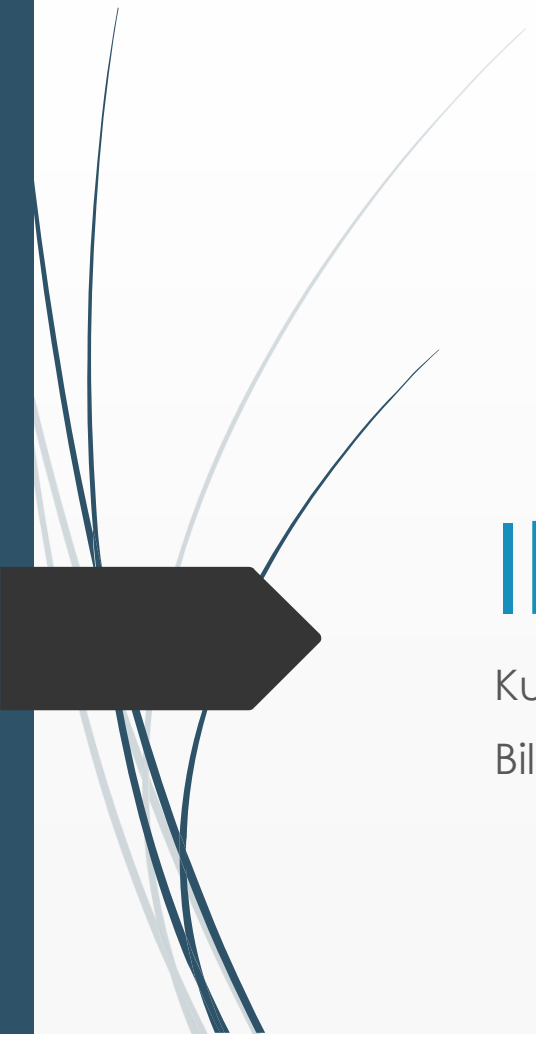


Ikerketa Operatiboa

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritza

Bilboko Ingeniaritza Eskola

Programazio linealeko problema bereziak- II

5.5 Distantzia minimoko problema

- 5.5.1 Dijkstra algoritmoa
- 5.5.2 Floyd-en algoritmoa

Programazio linealeko problema bereziak - II

5.5 Distantzia minimoko problema

Distantzia minimoko problema **sare** baten moduan adieraz daitezke. Kasu hauetan **o** hasierako nodoaren eta **f** azken nodoaren arteko distantzia minimoa kalkula daiteke, i. eta j. nodoen arteko distantzia zehazten duten kostu-matrizeko elementuak c_{ij} izanik. Sareak baliorik ez balu, arku guztientzat $c_{ij} = 1$ da.

Problema hau PLO motako problema bat da. Hala ere, problema hau Simplexen bidez osotasun baldintzak ezarri gabe ebatz daiteke, sareko problemei dagozkien murrizketa matrizeak guztiz modulu bakarrekoak baitira (matrize bat modulu bakarrekoa da, baldin bere azpimatrize erregular eta karratu guztien determinantearen balioa +1 edo −1 bada). Orduan, erlaxatutako problemaren (osotasun baldintzarik gabeko problemaren) soluzioa ere osoa da

Programazio linealeko problema bereziak - II

Problemaren formulazioa honako hau da:

$$\text{Min } Z = \sum_i \sum_j c_{ij} x_{ij}$$

$$\sum_i x_{ij} = \sum_k x_{jk} \quad \forall \text{j} \neq \text{hasierako nodoa, azken nodoa}$$

$$\sum_i x_{if} = 1 \qquad \text{azken nodoa}$$

$$\sum_k x_{ok} = 1 \qquad \text{hasierako nodoa}$$

Erabaki-aldagaiak ondorengoak izanik:

$$x_{ij} = \begin{cases} 1 & \text{i. eta j. nodoak elkartzen dituen arkua erabili bada} \\ 0 & \text{beste kasuetan} \end{cases}$$

Programazio linealeko problema bereziak - II

5.5.1 Dijkstra algoritmoa:

Helburua:

Izan bedi G=(V,A,ϕ) grafo haztatua (zuzendua edo ez), kiribil gabea eta n erpin dituena.

Dijkstra algoritmoaren bidez hasierako erpin finko batetik G-ren beste erpin guztietara dagoen distantzia minimoa kalkulatzen da. Baita erpin horretatik G-ren beste erpin guztietara dagoen ibilbide minimoa duen bidea ere.

Programazio linealeko problema bereziak- II

Izan bedi G=(V,A,ϕ) grafo haztatua (zuzendua edo ez), kiribil gabea eta n erpin dituena, hau da V=$\{v_0, v_1, \dots, v_n\}$.

5.5.1 Dijkstra algoritmoa:

1. pausua: Zenbatzaile bat hasieratu $i = 0$, eta S_0 lehenengo erpina duen multzoa definitu: $S_0 = \{v_0\}$.

   v_0 erpinari (E(v_0)=0, -) etiketa jarri. E(v_0) = 0 adierazpenak v_0-tik v_0-ra joateko distantzia 0 dela adierazten duelarik.

   Beste erpin guztietan $v_k \neq v_0$ (∞, -) etiketa jarri

Programazio linealeko problema bereziak- II

2. pausua: $\forall v_k \in V - S_i$ erpinei etiketak eguneratu:

$$(E(v_k), x)$$

non $E(v_k)=\min\{E(v_k), E(v_i) + p(v_i, v_k)\}$

$p(v_i, v_k) =$ v_i eta v_k erpinak lotzen dituen arkuaren pisua.

v_i eta v_k erpinak lortzen dituen arkurik ez badago: $p(v_i, v_k) = \infty$

x = $E(v_k)$ minimoa lortzeko erabiltzen den G-ren erpina da.

Programazio linealeko problema bereziak- II

3. pausua: E(v_k) minimoa duen v_k erpina aukeratu.

E(v_k) etiketa v_0-tik v_k-ra dagoen ibilbide minimoaren luzera da. v_k erpina aukeratutako erpinen multzora gehitu:

$S_{i+1} = S_i \cup \{v_k\}$.

Zenbatzaileari bat gehitu: $i++$

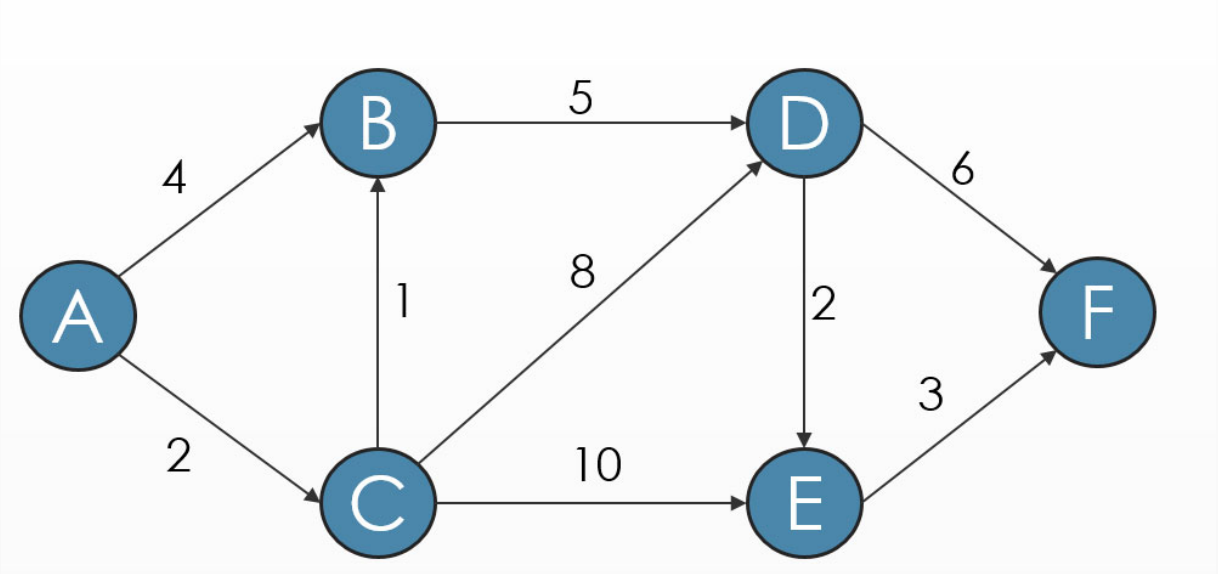
$i < n$ bada 2. pausura joan.

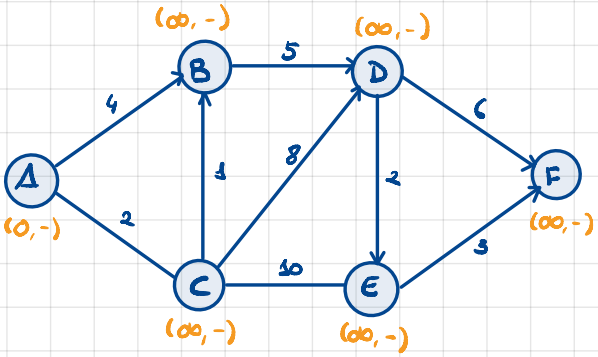
Bestela, prozedura bukatu egin da.

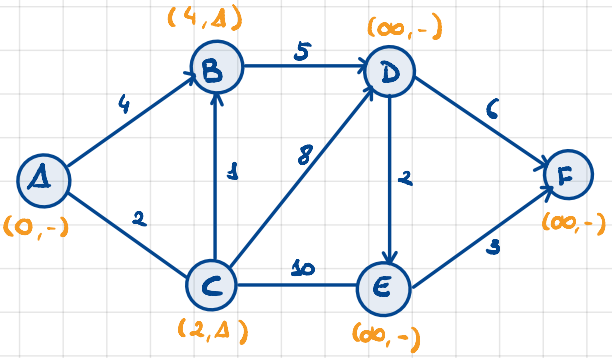
Programazio linealeko problema bereziak- II

Adibidea

Hurrengo sarean dauden A eta F nodoen arteko distantzia minimoa lortu:





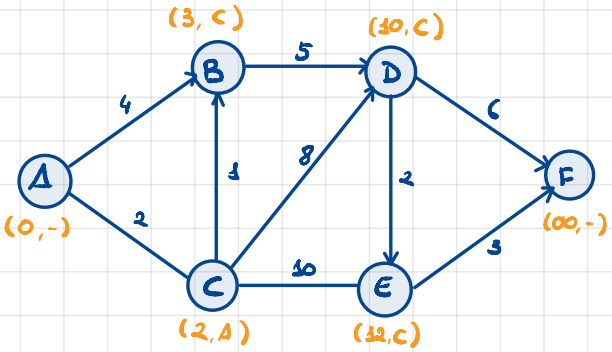


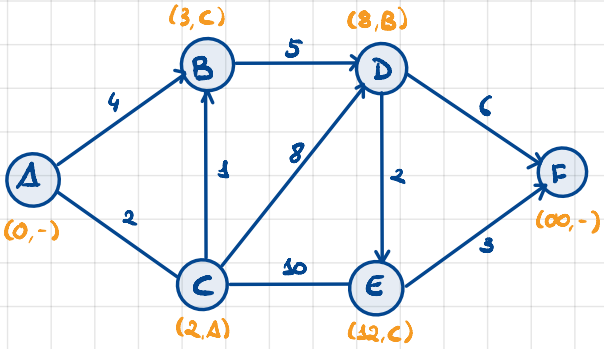
It. i=0

$S_0 = \{A\}$
$B = \min\{\infty, 0+4\} = 4$
$C = \min\{\infty, 0+2\} = 2$
$D = \min\{\infty, 0+\infty\} = \infty$
$E: \infty$
$F: \infty$
$V_1 = C$

It. 2

$S_2 = \{A, C\}$
$B = \min\{4, 2+1\} = 3$
$D = \min\{\infty, 2+8\} = 10$
$E = \min\{\infty, 2+10\} = 12$
$F: \infty$
$V_2 = B$



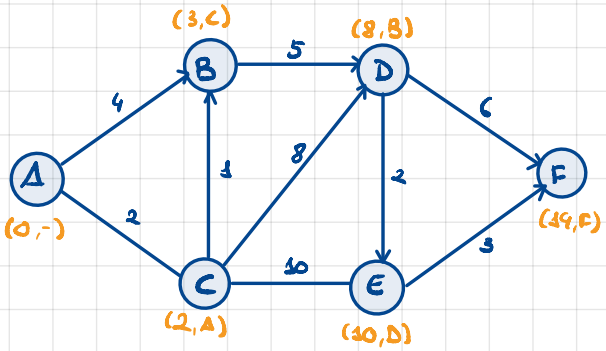


It. 3

$S_2 = \{A, C, B\}$
$D = \min\{10, 3+5\} = 8$
$E = \min\{12, \infty\} = 12$
$F = \min\{\infty, \infty\} = \infty$
$V_3 = D$

It. 4

$S_3 = \{A, C, B, D\}$
$E = \min\{12, 8+2\} = 10$
$F = \min\{\infty, 8+6\} = 14$
$V_4 = E$



$S_4 = \{A, C, B, D, E\}$
$F = \min\{14, 10+3\} = 13$

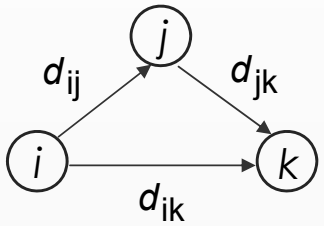
$d(A,B) = 3$	A-C-B
$d(A,C) = 2$	A-C
$d(A,D) = 8$	A-C-B-D
$d(A,E) = 10$	A-C-B-D-E
$d(A,F) = 13$	A-C-B-D-E-F

Programazio linealeko problema bereziak- II

5.5.2 Floyd-en algoritmoa

Algoritmo honen ezaugarri nagusia ondorengoa da:

Izan bitez i, j *eta* k nodoak, beraien arteko distantziak d_{ij}, d_{ik} eta d_{jk} izanik



i nodotik k nodora j nodoaren bidez joateko bidea motzagoa da $d_{ij} + d_{jk} < d_{ik}$ betetzen bada. Kasu honetan $i \rightarrow k$ egin beharrean $i \rightarrow j \rightarrow k$ egingo da.

Programazio linealeko problema bereziak- II

5.5.2 Floyd-en algoritmoa

Algoritmo hau Dijkstra algoritmoa baino orokorragoa da, nodo-sare bateko edozein bi nodoen arteko biderik motzena zehazten baitu.

Lehenengo eta behin, algoritmoa aplikatu baino lehen grafoko nodoak1-etik n-ra zerrendatzen dira.

Ondoren, D_{ij} $\forall i,j: i \neq j$ distantziak definitzen dira:

$$D_{ij} = \begin{cases} d_{ij} & i-tik\ j-ra\ doan\ arkuaren\ kostua \\ M\ (+\infty) & i-tik\ j-ra\ doan\ arkua\ existitzen\ ez\ bada \end{cases}$$

Programazio linealeko problema bereziak- II

Balio hauek erabiliz $\mathrm{D} = (D_{ij})_{\substack{i=1,\dots,n \\ j=1,\dots,n}}$ distantzien matrizea eraikitzen da.

Bestalde, n ordenako $S = (S_{ij})_{\substack{i=1,\dots,n \\ j=1,\dots,n}}$ beste matrize karratu bat definitzen da, ibilbide-matrizea deritzona.

Matrize horretako elementuak i erpinetik j erpinera doazen bideen azken-aurreko nodoak izango dira.

Hasiera batean $S_{ij} = i \ \forall i, j\colon\ i \neq j$

Programazio linealeko problema bereziak- II

Floyd algoritmoa:

1. pausua: Hasierako D (distantzien matrizea) eta S (ibilbide-matrizea sortu)

	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}	⋯	-

	1	2	⋯	j	⋯	n
1	-	2	⋯	j	⋯	n
2	1	-	⋯	j	⋯	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	1	2	⋯	j		n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	2	⋯	j	⋯	-

k=1 egin.

Programazio linealeko problema bereziak- II

2. pausua: k errenkada, errenkada pibote bezala, eta k zutabea, zutabe pibote bezala aukeratu eta ondorengoa egin:

$$D_{ik} + D_{kj} < D_{ij} \Rightarrow \begin{cases} S_{ij} = S_{kj} \\ D_{ij} = D_{ik} + D_{kj} \end{cases} \forall i, j: i \neq j \wedge i \neq k \wedge j \neq k$$

Aurreko baldintza betetzen ez bada, balioak dauden bezala utzi.

3. pausua: $k < n$ bada, k++ egin eta 2. pausura joan. Kontrako kasuan: Bukatu.

Programazio linealeko problema bereziak- II

Adibidea

Hurrengo sarea kontuan hartuta, aurkitu edozein bi nodoen arteko biderik motzena.