Kapitulua 1

Kriptografia eta Kodifikazioa

1.1 Sarrera eta historia

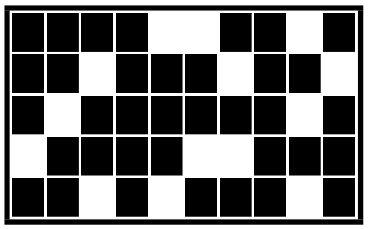
Betitik mezuak eta informazioa igorlengandik jasotzailearenganaino iritsi bitartean ikutzen dituen eskuek informazioaz ez jabetzeko beharra egon da. Horregatik mezuaren idazkera beste inorrentzako ezezaguna egiten zaion lengoaiaz edo ikur sortaz adierazteko estrategiak sortu dira. Hori bai, bide amaieran legezko jasotzaileak mezua jasotzen duenean informazioa berreskuratzeko formula izan behar du ambiguotasunik gabe. Horrek esan nahi du aurkitu behar direla aplikazioak non informazioaren hasierako idazkera beste itxura batean eta transformazioaren arauak oso zailak izan behar dute aplikazioaren aldaranzteko formularik ez daukanarentzat. Hori da kriptografia teoriaren motibazioa.

Historian ezagutu den mezuak enkriptatzeko beharra izan eta burutu egin zuen lehenengoetarikoa Julio Cesar izan zen mezuen letra bakoitzari beste letraz ordeztuz. Estrategia arrunt horietatik gaur egun erabiltzen diren RSA metodo sofistikatuetara iritsi arte bide luzea ibili da. Orokorrean bi motako kriptografia metodoak daude, ikurren ordezkapena erabiltzen dutenak eta ikurren orden aldaketa.

1.1.1 Teknika arrunt mekanikoak

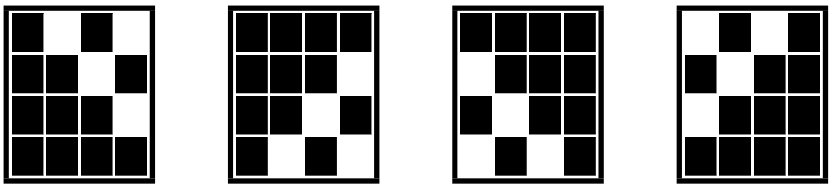
- **Cardano-ren saretoa**

  XVVI mendean Gerolamo Cardanok sortutako estrategia hau $n \times m$ laukitxoetako saretoa eraikitzean datza zoriz hartutako lauki batzuk zulatuta izanik. Bai mezu igorleak eta jasotzaileak sareto berdina izan behar dute eta mezuaren esaldiaren ordenari eutsiz, haien letrak banan bana soilik zulatuetako laukitxoan idatziko dira, beste lauki guztietan letra nahasketa erantsiz.



- **Sareto biratzailea**

  Saretoaren berezi baten laguntzaz sortutako beste metodo arrunta, mekanismo biratzailea erabiltzea da. Kasu honetan $4 \times 4 = 16$ laukitxoetako saretoa edo, $6 \times 6 = 36$-etakoa,... aukeratzen da zulo batzuekin eta 90^o biraketa bakoitzean agerian geratzen diren laukizuzentxoetan mezuaren letra-segida orden arruntean banan bana idatziko dugu.



Adibidea 1.1.1 *Idatz ezazu "kontaktua txamarra orlegidun gizona da" mezuaren aurreneko 16 letrak aurreko sareto biratzailearen laguntzaz.*

Adibidea 1.1.2 *Zenbaki itzazu laukitxo bakoitza zenbaki ezberdin batekin eta idatzi aurreko saretoaren laukitxo zurien multzoari dagokion biraketa bakoitzari dagokion aplikazioa eta aplikazioen konposaketa.*
Ikur orden aldaketari ze aplikazio mota dagokio?
Ze baldintza bete behar duen egiturak aplikazioa injektiboa izateko eta beraz alderantzizko aplikazioa existitu ahal izateko?
Ze erlazioa dago grafo teoriarekin?

- **Playfair enkriptazioa**

  Teknika hau Ingalaterran erabili ohi zen mezu sekretuak bidali baino lehen letra sorta nahasiak bihurtzeko. Metodoa ordezkapen motakoa da. Letra bakoitza bestearen ordez ordezten da, baina aplikazioa taula bereziaz emango da. Ordezkapen araua 25 laukitxoekiko taulaz emanda dago, non hasieran igorleak eta jasotzaileak ezagutzen duten letra ezberdinetaz osatutako hitz bat den, eta jarraian alfabetikoki osatutako beste letra guztiak. Adibidez, baldin hitz sekretua "gernika" bada, orduan zifratzeko taula hau da,

g	e	r	n	i
k	a	b	c	d
f	h	j	l	m
o	p	q	s	t
u	v	x	y	z

Mezua zifratzeko, lehenbizian letra segidaren osagaien pareak bereiziko ditugu. Adibidez "ogia egiteko galirina ura eta legamia behar dugu" mezua letra pare sorta honetan banatuko dugu {og - ia - eg - it - ek - og - al - ir - in - au - ra - et - al - eg - am - ia - be - ha - rd - ug - u} . Ondoren, ordezkapen arau hauek erabiliko dira letra bakoitza beste ikurraz ordezteko,

- Baldin bi letrak ilara berean badaude, bakoitza bere eskubialdean dagoen letraz ordeztuko da (ilarako azken letra bada, ilara bereko hasierakoaz ordeztuko da).
- Baldin bi letrak zutabe berean badaude, bakoitza bere azpian dagoen letraz ordeztuko da (zutabako azken letra bada, zutabe bereko lehenengoaz ordeztuko da).
- Baldin bi letrak ilara eta zutabe ezberdinean badaude, orduan horietariko bakoitza bien artean definitzen duten laukizuzentxoaren beste bi erpinetan dagoen letrataz ordeztuko dira, ordulariaren orratzen norabidea jarraituz.

Adibidea 1.1.3 *Aurkitu zein letretaz ordezten den "e" letra textu sekretuan agertzen den aldi bakoitzean.*

- **Jefferson-en zilindroa**

  Prozedura honen mamia alfabetoko 26 letretako zenbait diskoak biltzean datza. Disko bakoitzaren letra agerpen ordena ezberdina da. Baldin disko kopurua n bada, orduan mezu sekretua n letretako bildumetan banatzen da. Letre sorta bakoitza hartuko dugu eta diskoak egokituko ditugu segida hori irakur ahal izateko, gero disko guztiak biratzen dira $k/26$ bira zatikia, hau da, k letra aurrerago. Letra sorta berria ordezkapenaren emaitz ulertezina izango da. Adibidez, ondoko taula, Jefferson-en 10 diskoko zilindroaren irudi partziala erakusten du,

h	d	i	b	j	s	r	n	f	l
v	m	g	p	e	f	k	b	z	q
m	**a**	**t**	**e**	**m**	**a**	**t**	**i**	**k**	**a**
f	r	n	r	s	o	l	s	o	x
a	e	u	z	x	q	v	g	h	s

Jefferson-en zilindroan Estatu Batuetako armadak erabili zuen XX mende hasieran 2. Munduko Guda arte eta baita Alemaniak 2. Gudan erabilitako "*enigma*" gailu ospetsua honen hobekuntza baino ez zen.

1.2 RSA kriptografia sistema eta aplikazioak

Gaur egun gizartearen jarduera ekonomikoaren zein pribatuaren garapena neurri handi batean datu garraio eta trukearen menpean dago. Garrantzitsua da datu horietariko batzuk soilik igorleak eta jasotzaileak ezagutzea eta bidean atzeman dezaketen guztiei informazio hori izkutatzea. Baina XX menda erdirarte aurkitu zituzten metodo guztiak alderantzizko funtzioa daukan zifratzeko funtzio mota batean oinarrituta daude. Noski, funtzio edo kode hori informazio-bidearen bi muturretan daudenak ezagutu behar duten. Horregatik antzinan oso ohikoa zen zifratzeko kodeak oso maiz aldatu behar edo kode taula batzuk erabiltzea jakiteko egun bakoitzean zein erabili behar diren. Horrela norberaren kodeak etsai edo lapur batek eskuratu eta kalteak eragitea saisten zuten. Helburu horrekin 1977an Rivest, Shamir eta Adlemanek RSA enkriptazio algoritmo eraginkorra plazaratu zuten alderantzizkorik irauli ezin den prozesuaren bidez. Algoritmo onen oinarria bi zatiekiko kodea izatean datza, non zati bat publikoa den enkritatzeko eta beste zatia pribatua den eta iraultzeko balio duen. Edozein kasuan zati publikoa jakitea ez da nahikoa kodearen zati pribatua ondorioztatzeko. Algoritmoaren zati matematikoa zoriz aukeratutako bi zenbaki lehen itzelak (kode izkutua) aukeratzean datza eta haien biderkadura (kode publikoa). Zati publikoarekin ez da nahikoa zati pribatua ezagutzeko oraindik ez baitira aurkitu zenbaki oso handien faktorizatzeko algoritmo eraginkorrik.

Kodifikazio metodo hauek seguru etxe eta bankuetxe kontu zenbakiak, erakunde eta enpresen informazio digitala, sistema elektronikorako sarbide-pasahitzak eta beste motako informazio garrantzitsua bidaltzeko unean enkriptatzeko balio du.

Teorema 1.2.1 (Euler 1736) *Bitez $p \neq q$ zenbaki lehenak, $n = p \cdot q$ eta k zenbaki bat non $p-1$ eta $q-1$ balioak $k-1$-en faktoreak diren. Orduan edozein m zenbaki osorako m^k eta m modulu n hondar berdina emango dute,* $\mod(m^k, n) = \mod(m, n)$.

Euler-en teorematik ondoko korolario ondorioztatzen da.

Korolarioa 1.2.1 *Bedi k zenbakiak soilik bi faktoreak dauzkala, $k = d \cdot e$, beraz $c = m^e$ zenbakiaren c^d berreduraren hondarra modulu n berdin m-rena da,* $\mod(c^d, n) = \mod(m^{ed}, n) = \mod(m^k, n) = \mod(m, n)$.

Aurreko emaitzak ondoko biekin erabiliko dira enkriptatzeko algoritmoa eraikitzeko

- Baldin $m < n$ zenbaki osoak badira, orduan $\mod(m, n) = m$.
- Baldin c, d eta n zenbaki osoak badira eta $\mod(c, n) = r$, orduan $\mod(c^d, n) = \mod(r^d, n)$. Noski $c = h \cdot n + r$ eta Newton-en binomioaren formulaz $c^d = (h \cdot n + r)^d = q \cdot n + r^d$. Hau erabiliko dugu $c = m^e$ zenbakiarekin.

Adibidez $p = 3$, $q = 11$, $e = 3$ eta $d = 7$ teoremako baldintzak beteko dituzte eta $k = e \cdot d = 21$ eta $n = p \cdot q = 33$ izanik.

RSA enkriptazio algoritmoa

Behin p, q, e eta d zenbakiak aukeratu diren (praktikan p eta q zenbaki lehenak 10^{200} baino handiagoak dira) eta Euler-en teorema betetzen duten $n = p \cdot q$ eta $k = e \cdot d$ biderkadurak ere, orduan jasotzaileak kode publikoa argitaratzen du, e eta n zenbakiak hain zuzen, guztiek jakin dezaten ze araua jarraitu behar duten haien mezuak kodifikatzeko jasotzaileari bidali aurretik. Kodearen zati pribatua d, p eta q zenbakiak dira eta hauek gabe, artekari batek ezin izango du enkriptazio metodoa irauli. Aldiz, edonork bidal diezaioke mezu sekretuak jasotzaileari. Bidaltze/jasotze prozesua honela egiten da:

- Mezua enkriptatzeko ikur bakoitza zenbaki batetaz ordezten da. Adibidez alfabetikoki letra bakoitzari 1-26 zenbakia esleituko zaio, non zenbaki sorta honen osagai guztiak n baino txikiagoak diren. Behin mezuaren letra guztiak $\{\alpha_1, \ldots, \alpha_j\}$ zenbaki segida bat bihurtu diren, $\{m_1, \ldots, m_j\}$, hauetako osagai bakoitza e-gatik berretuko du eta honen hondarra modulu n kalkulatuko du, $r_i =$ mod (m_i^e, n) kalkulatuz. Orain $\{r_1, \ldots, r_j\}$ zenbaki bilduma jasotzaileari helduarazi behar zaio, nahiz eta bide publikoa erabili, adibidez interneteko blog bat.

- Mezua dekodifikatzeko jasotzaileak $\{r_1, \ldots, r_j\}$ zenbaki bildumaren osagai bakoitza d-gatik berretu eta honen hondarra modulu n kalkulatu baino ez du egin behar, mod $(r_i^d, n) =$ mod $((m_i^e)^d, n) =$ mod (m_i^k, n) eta azken ha Euler-en teoremaren ondorioz berdin m_i da. Beraz hasierako $\{m_1, \ldots, m_j\}$ bilduma eskuratuko du eta adostutako itzulketa irauliz $\{\alpha_1, \ldots, \alpha_j\}$ mezua.

Dokumentu baimenketa RSA algoritmoaz

RSA algoritmoak baita dokumentu edo textu bat benetako egileak idatzita izan dela eta argitaratzeko bidean beste inork aldaketarik eragin ez diola egin bermatuko du ere. Horretarako ondoko prozesua jarraituko da

- Dokumentuaren idazleak eta aldiberean horren jabeak mezua irakurriko dutenei mezua benetakoa dela eta harek idatzita dagoela sinistuarazi nahi die. Orduan idazleak dokumentuaren $\{\alpha_1, \ldots, \alpha_j\}$ ikurren bilduma zenbaki segida bihurtu du, $\{m_1, \ldots, m_j\}$ eta osagai bakoitza bere kode pribatuaz berretuko du m_i^d eta bere hondarra modulu n, $r_i =$ mod (m_i^d, n), $i = 1, \ldots, j$, lortuz. Orain Ikur bi segidak argitaratzen ditu (e, n) kodearen zati publikoarekin batera.

- Jasotzaileak bermatu nahi badu mezua legezko iturritik datorrela aldaketarik jasan gabe, orduan zati enkriptatuaren balioak r_i^e berretuko ditu eta egiaztetuko du ea ezaguna den m_i digitua betetzen den. Mezuak letra ugari baditu, orduan soilik d kode pribatua ezagutzen duenak ahal izan du bihurketa eragin.

Sinadura digitala RSA algoritmoaz

Askotan funtsezkoa da tresna edukitzea ondokoa zihurtzeko: igorle batek idatzitako eta bidalitako dokumentua egilearekin lotzea eta ukatzea ea beste inork egin omen duen. Hori da sinadura elektronikoa, dokumentuak eta egileak erlazionatzen duen prozedura hutsik egin gabe eta horretarako beste behin RSA algoritmoa erabiltzen da. Prozesua burutzeko sinadura elektronikoko ziurtagiriak banatzen dituen enpresa hornitzailea tarteko beharrezkoa da.

Demagun dokumentu bat partekatu behar duten bi erakunde, enpresa zein gizaki eta bermatu behar da igorleak besteari dokumentua helduarazi eta gero egiletza ukaezina izango dela. Adibidez hau gerta daiteke erakunde ofizialari edozein zerga aitorpenak ematean edo enpresa batek bezeroari eskeintza ekonomikoa proposatzean. Bitartean sinadura elektronikoko enpresa hornitzaileak $(p_e, q_e, e_e, d_e, n_e)$ RSA kodearen jabea da eta jasotzaileari $(p_2, q_2, e_2, d_2, n_2)$ RSA kodea ematen dio eta igorleari bere RSA kode pertsonala $(p_1, q_1, e_1, d_1, n_1)$ eta baita jasotzailearen zati publikoa (e_2, n_2). Nola Mezuaren karakter alfanumerikoak enkriptatzea konputazionalki garestia den, baita H "*hash funtzioa*" ematen zaie enkriptatzeko software-arekin batera mezu luzeak laburtzeko. Gainera H funtzioak bi sarrera ezberdinetarako bi irteera ezberdin itzuliko ditu ere, nahastu gabe iturri ezberdineko textuak. H funtzioa berdina izango da bai igorlearentzat eta baita jasotzailearentzat ere. Hauxek dira jarraitu beharreko urratsak:

- Igorleak bere T textuari $H(T)$ hash funtzioa aplikatuko dio eta edozein artekariarentzako irakurezina bihurtzeko $KPUB_2(H(T))$ jasotzailearen kode publikoaz enkriptatuko du. Jarraian bere kode pribatuaz enkriptatuko du $KPRIB_1(KPUB_2(H(T)))$. Orain bi fitxategi bidaliko ditu eta honi sinadura digitala esaten zaio:
  - Textu enkriptatua $KPRIB_1(KPUB_2(H(T)))$.
  - Sinadura elektronikoko enpresa hornitzailearen $KPRIB_{se}$ kode pribatuaz enkriptatutako, "*Ziurtagiri elektronikoa*" ZE fitxategia, $KPRIB_{se}(ZE)$. Honek aldiberean bi zati dauzka:
    - Igorlearen nortasuna
    - Igorlearen kode publikoa $KPUB_1 = (e_1, n_1)$.
- Jasotzaileak jasotako fitxategia informazio irakurgarri bihurtzeko, enpresa hornitzailearen kode publikoaz ziurtagiri elektronikoa irauliko du, $KPUBB_{se}(KPRIB_{se}(ZE)) = ZE$, eta irteera horretan agertzen den $KPUB_1$ kode publikoaren laguntzaz lehenengo iraultza eragin, $KPUB_1(KPRIB_1(KPUB_2(H(T)))) = KPRIB_1^{-1}(KPRIB_1(KPUB_2(H(T)))) = KPUB_2(H(T))$. Gero enkriptatutako textua jasotzailearen kodearen zati pribatuaz irauli, $KPRIB_2(KPUB_2(H(T))) = KPUB_2^{-1}(KPUB_2(H(T))) = H(T)$, eta amaitzeko textu garbia irauliko du hash funtzioaren alderantzizkoa aplikatuz. Horrela bermatzen da nortasun berezi hori daukan igorleak bidali ahal izan du mezu enkriptatu hori.

RSA algoritmorako balio duten parametroak

	kode pribatua	**kode publikoa**
1)	p=3, q=11, d=7	e=3, n=33
2)	p=3, q=23, d=9	e=5, n=69
3)	p=5, q=13, d=7	e=7, n=65
4)	p=5, q=37, d=29	e=5, n=185
5)	p=7, q=43, d=23	e=11, n=301
6)	p=7, q=79, d=67	e=7, n=553
7)	p=11, q=17, d=23	e=7, n=187
8)	p=11, q=59, d=83	e=7, n=649
9)	p=13, q=31, d=19	e=19, n=403
10)	p=13, q=47, d=79	e=7, n=611
11)	p=17, q=67, d=151	e=7, n=1139
12)	p=17, q=83, d=101	e=13, n=1411
13)	p=19, q=61, d=47	e=23, n=1159
14)	p=19, q=71, d=97	e=137, n=1349
15)	p=23, q=37, d=61	e=13, n=851
16)	p=23, q=53, d=229	e=5, n=1219

PRAKTIKA 1. RSA enkriptazio algoritmoa

1. Bidali email-ez kontaktuari zuen RSA kodeko zati publikoa.
2. Enkriptatu zuen mezua kontaktuaren RSA kodeko zati publikoaz eta hari bidali.
3. Irauli kontaktuarengandik jasotako mezu enkriptatua eta bidali irakasleari.

Laguntza (Mathematica-ko aginduak)

Agindua	**Adibidea**	**Emaitza**
{...}	zerr={{0,"T"},{1,"R"},...}	zerrenda sortu eta osatu
Mod	Mod[46787554,23]	modulua
If	If[zenbaki==1,konkod=...]	baldintza
Do	Do[Print[$i \wedge 2$],{i,1,5}]	errepikakorra
While	While[i<=5,Print[$i \wedge 2$];i++]	errepikakorra
Characters	Characters["mezua"]	hitzatik letra zerrendara
ToCharacterCode	ToCharacterCode["a"]	alfanumerikora zenbakizko kodigora
FromCharacterCode	FromCharacterCode[97]	zenbakizko kodigotik alfanumerikora
StringJoin	StringJoin["z","e","p","o"]	zerrenda alfanumerikotik katera
AppendTo	zer=; AppendTo[zer,"1"]	zerrenda atzetik osatu
PrependTo	PrependTo[zer,"0"]	zerrenda aurretik osatu

4. Saia zaitez aurkitzen automatikoki RSA $\{p, q, d, e, n\}$ kode sortak (adibidez 100 sorta), non $n = p \cdot q$ eta $k = (p-1) \cdot (q-1) + 1 = e \cdot e$ betetzen den. Mathematica-ko "PrimeQ[]" eta IntegerQ[]" aginduak erabilgarriak suerta daitezke.

PRAKTIKA 2. Cardano-ren saretoa

1. Finkatu saretoaren dimentsioa "$n \times m$" (gelxka kopurua zutabeka eta errenkadaka) eta aukeratu bete daitezken gelaxken koordenatuak. Kalkulatu saretoan behin bakoitzean sartzen diren letra kopurua, "K".
2. Idatzi mezua eta biurtu letra banako sorta batean hitz tarteko utsuneak kenduz. Kalkulatu mezuaren letra kopurua "L" eta zenbat aldiz erabili behar den saretoa mezu osoa idazteko "T". Banatu mezuaren letra multzoa T azpimultzoetan eta idatzi saretoaren idazteko dauden utsunetan beste utsuneak zoriz hautatutako letretaz osatuz.
3. Definitu $nT \times m$ dimentsiodun "Z" matrizea eta osatu mezu enkriptatuaren letra guztiaz.

Laguntza (Mathematica-ko aginduak)

Agindua	**Adibidea**	**Emaitza**
Random	Random[Integer, {1,26}]	zorizko zenbakia

1.3 Kontrol Kodeak

Munduan badaude produktu edota gizaki multzo handiak, non osagai edo bilduma bakoitza erregistro baten bitartez bereizi eta sailkatu behar den. Adibide ezagunak norberaren NAN, kotxeko matrikulak, salmentan dauden produktuen barra-kodeak edo izakien ADN-kode genetikoa da. Eginkizun horretan bi koska daude, lehenbizikoa erregistro sistema egitura ordenatua eta egokiaz hornituta egotea da, elementua identifikatzen duen kodeak bere ezaugarriei buruzko ahal den informazio eta ezaugarri gehienak eman behar duelarik. Beste koska ondokoa da, erregistroaren ikurrek elementu zuzena identifikatu behar dutela akatsik egin gabe. Erregistroa zuzena dela bermatzea edo haren ikurraren bat nahastuta badago sistema egotea akatsa zuzentzeko oso baliogarria izango da. Helburu horri ekiteko "*kontrol kodeak*"aurkitu zuten. Kontrol kodea zehazteko sistema ezberdin daude eta identifikazio kode bakoitzari laguntzen diona balio berezia izango du eta antzeko identifikazio kodeek daukatenaren ezberdina. Adibidez norberaren NAN dokumentuan agertzen den 8 digitoetako zenbakiarekin batera letra bat dator eta haren lana zenbakia ongi idatzi dugula bermatzea da. Baldin NAN zenbakiaren digito bat gaizki idazten badugu edota bi digito elkartrukatzen baditugu, orduan adierazitako zenbaki berriari dagokion kontrol-letra ezberdina izango litzateke eta horrela konturatzen gara akatsaz.

1.3.1 USAko Posta Zerbitzua

estatu batuetan bidalketa bakoitzari identifikazio zenbaki bat ematen diote, 11 digitodun zenbakia, lehenengo hamarrak erregistro izanik eta hamaikagarrena kontrol kodea. Azken hau finkatzeko aurreko 10ek osatzen dutenaren hondarra modulu 9 aterako dugu. Adibidez 4387365291-rena 3 da, beraz erregistro osoa 43873652913 litzateke. Noski, edozein digito bateko akatsa, adibidez, seigarren lekuan dagoen 6a beste edozein digitoaz ordezten badugu, adibidez, 4387325291, orduan zenbaki okerraren aldea zuzenarekin 4387365291-4387325291=40000 eta honen hondarra modulu 9 berdin $4 \neq 0$ da, beraz $\mod(4387365291, 9) = 3$ eta aldiberean $\mod(4387325291, 9) = 8$, ezberdinak izanik.

Hemen erabiltzen ari garen emaitza p eta q zenbakien hondarra berdina bada modulu n, orduan haien kendurarena $\mod(p - q, n) = 0$.

Sistema hau erraza da eta ohiko akats asko saiestuko ditu baina arazo bat dago eta $0 \leftrightarrow 9$ digito nahasketa da $\mod(9 \cdot 10, 9) = 0$ delako eta kontrol kodea ez du akatsa harrapatuko berdina delako erregistro zuzenean eta okerrean ere.

1.3.2 American Express txekeak

Aurrreko kasuaren adibide antzekoa da American Express konpainiak erabiltzen duena haren bidai txeketan 9 digitoaz osatutako erregistroa daukatela eta kontrol zenbakia finkatzen da beste digito guztien batura gehi kontrola berdin 0 modulu 9 izateko. Adibidez, 280427135 zenbakidun txekearen kontrol zenbakia 7 da, noski, $2+8+6+4+2+7+1+3+5 = 38+7 = 45 = 5 \cdot 9$.

1.3.3 UPC produktu kode-sistema orokorra

Elikadura produktuetarako mundu osoan erabili izan den kode-sistema UPC "universal product code" 12 digitoetako zenbakia da, non lehenengo 6ek egilea zehazten duten, hurrengo 5ek produktu mota eta azkena kontrolekoa den. Azken hau finkatzeko erizpidea hau da, 1go, 3., 5., 7., 9. eta 11. kokagunetan daudenak batu eta 3-gatik biderkatuko ditugu eta emaitz honi besteak batuko diogu. Kontrol zenbakia aurreko batura berdin 0 modulu 10 izateko gehitu behar zaion kopurua da. Adibidez 045000 11785 3, zenbakian $(0+5+0+1+7+5) \times 3 + (4+0+0+1+8) = 67+3 = 70 \equiv 0 \mod (10)$. Zergatik sistema honek saiesten dituen zenbaki akats arinak?

1.3.4 CODABAR kreditu-txarteletako kode-sistema

CODABAR kreditu txartelak zein odol bankuen sarrerak identifikatzeko erabiltzen den sitema da.

Aurkitu informazioa Estatu batuetako Banketxek darabiltzaten txekeak zenbakitzeko sitema eta azaldu.

Aurkitu informazioa CODABAR kreditu txartelak zein odol bankuen sarrerak identifikatzeko erabiltzen den sitema eta azaldu.

1.3.5 ISBN liburu kode-sistema

Mundu osoan ISBN "International standar book number" izendatutako sistema erabiltzen da liburuak identifikatzeko. 2007 urtetik erreferentziak 13 digitoetako zenbakiaz osatuta daude, $\alpha_1\alpha_2\cdots\alpha_{13}$. Horien artean prefijoa 3 digitoetaz osatuta dago eta dagoeneko 978 (germaniar sustraiko hizkuntz talderako) edo 979 (ingeles sustraiko hizkuntz talderako). Hurrengo 1-2 digito multzoa erregistro-taldea edo gune geografikoa zehaztuko dute. Hurrengo 1-2 digitoek editoriala zehaztuko dute. Hurrengo 6ek elementua edo liburuaren titulua eta edizioa adieraziko dute eta azkenekoa kontrol kodea da. Adibidez ISBN 978-951-45-9693-3. Kontrol-kodea zehazteko, k,

ondoko eragiketaren emaitza 10gatik zatigarria izan behar duen, $1\cdot\alpha_1+3\cdot\alpha_2+1\cdots+1\alpha_9+3\alpha_{12}$. Aurreko adibidean $(9+8+5+4+9+9)+3(7+9+1+5+6+3)=137\Rightarrow k=3$. Aitzinean, 2007 baino lehenago, soilik 10 digitutako zenbakia erabiltzen zen eta modulu 11 erabiltzen zen kontrol-kodea zehazteko.

`http://www.isbn-international.org/pages/media/Usermanuals/`

1.3.6 NAN identifikazio-kodea

Norberaren NAN dokumentuan 8 digitoetaz eta letra batetaz osatuta dago, Adibidez, 72811349-L. Ikur bilduma horretan, biztanlearen identifikazioa zenbakia da eta letra kontrol kodea. Azken honek saiesten du pertsona horren oroimen edo idaztearen akatsa digito bat nahasten badu edota bi digitoen elkartrukatzea, orduan adierazitako zenbaki okerrari dagokion letra ezberdina delako.

Letra kalkulatzeko prozesua hau da, aurrena zenbakiaren hondarra modulu 23-az, hau da 19a ikusitako adibidean, eta gero ondoko taulan zein letrari dagokion begiratuko dugu, "L" aurkituz. Taularen egitura Estatuko Administrazioak finkatutakoa da:

0	1	2	3	4	5	6	7	8	9	10	11
T	**R**	**W**	**A**	**G**	**M**	**Y**	**F**	**P**	**D**	**X**	**B**
12	13	14	15	16	17	18	19	20	21	22	
N	**J**	**Z**	**S**	**Q**	**V**	**H**	**L**	**C**	**K**	**E**	

Ikus dezagun nola sistema honek aurkitzen dituen digito batean egindako akatsa. Lehenbizian adieraz dezagun "k" digito kopuruko X zenbakiaren digituak $\mathrm{x}=(\mathrm{x}_k,\mathrm{x}_{k-1}\ldots,\mathrm{x}_1)$ bektoreaz adieraziko dugu eta bere balioa $(\lambda_k,\lambda_{k-1}\ldots,\lambda_1)$ bektorearekiko biderkadura eskalarraz, $\mathrm{X}=\mathrm{Xx}\cdot\lambda$, non $\lambda_i=10^{i-1}$. Orduan $\mathrm{x}=(\mathrm{x}_k,\ldots,\mathrm{x}_i,\ldots,\mathrm{x}_1)\to\overline{\mathrm{x}}=(\mathrm{x}_k,\ldots,\mathrm{y}_i,\ldots,\mathrm{x}_1)$ nahasketa eragiten badugu, orduan, orokortasuna galdu gabe $x_i>y_i$ hartuko dugu, eta benetako zenbakia eta okerraren arteko aldea $(x_i-y_i)\cdot 10^{i-1}$ da eta zenbaki hori ezin da 0 izan modulo 23, horren biderkagaiak soilik 2, 5 eta 1-9 zenbakiak izan bait daitezke, baina ez 23. Beraz, x-k eta $\overline{\mathrm{x}}$-ak ezin dute hondar bera izan modulu 23.

Hortaz gain, baldin bizenbakiak elkartrukatzen baditugu, orduan ondoko teorema erabiliko dugu frogatzeko zenbaki okerraren hondarra modulu 23 eta hasierakoa ezberdinak direla.

Teorema 1.3.1 *Bedi $n\neq 2,5$ zenbaki osoa, orduan $\mathrm{x}=(\mathrm{x}_k,\ldots,\mathrm{x}_i,\ldots,\mathrm{x}_1)$ identifikazio zenbakiaren digito bateko ordezkapena $x_i\to y_i$ ez da igertzen baldin eta soilik baldin* $\mod\Big((x_i-y_i),n\Big)=0$. *Baldin $x_i\leftrightarrow x_j$ osagaiak elkartrukatzen badira, nahasketa ez da igertzen baldin eta soilik baldin* $\mod\Big((x_i-x_j)\cdot(\lambda^{i-j}-1),n\Big)=0$.

Teoremaren frogapena erraza da eta zenbaki handienari txikiena kendu eta emaitza adierazi baino ez da egin behar. Emaitz honi esker ondorioztatzen da 23 hautaketa ona izan dela NAN-en zenbaki elkartrukaketaren koturatu gabeko nahasketa saiesteko, ba $0 \leq (x_i - x_j) \leq 9$ ezin da 23-ren biderkagaia eta $(\lambda^{i-j} - 1) \in \{9, 99, 999, \ldots\}$ ezta ere.

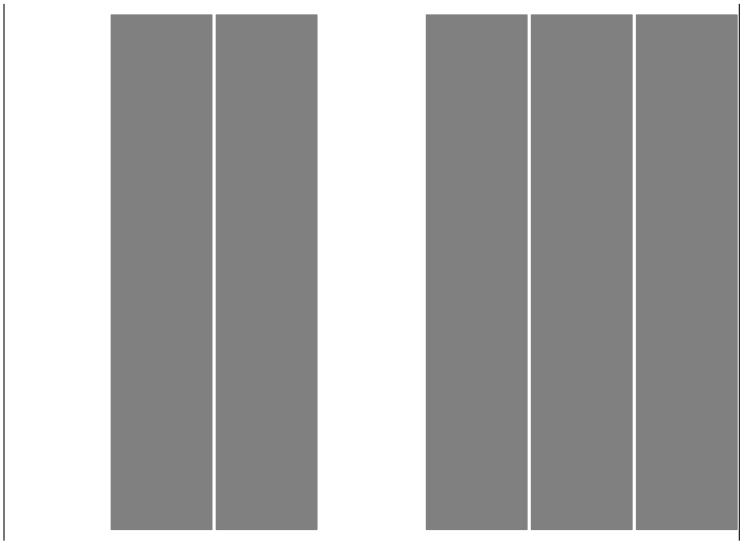
1.3.7 Barra-kodeko sistema

Salmentan dauden produktu gehienetan lodiera anitzeko marra beltz batuetaz osatutako egitura edo irudi angeluzuzenak ikusi ohi ditugu. Berauek gordetzen duten informazioa interpetatzeko eskaner edo bisore optikoak erabili ohi dira argi izpiak irudi horien gainean igortzeko. Marra beltzen artean dauden tarte zuriek islatzen duten argiaren intensitatearen arabera 0-en eta 1-en funtziopean adieraz daiteken informazio binarioa jasotzen da. Irudi angeluzuzena ertz bakoitzean pixka bat luzeagoak diren bi marretaz mugatuta dago eskanerrak tamainua eta lodiera kalibratzeko. Beraz edozein irakurgailuak jasoko du informazio bera barra-kode batetik.

Behin kode binarioak zenbaki hamartarretan bihurtu diren, guztira 12 digito agertuko dira eta produktu alearen informazioa zehazteko UPC kode-sistema unibertsala erabiliko da. Lehenengo digitoa zenbaki sistema aipatuko du, gehienetan 0a, 1a, 6a, 7a edo 8a da, baina bertako produktu galkorrak (fruta, barazkiak, okela, gazta, etabar) badira 2a erabiliko da, botika izatekotan 3a, fideltasun tarjetetarako 4a eta merkeldiko produktuetarako 5a edo 9a erabiliko da. Gero hurrengo 5 digitoetako bi multzoek hurrenez hurren egilea eta produktuaren xehetasunak zehaztuko dituzte. azkenik amaierako digitoa kontrol-kodea da.

Egilea zehazteko 5 digitoko multzoaren digito bakoitza 7 unitateko zabalera duen marra zuribeltzaz osatutako angeluzuzenaz adierazten da. 7 marretatik 3 edo 5 beltza daude eta beste 4 edo 2 zuriak. Ondoko irudian adibidea dago.

aldiz, produktuaren xehetasunak adierazten duen hurrengo 5 digitokoak zehazteko irudi multzoa 2 edo 4 marra beltzetaz eta 5 edo 3 marra zurietaz osatutak daude. Irudi bakoitzaren esanahia ondoko taulan ikusiko dudu egilearen kodea eta produktuaren kodearen kasuan, non 1-ak marra beltza eta 0-ak marra zuria adierazten duten.



Irudia 1.1: Barra-kode baten zati honek 8 digitoa adierazten du.

Balioa	**Egilea**	**Produktua**
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

Digito bakoitzari dagokion kode binarioa "*egilearen*" adierazpidetik "*produktuaren*" adierazpidea bihurtzeko 1-ak 0-en ordez ordeztuko ditugu eta alderantziz. Erizpide honi esker nahiz eta irakurle optikoa buelta emanda egon, hark badaki zein aldetik hasi behar duen irakurtzen, marra beltz (1 ikurrak) kopuru bakoitia egon behar baitu aurreneko 7 hutsuneen artean. Batzutan EAN-13 kodifikazioa erabiliz gero 5 digitoko bi multzo hauen aurretik bi digitoko multzoa dago errialdearen kodearekin, adibidez Espainiakoa 84 da. Amaieran kontrol kodea dago, EAN-13-ren kasuan, egile, produktu eta errialde 12 digituko kodeak kontutan hartuko ditugu eta eskubitik hasita, kokapen bakoitietan daudenak batuko ditugu eta emaitzari gehituko diogu kokapen bikoitietan dauden digitoak 3gatik biderkatutak. Kontrol kodea finkatuko dugu aurreko kopuruari gehitzeko eta batura osoa 0 izateko modulu 10.

PRAKTIKA. Kontrol-kodeak

1. Eraiki funtzio bat non parametroa NAN zenbakia den eta irteera zenbakiari dagokion kontrol-kodea edo letra den.

2. Eraiki funtzio bat non parametroa 10 digitoetako zenbakia den eta irteera zenbakiari dagokion barretako kodea (bost digito 3→0 eta 2→1). Gainera hasieran eta amaieran erantsi 1 ikurra.

3. (Hautazkoa) Sortu n barra luzekin (1-ak) eta m barra motzekin (0-ak) zenbaki sistema bitarrean adierazitako ikurrak eta zenbaki sistema hamartarreko zenbakiekiko identifikazioa.

Laguntza

Mathematica-ko aginduak:

Agindua	**Adibidea**	**Emaitza**
{...}	zerr={{0,"T"},{1,"R"},...}	zerrenda sortu eta osatu
Mod	Mod[46787554,23]	modulua
If	If[zenbaki==1,konkod=...]	baldintza
Do	Do[Print[i^2],{i,1,5}]	errepikakorra
IntegerDigits	IntegerDigits[396]	zenbakitik digito-zerrendara
ToCharacterCode	ToCharacterCode["a"]	alfanumerikotik zenbakizko kodigora
FromCharacterCode	FromCharacterCode[97]	zenbakizko kodigotik alfanumerikora
StringJoin	StringJoin["z","e","p","o"]	zerrenda alfanumerikotik katera
AppendTo	zer=; AppendTo[zer,"1"]	zerrenda atzetik osatu
PrependTo	PrependTo[zer,"0"]	zerrenda aurretik osatu

Kapitulua 2

Algebra lineala.

2.1 Google-ren oinarri matematikoa

Internet-en mila miloika webgune daude eta erabiltzaileak behar duen informazioa emanten duen orrialdea aurkitzeko estrategia edo algoritmo eraginkorrak programatutak daukaten bilatzaile azkarrak beharrezkoak dira. Horretarako 1998an lehenbizian baino askoz zehatzago eta azkarragoa zen internet-bilatzailea aurkitu zuten Stanford Unibertsitateko Sergei Brin eta Lawrence Page. Harrigarria badirudi ere, haien asmakizuna erabiltzen dugunean hitz-gako egokiak aukeratuz, gehienetan guk bilatutako webgunea Google-k aurkeztutako lehenengo iradokizunen artean dago. Algoritmoak haiek proposatutako Page-Rank erizpidea eta XX menda hasieran frogatutako algebra linealeko Perron-en eta Frobenius-en teoremetan oinarrituta dago.

Aurreneko erronka webgune bakoitzari garrantzitasun edo bisitatua izatearen probabilitatea ematea da eta hau badago erlazionatuta webgunerainoko lotura daukaten webgune kopuruarekin eta webgune horien garrantziekin. Adibidez ez dauka balio berdina webgune baterako lotura bizilagun elkarte baten blog-etik etortzea ala miloika bisita jasotzen dituen The Times egunkaritik edo Nasa Erakundetik etortzea. Beraz lehenengo iradokizuna da webgune bakoitzaren garrantzi-balioa hara igarotzeko lotura daukaten webgunen garrantzien araberakoa da.

Algoritmoaren ereduak M matrize karratu erraldoiaz ornituta egon beharko da T_i webgune bakoitzagatik errenkada bat eta zutabe bat daukana. Honen $m_{i,j}$ osagaia (i errenkadan eta j zutabean) 0 izango da baldin $T_j \not\to T_i$ loturrik ez badago eta $m_{i,j} \in [0,1]$ tarteko probabilitatea $T_j \to T_i$ lotura dagoenean.

2.1.1 Page-Rank

Internet erabiltzailea internet-eko ibilbidea zoriz hasten bada T webgunean sartzeko probabilitatearen estimazioa $0 \leq P(T) \leq 1$ hurbilduki kalkulatu nahi dugu. Munduan miloika webgune

daude eta estimazio hori balioztatzeko garrantsitsuak diren bi zenbaki hauek dira, T-rako lotura eskeintzen duten webgunen kopurua eta T-tik ateratzen diren lotura kopurua. Helburua betetzeko T-ri elkartutako ondoko formulako "*page-rank*" balioa disenatu zen,

$$\lambda PR(T_j) = \frac{1-d}{N} + d\left(K_{1j}\frac{PR(T_1)}{C(T_1)} + \cdots + K_{nj}\frac{PR(T_n)}{C(T_n)}\right), \tag{2.1}$$

non parametroen balioak ondoko hauek diren,

N	=	indexatutako webgune guztien kopurua,
n	=	T-rako loturarekiko webgunen kopurua,
$PR(T_i)$	=	T-rako loturarekiko i. webgunearen page-rank-a,
$C(T_i)$	=	T_i webgunetik lotura kopurua,
K_{ij}	=	1 baldin T_i-tik T_j-rako loturarik badago (bestela $K_{ij}=0$)
d	=	(0,1) tarteko magultasun parametroa,
λ	=	(0,1) proporzionaltasun konstantea.

Formula honek zenbait ezaugarri interesgarri ditu:

- Webgune bakoitzak $PR(T)$ berezia dauka eta hara doazen loturen webgunen $PR(T_i)$-ren menpekoa da.
- Webgune baten eragina handik lotutako webgunen gainean lotura kopuruaren menpekoa da alderantzizko erlazioaz, hainbat eta lotura gehiago, orduan eta eragin edo webgunea bisitatzeko probabilitate txikiagoa.
- Webgunera iristen diren lotura guztiak haren $PR(T)$ balioa handitzen dute.
- Webgune batetik irtetzen diren loturak beste webguneen balioak handitzen dute eta aldiberean haiek lotzen duten webgunen balioa handitzen dute. Internet grafoaren egitura dauka webguneak erpintzat eta loturak hertzatzat portatuz, beraz webgunearen $PR(T)$-a webgune guztien eragina jasotzen du.
- Webgune guztien "*page-rank*" puntuazio bektorea $p = (PR(T_1), \ldots, PR(T_N)) \in \mathbb{R}^N$ dago eta haren balioek ondoko berdintza matriziala beteko dute.

$$\lambda \begin{pmatrix} PR(T_1) \\ \vdots \\ PR(T_N) \end{pmatrix} = \begin{pmatrix} \frac{1-d}{N} \\ \vdots \\ \frac{1-d}{N} \end{pmatrix} + d \begin{pmatrix} \frac{K_{11}}{C(T_1)} & \cdots & \frac{K_{n1}}{C(T_n)} \\ \vdots & \ddots & \vdots \\ \frac{K_{1n}}{C(T_1)} & \cdots & \frac{K_{nn}}{C(T_n)} \end{pmatrix} \begin{pmatrix} PR(T_1) \\ \vdots \\ PR(T_N) \end{pmatrix} \tag{2.2}$$

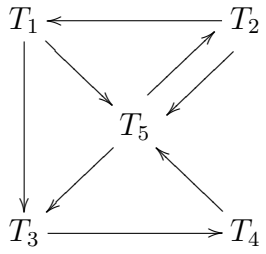
Aurreko ekuazioaren matrizea orokorrean ez da simetrikoa izango grafo norabideatuari dagokiolako. Baina matrize estokastikoa da osagaiak $0 \leq k_{ij}/C(T_i) \leq 1$ direlako eta zutabe bakoitzaren osagai batura 1 delako.

Egoera errazena $d = 1$ hartzean datza eta $PR(T)$ balio logikoen bektorea aurkitzeko problema edo eraberean aurreko ekuazio bektoriala autobalioko eta autobektoreko problema bihurtuko da,

$$\lambda \cdot p = \mathrm{A} \cdot p \tag{2.3}$$

Honek soluzioa (λ, p) da non λ autobalioa $\det(\mathrm{A} - \mathrm{I}) = 0$ ekuazio karakteristikoaren soluzioa den eta p bektorea λ-ri dagokion autobektorea.

Ikus dezagun ondoko adibidea 5 webgunekin eta $d = 1$ parametroarekin,



$$\begin{pmatrix} PR(T_1) \\ \vdots \\ PR(T_N) \end{pmatrix} = \begin{pmatrix} 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 \\ 1/2 & 1/2 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} PR(T_1) \\ \vdots \\ PR(T_N) \end{pmatrix} \tag{2.4}$$

Ereduaren erabilgarritasuna bermatzeko aurreko planteamenduan galdera garrantsitsu honen erantzuna falta zaigu. Ba al dago planteatutako autobalio problemaren soluziorik? Zorionez ereduaren baldintzapetan Perron-en eta Frobenius-en teoremek soluzioaren existentzia ziurtatzen dute, bai λ autobalioa eta hari elkartutako autobektorea p. Noski, interesatzen zaigu osagai positibodun autobektoreak, $p > 0$, probabilitatea adierazten dutelako.

Teorema 2.1.1 *(Perron 1907)*

Demagun $a_{ij} > 0$ osagai positibodun $\mathbf{A}$ matrizea, orduan existituko da $\lambda_1 > 0$ autobalioa, $\lambda_1 > |\lambda_i|$, $\forall i > 1$, non honi elkartutako autobektorea, $\mathbf{A}p = \lambda p$, $p > 0$ positiboa den eta $\mathbf{A}$-ren beste autobektore positibo guztiak beronen multiploak diren.

Teorema 2.1.2 *(Frobenius 1908-1912)*

Demagun $a_{ij} \geq 0$ osagai ez negatibodun $\mathbf{A}$ matrize laburtezina, orduan baldin $\mathbf{A}$ existituko da $\lambda_1 > 0$ autobalio positibo eta bakuna, $\lambda_1 \geq |\lambda_i|$, $\forall i > 1$, gainera norma bereko beste k autobalio badaude, orduan $x^k - \lambda^k = 0$ ekuazioaren erroak izango diren. Autobalio honi elkartutako autobektorea, $\mathbf{A}p = \lambda p$, $p > 0$ positiboa den eta $\mathbf{A}$-ren beste autobektore positibo guztiak beronen multiploak diren.

Gurentzat benetan erabilgarria Frobenius-en teorema izango da, Google-ko matrizea sarrera nuluak asko izango baititu, $a_{ij} = 0$. Autobalio nagusiari elkartutako autobektorea $p \in \mathbb{R}^N$ $\{T_i\}_{i=1}^N$ webgunen puntuazio kandidatotzat har daiteke.

$\mathbf{A}$ matrize laburtezina izateak ondokoa esan nahi du, ez da existitzen errenkada zein zutabe permutaziorik $\mathbf{A}$ matrizea ondoko egitura bihurtzen duena, non azpimatrizeak karratuak diren,

$$\left(\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{0} & \mathbf{A}_{22} \end{array}\right) \tag{2.5}$$

Hiru dimentsiotako espazioan Perron-en teoremaren frogapen erraza $\alpha(p) = \mathbf{A}p/\|\mathbf{A}p\| \in \mathbb{R}^3$ aplikazioa definitzean datza $\mathbb{R}^3$-ko oktante positiboa, $x \in \mathbb{R}^3$, $x \geq 0$, oktante positiboko 1 luzeradun bektoreen espaziora bidaltzen duena. Bereziki $D = \{x \in \mathbb{R}^3, \|x\| = 1\}$ espazioko bektoreen irudiak bertan daude eta jarraitua da, beraz Brouwer-en puntu finkoaren teoremaz $\alpha(\overline{x}) = \mathbf{A}\overline{x}/\|\mathbf{A}\overline{x}\| = \overline{x}$ existitzen da eta autobalioa $\lambda = \|\mathbf{A}\overline{x}\|$ da $\mathbf{A}\overline{x} = \|\mathbf{A}\overline{x}\|\overline{x}$ delako.

2.1.2 Autobalioak konputatzeko berredura metodoa

Aztertu behar dugu problemaren askapen praktikoa, hau da, autobalioko eta autobektoreko problemaren soluzioa. Hau konputazionalki berredura-metodoaz egingo dugu. Oinarri teorikoa autibalio nagusiaren tamainuarekin kan egitean datza. Demagun $\mathbf{A}$ matrize diagonalizagarria dela eta autobalio nagusi bakarra existitzen dela, $\lambda_1 > |\lambda_2| > \cdots > |\lambda_N|$. Matrizearen $\{p_1, \ldots, p_N\}$ autobektoreek $\mathbb{R}^N$-ko oinarria osatuko dute eta edozein $p_0 \in \mathbb{R}^N$ honela adieraz daiteke,

$$p_0 = a_1 p_1 + a_2 p_2 + \cdots + a_N p_N. \tag{2.6}$$

Orain p_i guztiak $\mathbf{A}$-ren autobektoreak direnez, $\mathbf{A}p_0 = a_1\lambda_1 p_1 + a_2\lambda_2 p_2 + \cdots + a_N\lambda_N p_N$ eta aldiberean $(1/\lambda_1)\mathbf{A}p_0 = a_1 p_1 + a_2(\lambda_2/\lambda_1)p_2 + \cdots + a_N(\lambda_N/\lambda_1)p_N$ eta prozesua zenbait aldiz errepikatzen badugu, $(1/\lambda_1)^m\mathbf{A}^m p_0 = a_1 p_1 + a_2(\lambda_2/\lambda_1)^m p_2 + \cdots + a_N(\lambda_N/\lambda_1)^m p_N$ da. Azken adierazpen honetatik batugai gehienak 0-rantza hurbilduko dir lehenengoa ezik,

$$\lim_{m\to\infty} \frac{1}{\lambda_1}^m \mathbf{A}^m p_0 = a_1 p_1. \tag{2.7}$$

Nahikoa da edozein $0 \neq p_0 \in \mathbb{R}^N$ bektorea aukeratzea eta begiratzea $(\mathbf{A}^m p_0)_i/(\mathbf{A}^{m-1}p_0)_i \approx \lambda_1$ balio bat hurbiltzen duten $\forall i = 1, \ldots, N$ guztietarako m handia denean. Orduan λ_1 bilatutako autobalio nagusia izango da eta $a_1 p_1 \approx (1/\lambda_1)^m \mathbf{A}^m p_0$ izango da. Amaieran $a_1 p_1/\|a_1 p_1\|_1$ bilatutako autobektorea webgune bakoitza bisitatzeko probabilitatearekin.

2.1.3 Adibidea. Apostuetako eredua.

Zenbait ekitaldi antolatzeko orduan, adibidez, txapelketak, apostuak, produktu-salmentak edo abesti-zerrendak, besteak beste, puntuaketa eta sailkapen erizpide eraginkorrak aurkitzea garrantzitsua da. Erizpide eraginkorren kontzeptuaz ondokoa adierazi nahi dugu, elementuen maila eta lehiatzeko gaitasuna aurretik balioztatzen duen neurri fidagarrienaz aukeratzea. Gaur egun kinielak edota apostuak oso ohizko jolasak dira eta arrakasta handiago izango lortuko dugu baldin emaitzak aurretik susmatzeko eredua aurkitzen badugu. Horregatik kirol talde baten arrakasta aurrez iragartzeko ez da nahikoa bere puntu pilaketa eta sailkapena ezagutzea, beste talden aurka lortutako emaitzaz ohartzea baizik. Aurkarit indartxuaren aurrean lortutako talde baten emaitz positiboak, nahiz eta beste talden aurkako norgehiagotan jarrera irregularra erakutsi, agian irabasteko probabilitate handiago emango dio. Problema honetan berriz talde bakoitzaren balorazioa beste taldeen balorazioaren menpean egongo da eta autobalioko eta autobektoreko problema baino ez da. Kasu honetan ereduaren erronka matrizearen osaigaiak zehazteko erizpidea da. Adibidez i. taldearen balorazioa $B(T_i)$ ondoko batukariaren emaitzaz finka dezakegu,

$$B(T_i) = \frac{g_{i1}}{k_i} B(T_1) + \cdots + \frac{g_{ii-1}}{k_i} B(T_{i-1}) + \frac{g_{ii+1}}{k_i} B(T_{i+1}) + \cdots + \frac{g_{in}}{k_i} B(T_n), \tag{2.8}$$

non g_{ij} zenbaki osoa T_i taldeak T_j taldearen aurka lortutako garaipen (edo puntu) kopurua den eta k_i zenbakia T_i taldeak jolastutako norgehiagoka kopuru osoa den.

1.4 JPG irudi murrizketa

Bideokamara eta argazki-kamara digitalak agertu zirenetik irudiak ateratzea erraza eta merkea da. Ondorioz gaur egun guztiok irudiak grabatu eta elektronikoki bidaltzen ditugu edozein motako egoera zein gertaera dokumentatzeko eta orioitzeko. Baina dinamika horrek arazoak sortuko ditu, irudi digitalak kolore ezberdinetako milaka puntu edo pixeletaz osatutak daudelako eta informazio gordin hori gordetzea asko okupatzen duelako eta bidalketa prozesua motelegia izan daitekelako. Horregatik irudiak egokitzeko eta gordetzeko modu ekonomikoa aurkitzea funtsezkoa da. Prozesu horretan irudien ezaugarri nabarienak edo behintzat gizakien begiek igartzen dituenak mantentzea bermatu behar da. Helburu horrekin aspaldi batean formulazio matematikoan oinarritutako "*jpeg*" formatua garatu zuten. Teknologia honek Fourier-en transformatuaren bidez informazio digitala funtzio periodikoen koefizienteen moduan adierazten du.

1.4.1 Irudi-digitalizazioa

Demagun laukizuzen antzako irudi bat, hau da $[a,b]\times[c,d]\in\mathbb{R}^2$. Laukizuzenaren koordenatu bakoitza pixel batetaz okupatuta dago eta bere kolorea RGB (red, green, blue) indizeaz neurtzen da. Orain horizontalki eta bertikalki laukizuzena zatitu egiten da $n\times m$ tamainuko laukizuzentxoetan non $a=a_0<a_1<\cdots<a_n=b$ eta $c=c_0<c_1<\cdots<c_m=d$ diren eta $[a_i,a_{i+1}]\times[c_j,c_{j+1}]$. Laukizuzen txiki hauetan koloreak gehiegi ez direla aldatzen pentsa dezakegu. Nola hiru kolore hauetako bakoitzaren maila 1-etik 256-ra neur daiteken, konbinazio kopurua edo kolore kopurua 256^3 da baina 3 byte-z adieraz daitezke. Beraz, $\mathbb{R}^2$-ko irudia adibidez $n\cdot m$ pixel-etan digitalizatuta badago eta pixel bakoitzaren kolorea 3 byte-taz zehaztuta badago, guztira $3nm$ byte beharko ditugu. Guk aurkeztutako jpeg konpresio-algoritmoaz 300.000 byte-ko irudia 35.000 byte-ko artxiboan gorde daiteke doitasun handia mantenduz.

Lehenengo urratsa koordenatu aldaketa eragitea da irudi askoetan korrelazio handia dagoelako ikusitako hiru kolore nagusi horien hartean, beraz (gorria, berdea, urdina) erabili ordez, (argitasuna, tonalitatea$_{urdin}$,tonalitatea$_{gorri}$) erabiltzen da, (Y,C_b,C_r).

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1.1}$$

Gero irudia $8\times 8=64$ pixeletako sortatan bananduko dugu eta pixel horien kolorea zehazten duen koordenatu bakoitzarekin 8×8 matrizea eraikiko dugu, ondorioz sorta bakoitzagatik 3 matrize lortuko ditugu. Prozesua sinplifikatzeko bakarrik Y koordenatuari dagokion matrizea har dezakegu, gris tonoari dagokiona hain zuzen. Hau gertatzen da argazkia zuribeltzez gordetzen dugunean.

1.4.2 Funtzio espazioaren oinarria

Helburua da 8×8 pixeletako karratuaren zenbakizko informazioaren zati nabariena era merke batean gordetzea. Horretarako lerro bakoitzean dauden 8 zenbakiak funtzio jarraituaren baliotzat hartuko ditugu, horren neurri txikiko karratuaren barruan kolore edo tonalitate aldaketa handirik ez dagoelakoan. Balio horiei dagokien funtzioa hurbiltzeko funtzio trigonometrikoko oinarria darabilgu batugai gutxiekin oso hurbilketa zehatzak lortzen baitira eta gainera kalkuluko prozesua arinagoa izateko Fourier-en transformatu azkarra erabili daitekelako. Funtzio jarraitu eta periodikoen L luzeradun eta x aldagai espazialeko espazioaren oinarria $\sin(2\pi kx/L)$ eta $\cos(2\pi kx/L)$ dira. Nola $\sin(x+\pi/2) = \cos x$ den eta $L = 8$ den, eragiketa batzuk eginez $[0, L]$ tartean funtzioa hurbiltzeko soilik sin edo cos funtziopeko garapena erabil daitekela nahikoa da $\varphi_k(x) = \cos(\pi k(x+1/2)/L)$. F funtzioaren Fourier-en serie diskretua $\{x_0, \ldots, x_7\}$ puntuetan,

$$\hat{F}_{\omega_j} = \frac{1}{2}\sum_{n=0}^{7} C_j F(x_n) \cos\frac{\pi(2x_n+1)\omega_j}{16}, \qquad C_0 = 1/\sqrt{2}, \quad C_j = 1. \tag{1.2}$$

Eraberean F funtzioaren balioak errekuperatzeko Fourier-en transformatuaren bidez,

$$F(x_i) = \frac{1}{2}\sum_{\omega_j=0}^{7} C_j \hat{F}_{\omega_j} \cos\frac{\pi(2x_i+1)\omega_j}{16}. \tag{1.3}$$

Funtzio hauek ortogonalak dira,

$$\langle \varphi_j, \varphi_k \rangle = \sum_{n=0}^{7} \cos\frac{\pi(2x_n+1)\omega_j}{16} \cos\frac{\pi(2x_n+1)\omega_k}{16} = 0. \tag{1.4}$$

Eraberean, behin Fourier-en transformatua x ardatzean garatu den, prozesu bera errepika daiteke y ardatzean. Horrela balioak 8×8 karratuan hartzen dituzten funtzioen oinarria lortuko dugu,

$$\varphi_{jk}(nm) = \cos\frac{\pi j(2n+1)}{16} \cdot \cos\frac{\pi k(2m+1)}{16}. \tag{1.5}$$

Bidimentsioko Fourier-en transformatuaren koefizienteak $x_n = n$ eta $y_m = m$ direnean hauxek dira,

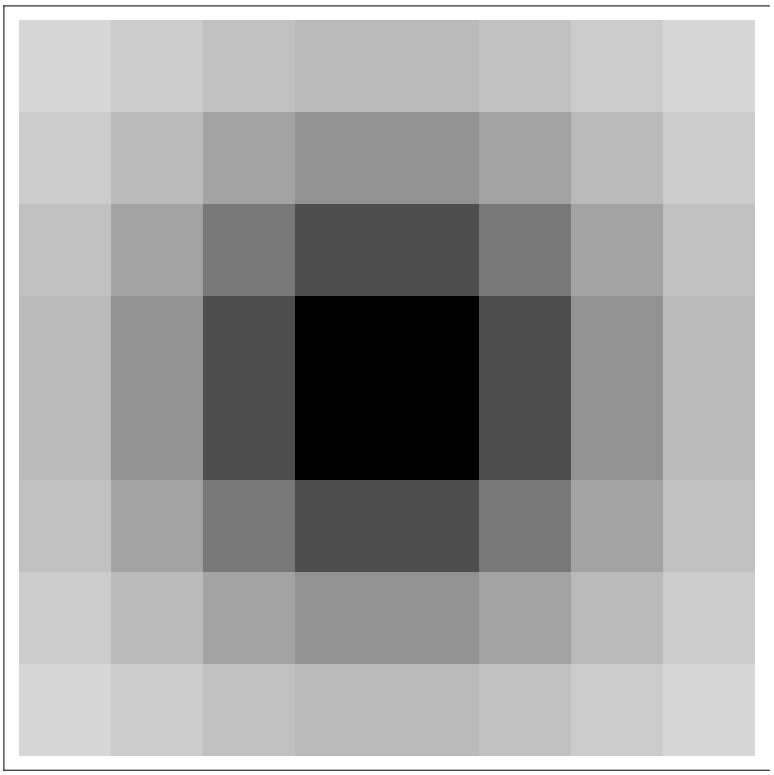
$$\hat{F}_{jk} = \frac{1}{4}\sum_{n,m=0}^{7} C_j C_k F(x_n, y_m) \cos\frac{\pi j(2n+1)}{16} \cdot \cos\frac{\pi k(2m+1)}{16}. \tag{1.6}$$

Orain koefiziente guzti horiek hasierako datuak gordetzen genituen dimentsioko matrizean gordeko ditugu,

(1.7)

F_{00}	F_{10}	F_{20}	F_{30}	F_{40}	F_{50}	F_{60}	F_{70}
F_{01}	F_{11}	F_{21}	F_{31}	F_{41}	F_{51}	F_{61}	F_{71}
F_{02}	F_{12}	F_{22}	F_{32}	F_{42}	F_{52}	F_{62}	F_{72}
F_{03}	F_{13}	F_{23}	F_{33}	F_{43}	F_{53}	F_{63}	F_{73}
F_{04}	F_{14}	F_{24}	F_{34}	F_{44}	F_{54}	F_{64}	F_{74}
F_{05}	F_{15}	F_{25}	F_{35}	F_{45}	F_{55}	F_{65}	F_{75}
F_{06}	F_{16}	F_{26}	F_{36}	F_{46}	F_{56}	F_{66}	F_{76}
F_{07}	F_{17}	F_{27}	F_{37}	F_{47}	F_{57}	F_{67}	F_{77}

Adibide batean ikus dezakegu 8×8 zuribeltzezko pixeletan marraztutako irudi baten parematroak, hau da soilik rgitasuneko parametroak $Y \neq 0$ dauzkana eta beste biak $C_b = C_r = 0$ analisia sinplifikatzeko,



Irudia 1.1: Irudi pixelatua.

$$Y = \begin{pmatrix} 35.79 & 45.33 & 55.14 & 61.82 & 61.82 & 55.14 & 45.33 & 35.79 \\ 45.33 & 61.82 & 81.6 & 97.14 & 97.14 & 81.6 & 61.82 & 45.33 \\ 55.14 & 81.6 & 120. & 156.9 & 156.9 & 120. & 81.6 & 55.14 \\ 61.82 & 97.14 & 156.9 & 226.7 & 226.7 & 156.9 & 97.14 & 61.82 \\ 61.82 & 97.14 & 156.9 & 226.7 & 226.7 & 156.9 & 97.14 & 61.82 \\ 55.14 & 81.6 & 120. & 156.9 & 156.9 & 120. & 81.6 & 55.14 \\ 45.33 & 61.82 & 81.6 & 97.14 & 97.14 & 81.6 & 61.82 & 45.33 \\ 35.79 & 45.33 & 55.14 & 61.82 & 61.82 & 55.14 & 45.33 & 35.79 \end{pmatrix} \tag{1.8}$$

eta horri dagokion Fourier-en transformatu diskretuaren koefizienteak

$$\begin{pmatrix} 720.1 & 0 & -259.6 & 0 & 20.54 & 0 & -9.748 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -259.6 & 0 & 156.6 & 0 & -27.44 & 0 & 7.158 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20.54 & 0 & -27.44 & 0 & 11.02 & 0 & -2.64 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9.748 & 0 & 7.158 & 0 & -2.64 & 0 & 0.8812 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.9}$$

Gizakiaren begiek ez dute maiztasun handiko uhin optikoek ongi bereizte, izan ere, irudiaren kolore eta tonoak pixkanaka aldatzen direnean, orduan badaukagu gaitasuna ezberdintasunak hautemateko. Berriz, aldaketak oso bapatekoak direnean, orduan oharkabe geratuko dira. Portaera fisiologiko horiek askotan analizatu eta aprobetxatu izan dira iragarki subliminalak zein desafio optikoak plazaratzeko.

Gure begiek gehien hautematen duten irudiaren ezaugarriak nabarmentzeko eta horrekin batera irudiaren fitxategi digitalizatua tamainu askoz txikiagoko fitxategia bihurtzeko, aurreko Fourier-en cosinuko transformatu diskretuko matrizearen osagaiak pisu bereziaz modulatuko ditugu, bereziki $\hat{F}_{jk}/(\alpha(s)q_{jk})$. Gainera lortutako zenbakiak adierazpen osoen moduan borobilduko ditugu, $\texttt{Round}[\hat{F}_{jk}/(\alpha(s)q_{jk})]$, haiek osatzen duten matrizeak byte gehiegi ez okupatzeko, zenbaki erreal batek osoa baino askoz oroimen gehiago behar duelako. Aurreko formulazioan $\alpha << 1$ koefizientea txikia hartuko dugu ($s >> 1$) irudiko artxiboaren murrizketa txikia nahi dugunean eta konpresio arina eta $\alpha >> 1$ murrizketa zorrotza ($s \sim 1$) lortu nahi dugunean. Adibidez α-ren ondoko balioak har daitezke $1 \leq s \leq 100$

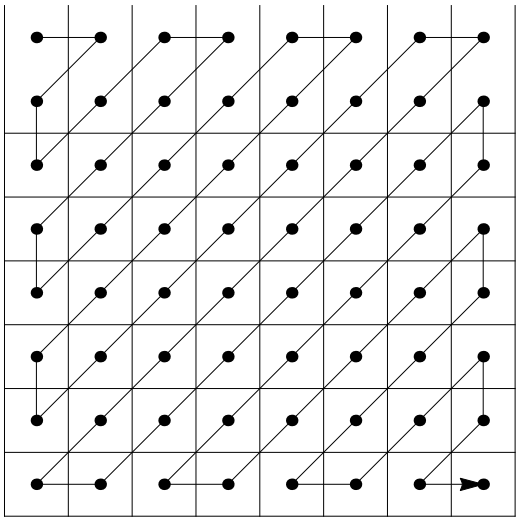
$$\alpha(s) = \begin{cases} 50/s, & 1 \leq s \leq 50 \\ 2 - s/50, & 50 < s \leq 100 \end{cases} \tag{1.10}$$

Eraberean Y argitasunerako eta C tonalitaterako erabiltzen diren $Q_Y = (q_{jk})$ eta $Q_C = (\overline{q}_{jk})$ ohiko koefiziente matrizeak antza honetakoak izan daitezke,
(1.11)

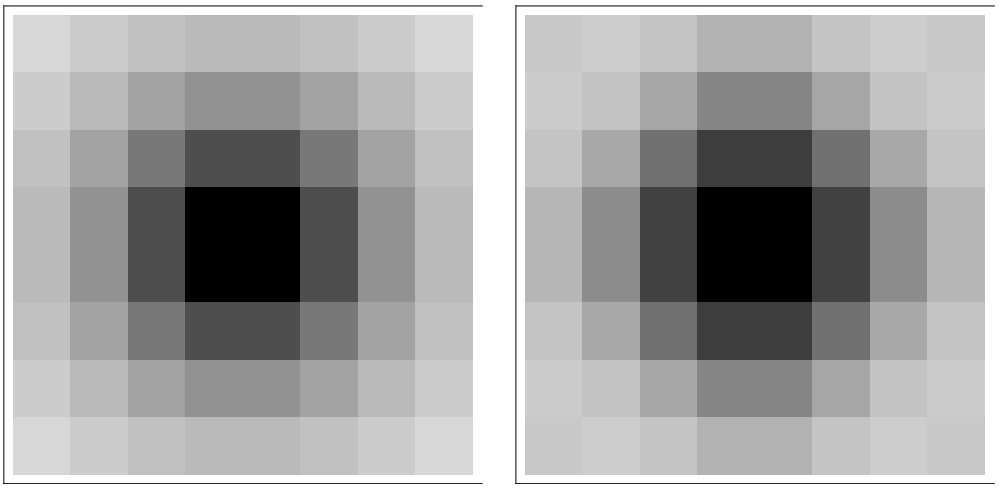
$$Q_Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Adibidez Fourier-en koefizienteen matrizeari aplikatutako $\alpha(30)$ zorroztasun ertaineko murrizketak ondoko matrizea itzuliko du eta haren informazioa bektore arruntaren moduan gordetzen badugu ondoko irudian ikusten den osagaien ordena hartuz, orduan 64 byte erabili ordez, soilik $(720, 0, 0, -267, 0, -257, 0, 0, 0, 0, 40, 0, 160, 0, 30)$ 15-tan gorde dezakegu ondorengoak 0 baitira

$$\begin{pmatrix} 720 & 0 & -267 & 0 & 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -257 & 0 & 160 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Adibidean geratu zaigun 15 osagaiko bektorean gordetako informaziotik irudia berreskuratzen badugu, ondoko irudiaren eskubiko zatia lortuko dugu eta konpara dezakegu ezkerrekoarekin, hasierakoa hain zuzen,



Irudia 1.2: Irudi pixelatua eta konparaketa irudi murriztuarekin.

JPEG2000 irudi murrizketa

JPEG irudi murrizketa arruntak bi eragozpen ditu, lehenengoa 8×8 lauki bakoitzaren mugetako pixelen balioak da ez direla hartzen inguruko laukien menpean edo jarraitutasun baldintzak gordetzeko, eta bestea da irudia berreskuratzekotan multzo osotzat hartu behar dugula eta sare informatikotik jeitsi behar badugu eta konexioa astiro badabil orduan ezin da doitasun gutxiko aurreirudia erakutsi. Eragozpenak gainditzeko ilara edo errenkada bakoitzeko pixelak binaka hartuko ditugu, $(f_0, f_1), (f_2, f_3), \ldots, (f_{n-1}, f_n)$ ta onnoko transformazio linealaz aldagai sorta berria bihurtzen da, $(a_0, a_1), (a_2, a_3), \ldots, (a_{n-1}, a_n)$. Azken hau wavelet koefizienteak izendatu dira,

$$
\begin{array}{rcl@{\qquad}c@{\qquad}rcl}
a_0 &=& (f_0 + f_1)/2 & & f_0 &=& a_0 + a_1 \\
a_1 &=& (f_0 - f_1)/2 & & f_1 &=& a_0 - a_1 \\
a_2 &=& (f_2 + f_3)/2 & \Longleftrightarrow & f_2 &=& a_2 + a_3 \\
a_3 &=& (f_2 - f_3)/2 & & f_3 &=& a_2 - a_3 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots
\end{array}
\tag{1.12}
$$

Koefiziente multzo berriaren azpiindize bikoitikoek jatorrizko pixelen batezbestekoa gordeko dute eta aldiz bakoitikoek pixelen aldea edo aldaketa. Beraz aldaketa leuneko edo maiztasun txikikoak bikoitietan geratuko dira eta horiek dira gure begiek gehien bereizten dituztenak eta aldaketa maiztasun handikoak bakoitietan geratuko dira. Eragiketa berdina errepikatzen badugu zutabeekin a_{ij} koefizienteak bi dimentsioetan kalkulatuko ditugu eta lau multzoetan gorde ditzakegu, bakoitzak hasierako pixel osoen multzoaren tamainuaren laurdena okupatzen. Bi azpiindize bikoitiak dituztenek $\mathrm{LL}^{(1)}$ (Low-Low frequency) multzoa osatuko dute, lehenengoa bikoitia eta bigarrena bakoitia dituztenek $\mathrm{LH}^{(1)}$, (Low-High frequency), eta azken bi multzoak

hurrenez hurren $\mathrm{HL}^{(1)}$ eta $\mathrm{HH}^{(1)}$ multzoetan. Irudiaren informazioa 4 multzo horietan bidaliko da, lehenengoz LL, eta irudiaren aurreikuspena berreskuratzeko aukera emango digu. Gainera LL multzoarekin prozesua errepika daiteke 4 azpimultzotan bananduz eta lehenbizian soilik $\mathrm{LL}^{(2)}$ bidaliz.

Mugetako ezjarraitasunaren eragozpena gainditzeko aldameneko laukiaren mugetako pixelen informazioarekiko batezbestekoa kalkulatuz egin daiteke,

$$\begin{array}{rcl} a_0 & = & -\frac{1}{8}f_{-2} + \frac{1}{4}f_{-1} + \frac{3}{4}f_0 + \frac{1}{4}f_1 - \frac{1}{8}f_2 \\ a_1 & = & -\frac{1}{2}f_{-1} + f_0 + -\frac{1}{2}f_1 \end{array} \tag{1.13}$$

Ariketak

1.- Aurkitu ze baldintzak bete behar dituen $f(x)$ funtzioak bere Fourier-en serieak funtziorantz konbergentea izateko.

2.- Teorikoki Fourier-en funtzio-oinarria $\Lambda = \{1, \sin x, \sin 2x, \ldots, \cos x, \cos 2x, \ldots\}$ da $x \in [0, 2\pi]$ tartean. Ze aldagai aldaketa eta ze oinarri funtzioak erabiliko dira $x \in [0, N]$ tartean?

3.- Frogatu Λ Fourier-en oinarriko funtzioak ortogonalak direla:

a) Kasu jarraituan: $\frac{1}{\pi}\int_0^{2\pi} \sin mx \cos nx \ \mathrm{d}x = 0$ eta $\frac{1}{\pi}\int_0^{2\pi} \sin mx \sin nx \ \mathrm{d}x = \delta_{nm}$

b) Kasu jarraituan: $\frac{2}{N}\sum_{k=0}^{N-1} \sin\frac{2\pi nk}{N} \cos\frac{2\pi mk}{N} = 0$ eta $\frac{2}{N}\sum_{k=0}^{N-1} \sin\frac{2\pi nk}{N} \sin\frac{2\pi mk}{N} = \delta_{nm}$

4.- Sinuak eta kosinuak erabili ordez *jpeg* prozesuak praktikan $\varphi_j(x_n)\cos\frac{\pi(2x_n+1)\omega_j}{16}$ non $\omega_j = j$, $j = 0, \ldots, 7$. Hemen izendatzailean agertzen den zenbakia 16 da eta ez 8, laukizuzentxo bakoitzaren alde bateko puntu kopurua bezela. Zergatik uste duzu bidezkoa dela oinarri funtzio horiek aukeratzea?

Kapitulua 3

Programazio Matematikoa

3.1 Optimizazioa

Bizitzako ekimen askoen helburua irabaziak hobetzea da eta horren ondorioz maiz agertzen dira kostuak minimizatzea eskatzen duten prozesuak, besteak beste, industrialak, garraioarekin lotutakoak eta komertzialak. Askoetan benetako errealitatea antzematen duen ereduari tresnak matematikoak aplikatuz etekinak lor daitezke. Aurrena jokalekuan agertzen diren elementuak aldagaien edo konstanteen moduan identifikatu behar dira eta ondoren eredu matematikoa eraiki, planteamendu hauei askoetan "logistikako ereduak" esaten zaie. Matematikak badauzka murrizketekiko funtzioaren minimizazio edo optimizazio problemaren soluzioa bilatzeko tresna eraginkorrak. Orokorrean $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ aldagai sortaren menpeko minimizatu (edo maximizatu) behar den $f : \mathbb{R}^n \to \mathbb{R}$ "helburuko funtzioa" eta soluzioak derrigorrez bete beharreko zenbait baldintza agertuko dira. Alde batetik, bai helburuko funtzioa zein murrizketak adierazpen ezlinealak (programazio ezlineala edo mixtoa) edo linealak (programazio lineala) izan daitezke eta bestetik aldagaien eremuaren eskakizuna zenbaki errealak $\mathbb{R}$, zenbaki osoak $\mathbb{Z}$ edo aldagai binarioak $\{0, 1\}$ izan daiteke. Azken erabaki hau problemaren esanahiaren araberakoa da, adibidez, x aldagaiaren esanahia artikulu baten ale kopurua bada, orduan $x \in \mathbb{Z}$ izango da, baina baldin x langile bat lanean ala atsedenaldian adierazten duen aldagaia bada, orduan aukera bikoitzeko aldagaia da, $x \in \{0, 1\}$.

Kapitulu honetan "programazio linealaren eta osoaren" problema aztertuko dugu. Demagun $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ aldagai bektorea eta bete beharreko baldintza edo murrizketa

linealak (x_i-en konbinazio linealak), orduan formulazio orokorra ondokoa da:

$$
\begin{aligned}
& Z = f(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j, && \text{(helburuko funtzioa)} \\
& \mathbf{Ax} = \mathbf{b} \Leftrightarrow \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, && \text{(murrizketak)}, \\
& \mathbf{x} \geq 0 && \text{(positibotasuna)}.
\end{aligned} \tag{3.1}
$$

Batzuetan berdintasun murrizketak agertu ordez, $\sum_{j=1}^{n} a_{ij}x_j = b_i$, ezberdintasun gisa plazaratuko dira, $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, baina orduan nahikoa da helburuko funtzioan ereginik ez daukan magultasun aldagai berri bat sartu, $x_{n+1} \geq 0$, non $\sum_{j=1}^{n} a_{ij}x_j + x_{n+1} = b_i$. Baldin Aldagai bat negatiboa izan beha badu, $x_i \leq 0$, orduan ereduan $\hat{x}_i = -x_i \geq 0$ erabiliko dugu positibotasun baldintza betetzeko. Eraberean $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ ezberdintasun baldintza berdintasun gisa idazteko $\sum_{j=1}^{n} a_{ij}x_j - x_{n+1} = b_i$ adierazpen baliokidea erabiliko dugu.

Problemaren soluzio onargarriari, "oinarrizko soluzioa" esango diogu, hau da $\mathbf{Ax} = \mathbf{b}$ etetzen dutenetako bektore bakoitza. Hori izango da abiapuntua $f(\mathbf{x})$ funtzioa minimizatzen duen $\mathbf{x}$ soluzioa bilatzeko metodo sistematikaoren bidez, simplex edo branch and bound algoritmoaren bidez. Batzuetan aldagai asko badaude (ehundaka ala milaka) algoritmoa oso astiro konbergi daiteke. Badaude progrma informatikoak programazio linealeko problemen soluzioak bilatzen dutenak, LPSolve (dohan), Excel Solver eta MPSolver Foundation, Gurobi, Lingo, etabar.

3.1.1 Programazio lineal binarioko adibidea, Sudokua

Aurreko mendaren 70 hamarkadan lehenbiziko aldiz publikatu zen Sudokuaren problema, hama urte geroago Japon herrialdean ohikoa bihurtu zen eta Europan 2005 urtetik aurrera egunkarietan denbora-pasatzeko jokuaren moduan agertzen hasi zen. Jokuaren oinarria 9×9 zenbaki taula da, non gelaxkak $h \in \{1, \ldots, 9\}$ zenbakietaz osatutak egon daitezke eta murrizketa batzuk bete behar dira:

- errenkada bakolitzean h zenbakia bakarrik behin ager daiteke,
- zutabe bakolitzean h zenbakia bakarrik behin ager daiteke,
- 3×3 gelaxketako karratutxo bakolitzean h zenbakia bakarrik behin ager daiteke.

Problema edo denbora-pasa hauetan zenbaki batzuk finkatutak datoz eta erronka da falta diren gelaxkak osatzea bete beharreko baldintza guztiak autsi gabe. Hainbat eta zenbaki gutxiago orduan eta zailagoa da buruz egitea askatasun maila gehiago daudelako. Hemen dago adibide bat,

				3		8		
		9					4	1
1	7		4		9			
		4				2		
				1		7		
9	2			7	3	6		
			5		1			
3	4							6
2					7			

Problema mota hauek programazio lineal binarioaren moduan idatz daitezke, non aldagaiak x_{ijk} diren eta (i, j) gelaxkan k zenbakia dagoen adierazten duten. Erantzuna baietza bada $x_{ijk} = 1$ izango da, eta bestela $x_{ijk} = 0$. Azpiindizeen existentzi eremua $(i, j, k) \in \{1, \ldots, 9\} \times \{1, \ldots, 9\} \times \{1, \ldots, 9\}$ da, beraz, guztira 729 aldagai dira. Begira dezagun zeintzu diren x_{ijk} aldagaien gainean dauden baldintzak:

1. Gelaxka bakoitzean bakarrik zenbaki bat egon daiteke, $\forall\, i, j = 1, \ldots, 9,\ \exists! k \in \{1, \ldots, 9\}$ non $x_{ijk} = 1$ eta beste $x_{ijk} = 0$. Baldintza hauek murrizketa linealen gisaz idatziko ditugu,

$$\sum_{k=1}^{9} x_{ijk} = 1, \quad \forall\, i = 1, \ldots, 9,\ j = 1, \ldots, 9 \qquad \text{(81 baldintza)}$$

2. Errenkada bakoitzean $h \in \{1, \ldots, 9\}$ digito bakoitza bakarrik behin ager daiteke, $\forall\, i = 1, \ldots, 9,\ \exists! j \in \{1, \ldots, 9\}$ non $x_{ijk} = 1$ eta hau gertatuko da $k = 1, \ldots, 9$ bakoitzerako,

$$\sum_{j=1}^{9} x_{ijk} = 1, \quad \forall\, i = 1, \ldots, 9,\ k = 1, \ldots, 9 \qquad \text{(81 baldintza)}$$

3. Eraberean zutabe bakoitzean $h \in \{1, \ldots, 9\}$ digito bakoitza bakarrik behin ager daiteke, $\forall\, j = 1, \ldots, 9,\ \exists! i \in \{1, \ldots, 9\}$ non $x_{ijk} = 1$ eta hau gertatuko da $k = 1, \ldots, 9$ bakoitzerako,

$$\sum_{i=1}^{9} x_{ijk} = 1, \quad \forall\, j = 1, \ldots, 9,\ k = 1, \ldots, 9 \qquad \text{(81 baldintza)}$$

4. Karratu osoaren barruan dauden 3×3 karratutxo bakoitzean $h \in \{1, \ldots, 9\}$ digito bakoitza bakarrik behin ager daiteke, $\forall\, k \in \{1, \ldots, 9\} \exists! i \in \{1, \ldots, 3\}$ eta $\exists! j \in \{1, \ldots, 3\}$, non $x_{(p+i)(q+j)k} = 1$ eta hau gertatuko da $p = 0, \ldots, 2$ eta $q = 0, \ldots, 2$ bakoitzetarako,

$$\sum_{i,j=1}^{3} x_{(p+i)(q+j)k} = 1, \quad \forall\, p = 0, \ldots, 2,\ q = 0, \ldots, 2 \qquad (81 \text{ baldintza})$$

Guztira $4 \times 81 = 324$ murriztapen dira eta problemaren magultasun graduak 729 dira, beraz murriztapen gehigarririk gabe, hasieran gelaxka guztiak utsik badaude, orduan sudokuak soluzio ugari dauzka. Baina sudokuan finkatutako zenbaki bakoitzarengatik murriztapen berriak sortzen dira. Ohikoa da gelaxka batzuk finkatutak dauzkan karratu batekin hastea non gutxienez oinarrizko soluzio bat dagoen (murriztapenak ez dira bateraezinak).

Amaieran (i, j) gelaxkako zenbakia formula honetaz finkatuko dugu:

$$\sum_{k=1}^{9} x_{ijk} \tag{3.2}$$

Galderak:

1. Zenbat baldintza gehitzen dira (2,3) gelaxkan 9 zenbakia ezartzen badugu, $x_{239} = 1$?
2. Zenbat baldintza gehitzen dira (2,8) gelaxkan 4 zenbakia ezartzen badugu, $x_{284} = 1$?
3. Zenbat baldintza gehitzen dira (3,6) gelaxkan 9 zenbakia ezartzen badugu, $x_{369} = 1$?

PRAKTIKA. Sudokua LPSolve-z eta Excel-ez

Programazio matematikoko eta optimizazio linealeko problemak ebazteko badago tresna informatiko ugari eta horien artean dohakoa den LPSolve erabiliko dugu. Badaude bi aukerak:

- Instalatu LPSolve ingurunea "lp_solve_5.5.2.0_IDE_Setup.exe" fitxategia exekutatuz,
- Deskonprimitu ordenagailuan LPSolve exekutablea "lp_solve_5.5.2.0_exe_win32.zip".

Eguneraketak http://sourceforge.net/projects/lpsolve/files/lpsolve/ webgunean daude.

Aginduak LPSolveren ingurunean

Hiru agindu mota erabiliko ditugu, funtzio helburua eta optimizazio mota ("max" edo "min"), murrizketak eta aldagai mota ("int" osoak badira edo utsik errealak badira). Oharrak "/*oharra*/" ikurren artean idatziko ditugu. Adibidez ondoko 2 dimentsioko problema:

```
/* Objective function */
min:  x1 + x2;
/* Variable bounds */
x1 + x2 <= 4;
4*x1 + x2 >= 4;
-x1 + 2*x2 >= 0;
/* Variable type */
int x1, x2;
```

Soluzioa "Result" lehioan agertuko da, kasu honetan: f(x1,x2)=2, x1=1, x2=1.

Zuzenean MSDos agindu lehiotxotik LPSolve exekutatu nahi badugu, orduan lehenbizian aurkeztutako problema idazkera berberaz textu fitxategian idatziko eta gordeko dugu eta fitxategiari luzapena aldatuko diogu "*.lp" ordez, adibidez "prob1.lp". Gero agindu hau exekutatuko dugu emaitza lehioan bertan plazaratzeko:

```
lp_solve -s prob1.lp
```

edo irteera fitxategi batean gorde nahi badugu, adibidez "prob1sol.txt" beste hau:

```
lp_solve -s prob1.lp > prob1sol.txt
```

Aginduak LPSolve Excel-etik erabiltzeko

LPSolve Excel-etik erabil ahal izateko ondoko bi fitxategiak eskuratu behar dira,

http://faculty.nps.edu/sebuttre/home/Software/Lpsolve/lpSolve.5.7.Xla
http://faculty.nps.edu/sebuttre/home/Software/Lpsolve/lpslink57.dll

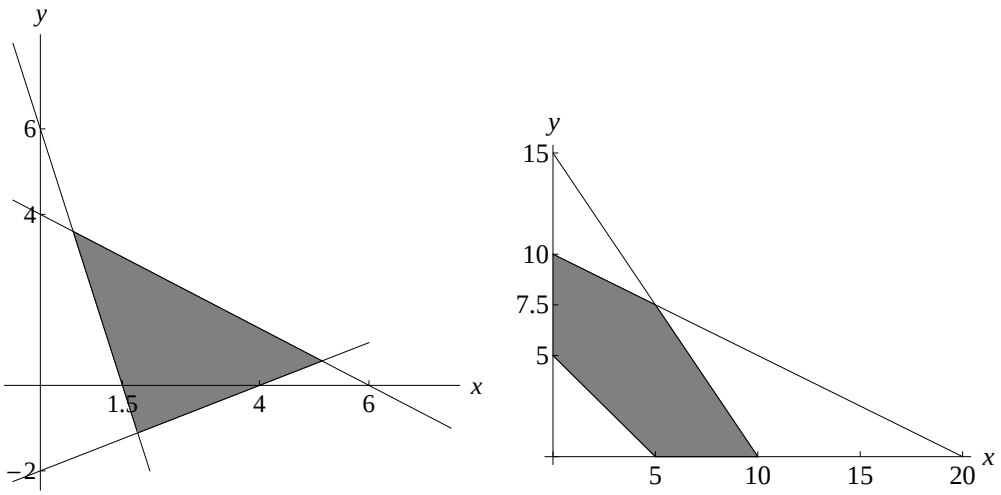
eta jarraitu ondoko urratsak:

1. Gehitu iezaiozu "C:/Archivos de programa/LPSolve IDE" kokapena "PATH" aldagaiari. Aldagai hau ondoko menupean dago:
   Inicio → PanelControl → Sistema → OpcionesAvanz → VariabEntorno → VariabSistema
2. Gorde "lpslink57.dll" fitxategia "c:/Windows/System32" karpetan.
3. Gorde "lpSolve.5.7.Xla" fitxategia ondoko karpetetan "OFFICE##" bertsioaren arabera
   "C:/Archivos de programa/LPSolve IDE/Testdir/"
   "C:/Archivos de programa/Microsoft Office/OFFICE##/ADDINS/"
   eta gehitu osagarria Excel-i goiko ezkerraldean dagoen Excel-en ikonotik menu hauetatik:
   Opciones de Excel → Complementos → Administrar (compl Excel) "Ir" → Examinar.
   Aukeratu ondoko fitxategia eta osagarri hori dagoen lehiotxoa piztuko da:
   "C:/Archivos de programa/Microsoft Office/OFFICE##/ADDINS/ lpSolve.5.7.Xla"

LPSolve optimizazio hiru problema hauek ebazteko aukera ematen du: "Lineal-osoa", "garraioa", esleipena. Baita utsune batzuk agertuko dira LPSolve programari adierazteko kalkulu-orrialdearen zein gelaxketan dauden murrizketen koefizienteak, ezberdintasunak, helburuko funtzioa eta agertu beharreko soluzioaren balioaren eta aldagaien koefizienteen kokapena.

1. Aukeratu "Max"edo "Min".
2. Adierazi helburuko koefizienteek osatzen duten gelaxka-rangoa "Enter Range containing..." utsunean, adibidez B7:D7.
3. Adierazi murrizketen koefizienteek, ezberdintasun zeinuek ("<", "<=", "=", "==", ">", ">=") eta termino konstanteek osatzen duten gelaxka-rangoa "Enter Constraint matrix" utsunean, adibidez B9:F10, murrizketak errenkadaka osatzea aukeratu badugu. Bloke honek aurrekoa baino bi zutabe gehiago izango ditu ezberdintasunen zeinuen zutabeagatik eta termino konstanteen zutabeagatik.
4. Aukeratu aldagai osoak edo binarioak "Enter vector..." utsunean.
5. Adierazi soluzioaren koefizienteek eta helburuko funtzioaren balioak osatuko duten gelaxka-rangoa "Enter location..." utsunean.

Adibidea 3.1.1 *Aurkitu $f(x,y) = y+0.5x$ funtzioaren maximoa eta minimoa irudietan dauden mugatutako eremuetan:*



Adibidea 3.1.2 *LPSolve-ri Sudokuren definizioagatik derrigortzen diren 324 murrizketak betetzen dituen soluzioetako bat eskatzen badiogu, ondoko hau emango digu,*

1	6	3	5	7	2	4	8	9
4	2	5	6	8	9	3	1	7
8	7	9	1	3	4	2	5	6
2	4	1	3	5	6	7	9	8
3	5	7	2	9	8	1	6	4
6	9	8	4	1	7	5	2	3
5	1	4	8	6	3	9	7	2
7	3	6	9	2	1	8	4	5
9	8	2	7	4	5	6	3	1

Baina zein da teorian agertutako sudokuaren soluzioa?

?	?	?	?	**3**	?	**8**	?	?
?	?	**9**	?	?	?	?	**4**	**1**
1	**7**	?	**4**	?	**9**	?	?	?
?	?	**4**	?	?	?	**2**	?	?
?	?	?	?	**1**	?	**7**	?	?
9	**2**	?	?	**7**	**3**	**6**	?	?
?	?	?	**5**	?	**1**	?	?	?
3	**4**	?	?	?	?	?	?	**6**
2	?	?	?	?	**7**	?	?	?

LPSolve eta Excel lotura

Programazio linealeko problema askatzeko LPSolve tresna erabilgarria da eta Excel soluzioa irudikatzeko eta azaltzeko eraginkorra da ere. Beraz, LPSolve-k kalkulatutako problemaren soluzioak Excel-ez adierazteko ondoko urratsak jarraitu:

1. Idatzi problemaren funtzio helburua, murrizketak eta aldagai motaprob.lp textu fitxategian LPSolve-ren idazkera-arauak jarraituz eta programa exekutatu.

2. Ikusi emaitzak "Result" lehioan eta "File" menutik aukeratu "Export Result" →"Objetive to CSV". Aukeratu izen bat, adibidez, "sudokusol.csv" eta sortu fitxategia.

3. Zabaldu "sudokusol.csv" eta menu nagusitik aukeratu "Guardar como" → "libro de Excel habilitado para macros" eta eman "sudokusol.xlsm" izena.

4. "sudokusol.xlsm" fitxategian sartu "Programador" → "visual basic" lehioan eta ezkerreko menuan "VBAProject (sudokusol.xlsm)" aukeratu "Insertar" → "modulo"

5. Orain visual basic-eko lehioan sartuko zarete. Excel-eko gelaxketan dauden dautuak prozesatzeko eta berridazteko itxura berezi batetaz erabili ondoko aginduetako batzukak:

   Sub sudokulehio()
   ' sudokulehio Macro
   Const N As Integer = 9 'errenkada eta zutabe kopurua
   Dim i, j, k As Integer 'zenbakitzaileak
   Dim varbin(1 To N, 1 To N, 1 To N) As Integer 'aldagai matrizea
   ⋮
   Sheets("sudokusol").Activate
   For i = 1 To N
   ⋮
   Next i
   *varbin(i, j, k) = Cells(2 + (i - 1) * N * N + (j - 1) * N + k, 3)*
   If (varbin(i, j, k) > 0.5) Then
   Cells(1 + i, 4 + j) = ...
   End If
   'kokatu emaitzak erdian eta eman kolorea gelaxkeei
   Range("E2:M10").HorizontalAlignment = xlCenter
   Range("E2:M10").Interior.Color = RGB(300, 100, 100)
   'ezker aldea
   Range("E2:E10").Borders(xlEdgeLeft).LineStyle = xlContinuous
   Range("E2:E10").Borders(xlEdgeLeft).Weight = xlThick
   End Sub

6. Exekutatu makroa goiko gezi berdea sakatuz

LPSolve eta Excel lotura

Programazio linealeko problema askatzeko LPSolve tresna erabilgarria da eta Excel soluzioa irudikatzeko eta azaltzeko eraginkorra da ere. Beraz, LPSolve-k kalkulatutako problemaren soluzioak Excel-ez adierazteko ondoko urratsak jarraitu:

1. Idatzi problemaren funtzio helburua, murrizketak eta aldagai motaprob.lp textu fitxategian LPSolve-ren idazkera-arauak jarraituz eta programa exekutatu.
2. Ikusi emaitzak "Result" lehioan eta "File" menutik aukeratu "Export Result" →"Objetive to CSV". Aukeratu izen bat, adibidez, "sudokusol.csv" eta sortu fitxategia.
3. Zabaldu "sudokusol.csv" eta menu nagusitik aukeratu "Guardar como" → "libro de Excel habilitado para macros" eta eman "sudokusol.xlsm" izena.
4. "sudokusol.xlsm" fitxategian sartu "Programador" → "visual basic" lehioan eta ezkerreko menuan "VBAProject (sudokusol.xlsm)" aukeratu "Insertar" → "modulo"
5. Orain visual basic-eko lehioan sartuko zarete. Excel-eko gelaxketan dauden dautuak prozesatzeko eta berridazteko itxura berezi batetaz erabili ondoko aginduetako batzuk:

   Sub sudokulehio()

   Const N As Integer = 9 'errenkada eta zutabe kopurua
   Dim varbin(1 To N, 1 To N) As Integer 'gelaxka kopurua
   Dim vargelaxka(1 To N, 1 To N) As Integer 'gelaxka multzoa

   Sheets("sudoku01").Activate

   ' gelaxka irakurketa zutabetik eta idazkera laukizuzentxoan
   For i = 1 To N
   For j = 1 To N
   If (varbin(i, j) > 0.5) Then
   vargelaxka(i, j) = varbin(i, j)
   Sheets("sudoku01").Cells(i, 4 + j) = vargelaxka(i, j)
   End If
   Next j
   Next i
   End Sub
6. Exekutatu makroa goiko gezi berdea sakatuz
7. Sudokuaren soluzioa idazteko laukizuzena apaintzeko ongi datoz visualbasic-eko agindu batzuk: `Borders(xlEdgeTop)`, `Interior.Color`, `HorizontalAlignment`, `LineStyle = xlContinuous`,...

Kapitulua 4

Fisika Matematikoa

4.1 Sarrera

Fisikako fenomeno gehienen portaera ekuazio matematikoen bidez adieraz daitezke. Hainbat eta errealitatearekin lege fisiko zilegiagoak eta neurri experimentalek emandako konstante fisiko zehatzagoak izan orduan eta eredu matematiko hobeagoak eraikiko ditugu. Eredu horietariko askok ekuazio diferentzial arrunten zein deribatu partzialetakoen menpean adieraziko dira. Oso maiz agertzen dira naturaren prozesuekin lotutak dauden "*kontserbazio legeak*". Hauek energia, momentua, momentu angeluarra, bolumena, etab kopuruen kontserbazioarekin erlazionatutak daude eta nolabait ekuazioen barruan agertu behar dira. Fenomeno ezagunenetako bat beroaren ekuazioa da, honen eraikuntzan agertzen diren lege fisikoak nahiko intuitiboak direlako

4.2 Beroaren ekuazioa

Demagun L luzeradun burnizko barra bat eta $u(x,t)$ funtzioa t unean eta x kokapenean ematen duen barraren temperatura,

$$u : \Omega \times [0,T] \longrightarrow \mathbb{R}, \qquad \text{non} \quad \Omega = [0,L] \quad \text{den.} \tag{4.1}$$

Barraren $[a,b] \subset \Omega$ segmentuan dagoen bero kopurua t unran ondoko integralaren bidez neur daiteke,

$$\int_a^b u(x,t)\ \mathrm{d}x. \tag{4.2}$$

Barraren $[a,b] \subset \Omega$ segmentuan bero fluxu norabideatua badago, $\varphi(x,t)$, orduan tarte horretan pilatutako beroaren denborarekiko aldaketa izkinetatik sartu eta ateratzen den fluxu kopurua-

ren diferentzia izango da,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_a^b u(x,t)\ \mathrm{d}x = \varphi(a,t) - \varphi(b,t) = -\int_a^b \varphi_x(a,t)\ \mathrm{d}x. \tag{4.3}$$

Nola $\mathrm{d}/\mathrm{d}t \int_a^b u(x,t)\ \mathrm{d}x = \int_a^b u_t(x,t)\ \mathrm{d}x$ har dezakegunez eta $[a,b]$ tartea nahi dugun neurrikoa ere, adibidez $[x, x+\Delta x]$, orduan $\Delta x \to 0$ limitea hartuz ondoko kontserazio legea lortuko dugu,

$$u_t(x,t) + \varphi_x(a,t) = 0, \qquad \forall (x,t) \in \Omega \times [0,T]. \tag{4.4}$$

Beste aldetik badago joera bero handiago egiten duen tokietatik hotzagoak daudenetara bero fluxu norabideatua egotea. Fluxu hau proportzionala izango da tenperatura aldearekin x norabide espazialean, ∇_x gradientea hain zuzen. Noski, tenperatura alde handia badago tarte txikian, gune horretan bero transferentzi handiagoa egongo da. Formulan ere D materialak daukan difusiok (eroankortasuna) parte hartuko du

$$\varphi(x,t) = -Du_x(x,t). \tag{4.5}$$

Hau da Fourier-en termodinamikako legea. Azken bi ekuazio hauek lotuz deribatu partzialen menpeko beroaren ekuazio diferentziala lortuko dugu:

$$u_t(x,t) - Du_{xx}(x,t) = 0, \qquad \forall (x,t) \in \Omega \times [0,T]. \tag{4.6}$$

Ekuazio honen hedapen bat bero iturria kontutan hartzea da x puntu bakoitzean eta denbora eta tenperatura beraren menpekoa, orduan ekuazioa hau izango da,

$$u_t(x,t) - Du_{xx}(x,t) = f(x,t,u), \qquad \forall (x,t) \in \Omega \times [0,T]. \tag{4.7}$$

Erlazio hau hasierako baldintzarekin batera, $u(x,0) = u_0(x)$, eta mugetako baldintzekin batera $u(0,t) = r$, $u(L,t) = s$ Cauchy-ren problema da,

$$\begin{cases} u_t(x,t) - Du_{xx}(x,t) = f(x,t,u), \qquad \forall (x,t) \in \Omega \times [0,T]. \\ u(x,0) = u_0(x) \\ u(0,t) = r,\ u(L,t) = s \end{cases} \tag{4.8}$$

4.2.1 DPE-an zenbakizko soluzioa. Diferentzia finituak.

Deriatu partzialen soluzioan zenbakizko hurbilketak kalkulatzeko teknika arrunta diferentzia finituak dira eta horren oinarria funtzioaren deribatuak oso gertu dauden bi puntuetan kalkulatutako funtzio beronen balioen aldeaz hurbiltzea da. Hasieran espazioa zatitu edo diskretizatu egiten da, $0 = x_0 < x_1 < \cdots < x_N = L$, non $x_{n+1} - x_n = \Delta x$ eta $0 = t_0 < t_1 < \cdots$, non

$t_{k+1} - x_k = \Delta t$. Funtzioaren balioen hurbilketak diskretizazioaren puntuetan honela idatziko ditugu $u_n^k \approx u(x_n, t_k)$ eta deribatu partzialen hurbilketen adierazpenak ondoko hauxek dira,

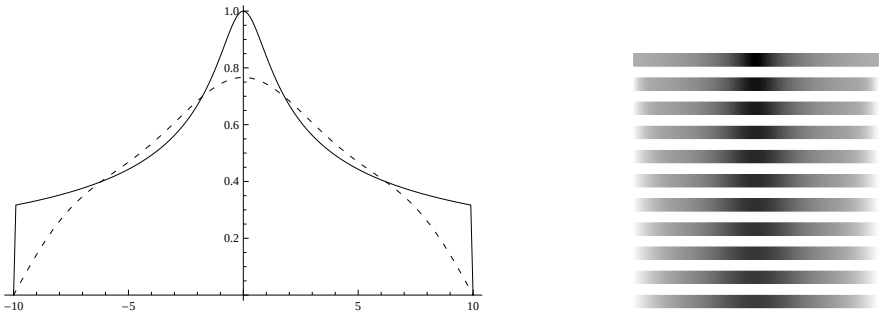
$$\begin{aligned} \frac{\partial u(x_n,t_k)}{\partial t} &\approx \frac{u(x_n,t_{k+1}) - u(x_n,t_k)}{\Delta t} \approx \frac{u_n^{k+1} - u_n^k}{\Delta t}, \\ \frac{\partial^2 u(x_n,t_k)}{\partial x^2} &\approx \frac{u(x_{n+1},t_k) - 2u(x_n,t_k) + u(x_{n-1},t_k)}{(\Delta x)^2} \approx \frac{u_{n+1}^k - 2u_n^k + u_{n-1}^k}{(\Delta x)^2}. \end{aligned} \tag{4.9}$$

Orduan t_k unean u^k soluzioaren hurbilketa ezagutuz hurrengo etapan u^{k+1} ondoriozta daiteke modu explizituan,

$$\begin{aligned} \frac{u_n^{k+1} - u_n^k}{\Delta t} - D\frac{u_{n+1}^k - 2u_n^k + u_{n-1}^k}{(\Delta x)^2} &= f(x_n, t_k, u_n^k) \Rightarrow \\ u_n^{k+1} = u_n^k + D\frac{\Delta t}{(\Delta x)^2}(u_{n+1}^k - 2u_n^k + u_{n-1}^k) &+ \Delta t f(x_n, t_k, u_n^k). \end{aligned} \tag{4.10}$$

Badaude hau baino zehatzagoak eta egonkorragoak diren beste zenbakizko eskemak, inplizituak, hain zuzen, baina hauek iterazio bakoitzean ekuazio ezlinealen sistemaren askapena derrigortzen dute, beraz zenbakizko soluzioaren doitasuna eta lan konputazioanalaren arteko oreka bilatu behar da. Baita metodo hauetan aztertu behar da $(\Delta t, \Delta x)$ urratsek bete behar duten baldintzak egonkortasunari eusteko, baina hori deribatu partzialetako zenbakizko metodoen kurtso berezian egin beharko litzateke.

Adibidez $u^0(x_i) = 1/(1+x^2)^{0.25}$ hastapen baldintza eta $u(-L,t) = u(L,t) = 0$ mugetako baldintzak betetzen dituen $D = 2 \cdot 10^{-1}$ disipazio koefizientearekiko bero ekuazioaren soluzioaren hurbilketa kalkulatzen badugu $\Delta t = 0.01$ eta $\Delta x = 0.1$ diskretizazioko urretsekin, ondoko irudian ikus daiteke bai $t = 0$ eta $t = 10$ uneetako soluzioen grafika bidimentsionalak konparatzen edo bai gris kolore mailen bidez (hainbat eta ilunagoa orduan eta bero handiagoa).



Irudia 4.1: Bero ekuazioaren zenbakizko soluzioa. Ezkerraldean t=0 (kurba jarraitua) eta t=10 (marraka) soluzioak eta eskubialdean soluzioaren bilakaera denboran zehar $0 \leq t \leq 10$.

PRAKTIKA. Beroaren ekuazioaren zenbakizko soluzioa

Mathematica lenguaiaz deribatu partzialetako beroaren ekuazioaren soluzioaren saiatu behar dugu teorian ikusitako diskretizazioen laguntzaz. Ekuazioa eta hastapen eta mugetako baldintzak hauxek dira,

$$\begin{cases} u_t(x,t) - Du_{xx}(x,t) = 0, \quad (x,t) \in [-L,L] \times [0,T], \\ u(-L) = u_1, u(L) = u_2, \\ u(x,0) = u_0(x). \end{cases} \tag{4.11}$$

- Zehaztu eredu espazialaren mugak, $[-L,L] = [-10,10]$, sarearen puntuen distantzia, $dx = 0.1$, puntu kopurua, $N = 2L/dx$, diskretizazioko puntu zerrenda, $\{x_1, \ldots, x_N\}$ non $x_i = -L + i \cdot dx, \quad i = 0, \ldots, N$,diskretizazioaren denbora-urratsa, $dt = 0.01$, amaierako denbora, $T = 10$, eta denborazko urrats kopurua $K = T/dt$.

- Zehaztu disipazio konstantea $D = 2 \cdot 10^{-1}$, mugetako balioak, "ezker=0", "eskubi=0", eta hastapen balioa $u^0(x_i) = 1/(1+x_i^2)^{0.25}$, $i = 0, \ldots, N$. Azken hau $u[0] = \{\ldots\}$ zerrendan sartuko dugu.

- Definitu $aurre(dt, dx, u0, D)$ funtzioa non dt eta dx denbora eta espazio urratsak diren, D disipazioa eta $u0$ une batean daukagun $\{u_{n-1}^k, u_n^k, u_{n+1}^k\}$ hurbilketak diren. Funtzioaren emaitza u_n^{k+1} izango da,

$$u_n^{k+1} = u_n^k + D\frac{\Delta t}{(\Delta x)^2}(u_{n+1}^k - 2u_n^k + u_{n-1}^k). \tag{4.12}$$

  Funtzioaren eraikuntza honelakoa da:

  ```
  aurre[ dt_, dx_, u0_, D_ ] := Module[ {emaitza} ,emaitza = ···; emaitza]
  ```

- Sortu $u[k]$ zerrenda eta osatu k. unean kalkulatutako soluzioaren hurbilketarekin. Kontutan hartu $aurre(dt, dx, u0, D)$ funtzioa $u[k-1]$ zerrendari aplikatuz eta mugetako balioak.

- Erabili ondko agindu grafikoa soluzioaren bilakaeraren irudia plazaratzeko,

  ```
  GraphicsGrid[Table[{Graphics[Table[{RGBColor[U[k][[i]],0,1-U[k][[i]]],
  Rectangle[{points[[i]]-dx/2, 0}, points[[i]] + dx/2, 1}]}, {i, 1, N}]],
  {k, 0, K, IntegerPart[1/dt]}]]
  ```

- Sortu $sol0$ eta $solK$ zerrendak eta osatu (x_i, u_i^0) eta (x_i, u_i^K) puntuekin hurrenez hurren. Marraztu bi funtzioak ikusteko $[0,T]$ denbora epean zenbat aldatu duen beroaren distribuzioa

  ```
  ListPlot[sol0, solK, Joined -> {True, True}, PlotStyle -> {Black, Brown}]
  ```

- (Hautazkoa) Errepikatu kalkuluak $f(x,t) = e^{-(10(x-7))^2}$ bero iturria erantsiz.

Kapitulua 5

Automata zelularrak

5.1 Sarrera

Badaude elementu homogeno ugarietaz osatutak dauden sistema fisikoak eta biologikoak non osagai bakoitzaren dinamika inguruko bizilagunekiko interakzioaren menpean dagoen. Askoetan elementu kopuru itzelagatik konputazionalki ez da bidezkoa haien eboluzioaren ekuazioak ikertzea, beraz antzekotasuna erakusten duen sistema diskretuak erabiltzen dira eredu matematikoa eraikitzeko, "*automata zelular*" izenekoak hain zuzen. Eredu matematiko mota hauetan denbora tarteak edo urratsak diskretuak dira eta une berezi batean dagoen egoeratik hurrengo unean dagoen egoera bihurtzeko urratsa "*transizio funtzioaren*" bitartez eragiten da. Elementuaren egoeraren eboluzioa bere egoera eta inguruko bizilagunen eraberakoa izango da finkatutako arau batzuk jarraituz. John Von Neumann-ek 1940 urtean automata zelularren teoria aurkeztu zuen lehenbiziko aldiz eta bai izaki bizidunen bildumako, trafikoko zein fluidoen mekanikako ereduak proposatzeko balio izan du, besteak beste.

Definizioa 5.1.1 *Automata zelularren ezaugarriak ondoko hauek dira:*

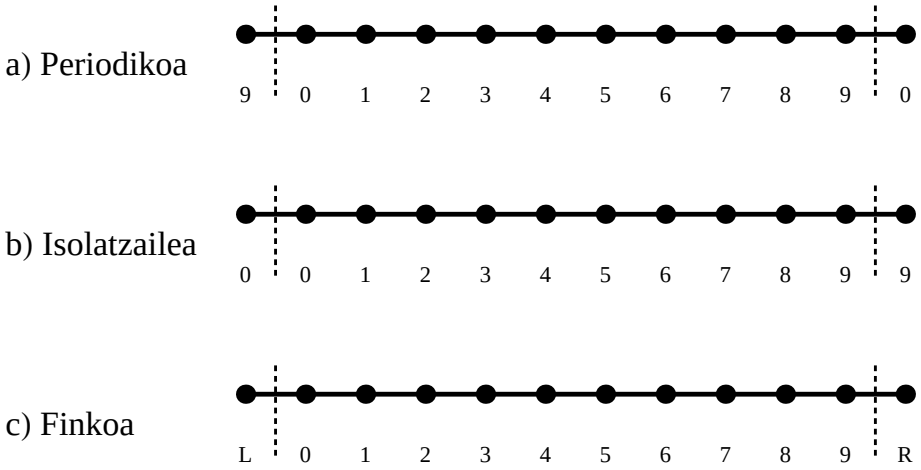
- *Zelulen edo nodoen $\mathcal{L}$ sare diskretua, homogenoa eta finitua, mugako baldintzaz ornituta.*
- *Zelulen egoera mota $\mathcal{E}$ multzo finitua eta txikia (Alfabetua).*
- *Zelularen eragin-ingurua osatzen duten zelula multzo finitua $\mathcal{N}^I$.*
- *Zelularen egoera-bihurketa adierazten duen $\mathcal{R}$ araua.*

5.2 Sarearen geometria

$\mathcal{L} \subset \mathbb{R}^d$ sareak espazio d dimentsiodunaren eskualde mugatu bat betetzen du eta bere barruan dauden zelula kopurua $|\mathcal{L}|$ da. Kapitulu honetan adibideak bakarrik $d = 1$ edo $d = 2$ espazioetan ikusiko ditugu. Zelula bakoitzaren erdigunearen kokapena zehazten duten koordenatu sorta r da. Zelula bakoitzarekin konektaturik dauden zelula kopurua b da eta "*koordinazio zenbakia*" esaten zaio. r-ren eragin ingurunea eta r-ren ondorengo egoera erabakitzen duten arau sorta zabaldu daiteke sareto eredugarriaren moduan edozein zelulari aplikatzeko.

5.3 1 dimentsiodun espazioa

Dimentsio batean, kasurik arruntena $\mathcal{L} \subset \mathbb{N}$ da eta kordinazio-zenbakia $b = 2$ izatea da, non $\mathcal{N}^I(r) = \{r - 1, r + 1\}$ den. Mugetako baldintza mota ezberdin daude eta ohizkoenak ondoko irudian ikus daitezkeenak dira, a) periodikoak, b) isolatzaileak eta c) finkoak, besteak beste.



Irudia 5.1: Dimentsio bateko eta automata zelular finituaren ohizko mugetako baldintzak.

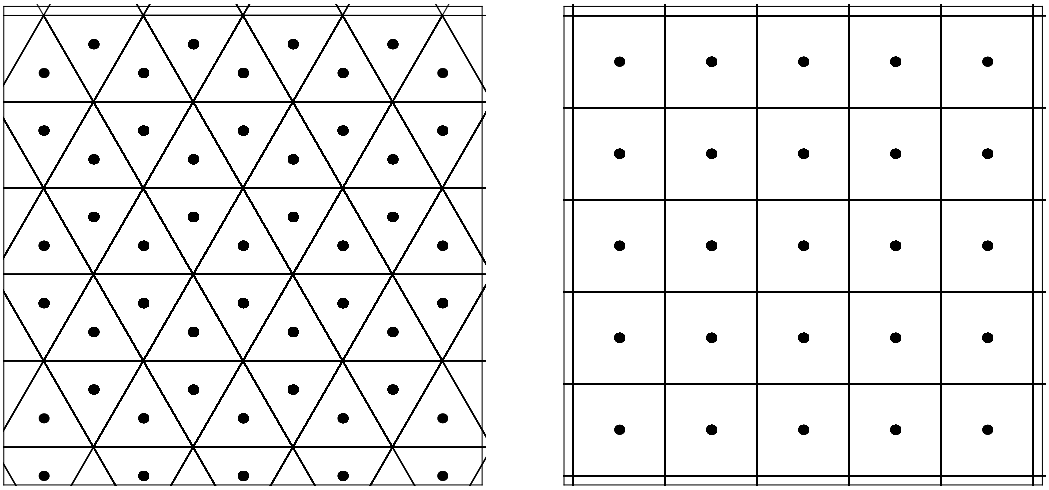
5.4 2 dimentsiodun espazioa

Bi dimentsiotan soilik hiru modu daude $\mathbb{R}^2$ espazioa estaltzeko zelula simetrikoetaz,

i) triangeluetaz, koordinazio zenbakia $b = 3$ izanik.

ii) karratuetaz, koordinazio zenbakia $b = 4$ izanik.

iii) hexagonoetaz, koordinazio zenbakia $b = 6$ izanik.



Irudia 5.2: Dimentsio bitan zelula simetrikoetaz eraiki daitekeen espazioaren bi banaketa, triangeluak eta karratuak erabiliz, hurrenez hurren.

Batzuetan zelula bera ez dago bere eragin-ingurunean, $r \notin \mathcal{N}_b^I$.

Adibidez bi dimentsioetan zelularen bihurketa erabakitzeko bakarrik zelula bera eta bere kokapenetik elementu hurbilenak hartzen baditugu, non zelula eredua $r = (0, 0)$ koordenatuan dagoen, orduan ondoko $\mathcal{N}_b^I$ saretoak (=zelula + eragin-ingurunea) sortuko dira:

geometria triangeluarra: $\mathcal{N}_3^I = \{(0,0),\ (0,-1\sqrt{3}),\ (1/2,1/2\sqrt{3}),\ (-1/2,1/2\sqrt{3})\}$

geometria karratua: $\mathcal{N}_4^I = \{(0,0),\ (0,-1),\ (1,0),\ (0,1),\ (-1,0)\}$

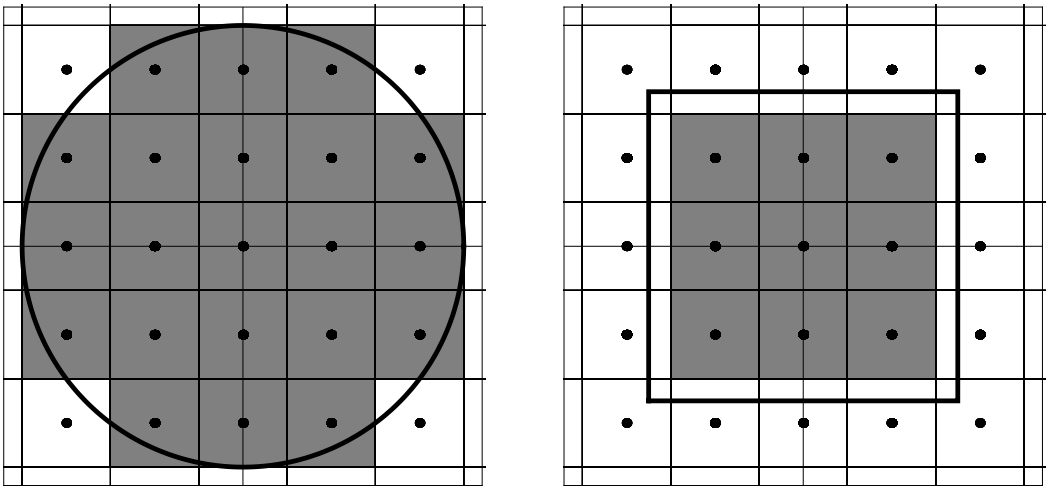
Hala ere modelizatu behar den sistemaren arabera eta bertan gertatzen diren interakzioaren arabera, batzuetan ingurune zabalagoak edo metrika ezberdinetaz neurtutakoak har daitezke eta bihurtzeko arauak ere gertutasunaren arabera konplexuagoak izan daitezke. Adibidez geometria karratuan honelako inguruneak hartzea oso ohikoa eta intuitiboa da:

Von Neumann: $\mathcal{N}_4^I = \{(0,0),\ (0,-1),\ (1,0),\ (1,1),\ (-1,1)\}$

Moore $\mathcal{N}_4^I = \{x \in \mathbb{R}^2 : ||x - r||_\infty \leq 1\} = \{(0,0),\ (0,-1),\ (1,-1),\ (1,0),\ (1,1),\ (0,1),\ (-1,1),\ (-1,0),\ (-1,-1)\}$

R-Radiala $\mathcal{N}_4^I = \{x \in \mathbb{R}^2 : ||x - r||_2 \leq R\}$

R-Axiala $\mathcal{N}_4^I = \{x \in \mathbb{R}^2 : ||x - r||_\infty \leq R\}$



Irudia 5.3: Dimentsio bitan eta egitura zelularra karratuan adierazitako eragin-inguruna R-Radiala eta R-Axiala.

$r \in \mathcal{L}$ zelula bakoitzaren egoera $s(r) \in \mathcal{E}$ adieraziko dugu eta baldin $\mathcal{L} \in \mathbb{Z}^d$ zelulen eremua $|\mathcal{L}|$ kardinalarekiko multzo finitua bada, orduan egoera edo konfigurazio posible guztien kopurua $|\mathcal{E}|^{|\mathcal{L}|}$. Gehienetan kopuru hau itzela da egoera orokorraren portaera erabat finkatzeko, horregatik r zelula ereduaren gainean eragiten duten $\mathcal{R}$ arau sorta zabalduko dugu sareto-ereduaren moduan espazio osoan. Baita metodo estadistikoak edota diferentzietako ekuazioak erabiltzen dira sistema osoaren joera aztertzeko. Joera hori periodikoa, egonkorra, kaotikoa edo zehaztugabea.

Arauak batzuetan deterministak dira inguruko baldintzek zelularen bihurketa erabat zehazten dutelako eta beste batzuetan probabilistikoak dira, adibidez egoera multzoa $\{0, 1\}$ denean, orduan zelularen bihurketa ingurunearen egoera eta banaketa binomialaren nahasketak erabakitzen duena.

5.5 Adibideak dimentsio batean

Demagun $\mathcal{L} = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$ espazioa eta $b = 2$ koordinazio zenbakia. Kasurik errazena $r = 0$-rekiko $\mathcal{N}_2 = \{-1, 1\}$ eragin-ingurunea hartzea da. Beroni gertatzen dena r orokorraren gainean heda daiteke $\mathcal{N}_2 = \{r - 1, r + 1\}$ hartuz. Bi egoera bakarreko espazioa hartuko dugu, $\mathcal{E} = \{0, 1\}$.

5.5.1 Landare ugalketa

Lursaila lerrokatutak dauden eremutxo karratuetan banatuta dago non $s(r) = 0$ utsik eta $s(r) = 1$ landatuta dagoela adierazten duen. Lehenengo adibide honetan mugarik gabeko landare ugalketa gertatzen da. Arauak honelakoak dira:

i) Landareak ez dira inoiz hilko.

ii) Inguruko bi eremutxoak utsik badaude eta bera ere, orduan utsik jarraituko du, bestela landatuta agertuko da.

Zelularen egoera $s(r)$ erabakitzen duen transizio funtzioa adierazpen arruntaz adieraz daiteke:

$$s(1,1,1) = 1,\ s(1,1,0) = 1,\ s(1,0,1) = 1,\ s(1,0,0) = 1,$$
$$s(0,1,1) = 1,\ s(0,1,0) = 1,\ s(0,0,1) = 1,\ s(0,0,0) = 0.$$

edo saretoetaz irudika daiteke:

5.5.2 Landare lehiakorrak

Lursaila lerrokatutak dauden eremutxo karratuetan banatuta dago non $s(r) = 0$ utsik eta $s(r) = 1$ landatuta dagoela adierazten duen. Bigarren adibidean landareak argia eta elikadurarengatik lehiatzen dute. Arauak honelakoak dira:

i) Landare berriak ez dira haziko.

ii) Inguruko bi eremutxoak landatutak badaude eta erdiko eremua ere landatuta badago, azken hau desagertuko da.

Zelularen egoera $s(r)$ erabakitzen duen transizio funtzioa adierazpen arruntaz adieraz daiteke:

$$s(1,1,1) = 0,\ s(1,1,0) = 1,\ s(1,0,1) = 0,\ s(1,0,0) = 0,$$
$$s(0,1,1) = 1,\ s(0,1,0) = 1,\ s(0,0,1) = 0,\ s(0,0,0) = 0.$$

edo saretoetaz irudika daiteke:

5.5.3 Trafikoarenfluxua

Demagun errepidean abiatzen diren ibiligailuak. Errepidea parametrizatuko dugu era diskretizatuan n kokapenetan eta horietariko bakoitza ibilgailu baten tamainukoa da. errepidearen kokapen bakoitza bi egoera erakutsi dezake, $r = 0$ (utsik) eta $r = 1$ (ibilgailuaz okupatuta) hain zuzen. Askotan trafikoa poliki dabil ibilgailu dentsitate handiagatik eta ilarak edo ataskoak dagoelak. Kotxeak aurreratzea ala geratzea erabakitzeko jarraibide hauek kontuan hartuko ditu:

i) Ibilgailu bakoitza denbora tarte batean gehienez urrats bat aurreratuko da.

ii) Nonbait ibilgailurik badago eta hurrengo kokapena utsik badago, orduan ibilgailua tarte bat aurreratuko da.

iii) Nonbait ibilgailurik badago eta hurrengo kokapena okupatuta badago, orduan ibilgailua dagoen kokapenean geratuko da.

Ibilgailuaren egoera $s(r)$ erabakitzen duen transizio funtzioa adierazpen arruntaz adieraz daiteke:

$$s(1,1,1) = 1,\ s(1,1,0) = 0,\ s(1,0,1) = 1,\ s(1,0,0) = 1,$$
$$s(0,1,1) = 1,\ s(0,1,0) = 0,\ s(0,0,1) = 0,\ s(0,0,0) = 0.$$

edo saretoetaz irudika daiteke:

Eredu hauek aplikagarriak dira eta zenbait aldakuntz onartzen dute mugetako baldintzen arabera. Adibidez aztertutako errepidearen tertean ibilgailu gehiago sartzen badira irtetzen direnak baino, orduan ataskoa zabaltzen da, bestela mantentzen edo murrizten da. Baita abiadura aldaketa har daiteke eta bere menpekotasuna aurreko ibilgailuarekin tartekatzen duen distantziarekiko. Noski, kasu horretan denbora tarte batean espazio bat baino gehiago aurreratzea posiblea da eta zelulen egoera espazioa handiagoa da ere.