# YÖNEYLEM ARAŞTIRMASI

*ŞEBEKE MODELLEMESİ*

*Prof.Dr.Birol ELEVLİ*

1

1

## Şebeke Modeli

Üretim, dağıtım, proje planlaması, kaynak yönetimi, finansal planlama ve benzeri problemlerde yaygın olarak kullanılır,

Sistemin parçaları arasındaki ilişkiyi görsel ve düşünsel olarak görmemize yarar,

Genel olarak beş gruba ayrılırlar:

1. Minimum dağılım(Minimum kapsayan ağaç)
2. En kısa yol problemi
3. Maksimum akış problemi
4. Minimum maliyetli akış problemi
5. Proje Planlama ve Kontrol (PERT/CPM

2

2

## Şebeke Kavramları

Nokta ve çizgilerden meydana gelmiştir,

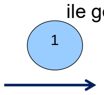
Noktalar *düğüm* olarak

Çizgiler *kenar* olarak

adlandırılır.

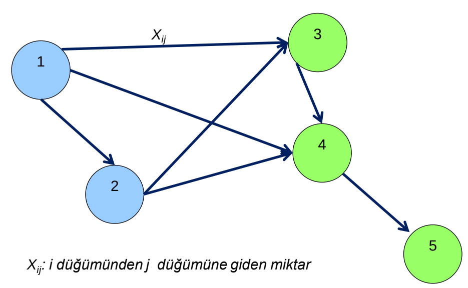
Düğümler ile

Kenarlar ile gösterilir




3

## Şebeke Gösterimi




$X_{ij}$: *i düğümünden j düğümüne giden miktar*

4

4

## Minimum Dağılım Problemi

Bir şebekedeki tüm düğümleri birbirine bağlayan ve toplamda en kısa mesafeyi veren bağlantıları/yolları bulma problemidir

Genel karakteristiği;

- Düğümler biliniyor, ancak potansiyel bağlantılar biliniyor. Potansiyel bağlantılarla ilgili mesafe/maliyet vs. biliniyor
- Her düğüm çiftinin bağlantısını sağlayacak şebeke oluşturulmalıdır,
- Amaç, isteneni minimum mesafe/maliyetle yerine getirmek
- Bu problemin optimum çözümü her zaman bir minimum kapsayan ağaçtır. Çünkü bağlantı sayısı herzaman düğüm sayısından bir eksiktir



## Çözüm Algoritması

- 1. En ucuz/kısa bağlantıyı seç,
- 2. Bağlantısı olan düğümlere bağlanabilecek en ucuz/kısa bağlantıyı seç,
- 3. Bağlantısı olmayan düğüm kalana kadar 2.adımı tekrarla
- Not:Eşitlik durumunda keyfi seçim yapılır



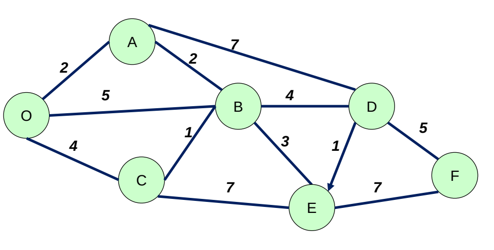
## Uygulama problemleri

- İletişim ağı tasarımında( internet bağlantısı,telefon hatları, kablolu tv, vs.)
- Bağlantı sağlamanın toplam maliyetini minimize etmek için az kullanılan ulaşım ağlarının tasarımında (demiryolu, karayolu, vs).
- Yüksek ve orta voltajlı elektik nakil hatlarının tasarımında,
- Toplam kablo uzunluğunu minimize etmek için elektrikli ekipmanlar için elektrik şebekesi tasarımında.
- Belirli noktaları birbirine bağlayan boru hatlarının tasarımında,
- Benzeri problemlerde



## Örnek Problem

*Aşağıda potansiyel bağlantıları gösterilen her kasabaya en az bir asfalt yol bağlantısı olması isteniyor. Hangi yolları asfalt yaparsınız?*



## Örnek_devam

| A | B | C | | D | | E | F |
|---|---|---|---|---|---|---|---|
| | | Düğ | Mes | Düğ | Mes | | |
| 1 | B | A,O,C,D,E | 2,5,1,4,3 | C | 1 | BC | 1 |
| 2 | B<br>C | A,O,D,E<br>O,E | 2,5,4,3<br>4,4 | A | 2 | BA | 1+2=3 |
| 3 | A<br>B<br>C | O,D<br>O,D,E<br>O,E | 2,7<br>5,4,3<br>4,4 | 0 | 2 | OA | 3+2=5 |

*A-Aşama; B- Seçilen düğüm;C-Seçili düğüme bağlanabilecek düğüm; D-Seçilen düğüme en yakın; E-Bağlantı; F-Toplam mesafe/maliyet*

9

## Örnek_devam

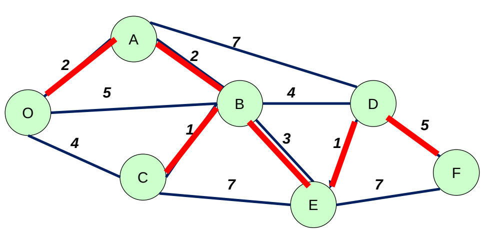
| A | B | C | | D | | E | F |
|---|---|---|---|---|---|---|---|
| | | Düğ | Mes | Düğ | Mes | | |
| 4 | A<br>B<br>C | D<br>D,E<br>E | 7<br>4,3<br>4 | E | 3 | BE | 5+3=8 |
| 5 | A<br>B<br>E | D<br>D<br>D | 7<br>4<br>1 | D | 1 | ED | 8+1=9 |
| 6 | D<br>E | F<br>F | 5<br>7 | F | 5 | DF | 9+5=14 |

*A-Aşama; B- Seçilen düğüm;C-Seçili düğüme bağlanabilecek düğüm; D-Seçilen düğüme en yakın; E-Bağlantı; F-Toplam mesafe/maliyet*

10

## Örnek Problem

*Bağlantılar; BC, BA, AO, BE, ED, ve DF*




11

## En Kısa Yol Problemi

- Şebeke içinde belirlenen bir başlangıç noktasından herhangi bir hedef noktaya giden en kısa yolu bulma problemidir
  - Başlangıç düğümü olmalı
  - Hedef düğüm olmalı
- Çözüm için 2 algoritma vardır
  - *Dijkstra Algoritması (Etiketleme algoritması)*
  - *Floyd-Marshall Algoritması*

12

12

## Dijkstra Algoritması

Özel bir etiketleme işlemleridir

- $u_i$ , 1. düğümden i. düğüme gelen en kısa uzaklık
- $[u_j,i] = [u_i+d_{ij},i]$ ifadesi de j düğümü için etikettir

1.Adım: Kaynak düğümünü kalıcı etiketle [0,-]. i=1 olsun.

2.Adım:

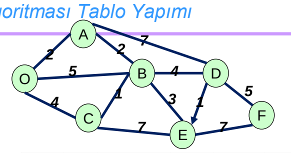
a) j'nin kalıcı etiketli olmaması koşulu ile i.düğümden ulaşılan j-her j düğümü için geçici $[u_i+d_{ij},i]$ etiketlerini hesapla. Eğer j düğümü başka bir k düğümü içinde $[u_j,k]$ ile zaten etiketlenmişse ve $u_i+d_{ij}<u_j$ ise, $[u_j,k]$ 'yi $[u_i+d_{ij},i]$ *ile değiştir.*

b) *Tüm düğümlerde kalıcı etiket varsa dur. Aksi halde tüm geçici etiketler arasından, $[u_r,s]$'nin en kısa mesafeli olanını seç ve i.adımı tekrarla.*

13

13

## Dijkstra Algoritması Tablo Yapımı




| Adım | A | B | | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|
| | | Düğ. | Mes | | | | | |
| 1 | O | A<br>B<br>C | 2<br>5<br>4 | A | 2 | A | 2 | OA |
| 2 | O | B<br>C | 5<br>4 | C | 4 | C | 4 | OC |
| | A | B<br>D | 2<br>7 | B | 2+2=4 | B | 4 | AB |
| 3 | A<br>B<br><br>C | D<br>D<br>E<br>E | 7<br>4<br>3<br>4 | D<br><br>E<br>E | 2+7=9<br><br>4+3=7<br>4+4=8 | E | 7 | BE |
| 4 | A<br>B<br>E | D<br>D<br>D<br>T | 7<br>4<br>1<br>7 | D<br>D<br>D | 2+7=9<br>4+4=8<br>7+1=8 | D<br>D | 8<br>8 | BD<br>ED |
| 5 | D | T | 5 | T | 8+5=13 | T | 13 | DT |
| | E | T | 7 | T | 7+7=14 | | | |

En Kısa Yol Probleminin Çözümü

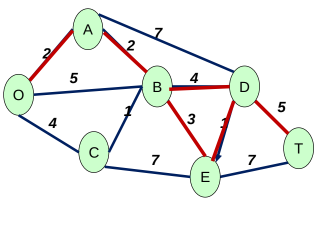
Sütunların açıklaması :A : Çözülmemiş düğüme bağlı çözülmüş düğümler, B: Çözülmüş düğüme bağlanabilecek çözülmemiş düğümler
C: En yakın çözülmemiş düğüm; D: Toplam Mesafe; E: En yakın düğüm; F: Min mesafe; G: Son bağlantı

14

14

## Dijkstra Algoritması Ssnuç


| Adım | A | B | | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|
| | | Düğ. | Mes | | | | | |
| 1 | O | A<br>B<br>C | 2<br>5<br>4 | A | 2 | A | 2 | OA |
| 2 | O | B<br>C | 5<br>4 | C | 4 | C | 4 | OC |
| | A | B<br>D | 2<br>7 | B | 2+2=4 | B | 4 | AB |
| 3 | A<br>B<br><br>C | D<br>D<br>E<br>E | 7<br>4<br>3<br>4 | D<br><br>E<br>E | 2+7=9<br><br>4+3=7<br>4+4=8 | E | 7 | BE |
| 4 | A<br>B<br>E | D<br>D<br>D<br>T | 7<br>4<br>1<br>7 | D<br>D<br>D | 2+7=9<br>4+4=8<br>7+1=8 | D<br>D | 8<br>8 | BD<br>ED |
| 5 | D | T | 5 | T | 8+5=13 | T | 13 | DT |
| | E | T | 7 | T | 7+7=14 | | | |

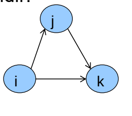




15

15

## Floyd Algoritması

- Dijkstra algoritmasından daha geneldir. Çünkü şebekedeki herhangi iki düğüm arasındaki en kısa yolu belirler.
- Algoritma, n düğümlü şebekeyi n satırlı ve n sütünlu kare matris olarak gösterir.
- Matrisin *(i,j)* elemanı, i.düğümden j.düğüme olan $d_{ij}$ uzaklığını verir. *i* doğrudan *j*'ye bağlı ise $d_{ij}$ sonlu yoksa sonsuzdur.
- Şekilde eğer $d_{ij}+d_{jk} < d_{ik}$ ise i'den başlayıp ,j'den geçerek k'ye ulaşmak daha kısadır. Bu üçlü işlemdir.




16

16

**0.Adım:** aşağıdaki gibi $D_0$ uzaklık matrisini hazırla. Bu matrisin çapraz elemanları sıfır olup, birbirine bağlantısı olmayan düğümler (sonsuz) işareti ile gösterilir.

$D_0$=i

| | | | | | |
|---|---|---|---|---|---|
| - | $d_{12}$ | … | $d_{1j}$ | … | $d_{1n}$ |
| $d_{21}$ | - | | | | |
| . . | | | | | |
| $d_{i1}$ | | | | | |
| . . | | | | | |
| $d_{n1}$ | | | | | |

17

17

**k.genel adım:** k.satırı ve k.sütunu anahtar satır ve anahtar sütun olarak tanımla. $D_k$ matrisine anahtar satır ve sütündaki değerleri yaz.

$D_k$ matrisindeki diğer elemanları aşağıdaki bağıntıya göre belirle:

$$D_k\,(i,j) = \min \{D_{k-1}\,(i,j)\ ,\ D_{k-1}\,(i,k) + D_{k-1}\,(k,j)\ \}$$

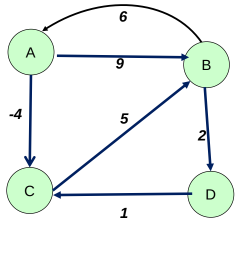
Bu işlemleri tüm düğümler için tekrarla.

18

18

## Floyd Algoritması Örnek

Aşağıdaki şebekedeki en kısa yolu bulunuz




*Bunun $D_0$ matrisini oluşturalım:*

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | -4 | $\propto$ |
| B | 6 | 0 | $\propto$ | 2 |
| C | $\propto$ | 5 | 0 | $\propto$ |
| D | $\propto$ | $\propto$ | 1 | 0 |

19

19

## Floyd Algoritması Örnek

*Aşağıdaki $D_0$ matrisinden D1 matrisi oluşturulmalı:*

| $D_O$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | -4 | $\propto$ |
| B | 6 | 0 | $\propto$ | 2 |
| C | $\propto$ | 5 | 0 | $\propto$ |
| D | $\propto$ | $\propto$ | 1 | 0 |

*Bunun $D_1$ matrisini oluşturalım: Bunun için öncelikle 1.satır ve 1. sütun anahtar satır kabul edilir*

| $D_1$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | $\propto$ | -4 |
| B | 6 | 0 | 2 | 2 |
| C | $\propto$ | 5 | 0 | $\propto$ |
| D | $\propto$ | $\propto$ | 1 | 0 |

*$D_1$ matrisini nin diğer elemanları aşağıdaki gibi hesaplanır:*

$D_1$ (BC) = min {$D_0$(BC) ; $D_0$(BA) + $D_0$(AC)} , *yerine yazarsak*

$D_1$ (BC) = min {2 ; 6+ $\propto$} $\Rightarrow D_1$ (BC) = 2

$D_1$ (BD) = min {$D_0$(BD) ; $D_0$(BA) + $D_0$(AD)} , *yerine yazarsak*

$D_1$ (BD) = min {2 ; 6+ $\propto$} $\Rightarrow D_1$ (BD) = 2

$D_1$ (CB) = min {$D_0$(CB) ; $D_0$(CA) + $D_0$(AB)} , *yerine yazarsak*

$D_1$ (CB) = min {5; $\propto$ + 9} $\Rightarrow D_1$ (BD) = 5

$D_1$ (CD) = min {$D_0$(CD) ; $D_0$(CA) + $D_0$(AD)} , *yerine yazarsak*

$D_1$ (CD) = min {$\propto$ ; $\propto$ + $\propto$} $\Rightarrow D_1$ (BD) = $\propto$

$D_1$ (DC) = min {1 ; $\propto$ + -4} $\Rightarrow D_1$ (DC) = 1

20

20

## Floyd Algoritması Örnek

*Aşağıdaki $D_1$ matrisinden $D_2$ matrisi oluşturulmalı:*

| $D_1$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | $\propto$ | -4 |
| B | 6 | 0 | 2 | 2 |
| C | $\propto$ | 5 | 0 | $\propto$ |
| D | $\propto$ | $\propto$ | 1 | 0 |

*Bunun $D_2$ matrisini oluşturalım: Bunun için öncelikle 2.satır ve 2. sütun anahtar satır kabul edilir*

| $D_2$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | 11 | 11 |
| B | 6 | 0 | 2 | 2 |
| C | 11 | 5 | 0 | 7 |
| D | $\propto$ | $\propto$ | 1 | 0 |

*$D_2$ matrisini nin diğer elemanları aşağıdaki gibi hesaplanır:*

$D_2$ (AC) = min {$D_1$(AC) ; $D_1$(AB) + $D_1$(BC)} , yerine yazarsak

$D_2$ (AC) = min {$\propto$ ; 9+ 2} $\Rightarrow$ $D_2$ (AC) = 11

$D_2$ (AD) = min {$D_1$(AD) ; $D_1$(AB) + $D_1$(BD)} , yerine yazarsak

$D_2$ (AD) = min {-4 ; 9+ 2} $\Rightarrow$ $D_2$ (AD) = 11

$D_2$ (CA) = min {$D_1$(CA) ; $D_1$(CB) + $D_1$(BA)} -> $D_2$ (CA) = min {$\propto$ ; 5+ 6} $\Rightarrow$ $D_2$ (CA) = 11

$D_2$ (CD) = min {$D_1$(CD) ; $D_1$(CB) + $D_1$(BD)} -> $D_2$ (CD) = min {$\propto$ ; 5+2} $\Rightarrow$ $D_2$ (CD) = 7

$D_2$ (DA) = min {$D_1$(DA) ; $D_1$(DB) + $D_1$(BA)} -> $D_2$ (DA) = min {$\propto$ ; $\propto$ + 6} $\Rightarrow$ $D_2$ (DA) = $\propto$

$D_2$ (DC) = min {$D_1$(DC) ; $D_1$(DB) + $D_1$(BC)} -> $D_2$ (DC) = min {1 ; $\propto$ + 2} $\Rightarrow$ $D_2$ (DC) = 1



## Floyd Algoritması Örnek

*Aşağıdaki $D_2$ matrisinden $D_3$ matrisi oluşturulmalı:*

| $D_2$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | 11 | 11 |
| B | 6 | 0 | 2 | 2 |
| C | 11 | 5 | 0 | 7 |
| D | $\propto$ | $\propto$ | 1 | 0 |

*Bunun $D_3$ matrisini oluşturalım: Bunun için öncelikle 3.satır ve 3. sütun anahtar satır kabul edilir*

| $D_3$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | 11 | 11 |
| B | 6 | 0 | 2 | 2 |
| C | 11 | 5 | 0 | 7 |
| D | 12 | 6 | 1 | 0 |

*$D_3$ matrisini nin diğer elemanları aşağıdaki gibi hesaplanır:*

$D_3$ (A**B**) = min {$D_2$(AB) ; $D_2$(AC) + $D_2$(CB)} , yerine yazarsak

$D_3$(AB) = min { 9 ; 11+ 5} $\Rightarrow$ $D_2$ (AC) = 9

$D_3$(AD) = min {$D_2$(AD) ; $D_2$(AC) + $D_2$(CD)} = $D_3$(AD) = min {11; 11+ 7} $\Rightarrow$ $D_3$ (AD) = 11

$D_3$ (BA) = min {$D_2$(BA) ; $D_2$(BC) + $D_2$(CA)} -> $D_3$ (BA) = min {6 ; 2+ 11} $\Rightarrow$ $D_3$(CA) = 6

$D_3$ (BD) = min {$D_2$(BD) ; $D_2$(BC) + $D_2$(CD)} -> $D_3$ (BD) = min {2 ; 2+7} $\Rightarrow$ $D_2$ (CD) = 2

$D_3$ (DA) = min {$D_2$(DA) ; $D_2$(DC) + $D_2$(CA)} -> $D_3$ (DA) = min {$\propto$ ; 1+ 11} $\Rightarrow$ $D_2$ (DA) = 12

$D_3$ (DB) = min {$D_2$(DB) ; $D_2$(DC) + $D_2$(CB)} -> $D_3$ (DB) = min {$\propto$ ; 1 + 5} $\Rightarrow$ $D_2$ (DC) = 6



## Floyd Algoritması Örnek

*Aşağıdaki $D_3$ matrisinden $D_4$ matrisi oluşturulmalı:*

| $D_3$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | 11 | 11 |
| B | 6 | 0 | 2 | 2 |
| C | 11 | 5 | 0 | 7 |
| D | 12 | 6 | 1 | 0 |

*Bunun $D_3$ matrisini oluşturalım: Bunun için öncelikle 4.satır ve 4. sütun anahtar satır kabul edilir*

| $D_4$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | 11 | 11 |
| B | 6 | 0 | 2 | 2 |
| C | 11 | 5 | 0 | 7 |
| D | 12 | 6 | 1 | 0 |

*$D_4$ matrisini nin diğer elemanları aşağıdaki gibi hesaplanır:*

$D_4$ (A**B**) = min {$D_3$(AB) ; $D_3$(AD) + $D_3$(DB)} , yerine yazarsak

$D_4$(AB) = min { 9 ; 11+ 6} $\Rightarrow$ $D_4$ (AB) = 9

$D_4$(AC) = min {$D_3$(AC) ; $D_3$(AD) + $D_3$(DC)} .. $D_4$(AC) = min {11; 11+ 1} $\Rightarrow$ $D_4$ (AC) = 11

$D_4$ (BA) = min {$D_3$(BA) ; $D_3$(BD) + $D_3$(DA)} -> $D_4$ (BA) = min {6 ; 2+ 12} $\Rightarrow$ $D_4$(BA) = 6

$D_4$ (BC) = min {$D_3$(BC) ; $D_3$(BD) + $D_3$(DC)} -> $D_4$ (BC) = min {2 ; 2+1} $\Rightarrow$ $D_2$ (CD) = 2

$D_4$ (CA) = min {$D_3$(CA) ; $D_3$(CD) + $D_3$(DA)} -> $D_4$ (CA) = min {11; 7+ 12} $\Rightarrow$ $D_2$ (DA) = 11

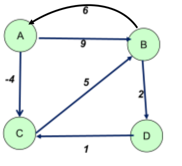
$D_4$ (CB) = min {$D_3$(CB) ; $D_3$(CD) + $D_3$(DB)} -> $D_4$ (CB) = min {5 ; 7 + 6} $\Rightarrow$ $D_2$ (DC) = 5



## Floyd Algoritması Örnek

*Nihai olarak $D_4$ matrisi oluşturulmuş olur.*

| $D_4$ | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 9 | 11 | 11 |
| B | 6 | 0 | 2 | 2 |
| C | 11 | 5 | 0 | 7 |
| D | 12 | 6 | 1 | 0 |




*$D_4$ matrisine bakarak herhangi iki düğüm arasındaki en kısa yolu bulmuş oluruz.*

## Örnek Problemler (Teçhizat yenileme)

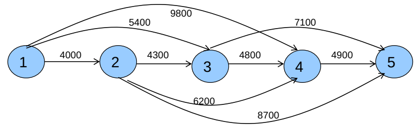
- Bir otomobil kiralama firması 5 yıllık dönem için araç filosu ile ilgili yenileme planı geliştirmektedir. Her yıl başında aracın yenilenmesi veya kullanılması kararı verilmektedir. Bir otomobil en az 1 yıl kullanılmalı ve en fazla 3 yıl sonunda yenilenmelidir. Tabloda ekonomik parametreler görülmektedir.

| Yıl | Kullanım Süresine bağlı yenileme maliyeti | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 2005 | 4000 | 5400 | 9800 |
| 2006 | 4300 | 6200 | 8700 |
| 2007 | 4800 | 7100 | |
| 2008 | 4900 | | |

- Önce problem şebeke halinde gösterilmelidir



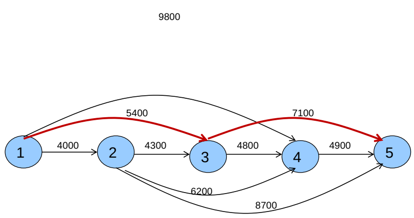





| Adım | A | B Düğ. | B Mes | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4000 | | | | | |
| | | 3 | 5400 | 2 | 4000 | 2 | 4000 | 1-2 |
| | | 4 | 9800 | | | | | |
| 2 | 1 | 3 | 5400 | 3 | 5400 | 3 | 5400 | 1-3 |
| | | 4 | 9800 | | | | | |
| | 2 | 4 | 6200 | 4 | 10200 | | | |
| | | 5 | 8700 | | | | | |
| 3 | 1 | 4 | 9800 | 4 | 9800 | 4 | 9800 | 1-4 |
| | 2 | 4 | 6200 | | | | | |
| | | 5 | 8700 | 4 | 10200 | | | |
| | 3 | 4 | 4800 | 4 | 10200 | | | |
| | | 5 | 7100 | | | | | |
| 4 | 2 | 5 | 8700 | 5 | 12700 | | | |
| | 3 | 5 | 7100 | 5 | 12500 | 5 | 3 | 3-5 |
| | 4 | 5 | 4900 | 5 | 14700 | | | |












## Maksimum Akış Modeli

- Bu tür problemlerde kaynak düğüm ve hedef düğüm birbirine sonsuz kapasiteli bir şebeke ile bağlıdır,
- Akış kaynaktan hedefe gidecek şekildedir,
- Ancak herhangi bir kenar gidiş ve dönüş olarak iki ayrı kapasiteye sahip olabilir,
- Bu tür problemlerde;
  - Kenar üzerindeki akış *(a,b)* olarak gösterilir.
    a…*i'den j'ye* olan akışı, b ise *j'den i'ye* olan akışı belirtir.
  - Herhangi bir yörüngedeki akış, o yörüngedeki kenarların minimum kapasitesine eşittir.
  - Çözüm yöntemi *etiketleme (Ford-Fulkerson Algoritması)* olarak tanımlanır
  - Her adımda şebekedeki düğümler (k,c) olarak etiketlenir,
    - k..akışın geldiği düğümü, c ise akış miktarını tanımlar
- Problemin çözümünde takip edilecek adımlar ise;

1. Başlangıç düğümüne gelen akışı (-,∞) olarak etiketle,
2. Başlangıç düğümünden sonraki her düğümü (kj,cj) olarak etiketle ve her düğümde $c_j = \min\{c_i, a_{ij}\}$ olmalıdır.
3. Bütün düğümler etiketlene kadar 2.adıma devam edilir,
4. Tüm düğümler etiketlendikte n sonra, sondan başa doğru akış miktarı kadar gidiş akıştan çıkarılıp, dönüş akışına eklenir.

$$(a_{ij}, b_{ij})^* = (a_{ij} - c_j, b_{ij} + c_j)$$

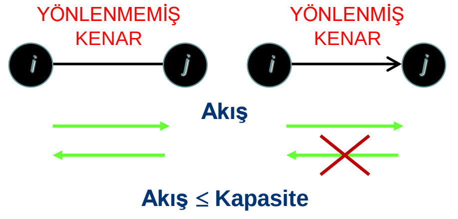
Bu koldaki yeni akışı gösterir

5. Bütün adımlar hedef düğümüne gidecek akış kalmayana kadar tekrarlanır

29

29

## Maksimum Akış Modeli




30

## Maksimum Akış Modeli

### Matematiksel Model

- Her kenardan geçen akış ≤ Kenar'ın Kapasitesi
- Düğüme gelen akış = Düğümden çıkan akış (Kaynak ve bitiş düğümü hariç)
- Kaynağa gelen toplam akış = 0
- Bitiş düğümünden çıkan akış = 0
- Kaynaktan çıkan toplam akış= Bitişe gelen toplam akış

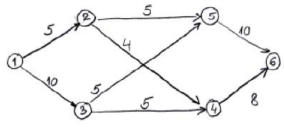
31

## Maks-Akış Problemi: *Tek-Kaynak Tek-Bitiş*

Kaynak düğümlerini bitiş düğümlerine bağlayan yönlendirilmiş kapasiteli Şebeke *(V,E,C)*,
Burada;
V , şebekedeki düğümler
E, şebekedeki yönlenmiş kenarlar (i, j)
C, (i,j) bağlantısının kapasitesi, $c_{ij} \geq 0$

Problem, kaynak düğümden bitiş düğüme olan akış miktarını maksimize etmektir.

Kaynak




Bitiş

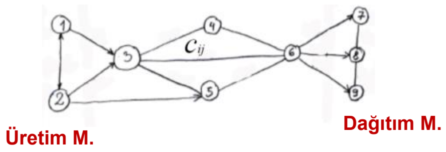
32

## Maks-Akış Problemi: *Çok-Kaynak Çok-Bitiş*

Kaynak düğümünü bitiş düğümüne bağlayan yönlendirilmiş kapasiteli Şebeke (V,E,C),
Burada;

V , şebekedeki düğümler
E, şebekedeki yönlenmiş kenarlar (i, j)
C, (i,j) bağlantısının kapasitesi, $c_{ij} \geq 0$

Problem, kaynak düğümlerinden bitiş düğümlerine olan akış miktarını maksimize etmektir.

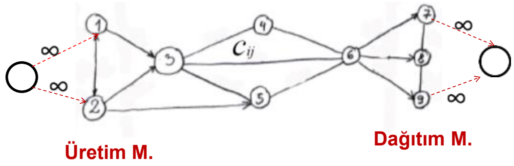



33

## Maks-Akış Problemi: *Çok-Kaynak Çok-Bitiş*

Bu problemler, aşağıdaki şekilde tek-kaynak tek-hedef problemine dönüştürülür;

- Kaynakların sonsuz kapasiteli kenarla bağlandığı bir *kukla (dummy) kaynak* tanımlanır,
- Bitişlerin sonsuz kapasiteli kenarla bağlandığı bir *kukla (dummy) bitiş* tanımlanır,

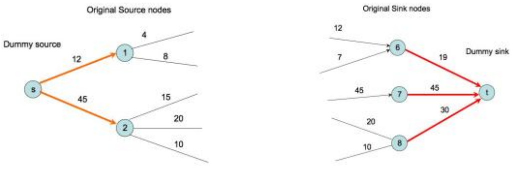



34

## Maks-Akış Problemi: *Çok-Kaynak Çok-Bitiş*

Benzer şekilde,

- Eğer çoklu kaynaklardan birden fazla çıkış varsa, kukla kaynağa çıkışların toplam kapasitesi kadar kapasite tanımlanır,
- Eğer bitişlere birden fazla akış gelen kenar varsa, gelen akışların toplamı kadar, kukla bitişe akış kapasitesi tanımlanır




35

## Maks-Akış Problemi: *Matematiksel Modelleme*

Yönlendirilmiş kapasiteli Şebeke G = (V,E,C)

Her kenar (i,j ) için $X_{ij}$…(i,j) kenarındaki akış miktarını,
Her kenar (i,j) için $C_{ij}$….(i,j) kenarındaki kapasiteyi ifade eder. Her kenardaki $X_{ij}$, $C_{ij}$ tarafından sınırlandırılmıştır. Yani

$c_{ij} \geq X_{ij} \geq 0$

Her düğüm geçiş düğümü olup (gelen akış = çıkan akış) yani

$$\sum x_{li} - \sum x_{ij} = 0 \quad tüm\ i \neq s$$

36

36

## Maks-Akış Problemi: *Matematiksel Modelleme*

Amaç ise;

Kaynak'dan gide akışı maksimize etmek;

$$\sum_{\{j|(s,j)\}} X_{sj}$$

veya

Bitişe giden akışı maksimize etmek

$$\sum_{\{l|(l,t)\}} X_{lt}$$



## Maks-Akış Problemi: *Matematiksel Modelleme*

Model

Maksimum $\sum_{\{j|(s,j)\}} X_{sj}$

Kısıtlar;

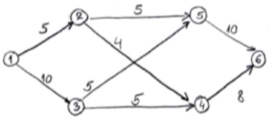
$$\sum x_{li} - \sum x_{ij} = 0$$

$0 \le x_{ij} \le c_{ij}$ tüm (i, j) $\in$ E.



## Maks-Akış Problemi: *Matematiksel Modelleme*

Örnek



Maksimum $x_{12} + x_{13}$

Kısıtlar

$X_{12}-X_{25}-X_{24} = 0$ 2.düğüm dengesi

$X_{13}-X_{35}-X_{34} = 0$ 3.düğüm dengesi

$X_{24} + X_{34}-X_{46} = 0$ 4.düğüm dengesi

$X_{25}+X_{35}-X_{56} = 0$ 5.düğüm dengesi

$0 \le X_{12} \le 5$ $0 \le X_{13} \le 10$ $0 \le X_{25} \le 5$

$0 \le X_{24} \le 4$ $0 \le X_{34} \le 5$ $0 \le X_{35} \le 5$

$0 \le X_{46} \le 8$ $0 \le X_{56} \le 10$

$X_{ij} \ge 0$



- Örnek 6.5.2 sayfa 236

## Minimum Maliyet Kapasiteli Akış Problemi(MMKAP)

- Şebeke modellerinin merkezidir,
  - Maksimum akış problemi gibi kollardaki akış kapasitesini ,
  - En kısa yol gibi, kollardaki minimum maliyeti

Göz önüne alır,

- Birden fazla kaynak ve hedef düğüm olabilir
- Doğrusal programlama problemi olarak modellenebilinir,

41

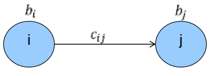
## MMKAP DP Modeli

$x_{ij}$ …….. $i$ düğümünden $j$ düğümüne akış miktarı, KARAR DEĞİŞKENİ

$c_{ij}$ ……… $i$'den $j$'ye olan akışın birim maliyeti

$u_{ij}, l_{ij}$ ……$ij$ kolunun ust ve alt kapasitesi …

$b_i$ …… $i$ düğümünde net akış miktarı
eğer düğüm $i$ kaynak düğüm ise $b_i > 0$
eğer düğüm $i$ hedef düğüm ise $b_i < 0$
eğer düğüm $i$ geçiş düğüm ise $b_i = 0$




$$Min\ Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$Kısıtlar\ \sum_{k=1}^{n} x_{jk} - \sum_{i=1}^{n} x_{ij} = b_i, \qquad j \in N$$

$$l_{ij} \le x_{ij} \le u_{ij}$$

42

42

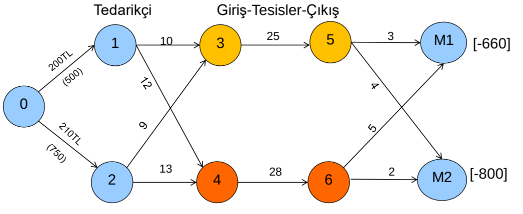
## Örnek (Sayfa 245- 6.6a-4) DKS Kimyasal

- 2 tesiste üretilen kimyasal için 2 müşteri, 660 ve 800 ton/ay talep etmekte
- 1. tesis
  - kapasitesi 400-800 ton/ay, üretim maliyeti 25TL/ton, müşteriye ulaştırma sırayla 3 TL/ton ve 4 TL/ton,
- 2.tesis
  - kapasitesi 450-900 ton/ay, üretim maliyeti 28 TL/ton, müşteriye ulaştırma sırayla 5TL/ton ve 2TL/ton ,
- Hammaddeler için 2 tedarikçi var,
  - 1. tedarikçiden tesislere nakliye sırayla 10 TL/ton ve 12 TL/ton, hammaddenin buraya gelişi 200 TL ve ayda en az 500 ton geliyor
  - 2. tedarikçiden tesislere nakliye sırayla 9 TL/ton ve 13 TL/ton, , hammaddenin buraya gelişi 210 TL ve ayda en az 750 ton geliyor
- 2 ton hammaddeden 1 ton nihai ürün elde edilmekte. Problemi şebeke modeli halinde gösteriniz.

43

43

## Model Çözümü




44

44

*Min Z = $200X_{01}+210X_{02}+10X_{13}+12X_{14}+9X_{23}+13X_{24}+25X_{35}+28X_{46}+3X_{57}+4X_{58}+5X_{67}+2X_{68}$*

*Kısıtlar:*

*Kaynak Kısıtı*

$X_{01} \geq 500$ $\quad X_{02} \geq 750$

*Düğüm denge kısıtı*

$X_{01} - X_{13} - X_{14} = 0$ $\quad X_{02} - X_{23} - X_{24} = 0$

$X_{13} - X_{23} - X_{35} = 0$ $\quad X_{14} - X_{24} - X_{46} = 0$

*Üretim Oranı*

$X_{35} - 2(X_{57} + X_{58}) = 0$ $\quad X_{46} - 2(X_{67} + X_{68}) = 0$

*Talep Kısıtı*

$X_{57} + X_{67} = 660$ $\quad X_{58} + X_{68} = 800$

*Tesis Kapasite Kısıtı*

$400 \leq X_{25} \leq 800$ $\quad 450 \leq X_{46} \leq 900$

$X_{ij} \geq 0$

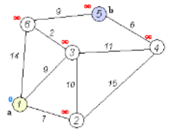
45

45


46

## Dijkstra Algoritması

- Bilgisayar bilimcisi Edsger Dijksta tarafında 1956'da geliştirilip 1959'da yayınlanmıştır.
- 11 Mayıs 1930'da Rotterdam'da doğmuş, 2002'ye kadar yaşamıştır.
- Bir graf'taki en kısa yolu bulmaktadır



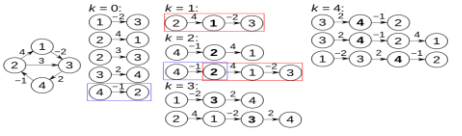
Dijkstra's algorithm. It picks the unvisited vertex with the lowest-distance, calculates the distance through it to each unvisited neighbor, and updates the neighbor's distance if smaller. Mark visited (set to red) when done with neighbors.

47

47

## *Floyd Algoritması*

- **Robert W (Bob) Floyd** (1936 –2001) ünlü bir bilgisayar bilimcidir.
- The Floyd-Warshall algorithm Robert Floyd tarafından 1962'de yayınlandı. Ancak 1959'da Bernard Roy ve 1962'de Stephen Warshall tarafından yayınlanan algoritmaların aynısıdır.
- Ağırlıklandırılmış graf'taki en kısa yolu bulmak için etkilidir.




48

48