Ikerketa Operatiboa

Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritza

Bilboko Ingeniaritza Eskola

Programazio linealeko problema bereziak

5.1 Garraio-problema

5.2 Esleipen problema

5.3 Fluxu maximoko problema

5.4 CPM (Critical Path Method)

Programazio linealeko problema bereziak

5.1 Garraio-problema

Ekoizle batek produktu bat A_i $(1 \leq i \leq m)$ m biltegitan du, a_i i. biltegian dagoen produktu kantitatea izanik. B_j $(1 \leq j \leq n)$ helmuga−puntura b_j __unitate eramaten dira. Bestalde A_i biltegitik B_j helmuga-puntura unitate bakoitza garraiatzean sortzen den kostua c_{ij} da. Problema honen helburua i. biltegitik j. helmuga-puntura garraiatu beharreko x_{ij} unitateen kopurua zehaztea da, garraio-kostua minimoa izan dadin.

Jatorrietan dauden artikuluen kopurua eta helmugetako eskaera bat datozela suposatzen da, hau da, ***problema orekatua*** dela suposatzen da.

Programazio linealeko problema bereziak

Problemaren formulazioa honako hau da:

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

$$non \quad \sum_{j=1}^{n} x_{ij} = a_i \quad i = 1, 2, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, 2, \ldots, n$$

$$x_{ij} \geq 0 \quad i = 1, 2, \ldots, m \,;\; j = 1, 2, \ldots, n$$

Programazio linealeko problema bereziak

Lehenengo m murrizketak $A_i\ (1 \leq i \leq m)$ m biltegietan auden artikulu kopuruekin erlazionatuta daude:

$$\sum_{j=1}^{n} x_{ij} = a_i \quad i = 1, 2, \ldots, m$$

Hurrengo n murrizketak berriz, $B_j\ (1 \leq j \leq n)$ helmuga−puntuetako eskaerekin erlazionatuta daude.

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, 2, \ldots, n$$

Problema orekatua ez bada, problema orekatu arte gezurretako jatorriak edo gezurretako helmugak sortu behar dira.

Programazio linealeko problema bereziak

5.1.1 Matrize-forma:

Garraio-problemaren datuak taula moduan jasotzen dira. Taula horri **garraio-problemarako matrize-forma edo garraio kostuen** taula deitzen zaio. Bertan iturburuak eskaintzekin, helburuak eskariekin eta unitateko garraio-kostuak agertzen dira.

	B_1	B_2	…	B_n	**Eskaintza**
A_1	c_{11}	c_{12}	…	c_{1n}	a_1
A_2	c_{21}	c_{22}	…	c_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
A_m	c_{m1}	c_{m2}	…	c_{mn}	a_m
Eskaria	b_1	b_2	…	b_n	

Programazio linealeko problema bereziak

5.1.2 Teoremak:

Teorema 1:

$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j < \infty$ betetzen bada, hau da, problema orekatua bada, orduan problemak soluzio bideragarri bornatua du.

Teorema 2:

Problema orekatu baten koefizienteen matrizearen heina $m+n-1$ da.

Problema orekatuak ondorioz, $m+n-1$ ekuazio linealki independente ditu, hortaz hasierako soluzioak $m+n-1$ oinarrizko aldagai izango ditu.

Programazio linealeko problema bereziak

Teorema 3:

Problema orekatuaren a_i, b_j koefiziente guztiak ez-negatiboak badira, oinarrizko soluzio bideragarri optimo oso bat existitzen da.

Problema honen egitura bereziari esker metodo desberdinak aplikatuz, hasierako oinarrizko soluzio bideragarria kalkula dezakegu. Bi metodo aztertuko ditugu:

- Ipar-mendebaldeko ertzaren metodoa
- Vogel-en metodoa

Programazio linealeko problema bereziak

Orokorrean, Vogel-en metodoak oinarrizko soluzio bideragarri hobea ematen du. Ipar-mendebaldeko metodoak berriz, kalkulu gutxiago behar izaten ditu.

Programazio linealeko problema bereziak

5.1.3 Hasierako oinarrizko soluzio bideragarria:

Garraio-problemarako soluzio bat kalkulatzeko, garraio-kostuen taularen dimentsio berberak dituen beste taula bat eraikikko dugu. Taula horrek garraio-fluxuen taula izena du eta bertan kokatuko ditugu garraio-fluxuak, hau da, biltegia bakoitzetik helmuga-puntu bakoitzera garraiatuko den produktu unitate kopurua (hasierako soluzio bideragarria)

Programazio linealeko problema bereziak

Ipar-mendebaldeko ertzaren metodoa:

Garraio-problema orekatua izanik, ondoko urratsei jarraituz hasierako oinarrizko soluzio bideragarri bat lortzen da:

1. pausua: Garraio-fluxuen taulan ipar-mendebaldeko (goiko-ezkerraldeko) ertza aukeratu.

Programazio linealeko problema bereziak

2. pausua: Aukeratutako posizioan x_{ij} aldagaiari ahal den fluxurik handiena esleitu, $x_{ij} = min\{a_i, b_j\}$. Ondoren, a_i eskaintza eta b_j eskaria eguneratu:

- Minimoa a_i bada, A_i biltegian eskaintza zero bihurtuko da. Ondorengo kalkuluetarako taulako i. errenkada ezabatu behar da b_j eskaria eguneratu: b_j-a_i
- Minimoa b_j bada, B_j helmuga-puntuaren eskaria zerbitzatua geratzen da, eskaria zero bihurtuko da eta taulako j. zutabea ezabatuko da, aurrerantzean egingo diren kalkuluetan kontuan ez izateko. a_i eskaintza eguneratu: a_i-b_j

Programazio linealeko problema bereziak

$a_i = b_j$ bada, biltegiaren eskaitza eta helmuga-puntuaren eskaria aldi berean zero bihurtuko da. Aurrerantzean egingo diren kalkuluetarako i. errenkada eta j. zutabea ezatuko dira.

Programazio linealeko problema bereziak

3. pausua: Bi kasu gerta daitezke:

- Ezabatu izan ez den errenkada edo zutabe bakarra baldin badago taulan, geratzen diren produktuen eskaintza eta eskariak ezabatu gabeko posizioetara esleitzen dira. Amaitu.

- Bestela, zutabe bat ezabatu bada eskumako gelaxkara mugitu eta errenkada bat ezabatu bada behera mugitu eta 1. pausura joan.

Programazio linealeko problema bereziak

Adibidea:

Izan bedi ondorengo garraio-kostuen taula duen problema:

		Helmuga				
		1	**2**	**3**	**4**	**Eskaintza**
Jatorria	**A**	1	2	3	3	5
	B	4	3	2	0	7
	C	0	1	2	1	12
Eskaria		3	5	8	8	24

Programazio linealeko problema bereziak

1. iterazioa:

1. pausua: Ipar-mendebaldeko ertza aukeratu (1,1)

2. pausua: Ahal den fluxurik handiena esleitu:

$$x_{11} = min\{3,5\} = 3$$

		Helmuga				
		1	**2**	**3**	**4**	**Eskaintza**
Jatorria	**A**	1	2	3	3	$\boxed{5}$
	B	4	3	2	0	7
	C	0	1	2	1	12
Eskaria		$\boxed{3}$	5	8	8	24

Programazio linealeko problema bereziak

A iturriaren eskaintza eguneratu: $a_1 = 5 - 3 = 2$

1. helmugako eskaria zero bihurtu ⇒ 1. zutabea ezabatu

	1	2	3	4	Eskaintza
A	3				2
B					7
C					12
	3	5	8	8	24

Programazio linealeko problema bereziak

3. pausua: Zutabe bat ezabatu denez ⇒ eskumako gelaxkara mugitu ⇒(1,2)

<u>2. iterazioa:</u>

1. pausua: Taulan ezabatu gabeko ipar-mendebaldeko ertza aukeratu (1,2)

2. pausua: Ahal den fluxurik handiena esleitu:

$$x_{12} = min\{2,5\} = 2$$

	1	2	3	4	Eskaintza
A	3				2
B					7
C					12
	3	5	8	8	24

Programazio linealeko problema bereziak

A iturriaren eskaintza zero bihurtu ⇒ 1. errenkada ezabatu

2. helmugako eskaria eguneratu ⇒ $b_2 = 5 - 2 = 3$

	1	2	3	4	Eskaintza
A	3	2			2
B					7
C					12
	3	3	8	8	24

3. pausua: Errenkada bat ezabatu denez ⇒ beheko gelaxkara mugitu ⇒(2,2)

Programazio linealeko problema bereziak

3. iterazioa:

1. pausua: Taulan ezabatu gabeko ipar-mendebaldeko ertza aukeratu (2,2)

2. pausua: Ahal den fluxurik handiena esleitu:

$$x_{22} = min\{3,7\} = 3$$

	1	2	3	4	Eskaintza
A	3	2			2
B					7
C					12
	3	3	8	8	24

Programazio linealeko problema bereziak

B iturriaren eskaintza eguneratu: $a_2 = 7 - 3 = 4$

2. helmugako eskaria zero bihurtu ⇒ 2. zutabea ezabatu

	1	2	3	4	Eskaintza
A	3	2			2
B		**3**			**4**
C					12
	3	3	8	8	24

3. pausua: Zutabe bat ezabatu denez ⇒ eskumako gelaxkara mugitu ⇒(2,3)

Programazio linealeko problema bereziak

4. iterazioa:

1. pausua: Taulan ezabatu gabeko ipar-mendebaldeko ertza aukeratu (2,3)

2. pausua: Ahal den fluxurik handiena esleitu:

$$x_{23} = min\{4,8\} = 4$$

	1	2	3	4	Eskaintza
A	3	2			2
B		3			4
C					12
	3	3	8	8	24

Programazio linealeko problema bereziak

B iturriaren eskaintza zero bihurtu ⇒ 2. errenkada ezabatu

3. helmugako eskaria eguneratu ⇒ $b_3 = 8 - 4 = 4$

	1	2	3	4	Eskaintza
A	3	2			2
B		3	4		4
C					12
	3	3	4	8	24

3. pausua: Ezabatua izan ez den errenkada bakarra dago ⇒ geratzen diren produktuen eskaintzak eta eskariak ezabatu gabeko posizioetan esleitu

Programazio linealeko problema bereziak

3. pausua: Ezabatua izan ez den errenkada bakarra dago ⇒ geratzen diren produktuen eskaintzak eta eskariak ezabatu gabeko posizioetan esleitu

	1	2	3	4	Eskaintza
A	3	2			2
B		3	4		4
C			**4**	**8**	12
	3	3	4	8	24

Hasierako oinarrizko soluzio bideragarria:

$$x_{11} = 3; x_{12} = 2; x_{22} = 3; \; x_{23} = 4; x_{33} = 4; \; x_{34} = 8$$

Programazio linealeko problema bereziak

		Helmuga				
		1	**2**	**3**	**4**	**Eskaintza**
	A	1	2	3	3	5
Jatorria	**B**	4	3	2	0	7
	C	0	1	2	1	12
Eskaria		3	5	8	8	24

Garraio-kostua:

$$1 \times 3 + 2 \times 2 + 3 \times 3 + 2 \times 4 + 4 \times 2 + 1 \times 8 = 40$$

Programazio linealeko problema bereziak

Vogel-en metodoa:

Ipar-mendebaldeko ertzaren metodoaren eta Vogel-en metodoa garraio-fluxua kokatzeko posizioa aukeratzeko moduan desberdintzen dira, hau da, algoritmoaren 1. urratsean. Vogel-en metodoak posizioa aukeratzeko errenkadetako eta zutabetako diferentziak kalkulatzen ditu:

$ED_i =$ i. errenkadako bi kosturik txikienen arteko diferentzia balio absolutuan $i = 1, \dots, m$

$ZD_j =$ j. zutabeko bi kosturik txikienen arteko diferentzia balio absolutuan j $= 1, \dots, n$

Programazio linealeko problema bereziak

1. pausua: Garraio-kostuen taulan ED_i eta ZD_j diferentziak kalkulatu. Diferentziarik handieneko errenkada edo zutabea aukeratu, eta bertan c_{ij} kosturik txikieneko (i, j) posizioa.
2. pausua: Ipar-mendebaldeko ertzaren metodoaren 2. pausua.
3. pausua: Bi kasu gerta daitezke:
   - Ezabatua izan ez den errenkada edo zutabe bakarra baldin badago taulan, geratzen diren produktuen eskaintza eta eskariak ezabatu gabeko posizioetara esleitu eta amaitu.
   - Bestela, 1. pausura joan

Programazio linealeko problema bereziak

Adibidea:

Izan bedi ondorengo garraio-kostuen taula duen problema:

		Helmuga				
		1	**2**	**3**	**4**	**Eskaintza**
Jatorria	**A**	1	2	3	3	5
	B	4	3	2	0	7
	C	0	1	2	1	12
Eskaria		3	5	8	8	24

Programazio linealeko problema bereziak

1. iterazioa:

1. pausua: ED_i eta ZD_j kalkulatu.

Diferentziarik handieneko errenkada edo zutabea aukeratu ⇒ B errenkada

B errenkadan c_{ij} kosturik txikiena aukeratu ⇒(2,4)

		Garraio-kostuen taula					ED_i
		1	**2**	**3**	**4**	**Eskaintza**	
Jatorria	**A**	1	2	3	3	5	1
	B	4	3	2	$\boxed{0}$	7	2
	C	0	1	2	1	12	1
Eskaria		3	5	8	8		
ZD_j		1	1	0	1		

Programazio linealeko problema bereziak

2. pausua:

$x_{24} = min\{7{,}8\} = 7$

$b_4 = 1$ eta $a_2 = 0 \Rightarrow$ 2. errenkada ezabatu

	Garraio-fluxuen taula				
	1	2	3	4	
A					5
B				7	
C					12
	3	5	8	1	

3. pausua: 1. pausura joan

x_{ij}

Programazio linealeko problema bereziak

2. iterazioa:

1. pausua: Ezabatu gabeko kostuak kontuan hartuz ED_i eta ZD_j kalkulatu. Diferentziarik handieneko errenkada edo zutabea aukeratu ⇒ 4. zutabea

B errenkadan c_{ij} kosturik txikiena aukeratu ⇒(3,4)

		Helmuga					ED_i
Jatorria		**1**	**2**	**3**	**4**	**Eskaintza**	
	A	1	2	3	3	5	1
	B	4	3	2	0	0	
	C	0	1	2	1	12	1
Eskaria		3	5	8	1		
ZD_j		1	1	1	2		

Programazio linealeko problema bereziak

2. pausua:

$x_{34} = min\{1,12\} = 1$

$b_4 = 0$ eta $a_3 = 12 - 1 = 11 \Rightarrow$ 4. zutabea ezabatu

	Garraio-fluxuen taula				
	1	2	3	4	
A					5
B				7	
C				**1**	**11**
	3	5	8		

3. pausua: 1. pausura joan

Programazio linealeko problema bereziak

3. iterazioa:

1. pausua: Ezabatu gabeko kostuak kontuan hartuz ED_i eta ZD_j kalkulatu. Diferentziarik guztiak berdinak direnez⇒ bat zoriz aukeratu ⇒ C errenkada

C errenkadan c_{ij} kosturik txikiena aukeratu ⇒(3,1)

		Helmuga					ED_i
Jatorria		1	2	3	4	Eskaintza	
	A	1	2	3	3	5	1
	B	4	3	2	0	7	
	C	$\boxed{0}$	1	2	1	11	1
Eskaria		3	5	8	0		
ZD_j		1	1	1			

Programazio linealeko problema bereziak

2. pausua:

$x_{31} = min\{3,11\} = 3$

$b_1 = 0$ eta $a_3 = 11 - 3 = 8 \Rightarrow$ 1. zutabea ezabatu

	Garraio-fluxuen taula				
	1	2	3	4	
A					5
B				7	
C	3			1	8
		5	8		

3. pausua: 1. pausura joan

Programazio linealeko problema bereziak

4. iterazioa:

1. pausua: Ezabatu gabeko kostuak kontuan hartuz ED_i eta ZD_j kalkulatu. Diferentziarik guztiak berdinak direnez⇒ bat zoriz aukeratu ⇒ C errenkada

C errenkadan c_{ij} kosturik txikiena aukeratu ⇒(3,2)

		Helmuga					ED_i
		1	**2**	**3**	**4**	**Eskaintza**	
Jatorria	**A**	1	2	3	3	5	1
	B	4	3	2	0	7	
	C	0	$\boxed{1}$	2	1	**8**	1
Eskaria		3	5	8	**0**		
ZD_j		**0**	1	1			

Programazio linealeko problema bereziak

2. pausua:

$x_{32} = min\{8,5\} = 5$

$b_2 = 0$ eta $a_3 = 8 - 5 = 3 \Rightarrow$ 2. zutabea ezabatu

	Garraio-fluxuen taula				
	1	2	3	4	
A					5
B				7	
C	3	**5**		1	**3**
			8		

3. pausua: Ezabatua izan ez den zutabe bakarra dago⇒ geratzen diren eskaintza eta eskariak esleitu

Programazio linealeko problema bereziak

3. pausua: Ezabatua izan ez den zutabe bakarra dago⇒ geratzen diren eskaintza eta eskariak esleitu

	Garraio-fluxuen taula				
	1	2	3	4	
A			**5**		5
B				7	
C	3	5	**3**	1	3
			8		

Hasierako oinarrizko soluzio bideragarria:

$$x_{13} = 5; x_{24} = 7; x_{31} = 3; x_{32} = 5; x_{33} = 3; x_{34} = 1$$

Programazio linealeko problema bereziak

		Helmuga				
		1	**2**	**3**	**4**	**Eskaintza**
	A	1	2	3	3	5
Jatorria	**B**	4	3	2	0	7
	C	0	1	2	1	12
Eskaria		3	5	8	8	24

Garraio-kostua:

$$3 \times 5 + 0 \times 7 + 0 \times 3 + 1 \times 5 + 2 \times 3 + 1 \times 1 = 27$$

Programazio linealeko problema bereziak

Garraio algoritmoa:

Hasierako oinarrizko soluzio bideragarria lortu ondoren, oinarrizko soluzio bideragarri hori hobetu egin behar da. Horretarako Simplex metodoaren egokitzapen bat erabiltzen da:

1. pausua: Simplex metodoaren optimaltasun baldintzak betetzen diren aztertu. Betetzen badira gelditu. Bestela oinarrira sartzen den aldagaia aukeratu.
2. pausua: Simplex metodoaren bideragarritasun baldintza erabiliz oinarritik irtetzen den aldagaia aztertu. Oinarria aldatu eta 1. pausura joan.

Programazio linealeko problema bereziak

Oinarrira sartuko den aldagaiaren aukeraketa:

Oinarrizko soluzio bideragarri bat hobetzeko, garraio-problemari dagokion eredu duala erabiltzen da.

Izan bedi ondorengo garraio-problema orekatua:

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

$$non \quad \sum_{j=1}^{n} x_{ij} = a_i \quad i = 1, 2, \ldots, m \longrightarrow u_i$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, 2, \ldots, n \longrightarrow v_j$$

$$x_{ij} \geq 0 \quad i = 1, 2, \ldots, m \,;\, j = 1, 2, \ldots, n$$

Programazio linealeko problema bereziak

Problema honekin erlazionatutako aldagai dualak u_1, $u_2,\ldots u_m$ eta v_1, $v_2,\ldots v_n$ izendatzen baditugu, dagokioen eredu duala honako hau da:

$$\text{Max} \sum_{i=1}^{m} a_i \cdot u_i + \sum_{j=1}^{n} b_j \cdot v_j$$

$$non \quad u_i + v_j \le c_{ij} \quad i = 1,2,\ldots,m\,;\ j = 1,2,\ldots,n$$

$$u_i,\ v_j \ \text{ez-murriztuak} \ \ i = 1,2,\ldots,m\,;\ j = 1,2,\ldots,n$$

Programazio linealeko problema bereziak

Oinarrizko soluzio bideragarri bat hobe daitekeen edo ez erabakitzeko, oinarrizkoak ez diren aldagaiei dagozkien kostu-murriztuak $W_{ij} = z_{ij} - c_{ij}$ kalkulatu behar dira.

$$W_{ij} = z_{ij} - c_{ij} = c_B^T \cdot B^{-1} \cdot a_{ij} - c_{ij}$$

Aldagai dualen bektorea $c_B^T \cdot B^{-1}$ dela kontuan hartuz:

$$c_B^T \cdot B^{-1} = (u_1, u_2, \ldots u_m, v_1, v_2, \ldots v_n)$$

Programazio linealeko problema bereziak

Aurreko berdintza ondorengo eran lortzen da:

Oharra:

Problema primala eta Problema duala

$$\begin{aligned} \min \quad & C^T X \\ \text{non} \quad & AX \geq b \\ & X \geq 0 \end{aligned}$$

$$\begin{aligned} \max \quad & b^T U \\ \text{non} \quad & A^T U \leq C^T \\ & U \geq 0 \end{aligned}$$

Programazio linealeko problema bereziak

Problema primalak soluzio finitua du $\Leftrightarrow$ problema dualak soluzio finitua du (helburu-funtzioaren balioa berdina da)

$$Z_{opt} = c_B^T \cdot B^{-1} \cdot b = b^T \cdot U \Rightarrow$$

$b^T \cdot U = U^T \cdot b$ dela kontuan izanik:

$$Z_{opt} = \boxed{c_B^T \cdot B^{-1}} \cdot b = \boxed{U^T} \cdot b \Rightarrow$$

Ondorioz, $c_B^T \cdot B^{-1}$ aldagai dualen bektorea da, hau da:

$$c_B^T \cdot B^{-1} = (u_1, u_2, \ldots u_m, v_1, v_2, \ldots v_n)$$

Programazio linealeko problema bereziak

Orduan,

$$W_{ij} = z_{ij} - c_{ij} = (u_1, u_2, \ldots u_m, v_1, v_2, \ldots v_n) \cdot a_{ij} - c_{ij}$$

a_{ij} bektoreak dituen 1eko bakarrak i eta m+j posizioetan daude, bektorearen gaineko osagaiak 0 izanik (ikusi 46. gardenkiko oharra)⇒

$$W_{ij} = z_{ij} - c_{ij} = u_i + v_j - c_{ij}$$

Oinarrian dauden aldagaien kostu murriztuak 0 direnez, oinarrian dauden m+n-1 aldagaien kasuan:

$$u_i + v_j - c_{ij} = 0$$

Programazio linealeko problema bereziak

Oharra:

$$W_{ij} = z_{ij} - c_{ij} = (u_1, u_2, \ldots u_m, v_1, v_2, \ldots v_n) \cdot a_{ij} - c_{ij}$$

a_{ij} bektoreak dituen 1eko bakarrak i eta m+j posizioetan daude, bektorearen gaineko osagaiak 0 izanik

Adibidea: Garraio-problema

	1 helmuga-puntua	2 helmuga-puntua
A	x_{11}	x_{12}
B	x_{21}	x_{22}

Programazio linealeko problema bereziak

Problema primala:

$$\text{Min} \quad Z = \sum_{i=1}^{2}\sum_{j=1}^{2} c_{ij}x_{ij}$$

$$x_{11} + x_{12} = a_1$$

$$x_{21} + x_{22} = a_2$$

$$x_{11} + x_{21} = b_1$$

$$x_{12} + x_{22} = b_2$$

$$x_{ij} \geq 0 \quad i = 1,2\,;\; j = 1,2$$

Forma matriziala

$$\text{Min} \quad Z = \sum_{i=1}^{2}\sum_{j=1}^{2} c_{ij}x_{ij}$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{11} \\ x_{12} \\ x_{21} \\ x_{22} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix}$$

x_{11} 1 eta m + 1 errenkadan

Programazio linealeko problema bereziak

- Modu honetako $m + n - 1$ ekuazioak sortzen duten sistema ebatzi behar da. Sistemak m+n aldagai dituenez, ekuazio-sistema hori aldagairen bati balio bat emanez ebatzi behar da (soluzio zehatz bat lortzeko)
- Sistema ebatzi ondoren, hau da, aldagai dualen balioak kalkulatuak izan direnean, balio adierazle guztiak kalkulatzen dira.

Helburua minimizatzea denez, bi kasu gerta daitezke:

- $W_{ij} = z_{ij} - c_{ij} \leq 0 \ \ \forall i = 1, \dots, m, j = 1, \dots, n \ \Rightarrow$ Soluzioa optimoa da
- $\exists W_{ij} = z_{ij} - c_{ij} \geq 0 \ \Rightarrow W_{ij} = z_{ij} - c_{ij}$ positiboen artean maximoa duen aldagaia oinarrira sartuko da

Programazio linealeko problema bereziak

Oinarritik irtengo den aldagaiaren aukeraketa:

Oinarritik irtengo den aldagaia zehazteko oinarrizkoak diren aldagaien eta oinarrira sartuko den aldagaien artean ziklo bat sortzen da.

Zikloa zein fluxuk osatzen duten zehaztu denean, oinarrira sartzea erabaki den aldagaiari fluxu positibo bat esleitu behar zaio, gainera, zikloa osatzen duten aldagaien artetik batek zero balioa hartu beharko du, oinarritik irtengo denak hain zuzen ere.

Programazio linealeko problema bereziak

Adibidea:

Ipar-mendebaldeko ertzaren metodoaren bidez lortutako hasierako oinarrizko soluzio bideragarria erabiliz problemaren soluzio optimoa lortu

		Helmuga				
		1	**2**	**3**	**4**	**Eskaintza**
Jatorria	**A**	1	2	3	3	5
	B	4	3	2	0	7
	C	0	1	2	1	12
Eskaria		3	5	8	8	24

Programazio linealeko problema bereziak

Hasierako oinarrizko soluzio bideragarria honako hau da:

	1	2	3	4
A	3	2		
B		3	4	
C			4	8

Garraio-algoritmoa:

1. iterazioa:

1. pausua: Optimaltasun baldintzak betetzen diren aztertu, horretarako oinarrizkoak diren aldagaiak kontuan hartuz

$$u_i + v_j - c_{ij} = 0$$

Programazio linealeko problema bereziak

$$\left.\begin{array}{l} x_{11} \rightarrow u_1 + v_1 - c_{11} = 0 \Rightarrow u_1 + v_1 - 1 = 0 \\ x_{12} \rightarrow u_1 + v_2 - c_{12} = 0 \Rightarrow u_1 + v_2 - 2 = 0 \\ x_{22} \rightarrow u_2 + v_2 - c_{22} = 0 \Rightarrow u_2 + v_2 - 3 = 0 \\ x_{23} \rightarrow u_2 + v_3 - c_{23} = 0 \Rightarrow u_2 + v_3 - 2 = 0 \\ x_{33} \rightarrow u_3 + v_3 - c_{33} = 0 \Rightarrow u_3 + v_3 - 2 = 0 \\ x_{34} \rightarrow u_3 + v_4 - c_{34} = 0 \Rightarrow u_3 + v_4 - 1 = 0 \end{array}\right\}$$

Sistema ebatzi v_1=0 eginez

Garraio-kostuak				
	1	**2**	**3**	**4**
A	$\boxed{1}$	$\boxed{2}$	3	3
B	4	$\boxed{3}$	$\boxed{2}$	0
C	0	1	$\boxed{2}$	$\boxed{1}$

Programazio linealeko problema bereziak

Sistema v_1=0 eginez ebatzi:

$$u_1 = 1, v_2 = 1, u_3 = 2, u_2 = 2, v_3 = 0, v_4 = -1$$

$z_{ij} = u_i + v_j$	$v_1 = \mathbf{0}$	$v_2 = 1$	$v_3 = 0$	$v_4 = -\mathbf{1}$
$u_1 = 1$	1	2	1	0
$u_2 = 2$	2	3	2	1
$u_3 = 2$	2	3	2	1

$z_{ij} - c_{ij}$ taula eraiki:

W_{ij}	**1**	**2**	**3**	**4**
A	$\boxed{0}$	$\boxed{0}$	-2	-3
B	-2	$\boxed{0}$	$\boxed{0}$	1
C	2	2	$\boxed{0}$	$\boxed{0}$

Programazio linealeko problema bereziak

$\exists W_{ij}= z_{ij} - c_{ij} \geq 0 \Rightarrow W_{ij} = z_{ij} - c_{ij}$ positiboen artean maximoa duen aldagaia oinarrira sartuko da.

Kasu honetan bi balio berdin daudenez, bat aukeratzen dugu:

$W_{32} = 2 \Rightarrow x_{32}$ aldagaia oinarrian sartuko da

Programazio linealeko problema bereziak

2. pausua: Oinarritik irtengo den aldagaia aukeratzeko oinarrizko aldagaien eta x_{32} aldagaiaren arteko ziklo bat eraikiko da:

Oinarrizko soluzio bideragarria:

Garraio-fluxuen taula					Eskaintza
	1	2	3	4	
A	3	2			5
B		$3-\tau_1$	$4+\tau_1$		7
C		τ_1	$4-\tau_1$	8	12
Eskaria	3	5	8	8	

Programazio linealeko problema bereziak

Eskaintzak eta eskariak bete behar direla kontuan izan behar da $\Rightarrow \tau_1$ taulan sartzean oinarrizko aldagai batzuen balioak aldatu behar dira.

x_{32} aldagaiari fluxu positibo bat esleitu behar zaio (τ_1), oinarrizkoa den aldagai bati 0 balioa emanez eta bideragarritasun mantenduz.

Kasu honetan aukera bakarra $\tau_1 = 3$ da $\Rightarrow x_{22}$ oinarritik irtetzen da.

Programazio linealeko problema bereziak

Ondorioz, bigarren oinarrizko soluzio bideragarria ondorengoa da:

Garraio-fluxuen taula					Eskaintza
	1	2	3	4	
A	3	2			5
B		0	7		7
C		3	1	8	12
Eskaria	3	5	8	8	

Programazio linealeko problema bereziak

2. iterazioa:

1. pausua: Optimaltasun baldintzak betetzen diren aztertu, horretarako oinarrizkoak diren aldagaiak kontuan hartuz

$$u_i + v_j - c_{ij} = 0$$

$$\left.\begin{array}{l}
x_{11} \rightarrow u_1 + v_1 - c_{11} = 0 \Rightarrow u_1 + v_1 - 1 = 0 \\
x_{12} \rightarrow u_1 + v_2 - c_{12} = 0 \Rightarrow u_1 + v_2 - 2 = 0 \\
x_{23} \rightarrow u_2 + v_3 - c_{23} = 0 \Rightarrow u_2 + v_3 - 2 = 0 \\
x_{32} \rightarrow u_3 + v_2 - c_{32} = 0 \Rightarrow u_3 + v_2 - 1 = 0 \\
x_{33} \rightarrow u_3 + v_3 - c_{33} = 0 \Rightarrow u_3 + v_3 - 2 = 0 \\
x_{34} \rightarrow u_3 + v_4 - c_{34} = 0 \Rightarrow u_3 + v_4 - 1 = 0
\end{array}\right\}$$

Sistema ebatzi v_1=0 eginez

Programazio linealeko problema bereziak

Sistema v_1=0 eginez ebatzi:

$$u_1 = 1, v_2 = 1, u_3 = 0, u_2 = 0, v_3 = 2, v_4 = 1$$

$z_{ij} = u_i + v_j$	$v_1 = 0$	$v_2 = 1$	$v_3 = 2$	$v_4 = 1$
$u_1 = 1$	1	2	3	2
$u_2 = 0$	0	1	2	1
$u_3 = 0$	0	1	2	1

$z_{ij} - c_{ij}$ taula eraiki:

W_{ij}	1	2	3	4
A	$\boxed{0}$	$\boxed{0}$	0	-1
B	-4	-2	$\boxed{0}$	1
C	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Programazio linealeko problema bereziak

$\exists W_{ij} = z_{ij} - c_{ij} \geq 0 \ \Rightarrow W_{ij} = z_{ij} - c_{ij}$ positiboen artean maximoa duen aldagaia oinarrira sartuko da.

$W_{24} = 1 \ \Rightarrow x_{24}$ aldagaia oinarrian sartuko da

Programazio linealeko problema bereziak

2. pausua: Oinarritik irtengo den aldagaia aukeratzeko oinarrizko aldagaien eta x_{24} aldagaiaren arteko ziklo bat eraikiko da:

Oinarrizko soluzio bideragarria:

Garraio-fluxuen taula					Eskaintza
	1	**2**	**3**	**4**	
A	3	2			5
B			$7-\tau_2$	τ_2	7
C		3	$1+\tau_2$	$8-\tau_2$	12
Eskaria	3	5	8	8	

Programazio linealeko problema bereziak

Eskaintzak eta eskariak bete behar direla kontuan izan behar da ⇒ τ_2 taulan sartzean oinarrizko aldagai batzuen balioak aldatu behar dira.

x_{24} aldagaiari fluxu positibo bat esleitu behar zaio (τ_2), oinarrizkoa den aldagai bati 0 balioa emanez eta bideragarritasun mantenduz.

$\tau_2 = 7$ da ⇒ x_{23} oinarritik irtetzen da.

Programazio linealeko problema bereziak

Ondorioz, hirugarren oinarrizko soluzio bideragarria ondorengoa da:

Garraio-fluxuen taula					Eskaintza
	1	**2**	**3**	**4**	
A	3	2			5
B				7	7
C		3	8	1	12
Eskaria	3	5	8	8	

Programazio linealeko problema bereziak

<u>3. iterazioa:</u>

1. <u>pausua:</u> Optimaltasun baldintzak betetzen diren aztertu, horretarako oinarrizkoak diren aldagaiak kontuan hartuz

$$u_i + v_j - c_{ij} = 0$$

$$\left.\begin{array}{l}
x_{11} \to u_1 + v_1 - c_{11} = 0 \Rightarrow u_1 + v_1 - 1 = 0 \\
x_{12} \to u_1 + v_2 - c_{12} = 0 \Rightarrow u_1 + v_2 - 2 = 0 \\
x_{24} \to u_2 + v_4 - c_{24} = 0 \Rightarrow u_2 + v_4 - 0 = 0 \\
x_{32} \to u_3 + v_2 - c_{32} = 0 \Rightarrow u_3 + v_2 - 1 = 0 \\
x_{33} \to u_3 + v_3 - c_{33} = 0 \Rightarrow u_3 + v_3 - 2 = 0 \\
x_{34} \to u_3 + v_4 - c_{34} = 0 \Rightarrow u_3 + v_4 - 1 = 0
\end{array}\right\} \text{Sistema ebatzi } v_1\text{=0 eginez}$$

Programazio linealeko problema bereziak

Sistema v_1=0 eginez ebatzi:

$$u_1 = 1, v_2 = 1, u_3 = 0, u_2 = -1, v_3 = 2, v_4 = 1$$

$z_{ij} = u_i + v_j$	$v_1 = 0$	$v_2 = 1$	$v_3 = 2$	$v_4 = 1$
$u_1 = 1$	1	2	3	2
$u_2 = -1$	-1	0	1	0
$u_3 = 0$	0	1	2	1

$z_{ij} - c_{ij}$ taula eraiki:

W_{ij}	1	2	3	4
A	$\boxed{0}$	$\boxed{0}$	0	-1
B	-5	-3	-1	$\boxed{0}$
C	0	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$

Programazio linealeko problema bereziak

$W_{ij} = z_{ij} - c_{ij} \leq 0 \quad \forall i = 1, \dots, m, j = 1, \dots, n \Rightarrow$ soluzioa optimoa Ondorioz, hirugarren oinarrizko soluzio bideragarria ondorengoa da:

Garraio-fluxuen taula					Eskaintza
	1	**2**	**3**	**4**	
A	3	2			5
B				7	7
C		3	8	1	12
Eskaria	3	5	8	8	

Hau da, $x_{11} = 3; x_{12} = 2; x_{24} = 7; x_{32} = 3; x_{33} = 8; x_{34} = 1$

Helburu-funtzioa $1 \times 3 + 2 \times 2 + 0 \times 7 + 1 \times 3 + 2 \times 8 + 1 \times 1 = 27$

Programazio linealeko problema bereziak

➨ 5.2 Esleipen problema

Kostu osoa minimizatzeko edo baliagarritasun funtzioa maximizatzeko asmoz *n* eginkizun zehatz *n* pertsonari esleitzen datza. *i.* pertsonari *j.* eginkizuna esleitzean 1 balioa hartzen duen eta esleipen hori ematen ez bada berriz, zero balio hartzen duen aldagai bitarra x_{ij} adieraziz eta esleipen honi dagokion irabazia c_{ij} denotatuz, problemaren formulazioa ondorengoa da:

Programazio linealeko problema bereziak

Problemaren formulazioa honako hau da:

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

$$non \quad \sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, 2, \ldots, n$$

$$x_{ij} \in \{0,1\} \quad i = 1, 2, \ldots, n \,;\; j = 1, 2, \ldots, n$$

Programazio linealeko problema bereziak

- Murrizketen lehenengo azpimultzoak eginkizun bakoitza pertsona bakar bati esleitzen zaiola adierazten du
- Bigarren azpimultzoak ordea, pertsona bakoitzak eginkizun bakar bat egingo duela esan nahi du.
- Esleipen problema garraio problemaren kasu partikular bat bezala uler daiteke
- Problema orekatua ez dagoenean, hau da, eginkizun kopurua eta pertsona kopurua bat ez datozenean, behar adina eginkizun edo pertsona erantsi behar dira.
- Gezurretako pertsona edo eginkizun horiek esleipen-kostu nulua izango dute.

Programazio linealeko problema bereziak

Esleipen problemaren informazio esanguratsua **esleipen-kostuen taula**n ematen da:

	H_1	H_2	$\dots$	H_n
I_1	c_{11}	c_{12}	$\dots$	c_{1n}
I_2	c_{21}	c_{22}	$\dots$	c_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_n	c_{m1}	c_{m2}	$\dots$	c_{mn}

I_i, H_j iturburu-puntuak eta helburu-puntuak (lanak eta pertsonak) izanik.

Programazio linealeko problema bereziak

5.2.1 Metodo hungariarra:

Metodo hungariarra **minimizazio** kasuan aplika daiteke.

1. pausua: Problema orekatu
   $i = 1, \dots, n$
2. pausua: Zeroak lortu esleipen-kostuen taulako errenkadatan.

   Errenkada bakoitzeko elementu guztiei errenkada horretako minimoa kendu:

   $$u_i = \min_j \{c_{ij}\} \quad i = 1, \dots, n \text{ kalkulatu}$$

   $$c_{ij}' = c_{ij} - u_i \quad i, j = 1, \dots, n \text{ taulako balio berriak lortu}$$

Programazio linealeko problema bereziak

3. pausua: Zeroak lortu esleipen-kostuen taulako zutabeetan.

Zutabe bakoitzeko elementu guztiei zutabe horretako minimoa kendu:

$$v_j = \min_i \{c_{ij}'\} \quad j = 1, \dots, n \text{ kalkulatu}$$

$$c_{ij}'' = c_{ij}' - v_j \quad i,j = 1, \dots, n \text{ taulako balio berriak lortu}$$

Programazio linealeko problema bereziak

4. pausua: Zeroak esleitu.

Zero kopuru txikieneko errenkada edo zutabea aukeratu.

Bertan zero bat esleitu eta errenkada edo zutabe berean dauden gainerako zeroak ezabatu.

Zeroen esleipenarekin jarraitu, ezabatu gabeko zero kopuru txikiena duen errenkadatik edo zutabetik hasita

- Zeroak esleitzearen prozesuaren amaieran errenkada guztiek esleitutako zero bat badute, soluzio optimoa da. **Amaitu**
- Zeroak esleitzearen prozesuaren amaieran esleitutako zerorik ez duen errenkadaren edo zutaberen bat existitzen bada, **5. pausura joan**

Programazio linealeko problema bereziak

<u>5. pausua:</u> Taulako zero guztiak estaltzen dituen errenkada edo zutabe kopuru minimoa aukeratu behar da. Aukeraketarako ondoko prozedura erabiltzen da.

5.1 Esleitutako zerorik ez duen errenkada oro markatu

5.2 Aurreko 5.1 pausuan markatutako errenkadetan ezabatutako zeroa duten zutabeak markatu.

5.3 Aurreko 5.2 pausuan markatutako zutabeetan zero bat esleitu duten errenkadak markatu.

5.2 eta 5.3 pausuak errepikatu markatu izan **ez** diren errenkada eta markatutako zutabeek zero guztiak estaltzen dituzte. Errenkada eta zutabe horiek estali eta 6. pausura joan.

Programazio linealeko problema bereziak

6. pausua: Zero berriak sortu

Estali gabeko elementuen artetik minimoa aukeratu

Estali gabeko errenkadetako elementuei balio hori kendu eta estalitako zutabeetako elementuei gehitu.

4. Pausura joan

Programazio linealeko problema bereziak

Adibidea:

Demagun A, B, C eta D eraikinen eraikuntzarako lehiaketara lau kontrastitek beraien proiektuak aurkeztu dituztela, kontratista bakoitzari eraikin baten eraikuntza esleituko zaiolarik. Ondoko taulan kontratista bakoitzak eraikuntza bakoitza altxatzeko beharko lukeen denbora agertzen da.

Helburua lau eraikuntzak denbora minimoan altxatzea bada, kalkula ezazu eraikuntzen eta kontratisten arteko esleipen optimoa.

Programazio linealeko problema bereziak

Adibidea:

	1	2	3	4
A	58	58	60	54
B	66	70	70	78
C	106	104	100	95
D	52	54	64	54

1. iterazioa:

- 1. pausua: 4 Kontratista eta 4 eraikin ⟶ orekatua
- 2. pausua:

	1	2	3	4	
A	4	4	6	0	⟶ $u_1 = 54$
B	0	4	4	12	⟶ $u_2 = 66$
C	11	9	5	0	⟶ $u_3 = 95$
D	0	2	12	2	⟶ $u_4 = 52$

- 3. pausua:

	1	2	3	4
A	4	2	2	0
B	~~0~~	2	0	12
C	11	7	1	~~0~~
D	~~0~~	0	8	2
	↓ $v_1 = 0$	↓ $v_2 = 2$	↓ $v_3 = 4$	↓ $v_4 = 0$

- 4. pausua:

(A, 4) aukeratu ⟶ (C, 4) ezabatu
(D, 2) aukeratu ⟶ (D, 1) ezabatu
(B, 3) aukeratu ⟶ (B, 1) ezabatu

- 5. pausua:
  - 5.1: esleitutako zerorik gabeko errenkadak: C
  - 5.2: C errenkadan ⟶ ezabatutako zeroa duen zutabea: 4
  - 5.3: 4. zutabean ⟶ zeroa esleituta duen errenkada: A

	1	2	3	4	
A	4	2	2	0	X
B	~~0~~	2	0	12	
C	11	7	1	~~0~~	X
D	~~0~~	0	8	2	
				X	

- 6. pausua: estali gabeko elementuen minimoa: 1
  - estali gabeko errenkadako elementuei 1 kendu
  - estalitako zutabeetako elementuei 1 gehitu

	1	2	3	4
A	(4-1) 3	(2-1) 1	(2-1) 1	(0-1+1) 0
B	0	2	0	(12+1) 13
C	(11-1) 10	(7-1) 6	(1-1) 0	(0-1+1) 0
D	0	0	8	(2+1) 3

- 4. pausua:

(A,4) aukeratu ⟶ (C,4)
(C,3) aukeratu ⟶ (B,3)
(B,1) aukeratu ⟶ (D,1)
(D,2) aukeratu

Errenkada guztiek esleitutako zero bat dute ⟶ Amaitu ⟶ Optimoa lortu dugu.

A eraikina 4. kontratistak
B eraikina 1. kontratistak
C eraikina 3. kontratistak
D eraikina 2. kontratistak

$z = c_{A4} + c_{B1} + c_{C3} + c_{D2} = 54 + 66 + 100 + 54 = 274$

Programazio linealeko problema bereziak

Oharra:

Metodo hungariarra helburu-funtzioa minimizatzea denean bakarrik aplika daiteke. Problema maximizazio problema bat bada helburu funtzioan ondorengo aldaketa egin behar da:

$$\text{Max } Z \quad \Rightarrow \quad \text{Min } -Z \quad \Rightarrow \quad \text{Min } Z'$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad -Z = \sum_{i=1}^{m}\sum_{j=1}^{n} -c_{ij}x_{ij} \qquad Z' = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}'x_{ij} \;\; \text{non} \;\; c_{ij}' = -c_{ij} + c_{kl}$$

$$-c_{kl} = \min\left\{-c_{ij} \;\; / \; -c_{ij} < 0\right\} \;\; \text{izanik}$$

Programazio linealeko problema bereziak

Adibidea:

Demagun lau salmenta-agente eta 4 hiri ditugula eta salmenta-agente bakoitza hiri batera esleitu behar dugula.

Hurrengo taulan salmentetan espero diren gehikuntza (%tan) agertzen dira:

	Bartzelona	Madril	Sevilla	Bilbao
1. salmenta-agentea	34	10	15	28
2. salmenta-agentea	16	15	22	12
3. salmenta-agentea	10	25	13	20
4. salmenta-agentea	30	19	27	31

Programazio linealeko problema bereziak

➨ 5.3 Fluxu maximoko problema

Fluxu maximoko problema sare bat bezala planteatzen da, hortaz murrizketen matrizea modulu bakarrekoa da (hau da, bere azpimatrize erregular eta karratu guztien determinantearen balioa +1 edo -1 da) eta ondorioz soluzio optimoa osoa da.

Problema sare batean, 1. hasierako nodotik (iturritik) n. azken nodora (helmugara) bitarteko nodoetatik higitzen den fluxu maximoa lortzen datza, k_{ij} i. eta j. nodoak elkartzen dituen arkuak jasan dezaketen fluxu maximoa izanik.

Programazio linealeko problema bereziak

Problemaren formulazioa honako hau da:

$$\max \ Z = \sum_i x_{in}$$

$$\text{non} \quad \boxed{\sum_{k=1}^{n} x_{ki} = \sum_{r=1}^{n} x_{ir} \qquad i = 2,3,\ldots,n-1}$$

Fluxuaren kontserbazioa

$$0 \le x_{ij} \le c_{ij} \quad eta \quad x_{ij} = 0 \quad (i,j) \text{ arkua ez badago}$$

x_{ij} i. eta j. nodoak elkartzen dituen arkuaren bidez garraiatzen den fluxua izanik

Programazio linealeko problema bereziak

Definizioa:

Izan bedi n nodo dituen sarea, 1 hasierako nodoa eta n azken nodoa izanik. N nodo guztien multzoa P eta P^c bi azpimultzoetan banatzen duen edozein partiketa, $1 \in P$ eta $\mathrm{n} \in P^c$ izanik, hasierako nodoaren eta azken nodoaren arteko **mozketa** bat bezala ezagutzen da.

Programazio linealeko problema bereziak

Definizioa:

(P, P^c) **mozketa baten kapazitatea** $C(P, P^c)$ erabiliz denotatuko duguna, P azpimultzoko nodoetatik irtetzen diren eta P^c azpimultzoko nodoetara heltzen diren arkuen kapazitateen batura da, hau da:

$$C(P, P^c) = \sum_{i \in P, j \in P^c} k_{ij}$$

Programazio linealeko problema bereziak

Bideratutako sare bateko fluxuaren eta mozketen erlazioa ondorengoa da:

Teorema 1:

Izan bitez F fluxu bideragarri bat eta $C(P, P^c)$ (P, P^c) mozketa baten kapazitatea. Orduan:

$$F \leq C(P, P^c)$$

Ondorioz, sare bat emanik, sare horretako edozein fluxu bideragarri edozein mozketaren kapazitatea baino txikiagoa edo berdina da.

Programazio linealeko problema bereziak

Hortaz, fluxu maximoko kapazitate minimoko mozketak bornatzen du, mozketa hau **mozketa-minimoa** bezala ezagutzen da.

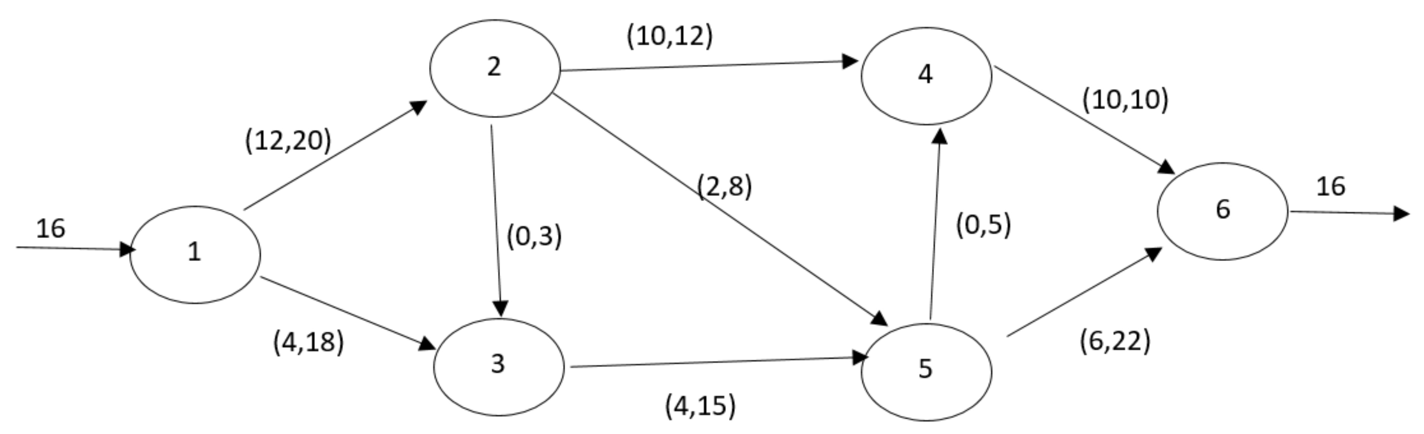
Teorema 2 (Fluxu-maximoa mozketa minimoa)

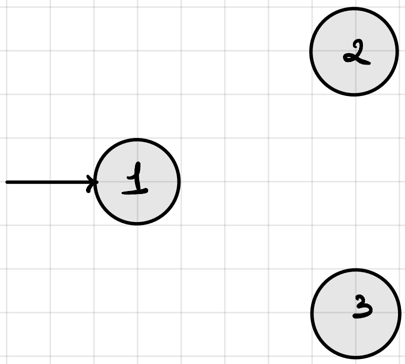
Edozein sare emanik, hasierako nodotik azken nodora higitzen den fluxu-maximoaren balioa mozketa minimoaren kapazitatearen berdina da.

Programazio linealeko problema bereziak

Adibidea

Izan bedi ondorengo sarea, arkuen artean (x_{ij}, k_{ij}) bikotea idatzi dugu, x_{ij} fluxu bideragarri baten balioa izanik:





$\boxed{F = 16}$

(1) $p = \{1,2\}$ $\quad p^c = \{3,4,5,6\}$

Mozketaren Kapazitatea: $C(p,p^c) = K_{13} + K_{24} + K_{25} = 18 + 3 + 12 + 8 = 41$

(2) $p = \{1,2,3,4,5\}$ $\quad p^c = \{6\}$

Mozketaren Kapazitatea: $C(p,p^c) = K_{46} + K_{56} = 10 + 22 = 32 \longrightarrow (\overset{F}{16} \leq 32)$

Programazio linealeko problema bereziak

Ondorioz:

Fluxu-maximoko problemaren optimo lortzeko mozketa guztien kapazitatea kalkulatu eta minimoa aukeratzea nahikoa litzateke.

Arazoa: Arku bakoitzetik higituko den fluxua ezezaguna da.

Konponbidea:

Fluxu-maximoko algoritmoa: Arku bakoitzetik higituko den fluxua ezaguna da.

Algorimoaren oinarrizko estrategia fluxuaren ibilbide gorakor bat zehazten datza.

Programazio linealeko problema bereziak

Izan bedi, hasierako nodoan hasten den ibilbide bat eta posible bada bukaerako nodoan bukatzen dena. Hasierako nodotik hasita ondorengoa egiten da:

Fluxu-gorakor ibilbidearen araua:

i. nodoa emanik, j. nodoa ibilbidean (katean*) egongo da baldin eta:

1. (i,j) arku bat existitzen bada non, $x_{ij} < k_{ij}$ (garraiatu beharreko fluxua bere kapazitatea baino txikiagoa da)
2. (j,i) arku bat existitzen bada non $x_{ij} > 0$ (fluxu positiboa)

Programazio linealeko problema bereziak

(*) Ibilide bezala jarri dugun arren, benetan kate bat da, izan ere nodo segidak norazko berdina ez izatea posible baita.

(i,j) arkua aurrerantza doala esango dugu, hasierako nodotik azken nodorako noranzkoa mantentzen badu eta (j,i) arkua berriz atzerantza doala esango dugu:

Programazio linealeko problema bereziak

5.3.1 Fluxu-maximoko algoritmoa:

1. pausua: $x_{ij}=0$ (hasierako soluzio bideragarria) eta F=0 zehaztu
2. pausua: Arku bakoitzean (x_{ij}, k_{ij}) etiketak jarri.

Fluxu-gorakor ibilbide bat existitzen ez bada amaitu ⇒ Fluxu maximoa lortu dugu

Fluxu-gorakor ibilbide bat existitzen bada, gorakorra den fluxua zehaztu:

$$\Delta = min\left\{\min_{(i,j)\in D}\left(k_{ij}-x_{ij}\right), \min_{(i,j)\in A} x_{ij}\right\}$$

D aurrentza doazen nodoen multzoa

A atzerantza doazen nodoen multzoa

Programazio linealeko problema bereziak

3. pausua: Fluxu-gorakor ibilbidean parte hartzen duten arkuen fluxua eguneratu:

- x_{ij} aurrerantza doazen arkuen fluxua Δ handitu
- x_{ji} atzerantza doazen arkuen fluxua Δ txikitu

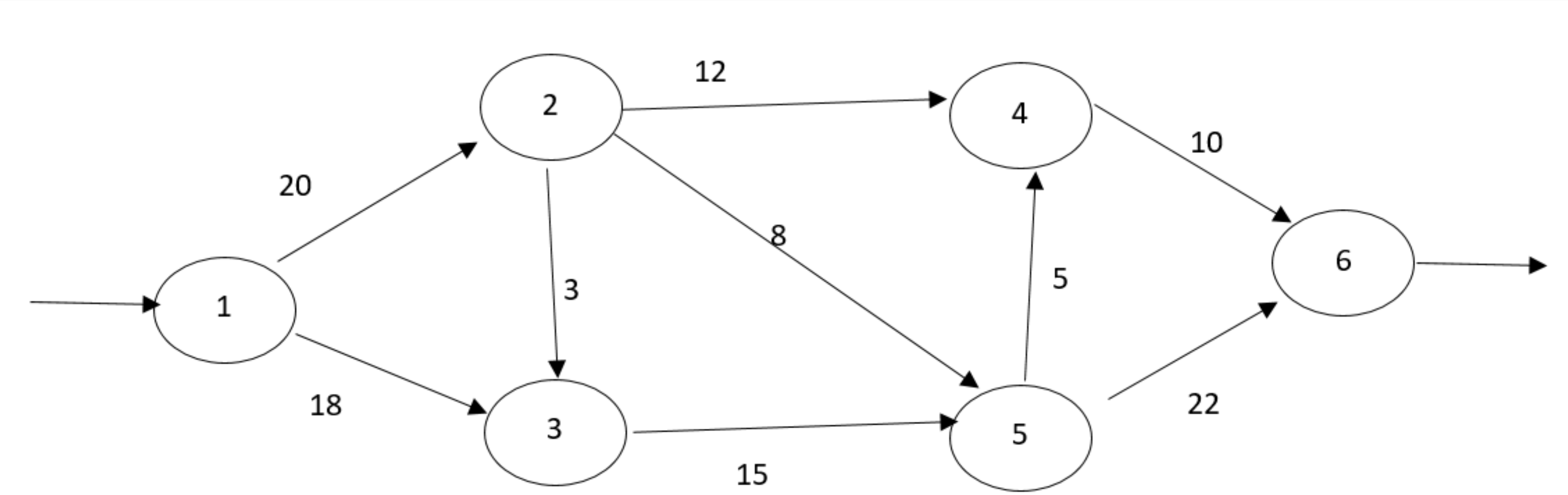
Fluxu-gorakor ibilbidean parte hartzen ez duten arkuen fluxua berdin mantentzen da

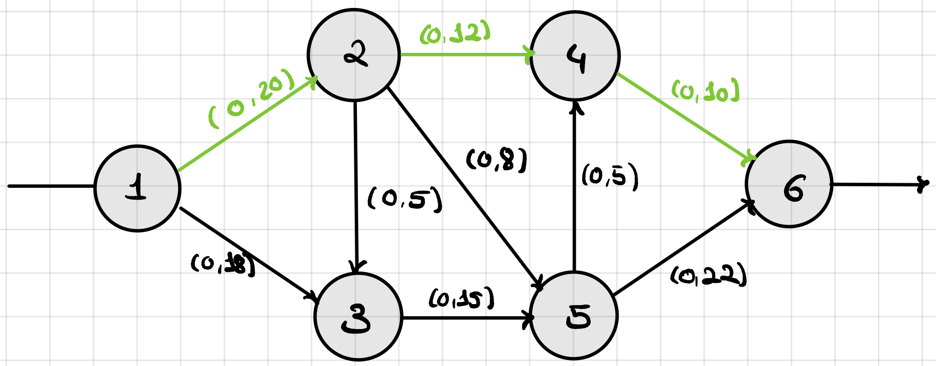
4. pausua: $Z = Z + \Delta$ egin eta 2. pausura joan.

Programazio linealeko problema bereziak

Adibidea

Izan bedi ondorengo sarea, arkuen artean k_{ij} bikotea idatzi dugu, x_{ij} fluxu bideragarri baten balioa izanik:





1) $x_{ij} = 0$ eta $F = 0$

2) 1-2-4-6 fluxu gorakor ibilbidea

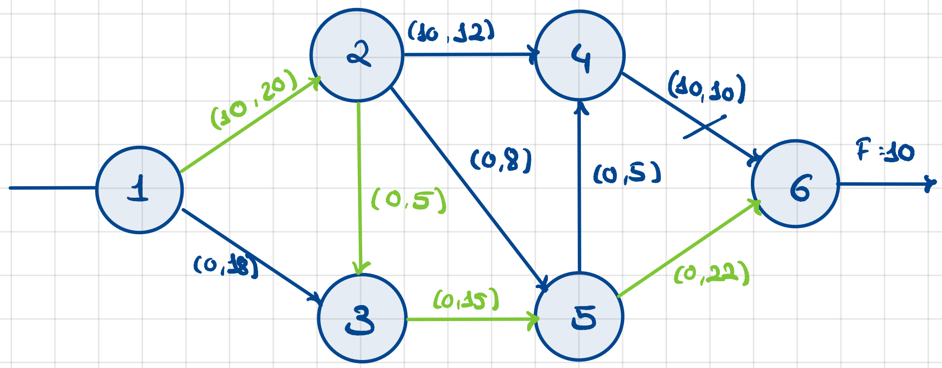
$x_{12} = 0 < 20 = k_{12}$
$x_{24} = 0 < 12 = k_{24}$
$x_{46} = 0 < 10 = k_{46}$

D ← nodo guztiek aurrerantz
$A = \emptyset$

$\Delta = \min\{20-0,\ 12-0,\ 10-0\} = 10$

3) $x_{12} = 0+10$
$x_{24} = 0+10$
$x_{46} = 0+10$

4) $F = 0+10 = 10$ → 2. pausua



2. iterazioa

2) Fluxu-gorakor ibilbidea: 1-2-3-5-6

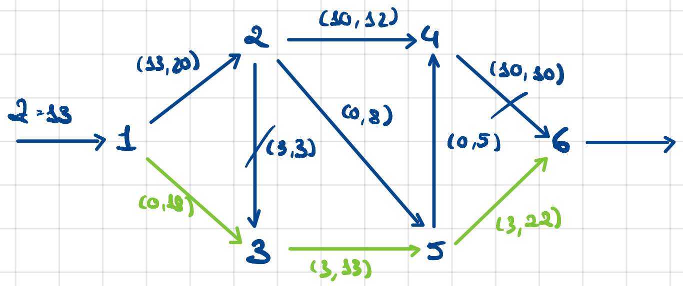
Arku guztiak aurrerantza ($A = \emptyset$)

$\Delta = \min\{20-10,\ 3-0,\ 15-0,\ 22,0\} = 3$

3) Fluxu-gorakor ibilbidean parte hartzen duten arkuen fluxua zehaztu

$x_{12} = 10+3 = 13$
$x_{23} = 0+3 = 3$
$x_{35} = 0+3 = 3$
$x_{56} = 0+3 = 3$

4)

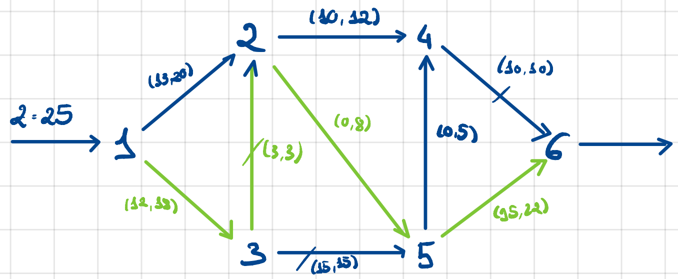


3. iterazioa

2) Fluxu gorakor ibilbidea: 1-3-5-6

Arku guztiak aurrera $\longrightarrow A = \emptyset$

$\Delta = \min\{18-0,\ 15-3,\ 22-3\} = 12$



4. iterazioa:

2) Fluxu gorakor ibilbidea: 1-3-2-5-6

$A = \{(3,2)\}$ atzerantza, beste guztiak aurrerantza

$\Delta = \min\{\min\{18-12, 8-0, 22-15\}, \min\{3\}\} =$

$\min\{7, 3\} = 3$

3) Aurrerantza doazen arkuetan +3

$x_{13} = 12+3 = 15$
$x_{25} = 0+3 = 3$
$x_{56} = 15+3 = 18$

Atzerantza doan arkuan -3: $x_{23} = 3-3 = 0$

Programazio linealeko problema bereziak

➨ <u>5.4 CPM (Critical Path Method)</u>

CPM metodoa edo ibilbide kritikoaren metodoa (Critical Path Method ingelesez) proiektuen garapenean eta kontolean erabiltzen den metodo bat da.

<u>Helburu nagusia:</u> Proiektu baten iraupena zehaztea. Proiektu bat elkarrekin erlazionatuta dauden jarduera edo ekintzen segida bat bezala ulertuz eta jarduera bakoitzak iraupen estimatu bat izanik.

Ibilbide bat proiektuearen hasieratik bukaerara doan bide bat da. Zentzu honetan, ibilide kritikoaren luzera proiektuko bide handienaren luzerarekin bat dator.

Proiektua baten iraupena ibilbide kritikoaren luzera da

Programazio linealeko problema bereziak

CPM metodoa erabiltzeko ondorengo pausuak jarraitu behar dira:

1. Proiektua eta proiektua osatzen duten jarduerak definitu
2. Jardueren arteko erlazioa ezarri
3. Grafo bat irudikatu. Grafo honetan jarduerak nodoen bidez adieraziko dira eta nodoak jatorri-erlazioak erabiliz konektatuko dira.
4. Jarduera bakoitzarekin erlazionatutako **denborak** definitu eta dagokion lasaiera lortu
5. Proiektuko ibilbide luzeena lortu. Ibilbide honek proiektuaren luzera zehaztuko du.

Programazio linealeko problema bereziak

CPM metodoa erabiltzeko ondorengo pausuak jarraitu behar dira:

1. Proiektua eta proiektua osatzen duten jarduerak definitu
2. Jardueren arteko erlazioa ezarri
3. Grafo bat irudikatu. Grafo honetan jarduerak nodoen bidez adieraziko dira eta nodoak jatorri-erlazioak erabiliz konektatuko dira.
4. Jarduera bakoitzarekin erlazionatutako **denborak** definitu eta dagokion lasaiera lortu
5. Proiektuko ibilbide luzeena lortu. Ibilbide honek proiektuaren luzera zehaztuko du.

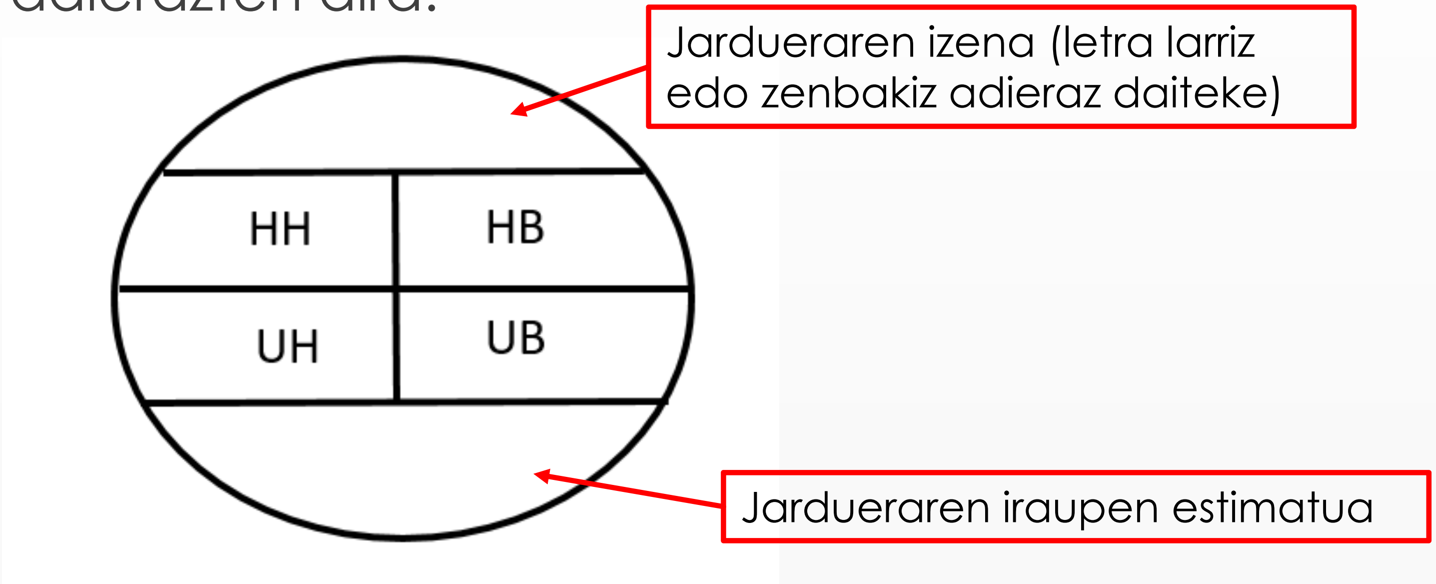
Programazio linealeko problema bereziak

Nodo bakoitzarekin (jarduera bakoitzarekin) ondorengo **denborak** erlazionatuta daude:

- **HH** (hurbilen dagoen hasiera unea): jarduera bat hasi ahal den unetik hurbilena da (aurretik dauden jarduerak osatuak izan direla suposatuz)
- **HB** (hurbilen dagoen bukaera unea): jarduera bat bukatua izan ahal den unetik hurbilena da
- **UB** (urrutien dagoen bukaera unea): proiektua atzeratu gabe, jarduera bat bukatua izan ahal den unetik urrutiena da
- **UH** (urrutien dagoen hasiera unea): proiektua atzeratu gabe, jarduera bat hasi ahal den unetik urrutiena da

Programazio linealeko problema bereziak

Lau une hauek nodo bakoitzean ondorengo notazioa jarraituz adierazten dira:



- **L** (lasaiera-denbora): proiektua atzeratu gabe jarduera bat atzeratu daitekeen epea da

Programazio linealeko problema bereziak

5.4.1 Ebazpen metodoa:

Metodo hungariarra **minimizazio** kasuan aplika daiteke.

1. pausua: Lehenengo jardueraren $HH = 0$ eta $HB = HH + t$ egin. t jarduera hori egiteko behar den denboraren estimazioa izanik.
2. pausua: Bigarren jarduerara joan
   - Edozein jardueraren HH balioa bere aurrekariak diren jardueren HB balioetatik maximoa da
   - Edozein jardueraren HB balioak: $HB = HH + t$ betetzen du

   Grafoan aurrerantz mugitu, hau da, ezkerretik eskumara mugitu eta jarduera guztien HH eta HB balioak lortu.

Programazio linealeko problema bereziak

3. pausua: Azken jardueratik hasi: Jarduera honen UB balioa HB balioaren berdina da: $UB = HB$ eta UH balioak $UH = UB - t$ betetzen du.

4. pausua: Azken-aurreko jarduerara joan. Edozein jardueraren UB balioa bere ondorengoak diren jardueren UH balioetatik minimoa da. Bestalde UH balioak $UH = UB - t$ betetzen du. Azken hirugarren jarduerara joan, …

Hau da grafoan atzerantz mugitu, hau da, eskumatik ezkerrera mugitu, lehenengo jardueraren UB eta UH balioak lortu arte.

Programazio linealeko problema bereziak

5. pausua: Jarduera bakoitzaren lasaiera-denbora (L) zehaztu: L=UB-HB (edo UU-HH) eginez.

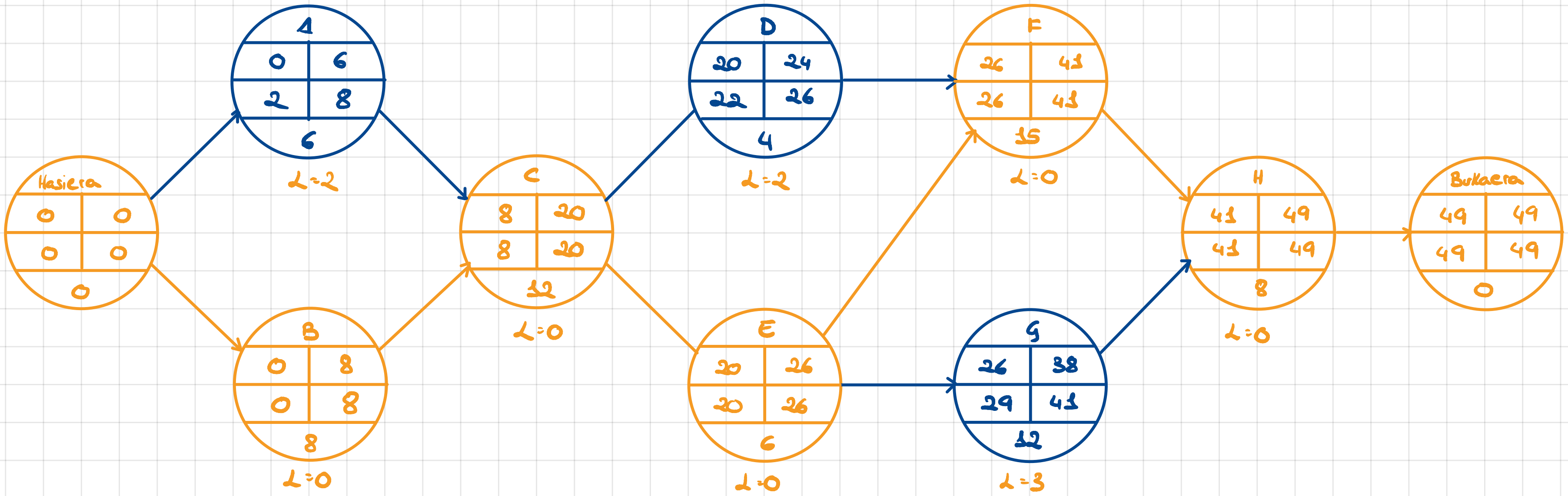
Ibilbide kritikoa lasaiera-denbora nulua duen ibilbidea da.

Programazio linealeko problema bereziak

Adibidea

Ondorengo taulan proiektu bat aurrera ateratzeko jarduerak ondorengo taulan laburbiltzen dira:

Jarduera	Aurrekaria	Iraupena (egunak)
A	-	6
B	-	8
C	A,B	12
D	C	4
E	C	6
F	D,E	15
G	E	12
H	F,G	8



Ibilbide Kritikoa: B - C - E - F - H

Proiektuaren iraupena: 49 egun