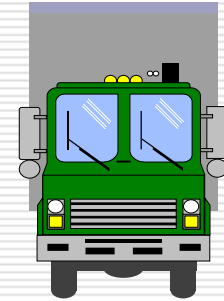
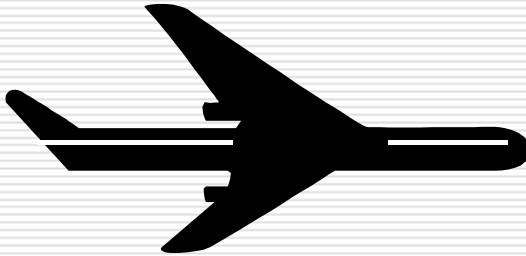


Lojistik Eniyilemesi (Optimizasyon)



Kaynak: Dr. Y. İlker TOPCU
Operations Research: Applications and Algorithms
by Wayne L. Winston

Copyright (c) 2004 Brooks/Cole, a division of Thomson Learning, Inc.

İçindekiler

- OR / MS, eniyileme
- Ulaştırma
- Geçici konaklama
- Atama
- Gezgin satıcı
- Kargo yükleme

OR / MS

- **Yöneylem arařtırması** (yönetim bilimi) genellikle kıt kaynakların tahsisini gerektiren kořullarda bir sistemin iřletilmesi ve en iyi tasarımının yapılmasına iliřkin karar vermeye yönelik bilimsel bir yaklařımdır.
 - Söz konusu terim, liderlerin bilim adamlarından ve mühendislerden çeřitli askeri sorunların analiz edilmesini istedikleri 2. dünya savařı sırasında türetilmiřtir.

-
- Karar vermeye yönelik bilimsel yaklaşım bir veya daha fazla sayıda **matematiksel modele** gereksinim duyar.
 - Matematiksel model, daha iyi karar verebilmek veya gerçek durumu daha anlaşılır hale getirmek için kullanılacak bir gösterimdir.

-
- Bir **eniyileme (optimizasyon) modeli** karar deęişkenlerinin verilen tüm kısıtları saęlayan deęerleri arasında amaç fonksiyonunu eniyiledięi (en bykledięi veya en kkledięi) deęerleri bulmaya alıřır
 - Sistem performansını etkileyen ve kontrolumuz altındaki deęerleri olan deęişkenlere **karar deęişkenleri** denilir.
 - oęu durumda, karar deęişkenlerinin sadece belirli deęerleri olasıdır. Karar deęişkenleri deęerleri zerindeki bu sınırlandırmalara **kısıt** denilir.

OR/MS Teknikleri

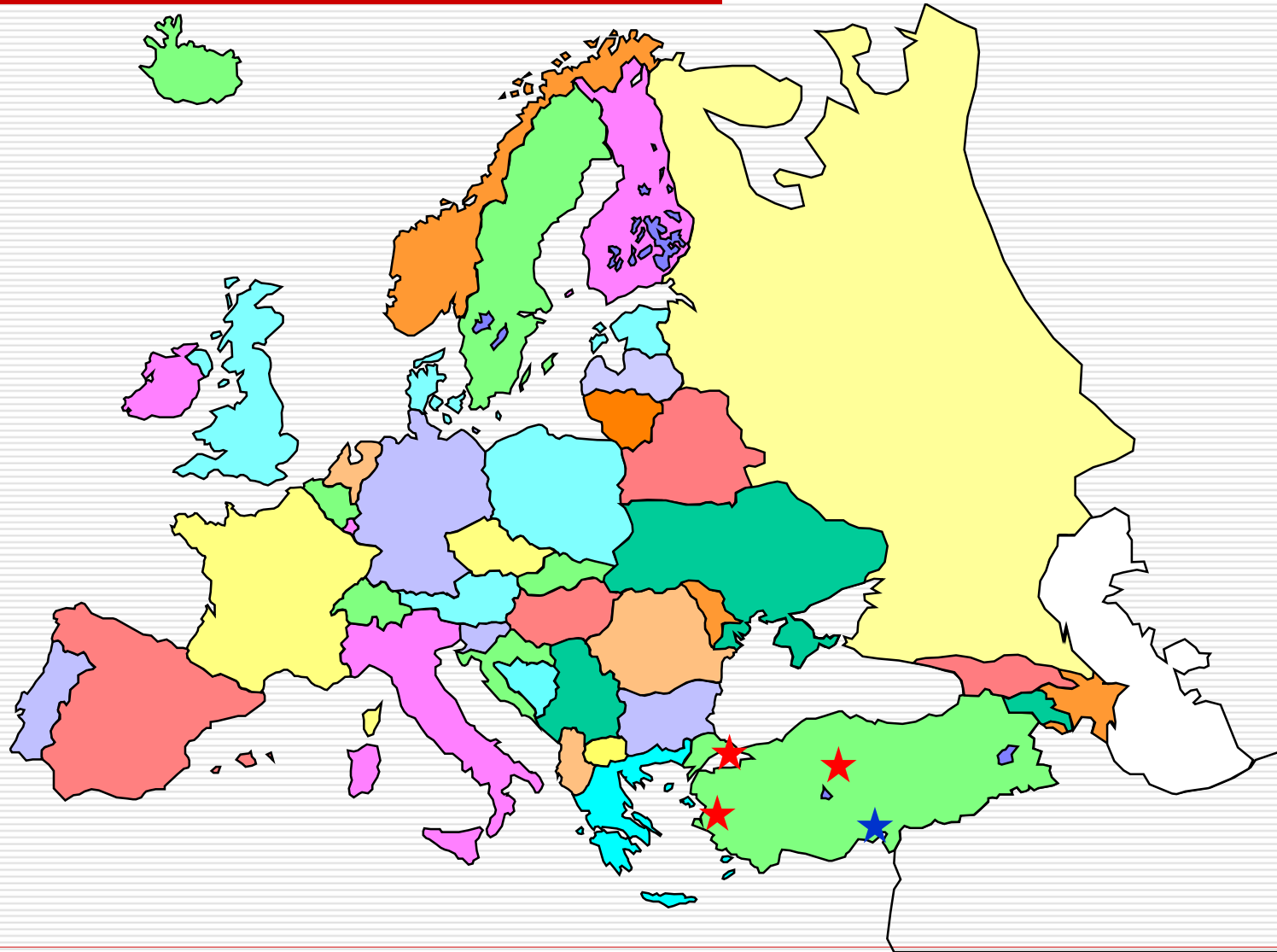
- Matematiksel programlama
 - Doğrusal programlama
 - Tamsayı programlama
 - Grafik analizi
 - Duyarlılık analizi
 - Ulaştırma
 - Atama
 - Hedef programlama
- Olasılık Teknikleri
 - İstatistik
 - Karar analizi
 - Kuyruk modelleri
- Ağ Teknikleri
 - Proje yönetimi (CPM/PERT)
 - Ağ modelleri
- ÇKKV (MCDM)
 - Değer/yarar temelli
 - Etkileşimli
 - Üstünlük
 - Basit
- Diğer
 - Simülasyon
 - Tahmin
 - Envanter
 - Doğrusal olmayan programlama

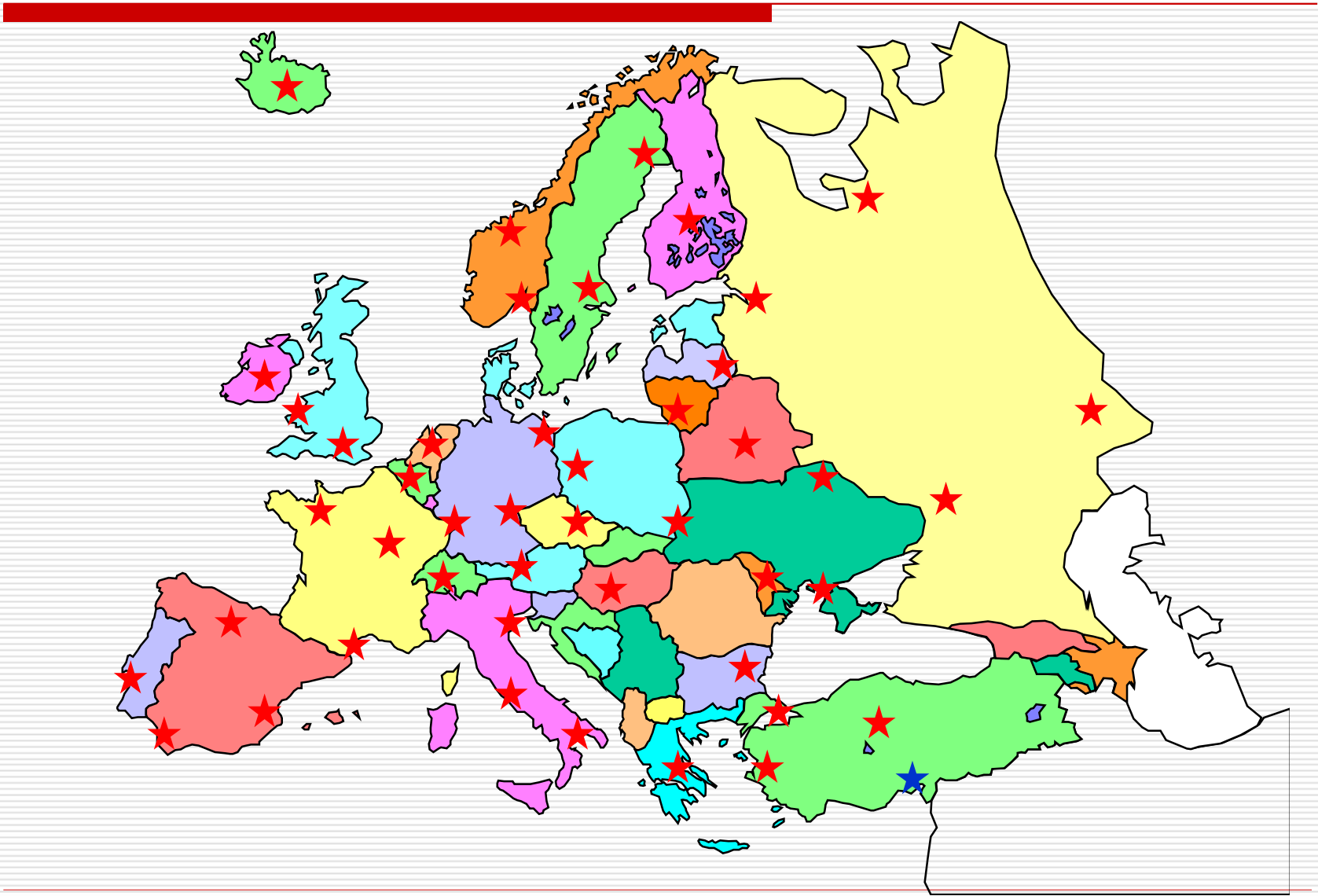
Ulaştırma eniyileme

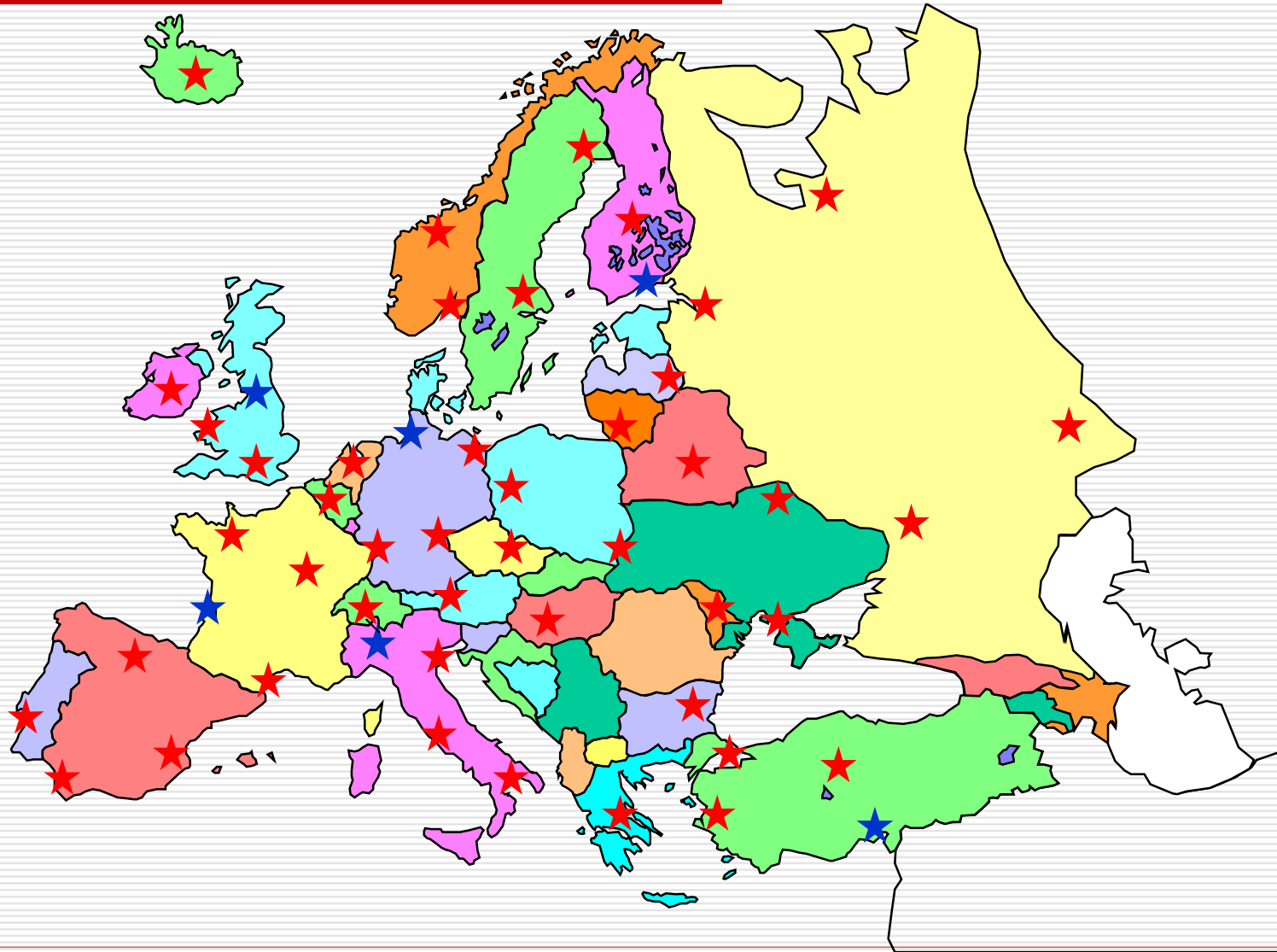
1. Ulaştırma sorununun formülasyonu
2. Ulaştırma sorununa başlangıç Temel Olurlu Çözüm (bfs) bulma
3. Ulaştırma Simpleks algoritması ile en iyi çözümü bulma

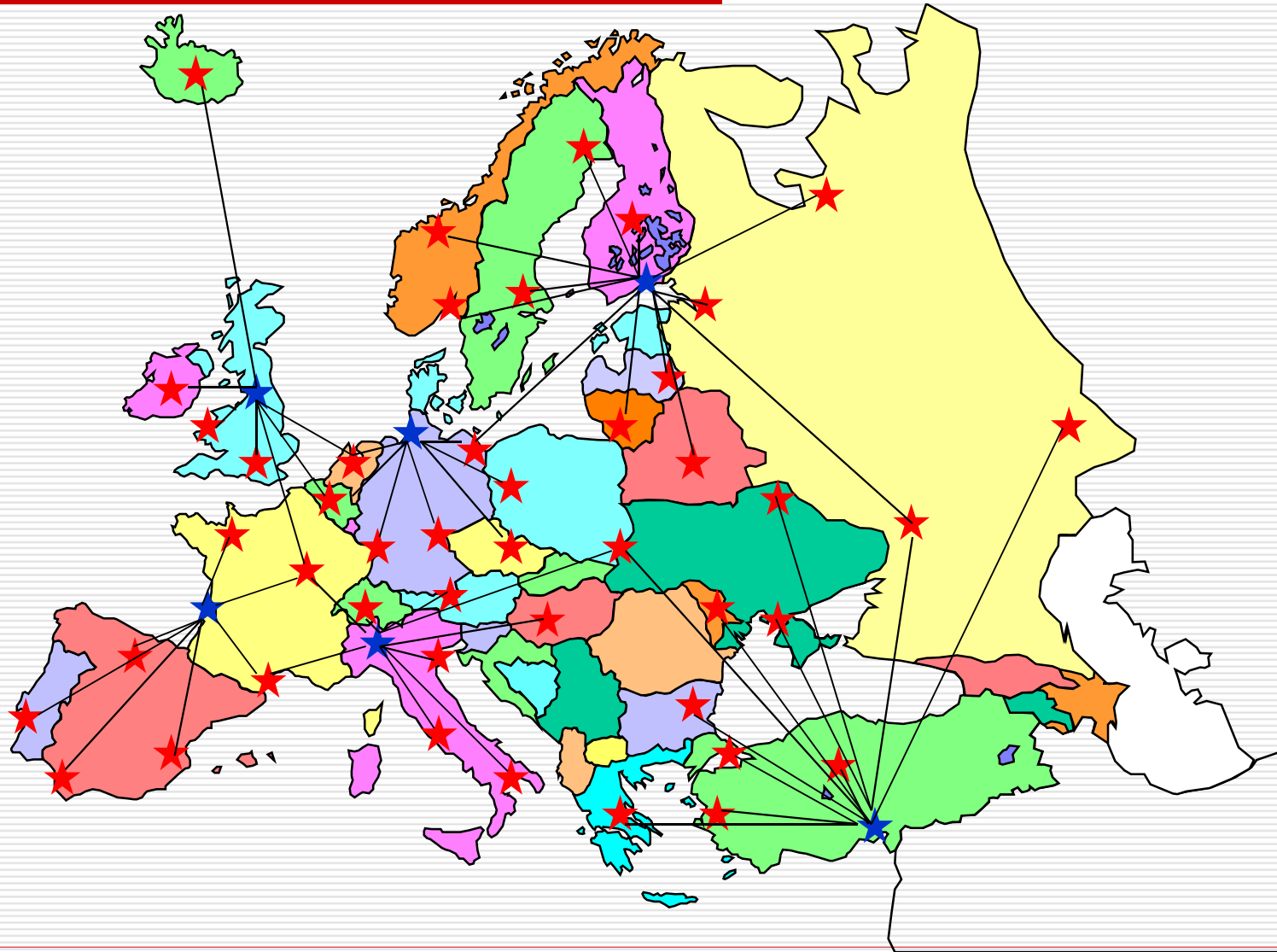
Ulařtırma Sorununun Formülasyonu

- Bir ulařtırma sorunu, temel olarak m arz noktasının kapasitesini kullanarak n talep noktasının talebini karşılamak için en iyi yolun bulunmasını hedefleyen sorundur.
- En iyi yolu bulmaya çalışırken, genel olarak bir arz noktasından bir talep noktasına bir ürün/hizmet taşımanın deęişken maliyeti göz önüne alınır.



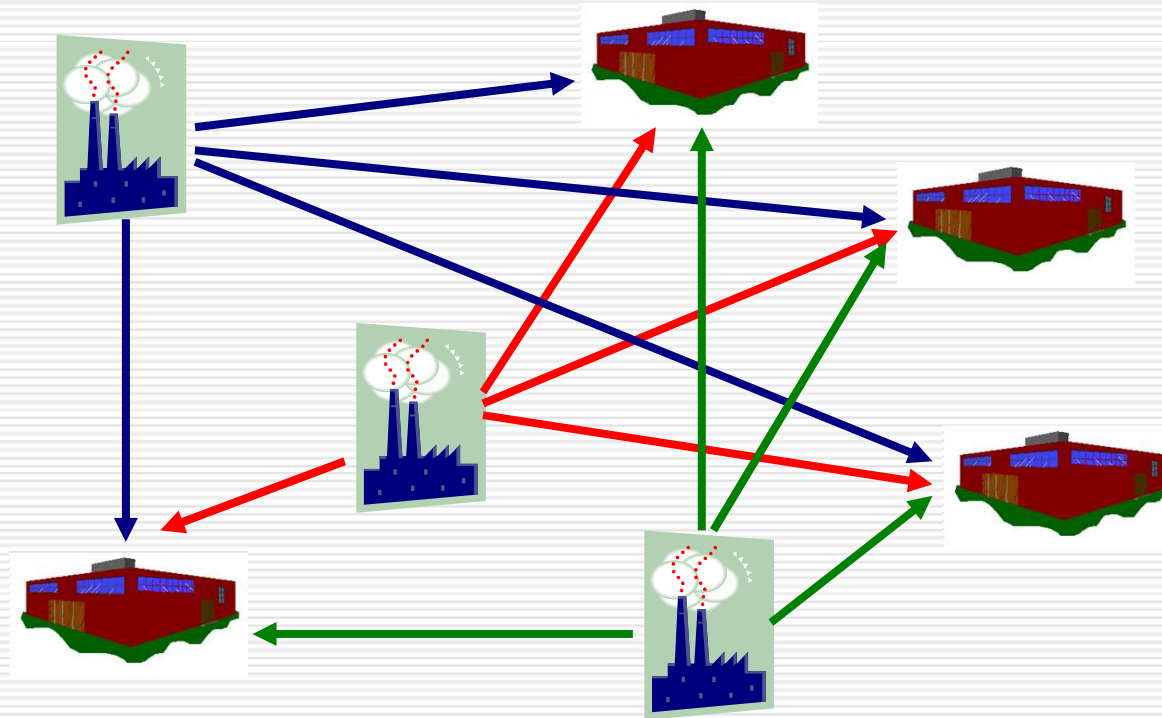






Örnek 1: Powerco

- Powerco şirketinin dört şehre hizmet veren üç adet elektrik santrali vardır.



Ör. 1 - Ulaştırma tablosu

- Her santralin ürettiği ve her şehrin en yoğun saatlerde talep ettiği elektrik miktarı tabloda verilmiştir.
- 1 milyon kWh elektriğin bir santralden bir şehre gönderilmesinin maliyeti uzaklığa bağlıdır.

| Nereden | Nereye | | | | |
|-----------------------|---------|---------|---------|---------|---------------------|
| | Şehir 1 | Şehir 2 | Şehir 3 | Şehir 4 | Arz (Milyon kwh) |
| Santral 1 | €8 | €6 | €10 | €9 | 35 |
| Santral 2 | €9 | €12 | €13 | €7 | 50 |
| Santral 3 | €14 | €9 | €16 | €5 | 40 |
| Talep (Milyon kwh) | 45 | 20 | 30 | 30 | |

Ör. 1 - Formülasyon

■ Karar değişkenleri

□ Powerco her santralden her şehre ne kadar güç gönderilmesi gerektiğini belirlemelidir: x_{ij} = santral i 'de üretilen ve şehir j 'ye gönderilen elektrik miktarı

■ x_{14} = Santral 1'de üretilen ve Şehir 4'e gönderilen elektrik miktarı

■ Kısıtlar

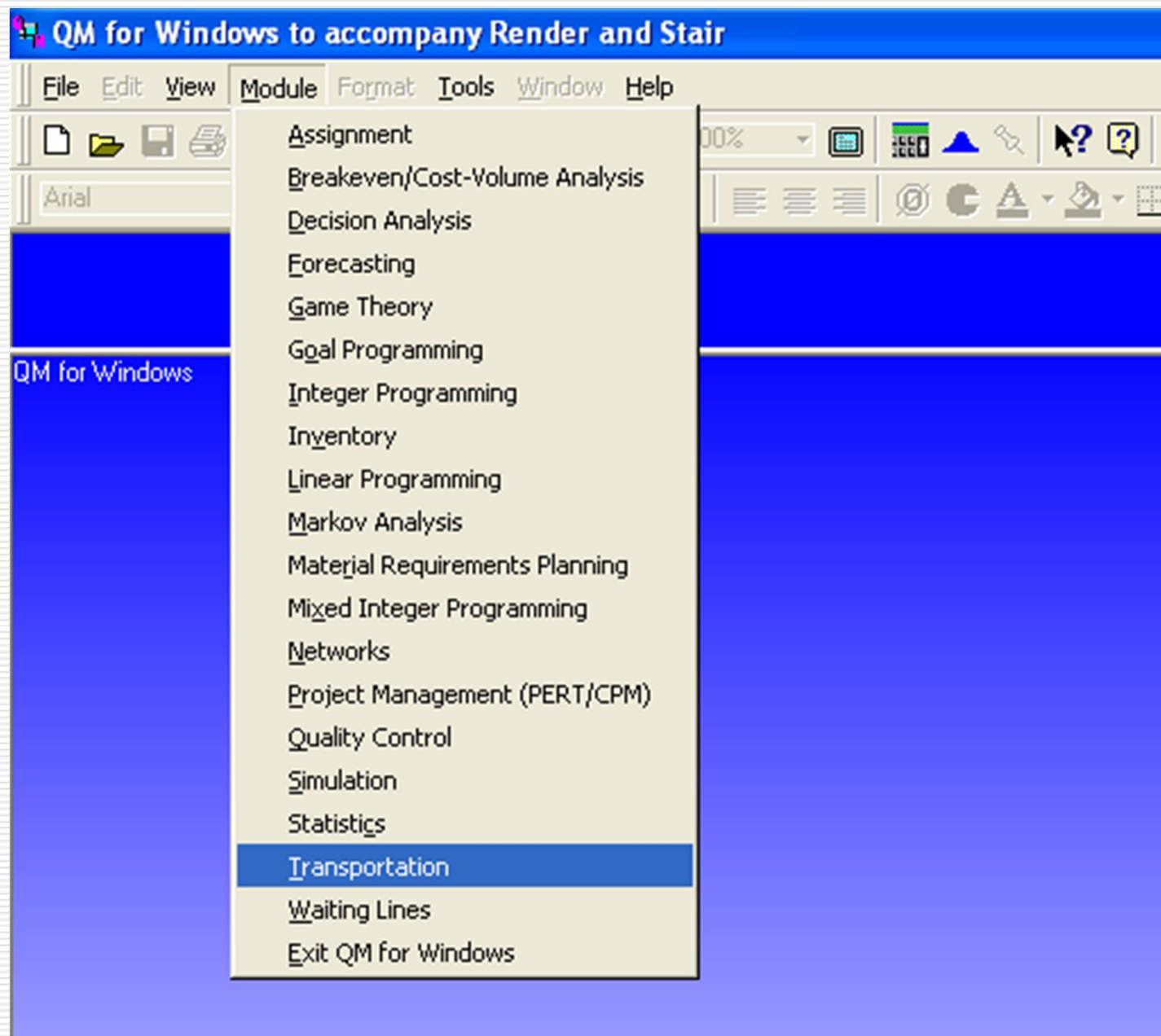
□ **Arz kısıtı** üretilen toplam miktarın fabrika kapasitesini geçmemesini sağlar. Her santral bir **arz noktası**dır.

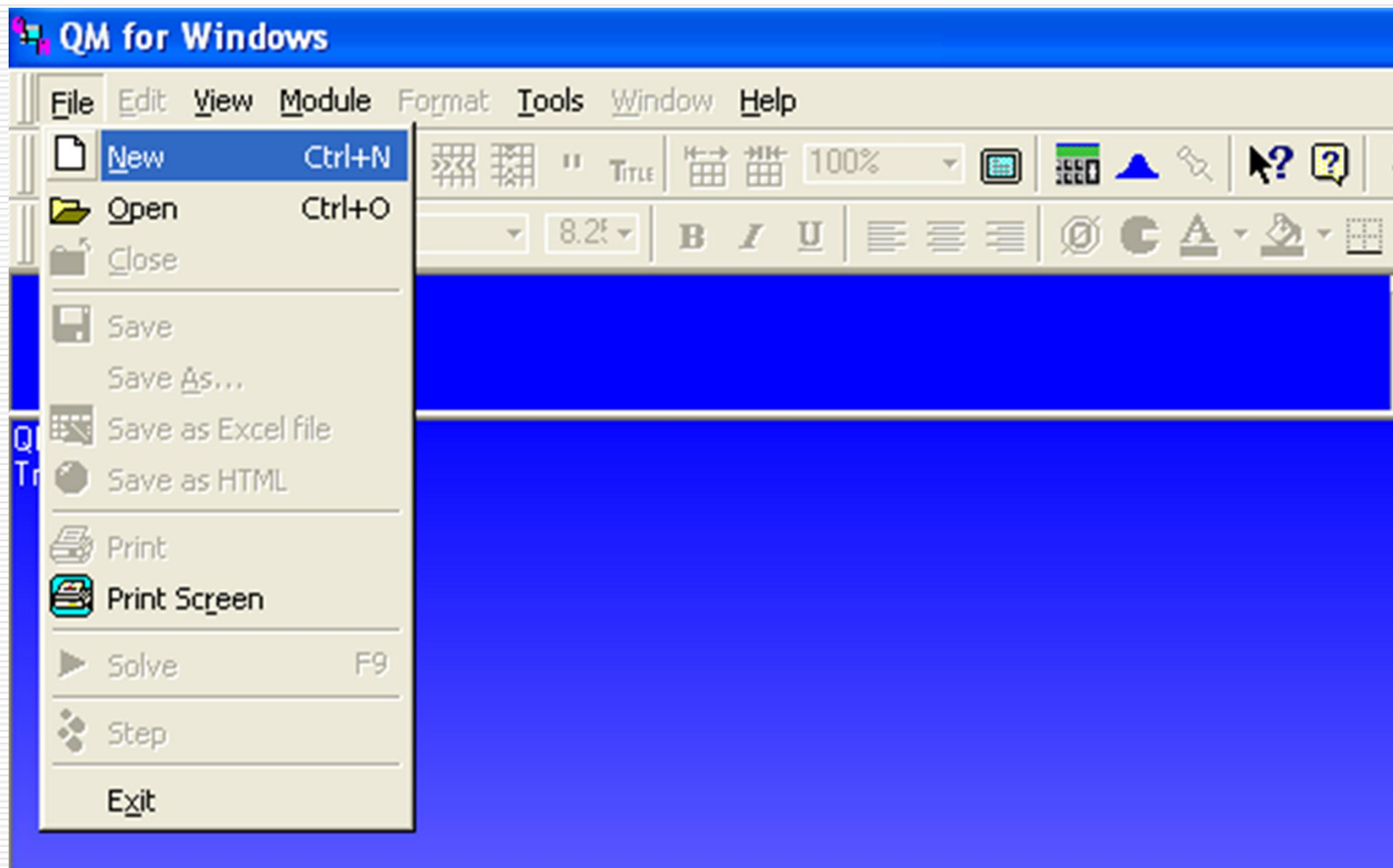
□ **Talep kısıtı** bir bölgenin talebinin karşılanmasını sağlar. Her şehir bir **talep noktası**dır.

□ Negatif miktarda elektrik gönderilemeyeceği için tüm x_{ij} 'ler negatif olmayan (≥ 0) değişkendir

-
- Genel olarak, bir ulaştırma sorunu aşağıdaki bilgi ile belirlenir:
 - Bir ürün/hizmet gönderen m adet arz noktası. i arz noktası en fazla s_i birim arz edebilir
 - Ürünün/hizmetin gönderildiği n adet talep noktası. j talep noktası en az d_j birime gereksinim duyar
 - Bir birimin i arz noktasından j talep noktasına gönderilmesi maliyeti c_{ij}
 - Karar değişkeni x_{ij} : i arz noktasından j talep noktasına gönderilen miktar

| | Talep noktası 1 | Talep noktası 2 | | Talep noktası n | ARZ |
|-----------------|-----------------|-----------------|-------|-------------------|-------|
| Arz noktası 1 | x_{11} | x_{12} | | x_{1n} | s_1 |
| Arz noktası 2 | x_{21} | x_{22} | | x_{2n} | s_2 |
| | | | | | |
| Arz noktası m | x_{m1} | x_{m2} | | x_{mn} | s_m |
| TALEP | d_1 | d_2 | | d_n | |





Create data set for Transportation

Title: Powerco

Modify default title

Number of Sources

3

Number of Destinations

4

Objective

- Maximize
 Minimize

Row names

Column names

- Source 1, Source 2, Source 3, ...
 a, b, c, d, e, ...
 A, B, C, D, E, ...
 1, 2, 3, 4, 5, ...
 January, February, March, April, ...

 Other

Cancel

Help

OK

Objective: Maximize Minimize

Starting method:
 Any starting method (selected)
 Northwest Corner Method
 Minimum Cost Method
 Vogel's Approximation Method

Instruction: Choose the method that you wish to use by clicking on it.

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 | SUPPLY |
|----------|---------------|---------------|---------------|---------------|--------|
| Source 1 | 0 | 0 | 0 | 0 | 0 |
| Source 2 | 0 | 0 | 0 | 0 | 0 |
| Source 3 | 0 | 0 | 0 | 0 | 0 |
| DEMAND | 0 | 0 | 0 | 0 | |

Objective
 Maximize
 Minimize

Starting method
 Any starting method

Instruction
 Choose the method that you wish to use by clicking on it.

Powerco

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 | SUPPLY |
|----------|---------------|---------------|---------------|---------------|--------|
| Source 1 | 8 | 6 | 10 | 9 | 35 |
| Source 2 | 9 | 12 | 13 | 7 | 50 |
| Source 3 | 14 | 9 | 16 | 5 | 40 |
| DEMAND | 45 | 20 | 30 | 30 | |

Transportation Shipments

Powerco Solution

Optimal cost = \$1,020

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 |
|----------|---------------|---------------|---------------|---------------|
| Source 1 | | 10. | 25. | |
| Source 2 | 45. | | 5. | |
| Source 3 | | 10. | | 30. |

Shipping list

Powerco Solution

| From | To | Shipment | Cost per unit | Shipment cost |
|----------|---------------|----------|---------------|---------------|
| Source 1 | Destination 2 | 10. | 6. | 60. |
| Source 1 | Destination 3 | 25. | 10. | 250. |
| Source 2 | Destination 1 | 45. | 9. | 405. |
| Source 2 | Destination 3 | 5. | 13. | 65. |
| Source 3 | Destination 2 | 10. | 9. | 90. |
| Source 3 | Destination 4 | 30. | 5. | 150. |

Shipments with costs

Powerco Solution

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 |
|----------|---------------|---------------|---------------|---------------|
| Source 1 | | 10/\$60 | 25/\$250 | |
| Source 2 | 45/\$405 | | 5/\$65 | |
| Source 3 | | 10/\$90 | | 30/\$150 |

Iterations

Powerco Solution

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 |
|-------------|---------------|---------------|---------------|---------------|
| Iteration 1 | | | | |
| Source 1 | -2. | 10. | 25. | -7. |
| Source 2 | 45. | -3. | 5. | -2. |
| Source 3 | -5. | 10. | -3. | 30. |

QM for Windows - [Iterations]

File Edit View Module Format Tools Window Help

Objective: Maximize Minimize

Starting method: Northwest Corner Method

Instruction: There are more results available in additional windows. These may be opened by using the WINDOW option in the Main Menu.

Powerco Solution

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 |
|-------------|---------------|---------------|---------------|---------------|
| Iteration 1 | | | | |
| Source 1 | 35. | (-5) | (-2) | -8. |
| Source 2 | 10. | 20. | 20. | -5. |
| Source 3 | -2. | (-6) | 10. | 30. |
| Iteration 2 | | | | |
| Source 1 | 35. | (-5) | (-2) | -2. |
| Source 2 | 10. | 10. | 30. | (-1) |
| Source 3 | -8. | 10. | -6. | 30. |
| Iteration 3 | | | | |
| Source 1 | 25. | 10. | (-2) | -7. |
| Source 2 | 20. | -5. | 30. | -4. |
| Source 3 | -3. | 10. | -1. | 30. |
| Iteration 4 | | | | |
| Source 1 | -2. | 10. | 25. | -7. |
| Source 2 | 45. | -3. | 5. | -2. |
| Source 3 | -5. | 10. | -3. | 30. |

Transportation Solution Screen Y. Ilker Topcu

Print Screen Next file Previous file Save as Excel file Save as HTML

QM for Windows - [Iterations]

File Edit View Module Format Tools Window Help

8.2! 100% Step Edit

Objective: Maximize Minimize

Starting method: Minimum Cost Method

Instruction: There are more results available in additional windows. These may be opened by using the WINDOW option in the Main Menu.

Powerco Solution

| | Destination 1 | Destination 2 | Destination 3 | Destination 4 |
|-------------|---------------|---------------|---------------|---------------|
| Iteration 1 | | | | |
| Source 1 | 15. | 20. | (-2) | -8. |
| Source 2 | 30. | -5. | 20. | -5. |
| Source 3 | -2. | (-1) | 10. | 30. |
| Iteration 2 | | | | |
| Source 1 | -2. | 20. | 15. | -10. |
| Source 2 | 45. | -3. | 5. | -5. |
| Source 3 | -2. | (-3) | 10. | 30. |
| Iteration 3 | | | | |
| Source 1 | -2. | 10. | 25. | -7. |
| Source 2 | 45. | -3. | 5. | -2. |
| Source 3 | -5. | 10. | -3. | 30. |

Transportation Solution Screen Y. Ilker Topcu

Print Screen Next file Previous file Save as Excel file Save as HTML

■ Eğer

$$\sum_{i=1}^{i=m} s_i = \sum_{j=1}^{j=n} d_j$$

ise toplam arz toplam talebe eşittir ve sorun bir **dengeli ulaştırma sorunu**dur.

-
- Eğer toplam arz miktarı toplam talep miktarını geçerse, sorunu dengelemek için talep miktarı aradaki fark (fazla arz miktarı) kadar olan bir **yapay talep noktası** yaratırız.
 - Söz konusu noktaya yapılacak gönderimler aslında olmayacağı için bu noktaya arz noktalarından yapılacak ulaştırma maliyeti 0 olacaktır.

Örnek 2: Değiştirilmiş Powerco Örneği (Fazla Arz)

- Şehir 1'in talebinin 40 milyon kwh olduğunu (45 milyon kwh yerine) düşünelim. Bu durumda dengeli bir ulaştırma sorunu formüle edilirse:
- Toplam talep 120, toplam arz 125.

| | Şehir 1 | Şehir 2 | Şehir 3 | Şehir 4 | Yapay | ARZ |
|-----------|---------|---------|---------|---------|-------|-----|
| Santral 1 | 8 | 6 | 10 | 9 | 0 | 35 |
| Santral 2 | 9 | 12 | 13 | 7 | 0 | 50 |
| Santral 3 | 14 | 9 | 16 | 5 | 0 | 40 |
| TALEP | 40 | 20 | 30 | 30 | 5 | 125 |

-
- Eğer toplam arz miktarı toplam talep miktarından azsa, aslında olurlu bir çözüm yoktur (talepler karşılanamaz).
 - Bu durumda karşılanamayan talep kadar arzı olan bir **yapay arz noktası** yaratırız.
 - Talebin olmayan bir arz noktasından karşılanamaması beraberinde bir "ceza maliyeti" getirir.

Örnek 3: Değiştirilmiş Powerco Örneği (Karşılanmamış Talep)

- Şehir 1'in talebinin 50 milyon kwh olduğunu ve karşılanamayan her 1 milyon kWh elektrik için €80 ceza maliyeti kesildiğini düşünelim.
- Bu durumda dengeli bir ulaştırma sorunu formüle edilirse:

| | Şehir 1 | Şehir 2 | Şehir 3 | Şehir 4 | ARZ |
|---------------------|---------|---------|---------|---------|-----|
| Santral 1 | 8 | 6 | 10 | 9 | 35 |
| Santral 2 | 9 | 12 | 13 | 7 | 50 |
| Santral 3 | 14 | 9 | 16 | 5 | 40 |
| Yapay (Eksiklik) | 80 | 80 | 80 | 80 | 5 |
| TALEP | 50 | 20 | 30 | 30 | 130 |

Ulaştırma Sorununa Başlangıç Temel Olurlu Çözüm (bfs) Bulma

- Diğer Doğrusal Programlama sorunlarının aksine, m arz noktası ve n talep noktası olan ($m+n$ eşitlik kısıtı olan)dengeli bir ulaştırma sorununu çözmek kolaydır.
- Bunun nedeni, karar değişkenlerinin (x_{ij} 'ler) bir kısıt haricinde tüm kısıtları sağlamaları durumunda, kalan kısıtın da x_{ij} 'lerin değerlerince sağlanacak olmasıdır.

-
- Dengeli bir ulařtırma sorununa bfs bulmak için üç temel yöntem vardır:
 - Kuzeybatı Köře Yöntemi
 - Enküçük Maliyet Yöntemi
 - Vogel'in Yaklařımı (VAM)

■ Bir **bfs**'de

- $m+n-1$ adet temel deęişken vardır
- Söz konusu temel deęişkenlerin temsil edildięi hücreler bir döngü oluşturmaz

-
- Bir değer atanan (boş olmayan) hücre bir **temel değişkeni** temsil eder.
 - En az dört hücrenin bir **döngü** oluşturması için:
 - Herhangi ardışık iki hücrenin aynı satır veya sütunda olması gerekir
 - Aynı satır veya sütunda ardışık üç hücre olmamalıdır
 - Serinin son hücresi ilk hücre ile aynı satır veya sütunda olup döngüyü kapatmalıdır

-
- Kuzeybatı Köşe Yöntemi taşıma maliyetlerini kullanmaz. Başlangıç bfs'si çok kolay bulunur ama toplam taşıma maliyeti çok yüksek olabilir.
 - Enküçük maliyet yöntemi taşıma maliyetlerini kullanır ve düşük maliyetli bir bfs bulur.
 - Vogel'in yaklaşımı aşırı yüksek taşıma maliyetlerini devre dışı bırakmak üzerinedir. Çok fazla hesaplama gerektirse de bulunan bfs çok büyük olasılıkla en düşük toplam taşıma maliyetini (en iyi çözüm) verir.

Ulaştırma Simpleks Algoritması ile En İyi Çözümü Bulma

- Özel bir prosedür kullanılarak ulaştırma tablosu üzerinde pivot işlemler yapılır:
 1. Mevcut bfs'nin en iyi çözüm olup olmadığı kontrol edilir. Eğer en iyi ise durulur. Değilse, ikinci adıma geçilir.
 2. Temele girecek temel olmayan değişken ve temelden çıkacak temel değişken belirlenir: Yeni bir bfs elde edilir. İlk adıma dönülür.

Geçici Konaklama Sorunları

- Bir ulaştırma sorunu sadece arz noktasından talep noktasına doğrudan taşımaya izin verir.
- Çoğu durumda, arz noktaları arasında veya talep noktaları arasında taşıma da söz konusu olabilir.
- Bazen de arz noktasından talep noktasına taşıma yaparken ürün/hizmetlerin geçici olarak konakladıkları noktalar (*geçici konaklama noktaları*) var olabilir.
- Geçici konaklama sorunları ulaştırma tablosuna aktarıldıktan sonra ulaştırma sorunu gibi çözülebilirler.

-
- Geçici konaklama sorununun en iyi çözümünün bulunmaya hazır hale getirilmesi:

Adım 1 Eğer gerekli ise yapay bir nokta ekleyerek sorunu dengeleyiniz. s = dengeli ulaştırma sorunu için toplam arz (talep) miktarı olsun.

Adım 2 : Her arz noktası ve geçici konaklama noktası için bir satır; her talep noktası ve geçici konaklama noktası için bir sütun oluşturularak bir ulaştırma tablosu kurunuz.

-
- Her arz noktasının arzı kendi orijinal arz miktarına, her talep noktasının talebi kendi orijinal talep miktarına eşit olacaktır.
 - Her geçici konaklama noktasının arzı (o noktanın orijinal arzı) + s ve her talep noktasının talebi (o noktanın orijinal talebi) + s olacaktır.

Örnek 4: Boğaziçi

- Boğaziçi LCD TV'ler üretmektedir. Biri İstanbul'da biri Bruges'da olmak üzere iki fabrikası vardır.
- İstanbul fabrikası günde 150 TV, Bruges fabrikası ise günde 200 TV üretmektedir.
- TV'ler uçak ile Londra ve Paris'deki müşterilere gönderilmektedir. Her şehrin talebi günde 130 TV'dir.
- Havayolu taşıma ücretlerindeki düzensizlik nedeni ile Boğaziçi bazı TV'leri önce Amsterdam veya Münih'e gönderip oradan müşterilere ulaştırmayı düşünmektedir.

- Boğaziçi gerekli gönderimi yaparken toplam taşıma maliyetlerini enazlamak istemektedir.
- Bir TV'yi uçak ile taşımanın maliyeti tabloda verilmiştir:

| € Nereden | Nereye | | | | | |
|--------------|----------|--------|-----------|---------|--------|-------|
| | Istanbul | Bruges | Amsterdam | Munchen | London | Paris |
| Istanbul | 0 | - | 8 | 13 | 25 | 28 |
| Bruges | - | 0 | 15 | 12 | 26 | 25 |
| Amsterdam | - | - | 0 | 6 | 16 | 17 |
| Munchen | - | - | 6 | 0 | 14 | 16 |
| London | - | - | - | - | 0 | - |
| Paris | - | - | - | - | - | 0 |

Ör. 4 – Yanıt

- Bu sorunda Amsterdam ve Münih *geçici konaklama noktalarıdır*.
- Adım 1. Sorunu dengeleme
Toplam arz = $150 + 200 = 350$
Toplam talep = $130 + 130 = 260$
Yapay talep = $350 - 260 = 90$
 $s = 350$ (dengeli sorun için toplam talep veya arz)

■ Adım 2. Ulaştırma tablosunun kurulması

Geçici konaklama noktası talebi = Orijinal talep + $s = 0 + 350 = 350$

Geçici konaklama noktası arzı = Orijinal arz + $s = 0 + 350 = 350$

| | Amsterdam | Munchen | London | Paris | Yapay | Arz |
|-----------|-----------|---------|--------|-------|-------|-----|
| Istanbul | 8 | 13 | 25 | 28 | 0 | 150 |
| Bruges | 15 | 12 | 26 | 25 | 0 | 200 |
| Amsterdam | 0 | 6 | 16 | 17 | 0 | 350 |
| Munchen | 6 | 0 | 14 | 16 | 0 | 350 |
| Talep | 350 | 350 | 130 | 130 | 90 | |

■ Adım 3. Ulaştırma tablosunun çözümü

| | Amsterdam | Munchen | London | Paris | Yapay | Arz |
|-----------|-----------------|-----------------|------------------|------------------|----------------|-----|
| Istanbul | 8 130 | 13 | 25 | 28 | 0 20 | 150 |
| Bruges | 15 | 12 | 26 | 25 130 | 0 70 | 200 |
| Amsterdam | 0 220 | 6 | 16 130 | 17 | 0 | 350 |
| Munchen | 6 | 0 350 | 14 | 16 | 0 | 350 |
| Talep | 350 | 350 | 130 | 130 | 90 | |

- Boğaziçi İstanbul'da 130 TV üretip bunları Amsterdam üzerinden Londra'ya taşınmalıdır.
- Bruges ise 130 TV üretip bunları doğrudan Paris'e göndermelidir.
- Toplam taşıma maliyeti 6370 Euro'dur.

Atama Sorunları

- Genel olarak bir **atama sorunu** tüm talep ve arzları 1'e eşit olan dengeli bir ulaştırma sorunudur.
- Bu sorunda tüm arz ve talepler tamsayıdır ve en iyi çözüm de tamsayılı çözümdür.
- Ulaştırma sorunu şeklinde çözmek yerine Macar yöntemi kullanmak daha uygundur.

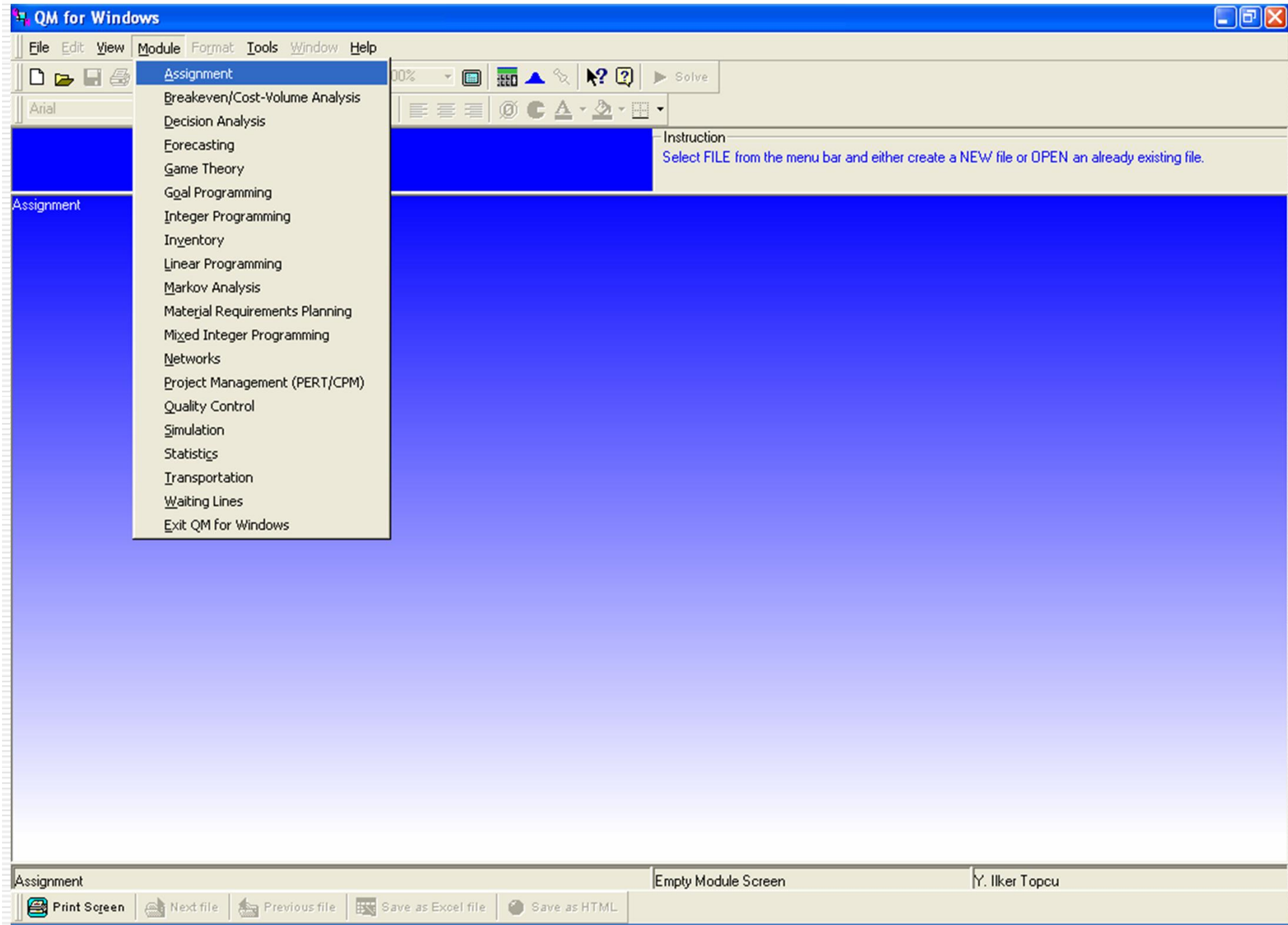
Örnek 5: Makine Atama

- Machineco şirketinin yapılması gereken dört işi vardır.
- Machineco şirketinin bu işler için kullanabileceği dört makinesi vardır.
- Her iş bir makineye atanacaktır.
- Machineco hangi işin hangi makineye atanacağını belirlemeye çalışmaktadır.

-
- Her iş için gerekli zaman aşağıda verilmiştir.

| | Time (Hours) | | | |
|-----------|--------------|------|------|------|
| | Job1 | Job2 | Job3 | Job4 |
| Machine 1 | 14 | 5 | 8 | 7 |
| Machine 2 | 2 | 12 | 6 | 5 |
| Machine 3 | 7 | 8 | 3 | 9 |
| Machine 4 | 2 | 4 | 6 | 10 |

- Machineco toplam üretim zamanını enküçükleme istemektedir.



Create data set for Assignment

Title:

Number of Jobs:

Number of Machines:

Objective

Maximize

Minimize

Row names | **Column names**

Job 1, Job 2, Job 3, ...

a, b, c, d, e, ...

A, B, C, D, E, ...

1, 2, 3, 4, 5, ...

January, February, March, April, ...

Other

Objective
 Maximize
 Minimize

Instruction
Enter the cost of assigning job 4 to machine 4. Any real value is permissible.

machineco

| | Machine 1 | Machine 2 | Machine 3 | Machine 4 |
|-------|-----------|-----------|-----------|-----------|
| Job 1 | 14 | 5 | 8 | 7 |
| Job 2 | 2 | 12 | 6 | 5 |
| Job 3 | 7 | 8 | 3 | 9 |
| Job 4 | 2 | 4 | 6 | 10 |

QM for Windows

File Edit View Module Format Tools Window Help

100% Edit

Arial 8.25 B I U

Objective: Maximize Minimize

Instruction: There are more results available in additional windows. These may be opened by using the WINDOW option in the Main Menu.

Assignment List

machineco solution

| JOB | Assigned to | Cost |
|-------|-------------|------|
| Job 1 | Machine 2 | 5. |
| Job 2 | Machine 4 | 5. |
| Job 3 | Machine 3 | 3. |
| Job 4 | Machine 1 | 2. |
| Total | | 15. |

Assignments

machineco solution

| Optimal cost = \$15 | Machine 1 | Machine 2 | Machine 3 | Machine 4 |
|---------------------|-----------|-----------|-----------|-----------|
| Job 1 | 14. | Assign 5 | 8. | 7. |
| Job 2 | 2. | 12. | 6. | Assign 5 |
| Job 3 | 7. | 8. | Assign 3 | 9. |
| Job 4 | Assign 2 | 4. | 6. | 10. |

Assignment Solution Screen Y. Ilker Topcu

Print Screen Next file Previous file Save as Excel file Save as HTML

■ Macar yöntemi adımları.

1. $m \times m$ 'lik maliyet matrisinin her satırındaki en küçük maliyeti bulunuz.
2. Her maliyetten kendi satırındaki en küçük maliyeti çıkararak bir matris kurunuz
3. Yeni matriste her sütunun en küçük maliyetini bulunuz
4. Bu sefer her maliyetten kendi sütunundaki en küçük maliyeti çıkararak yeni bir matris (indirgenmiş maliyet matrisi) kurunuz
5. İndirgenmiş maliyet matrisindeki tüm sıfırları örtecek şekilde en az sayıda (yatay veya düşey) çizgi çiziniz. Eğer bu işlem için m adet çizgi gerekli ise en iyi çözüm bulunmuştur. Eğer gerekli çizgi sayısı m adetten az ise bir sonraki adıma geçiniz
6. İndirgenmiş maliyet matrisinde Adım 5'de çizilen çizgiler ile örtülmemiş ve sıfır'dan farklı en küçük maliyeti (k) bulunuz
7. Her üstünden çizgi geçmeyen maliyetten k 'yı çıkarınız ve çift çizgi ile örtülen her maliyete k 'yı ekleyiniz. Adım 5'e dönünüz..

Gezgin Satıcı

- Hedeflenen noktaların tamamını ziyaret edip geri gelinmesini sağlayacak en az maliyetli (masraf, uzunluk veya zaman) bir rota bulmak
- Depodan yola çıkıp dağıtım yapıp geri dönecek kamyonlar
- Okul servisleri, çöp arabaları...
- Genelde sezgisel yaklaşımlarla iyi bir çözüm bulunur

En Yakın Komşu Sezgiseli

- Herhangi bir noktadan başla ve en yakın noktaya git.
- Son gidilen noktadan ziyaret edilmemiş en yakın noktaya git.
- Tüm noktalar ziyaret edilip bir tur oluşuncaya kadar devam et.
- Tek tek diğer noktaları da başlangıç noktası olarak alıp bulunan en iyi çözümü öner.

Örnek

- Burak'ın merkezi İzmir'de olan bir şirketi vardır. Her yıl sonunda İstanbul, Ankara, Antalya ve Adana'daki şubelerini ziyaret etmektedir.
- Yolculuğunun toplam süresini enazlamak için hangi sırada şubelerini ziyaret etmelidir?
- Süreler:

| | İzmir | İstanbul | Samsun | Antalya | Adana |
|----------|-------------|-------------|-------------|-------------|-------------|
| İzmir | - | 6 sa 18 dk | 11 sa 10 dk | 5 sa 35 dk | 10 sa 20 dk |
| İstanbul | 6 sa 18 dk | - | 7 sa 51 dk | 8 sa 18 dk | 10 sa 52 dk |
| Samsun | 11 sa 10 dk | 7 sa 51 dk | - | 10 sa 28 dk | 8 sa 10 dk |
| Antalya | 5 sa 35 dk | 8 sa 18 dk | 10 sa 28 dk | - | 8 sa 23 dk |
| Adana | 10 sa 20 dk | 10 sa 52 dk | 8 sa 10 dk | 8 sa 23 dk | - |

- İzmir'i başlangıç noktası olarak alalım
- Ziyaret edilmeyen şehirlerden İzmir'e en yakını Antalya'dır. İzmir'den Antalya'ya git.
- Ziyaret edilmeyen şehirlerden Antalya'ya en yakını İstanbul'dur. Antalya'dan İstanbul'a git.
- Ziyaret edilmeyen şehirlerden İstanbul'a en yakını Samsun'dur. İstanbul'dan Samsun'a git. Oradan Adana'ya git
- İzmir – Antalya – İstanbul – Samsun – Adana – İzmir
Toplam süre: 40 sa 14 dk

| | İzmir | İstanbul | Samsun | Antalya | Adana |
|----------|-------------|-------------|-------------|-------------|-------------|
| İzmir | - | 6 sa 18 dk | 11 sa 10 dk | 5 sa 35 dk | 10 sa 20 dk |
| İstanbul | 6 sa 18 dk | - | 7 sa 51 dk | 8 sa 18 dk | 10 sa 52 dk |
| Samsun | 11 sa 10 dk | 7 sa 51 dk | - | 10 sa 28 dk | 8 sa 10 dk |
| Antalya | 5 sa 35 dk | 8 sa 18 dk | 10 sa 28 dk | - | 8 sa 23 dk |
| Adana | 10 sa 20 dk | 10 sa 52 dk | 8 sa 10 dk | 8 sa 23 dk | - |

- İstanbul'u başlangıç noktası olarak alalım
- İstanbul'dan İzmir'e git.
- İzmir'den Antalya'ya git.
- Antalya'dan Adana'ya git.
- Adana'dan Samsun'a git
- İstanbul – İzmir – Antalya – Adana – Samsun – İstanbul
- İzmir – Antalya – Adana – Samsun – İstanbul – İzmir
Toplam süre: 36 sa 17 dk

| | İzmir | İstanbul | Samsun | Antalya | Adana |
|----------|-------------|-------------|-------------|-------------|-------------|
| İzmir | - | 6 sa 18 dk | 11 sa 10 dk | 5 sa 35 dk | 10 sa 20 dk |
| İstanbul | 6 sa 18 dk | - | 7 sa 51 dk | 8 sa 18 dk | 10 sa 52 dk |
| Samsun | 11 sa 10 dk | 7 sa 51 dk | - | 10 sa 28 dk | 8 sa 10 dk |
| Antalya | 5 sa 35 dk | 8 sa 18 dk | 10 sa 28 dk | - | 8 sa 23 dk |
| Adana | 10 sa 20 dk | 10 sa 52 dk | 8 sa 10 dk | 8 sa 23 dk | - |

- Samsun'u başlangıç noktası olarak alalım
- Samsun'dan İstanbul'a git.
- İstanbul'dan İzmir'e git
- İzmir'den Antalya'ya git.
- Antalya'dan Adana'ya git.
- Samsun – İstanbul – İzmir – Antalya – Adana – Samsun
- İzmir – Antalya – Adana – Samsun – İstanbul – İzmir
Toplam süre: 36 sa 17 dk

| | İzmir | İstanbul | Samsun | Antalya | Adana |
|----------|-------------|-------------|-------------|-------------|-------------|
| İzmir | - | 6 sa 18 dk | 11 sa 10 dk | 5 sa 35 dk | 10 sa 20 dk |
| İstanbul | 6 sa 18 dk | - | 7 sa 51 dk | 8 sa 18 dk | 10 sa 52 dk |
| Samsun | 11 sa 10 dk | 7 sa 51 dk | - | 10 sa 28 dk | 8 sa 10 dk |
| Antalya | 5 sa 35 dk | 8 sa 18 dk | 10 sa 28 dk | - | 8 sa 23 dk |
| Adana | 10 sa 20 dk | 10 sa 52 dk | 8 sa 10 dk | 8 sa 23 dk | - |

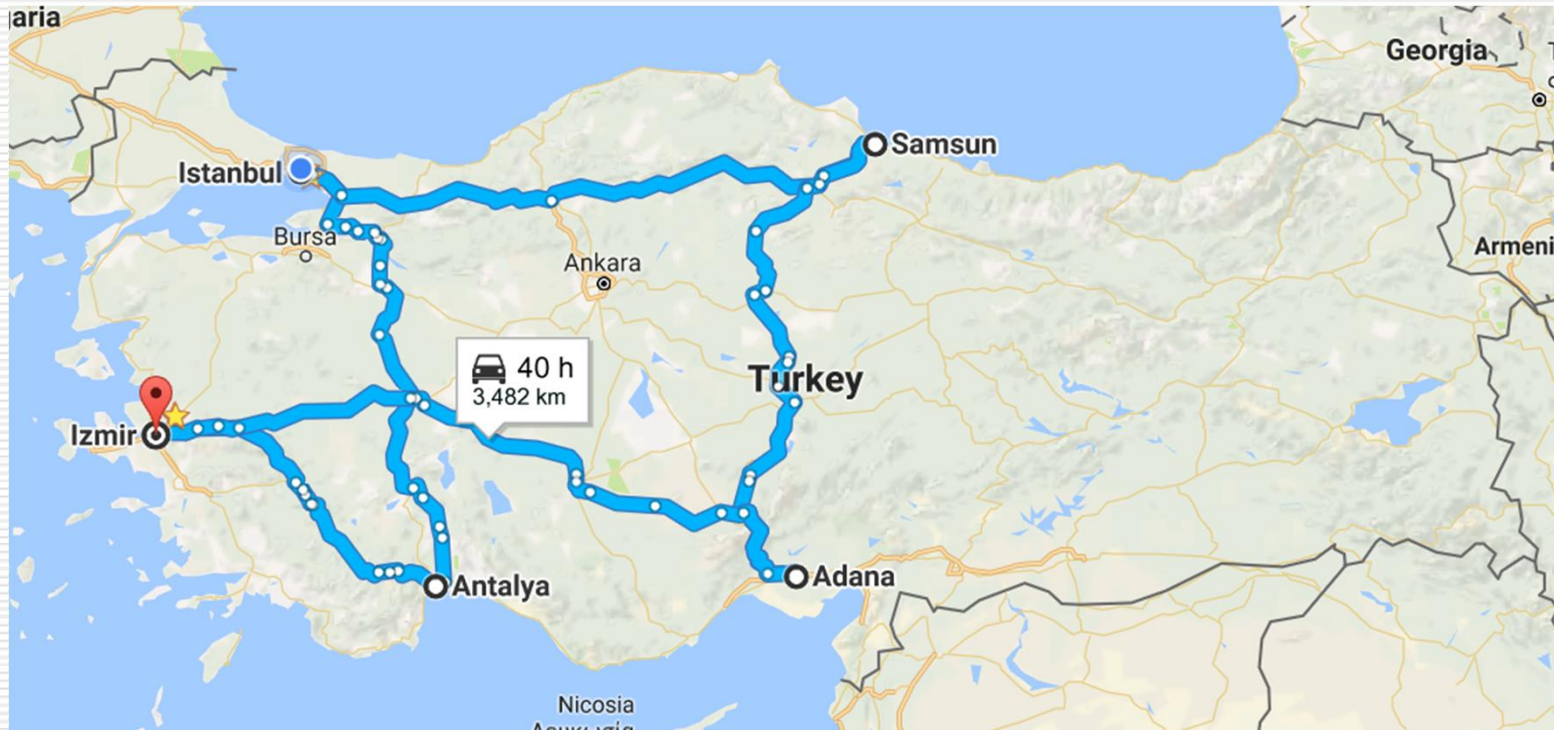
- Antalya'yı başlangıç noktası olarak alalım
- Antalya'dan İzmir'e git.
- İzmir'den İstanbul'a git.
- İstanbul'dan Samsun'a git.
- Samsun'dan Adana'ya git
- Antalya – İzmir – İstanbul – Samsun – Adana – Antalya
- İzmir – İstanbul – Samsun – Adana – Antalya – İzmir
Toplam süre: 36 sa 17 dk

| | İzmir | İstanbul | Samsun | Antalya | Adana |
|----------|-------------|-------------|-------------|-------------|-------------|
| İzmir | - | 6 sa 18 dk | 11 sa 10 dk | 5 sa 35 dk | 10 sa 20 dk |
| İstanbul | 6 sa 18 dk | - | 7 sa 51 dk | 8 sa 18 dk | 10 sa 52 dk |
| Samsun | 11 sa 10 dk | 7 sa 51 dk | - | 10 sa 28 dk | 8 sa 10 dk |
| Antalya | 5 sa 35 dk | 8 sa 18 dk | 10 sa 28 dk | - | 8 sa 23 dk |
| Adana | 10 sa 20 dk | 10 sa 52 dk | 8 sa 10 dk | 8 sa 23 dk | - |

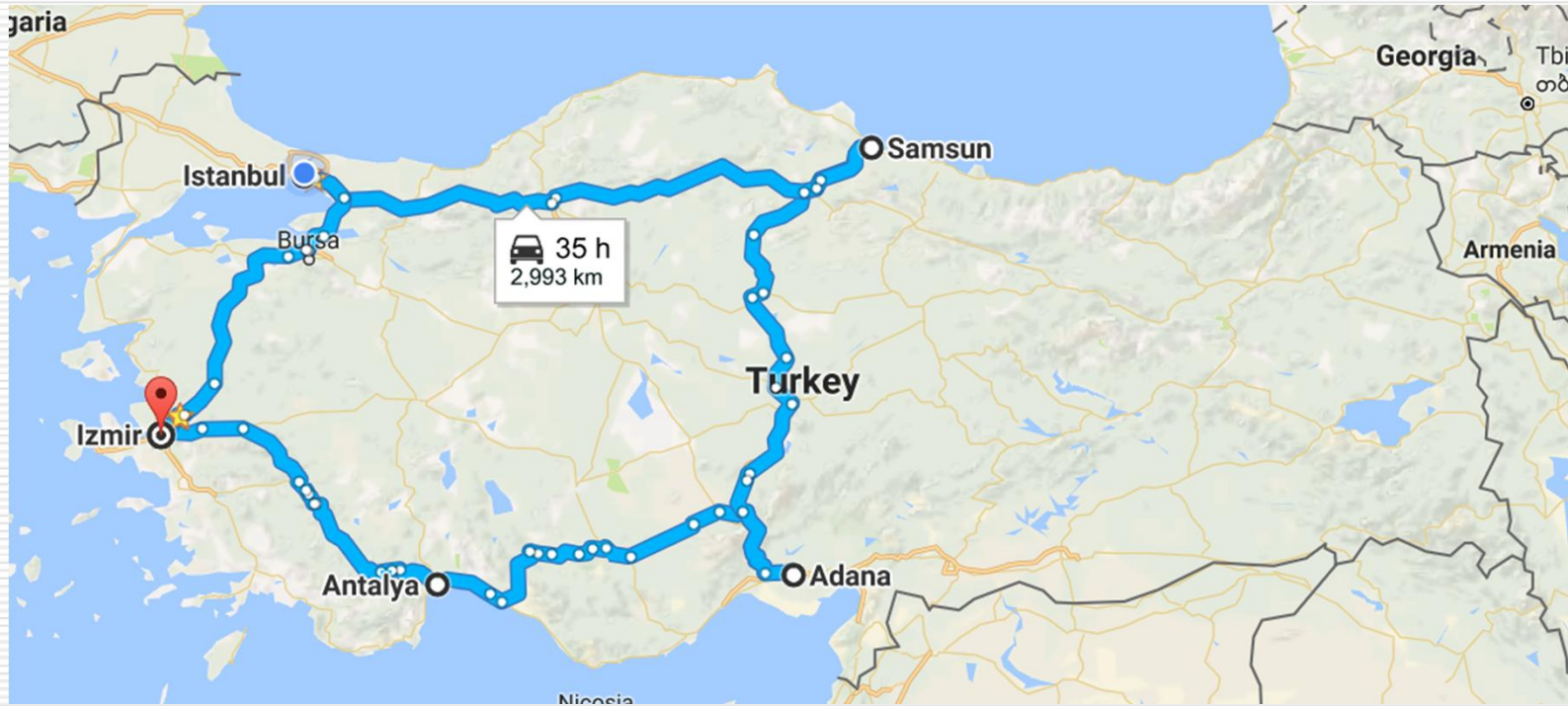
- Adana'yı başlangıç noktası olarak alalım
- Adana'dan Samsun'a git.
- Samsun'dan İstanbul'a git.
- İstanbul'dan İzmir'e git.
- İzmir'den Antalya'ya git
- Adana – Samsun – İstanbul – İzmir – Antalya – Adana
- İzmir – Antalya – Adana – Samsun – İstanbul – İzmir
Toplam süre: 36 sa 17 dk

| | İzmir | İstanbul | Samsun | Antalya | Adana |
|----------|-------------|-------------|-------------|-------------|-------------|
| İzmir | - | 6 sa 18 dk | 11 sa 10 dk | 5 sa 35 dk | 10 sa 20 dk |
| İstanbul | 6 sa 18 dk | - | 7 sa 51 dk | 8 sa 18 dk | 10 sa 52 dk |
| Samsun | 11 sa 10 dk | 7 sa 51 dk | - | 10 sa 28 dk | 8 sa 10 dk |
| Antalya | 5 sa 35 dk | 8 sa 18 dk | 10 sa 28 dk | - | 8 sa 23 dk |
| Adana | 10 sa 20 dk | 10 sa 52 dk | 8 sa 10 dk | 8 sa 23 dk | - |

- İzmir – Antalya – İstanbul – Samsun – Adana – İzmir
Toplam süre: 40 sa 14 dk



- İzmir – Antalya – Adana – Samsun – İstanbul – İzmir
 - İzmir – İstanbul – Samsun – Adana – Antalya – İzmir
- Toplam süre: 36 sa 17 dk

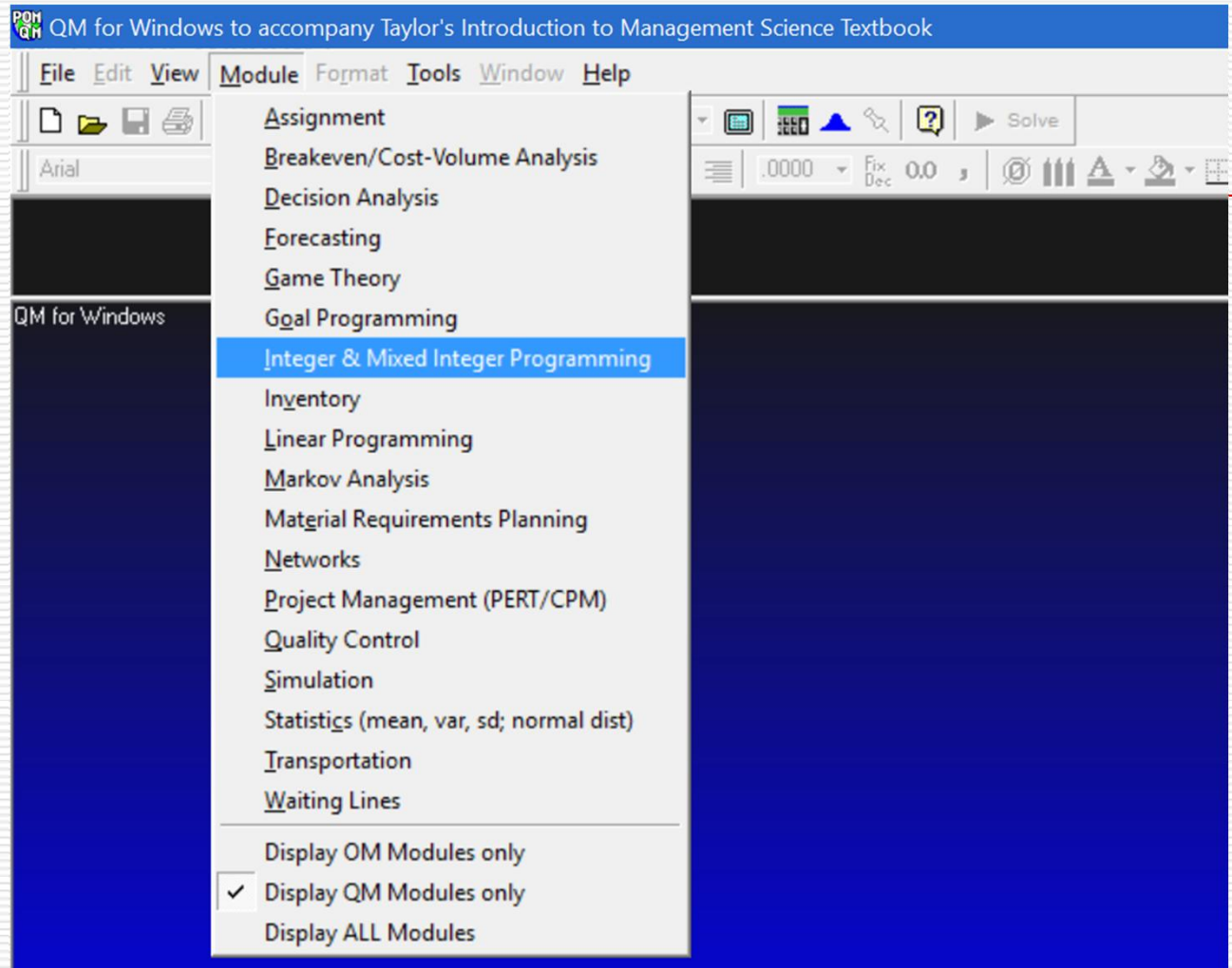


Kargo Yükleme

- Geleneksel sırt çantası problemi:
 - Bir sırt çantası var
 - Taşınması gereken ölçüleri (hacim, ağırlık...) ve değeri / yararı belli eşyalar var
 - Amaç taşınabilecek toplam eşya değerini en büyükmek
- 0-1 Tamsayı Programlama Modeli:
 - Tek kısıt var
 - Katsayılar negatif değil

$$\begin{aligned} \max z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{s.t.} \quad &a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b \\ &x_i = 0 \text{ or } 1 \quad (i = 1, 2, \dots, n) \end{aligned}$$

- c_i i . eşyanın taşınmasının yararı
- b taşınabilecek toplam hacim veya ağırlık
- a_i i . eşyanın hacim veya ağırlığı



Create data set for Integer && Mixed Integer Programming

Title: (untitled) Modify default title

Number of Constraints 1

Number of Variables 4

Objective

- Maximize
- Minimize

Row names Column names Overview

- Constraint 1, Constraint 2, Constraint 3,...
- a, b, c, d, e, ...
- A, B, C, D, E, ...
- 1, 2, 3, 4, 5, ...
- January, February, March, April, ...
Click here to set start month
- Other

Cancel Help OK

Objective

Maximize
 Minimize

Maximum number of iterations:

Maximum level (depth) in procedure:

(untitled)

| | X1 | X2 | X3 | X4 | | RHS | Equation form |
|---------------|---------|---------|---------|---------|----|-----|-------------------------------------|
| Maximize | 8 | 11 | 6 | 4 | | | Max $8X_1 + 11X_2 + 6X_3 + 4X_4$ |
| Constraint 1 | 5 | 7 | 4 | 3 | <= | 14 | $5X_1 + 7X_2 + 4X_3 + 3X_4 \leq 14$ |
| Variable type | Integer | Integer | Integer | Integer | <= | | |
| | | | | | = | | |
| | | | | | >= | | |

| Variable | Type | Value |
|----------------|------|-------|
| X1 | 0/1 | 0 |
| X2 | 0/1 | 1 |
| X3 | 0/1 | 1 |
| X4 | 0/1 | 1 |
| Solution value | | 21 |

$$\begin{aligned} \max z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{s.t.} \quad &a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b \\ &x_i = 0 \text{ or } 1 \quad (i = 1, 2, \dots, n) \end{aligned}$$

- c_i i . eşyanın taşınmasının yararı
- b taşınabilecek toplam hacim veya ağırlık
- a_i i . eşyanın hacim veya ağırlığı

- $r_i = c_i / a_i$
- Daha iyi eşyaların r oranı yüksek

-
- Tüm oranları hesapla
 - En iyi eşyayı sırt çantasına koy
 - Daha sonra ikinci iyi eşyayı koy
 - Bu şekilde çantada yer kalmayıncaya kadar devam et
 - Kalan kısma son eşyanın bir kısmını koyduğunu varsay

■ Oranlar:

$$r_1 = 8 / 5 = 1.6$$

$$r_2 = 11 / 7 = 1.57$$

$$r_3 = 6 / 4 = 1.5$$

$$r_4 = 4 / 3 = 1.33$$

■ İlk çözüm:

$$x_1 = 1, x_2 = 1, x_3 = 0.5, x_4 = 0, z = 22$$

$$\begin{array}{l} \max z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \end{array}$$

-
- x_3 için geliştirilen problemler:
 - P1 ($x_3=0$) çözüm: $x_3=0, x_1=x_2=1, x_4=2/3, z=21.67$
 - P2 ($x_3=1$) çözüm: $x_3=x_1=1, x_2=5/7, x_4=0, z=21.85$
 - P2 çözüm değeri daha iyi olduğu için bu sefer x_2 için:
 - P3 (P2 + $x_2=0$) çözüm: $x_3=1, x_2=0, x_1=1, x_4=1, z=18$
 - P4 (P2 + $x_2=1$) çözüm: $x_3=x_2=1, x_1=3/5, x_4=0, z=21.8$
 - P3 çözümü uygun bir çözüm (0/1)
 - Aday çözüm = 18

$$\begin{array}{l} \max z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \end{array}$$

-
- P4 seçilirse x_1 için problemler geliştirilir:
 - P5 (P4 + $x_1=0$) **çözüm:** $x_3=x_2=1, x_1=0, x_4=1, z=21$
 - P6 (P4 + $x_1=1$) **çözüm:** Uygun değil ($x_3=x_2=x_1=1 \rightarrow 16$)
 - P5 çözümü uygun
→ Yeni aday çözüm = 21

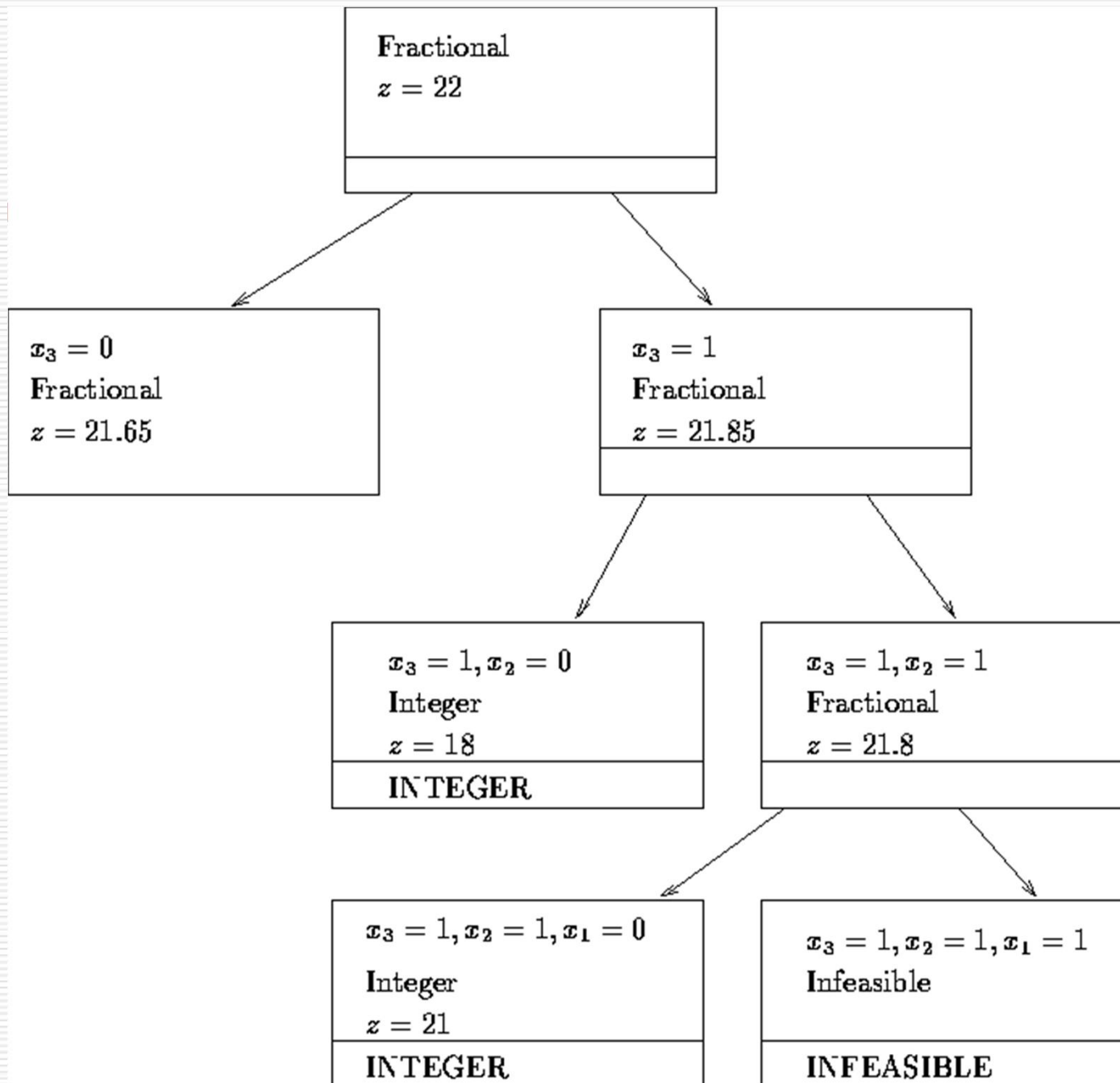
$$\begin{array}{l} \max z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{s.t.} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \end{array}$$

-
- P1 için problem yaratılmadı ama P1 için toplam yarar 21.67

- Bu durumda toplam yararı 21 olan uygun çözüm en iyi çözüm:

$$z=21, x_1=0, x_2=1, x_3=1, x_4=1$$

- 2., 3. ve 4. eşyalar taşınmalı
- Toplam değer 21 olur



Teşekkürler...

Yazılımı indirmek için web adresi:

www.isl.itu.edu.tr/ya

(yöntem/yazılım bağlantısı)