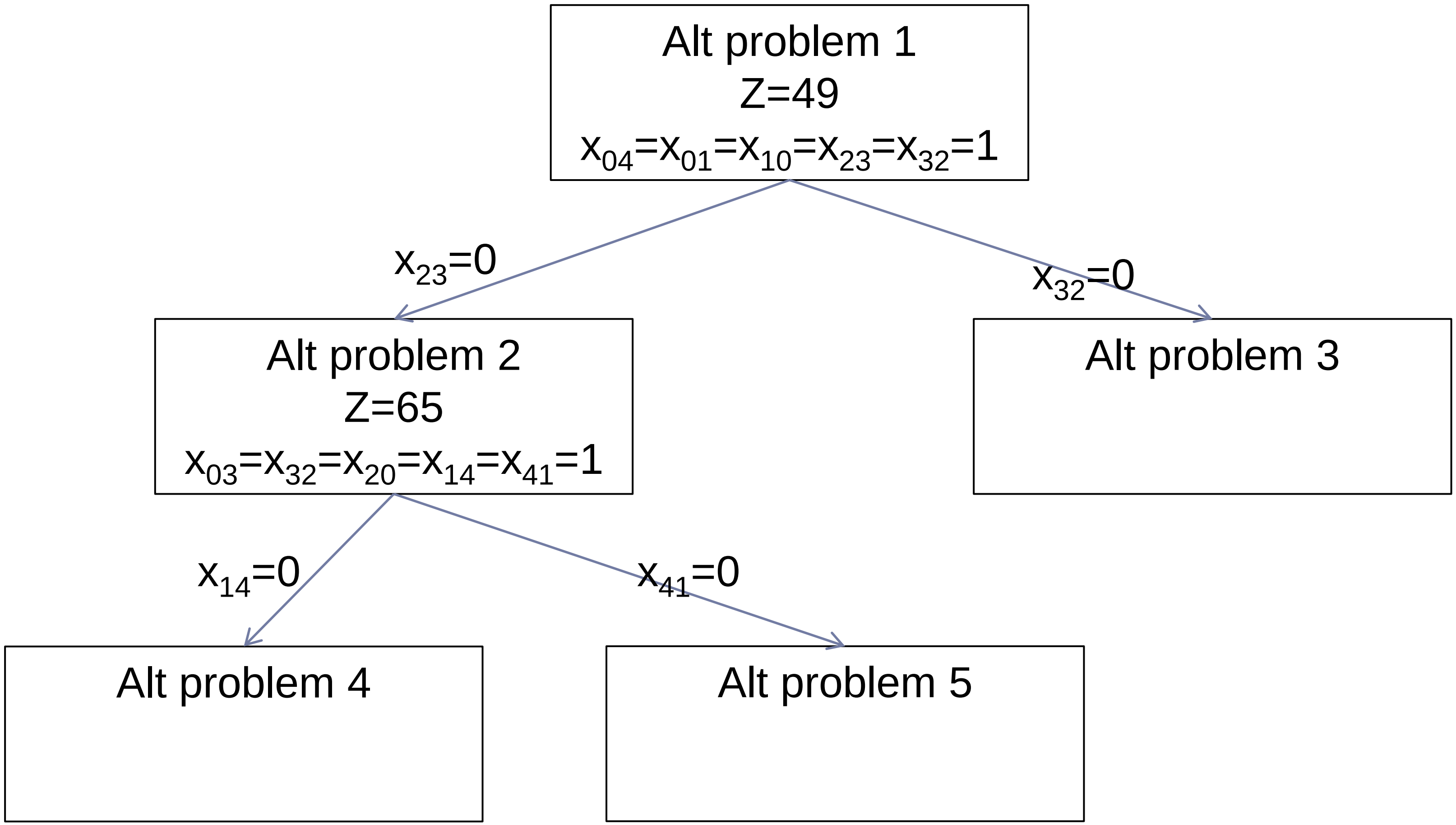
Alt problem 1
Z=49
$x_{04}=x_{01}=x_{10}=x_{23}=x_{32}=1$

$x_{23}=0$ → Alt problem 2

$x_{32}=0$ → Alt problem 3

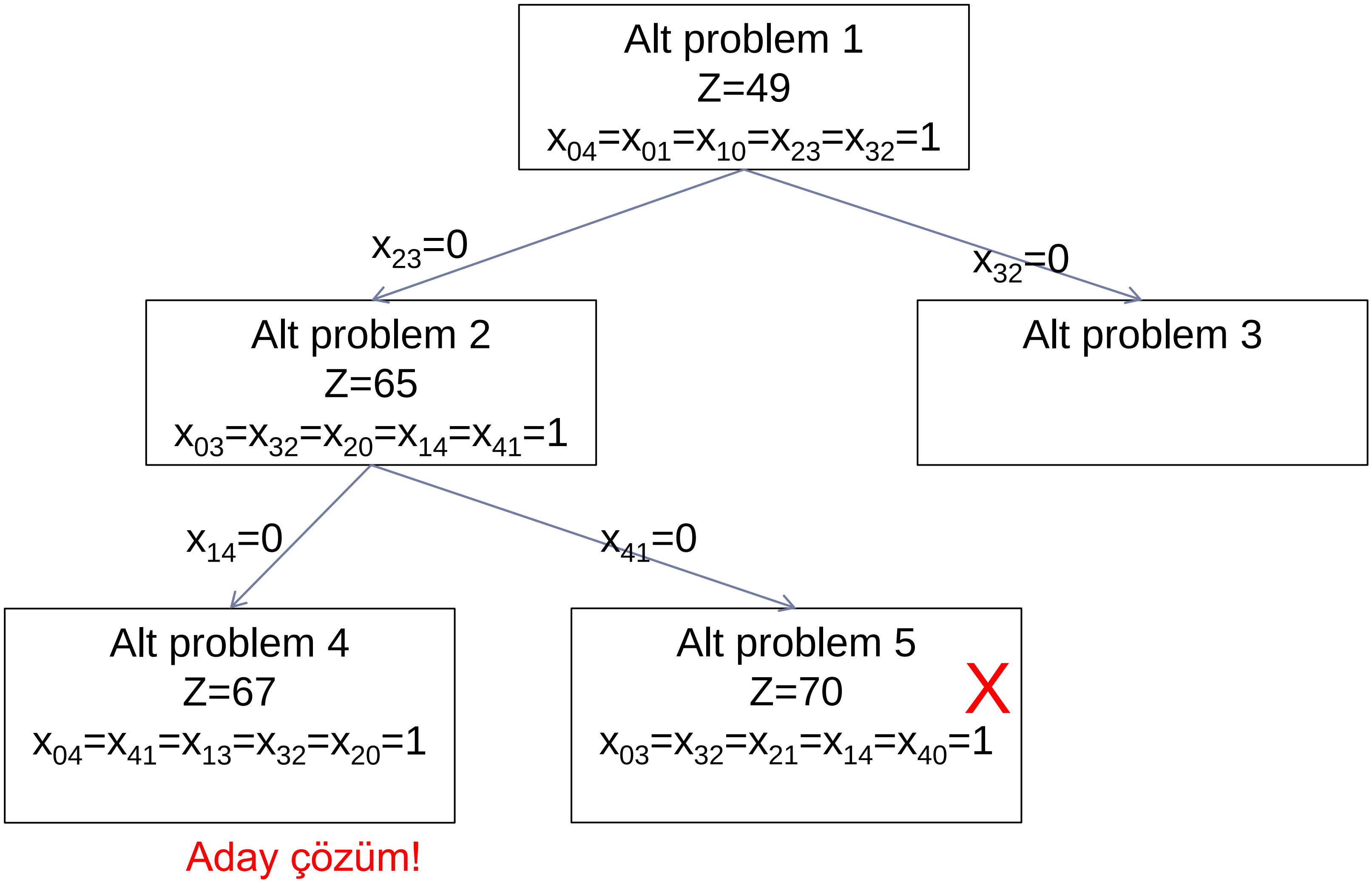
	Depo	M1	M2	M3	M4
Depo	M	13	22	16	6
M1	13	M	29	20	8
M2	22	29	M	**M**	30
M3	16	20	11	M	20
M4	6	8	30	20	M

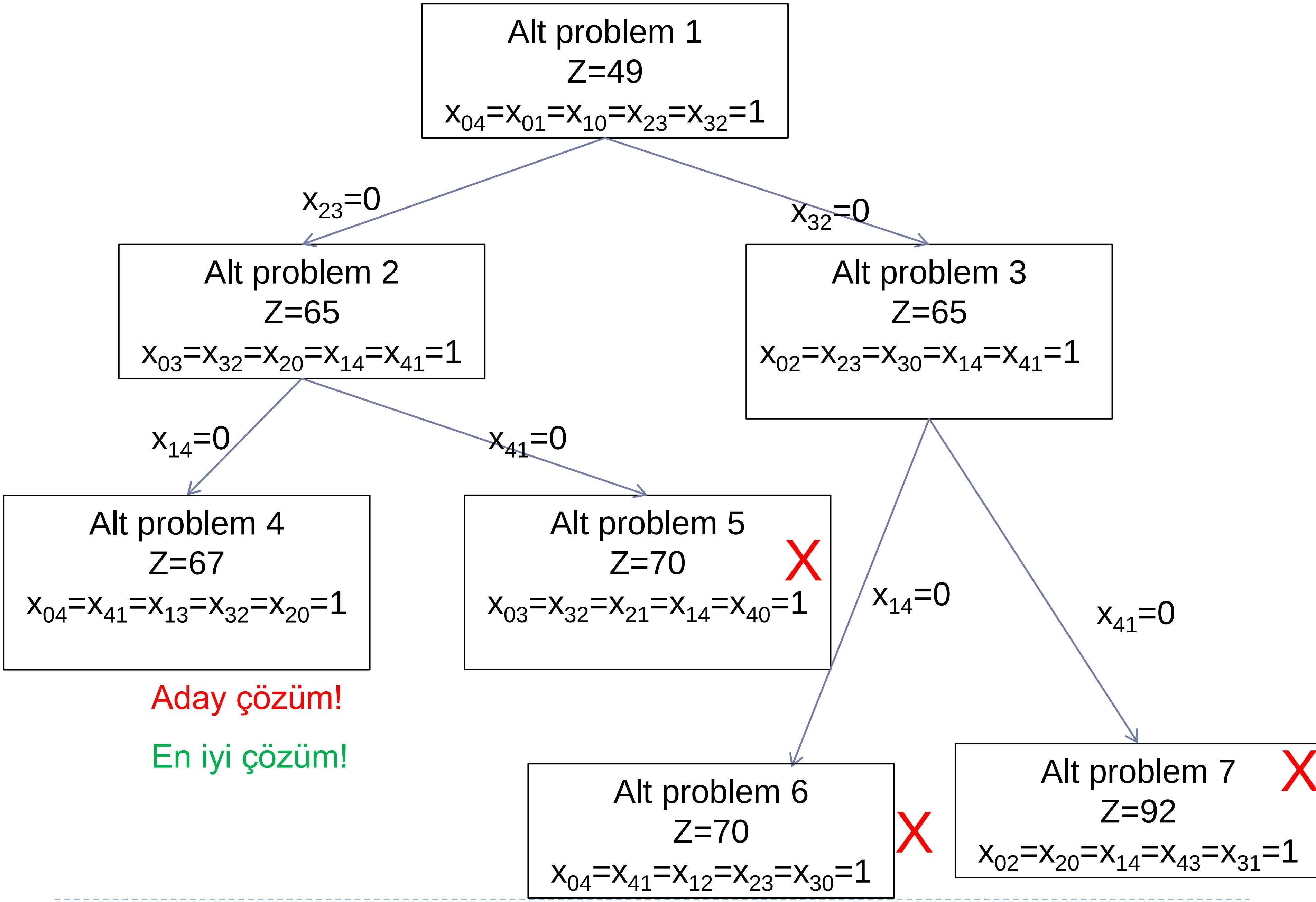
	Depo	M1	M2	M3	M4
Depo	M	13	22	16	6
M1	13	M	29	20	8
M2	22	29	M	11	30
M3	16	20	**M**	M	20
M4	6	8	30	20	M



	Depo	M1	M2	M3	M4
Depo	M	13	22	16	6
M1	13	M	29	20	**M**
M2	22	29	M	**M**	30
M3	16	20	11	M	20
M4	6	8	30	20	M

	Depo	M1	M2	M3	M4
Depo	M	13	22	16	6
M1	13	M	29	20	8
M2	22	29	M	**M**	30
M3	16	20	11	M	20
M4	6	**M**	30	20	M


Alt problem 1
Z=49
$x_{04}=x_{01}=x_{10}=x_{23}=x_{32}=1$
$x_{23}=0$
$x_{32}=0$
Alt problem 2
Z=65
$x_{03}=x_{32}=x_{20}=x_{14}=x_{41}=1$
Alt problem 3
$x_{14}=0$
$x_{41}=0$
Alt problem 4
Z=67
$x_{04}=x_{41}=x_{13}=x_{32}=x_{20}=1$
Alt problem 5
Z=70
X
$x_{03}=x_{32}=x_{21}=x_{14}=x_{40}=1$
Aday çözüm!


Alt problem 1
Z=49
$x_{04}=x_{01}=x_{10}=x_{23}=x_{32}=1$
$x_{23}=0$
$x_{32}=0$
Alt problem 2
Z=65
$x_{03}=x_{32}=x_{20}=x_{14}=x_{41}=1$
Alt problem 3
Z=65
$x_{02}=x_{23}=x_{30}=x_{14}=x_{41}=1$
$x_{14}=0$
$x_{41}=0$
Alt problem 4
Z=67
$x_{04}=x_{41}=x_{13}=x_{32}=x_{20}=1$
Alt problem 5
Z=70
X
$x_{03}=x_{32}=x_{21}=x_{14}=x_{40}=1$
$x_{14}=0$
$x_{41}=0$
Aday çözüm!
En iyi çözüm!
Alt problem 6
Z=70
$x_{04}=x_{41}=x_{12}=x_{23}=x_{30}=1$
X
Alt problem 7
Z=92
$x_{02}=x_{20}=x_{14}=x_{43}=x_{31}=1$
X