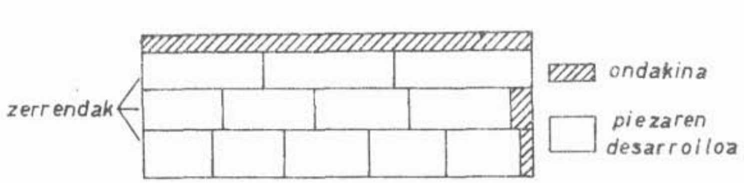
XAFLA EBAKI-AURREKO ALGORITMO BAT

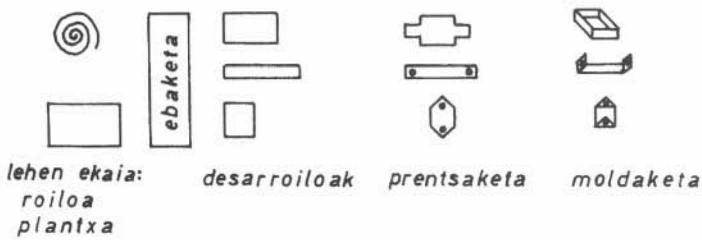
Mikel Lasa
Ramón Arana

Xafla ebaketa optimoa lortzeko erabiltzen diren algoritmo klasikoen zeregina modu honetan labur defini liteke: hedadura edo dimentsio biko pieza desberdin multzo bat ebaki hedadura jakineko roilo edo plantxetatik, ahalik eta ondakin gutien lortuaz. Ebaketa bi alditan egin ohi da: Lehenengoz, plantxa edo roila zerrendatan ebakiaz piezen zabaleraren arauera eta bigarren, zerrenda bakoitza pieza luzeraren arauera aldiko moztuaz.



Ulgor kooperatiben elektrodomestik produktoak fabrikatzen dira. Pieza batzu (konponenteak) kanpotik, probedoreenetik emendatzen dira, gainontzekoak (barneko piezak) bertan fabrikatzen direlarik, gehienetan lehen ekai bezala altzairuzko xafla erabiliaz.

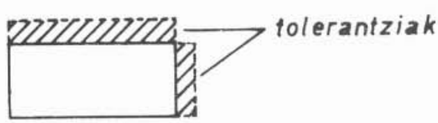
Xafla, roilo ala plantxa gisan egon daiteke. Barne piezei azken forma operazio lerrokada baten bitartez ematen zaie eta lehen operazioa, gisa denez, xafla ebakitzea da, piezaren lehen desarroiloa lortuaz.



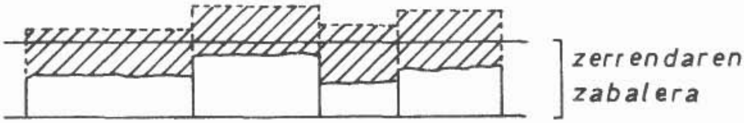
Gure lan honen helburua roiloetatik ateratzen diren piezen ebaketa hoberena (optimoa) iristean datza. Baina hemen azaltzera goazen algoritmoa problema guztiaren zati bat besterik ez da, hain zuzen ere ebaketa aurrean sortzen zaigun oztopo bati soluzioa emateko bide bat.

Pieza bakoitzetik ebaki beharreko kopurua, epe motzeko (aste bat edo bi) fabrikazio programari loturik dago; hots, ezin liteke nahi haina moztu eta kasu batzutan ebaki beharrekoa oso guti izan liteke. Horretzaz gainera roiloetatik ateratzen diren zerrendak roiloen luzera guztian zabalera berdina dute noski, definizioz. Zerrendaren zabalera berdineko pieza bakar bat baldin badaukagu mozteko eta honen fabrikazio beharra zerrendaren laurdenaz (eman dezagun) betetzen bada, desarroiloen stocka hanpatu egingo litzateke neurriz gain eta hori arrazoi bat baino gehiagogatik ezin liteke ameti.

Nola ebaki hauzia?. Zabalera berdineko pieza desberdinak ebakiz zerrenda bakoitzetik, denen arteko fabrikazio beharrak zerrendaren luzera guzia estali ahal izateko moduan. Soluzio hau praktikan ez da nahikoa, oso guti baitira zabalera berdineko piezak. Zorionez pieza bakoitzen desarroiloa (zabalera x luzera) ez da aldagaitza: minimun batetik gora tolerantzi edo gehigarri bat eduki dezake dimentsio bakoitzean.



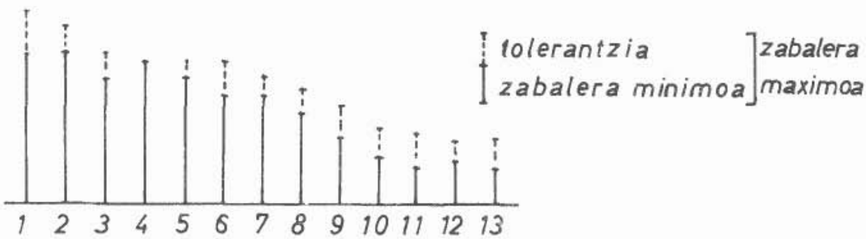
Tolerantzia kontutan harturik elkar daitezkeen piezen konbinaketa aberastu liteke eta joko erosoagoa egin.



Irud. 1

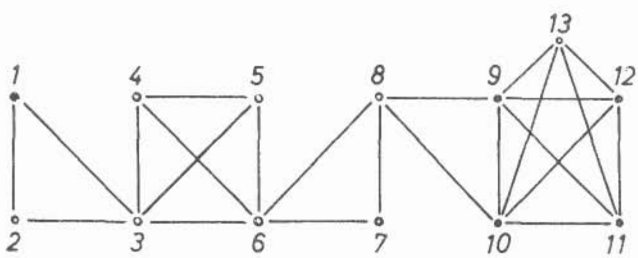
Hain zuzen ere azaltzera goazen algoritmoa piezen konbinaketa matematikoki lortzeko asmatua dugu, gero ordinatoreaz programatu ahal izateko. Ondoren algoritmoa azaltzera goaz.

Demagun pieza multzo bat. Horien zabalera + tolerantzia grafikoki modu honetan adieraz daiteke, luzeraren arauera ordenatuz gero:



Irud. 2

Grapho baten bitartez ere gauza bera adieraz liteke:



Graphoaren puntu bakoitzak pieza bat idurikatzen du. Puntu batetik bestera doan marrak "Konbinagaitasun" elkarpidea adierazten du. Beraz loturik gabeko bi puntu konbinagaitzak dira.

Bestalde guztiz inportatea da elkarpide honen propietateak definitzea:

- **symetrikoa** da, hots: 1 puntua 2 puntuarekin konbina badaiteke, 2 puntua 1 puntuarekin konbina daiteke;

- **intransitiboa** (iragangaitza) da, hots: 1 p. 2 p.rekin konbina badaiteke eta 2 p. 3 p.rekin, 1 p. besterik gabe ezin daiteke konbina 3 p.rekin;

- **reflexiboa** da, hots: pieza bakoitza bere buruarekin konbina daiteke.

Propietatearen bitartez graphoa zehatz definitu ondoren, graphoa begiratu hutsa nahiko dugu problemari buruz zenbait gauza oztoporik gabe esateko: e.b., garbi dakusgu 9 eta 10 p.k. 8, 10, 11, 12 eta 13 p.rekin nolanahi den konbina daitezkela; baina 8 p. ezin daiteke 10, 11, 12, 13 p.rekin konbina, bai aldiz 9 eta 11 p.rekin, etc...

Bestalde, tresna paregabea eskaintzen digu soluzioari hedapen berriak matematikoki zabaldu ahal izateko.

Goiko graphoan 1, 2, 3 puntuek, besteak beste, "**bortizki loturiko subgrapho**", (BLSG) bat osatzen dute, esan nahi da, puntu bakoitza subgraphoko (SG) beste edozeinekin loturik dagoela edo hirurak batera konbina daitezkela.

6 Irud. dauden beste BLSG-ak dira:

(3, 4, 5, 6)
(6, 7, 8)
(8, 9, 10)
(9, 10, 11, 12, 13)

Gure problemaren kako nagusienetakoa, hain zuzen algoritmoaren bitartez argitu nahi duguna, grapho osoaren barnean dauden **BLSG guztiak aurkitzean datza**. Problemaren muina ulertu duenak garbi ikusiko du BLSG-en beharra: grapho osoaren baitan puntu (pieza) bat bakarrik badago eta gainera pieza honen ebaki beharrak zerrenda osoa ez badu estaltzen, ondakin asko sortuko du piezaren ebaketak. Eta alderantziz, zenbat eta ugariago BLSG-ko puntuak orduan eta soluzio erosoago. Proba puntu bat BLSG bi edo gahiagokoa baldin bada, orduan eta konbinaketa joko ugariagoa puntu horri soluzio egoki bat eman ahal izateko.

BLSG guztien bilaketa

Lehenengo momentuan matrizen teoriaz baliatzea pentsatu genuen BLSG-ak bilakatzeko, gauza jakina bait da edozein grapho matriz binomial modura adierazteko gai dela. Hona hemen bada, pieza multzo baten zabalera + tolerantzia eta heuren arteko "konbinagaitasun" elkarpidea idurikatzeko hirugarren modua:

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	0	0	0	0	0	0	0
4	0	0	1	1	1	1	0	0	0	0	0	0	0
5	0	0	1	1	1	1	0	0	0	0	0	0	0
6	0	0	1	1	1	1	1	1	0	0	0	0	0
7	0	0	0	0	0	1	1	1	0	0	0	0	0
8	0	0	0	0	0	1	1	1	1	1	0	0	0
9	0	0	0	0	0	0	0	1	1	1	1	1	1
10	0	0	0	0	0	0	0	1	1	1	1	1	1
11	0	0	0	0	0	0	0	0	0	1	1	1	1
12	0	0	0	0	0	0	0	0	0	1	1	1	1
13	0	0	0	0	0	0	0	0	0	1	1	1	1

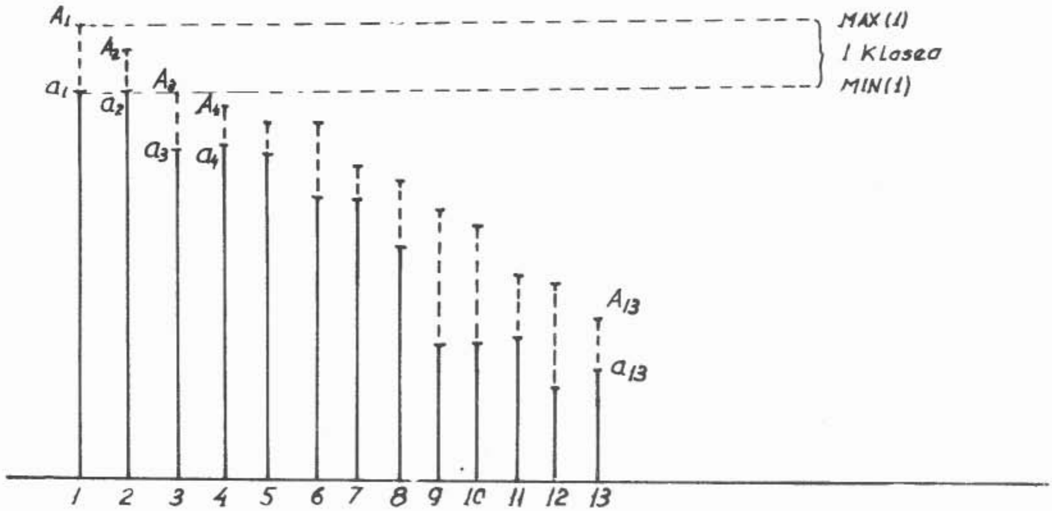
i lerro eta j zutabe bakoitzaren guruztzapuntuan bi balore besterik ez dukegu:

0 i p. eta j p.ren artean loturik ez badago
1 " " " " " " " badago

Beraz, matriz binomial symetriko bat da. Matriz honen baitan, BLSG bakoitzak 1-ez betetako submatriz koadratu maximal (SKM) bat osatzen du.

Planteamendua modu honetan egin ondoren, lege zen, matrizen teoriaz diharduten textuetan bilatu nahi izatea algoritmoren bat SKM guztiak aurkitu ahal izateko. Zoritxarrez gure lantegiko liburutegi matematikoan ez dugu aurkitu ahal izan nahi genuen algoritmorik.

Beste bide batetik topatu dugu soluziobidea. BLSG edo, berdina dena, SKM guztiak topatzeko asmatu dugun algoritmoa azaltzeko irud. 5-az baliatuko gara berriro.



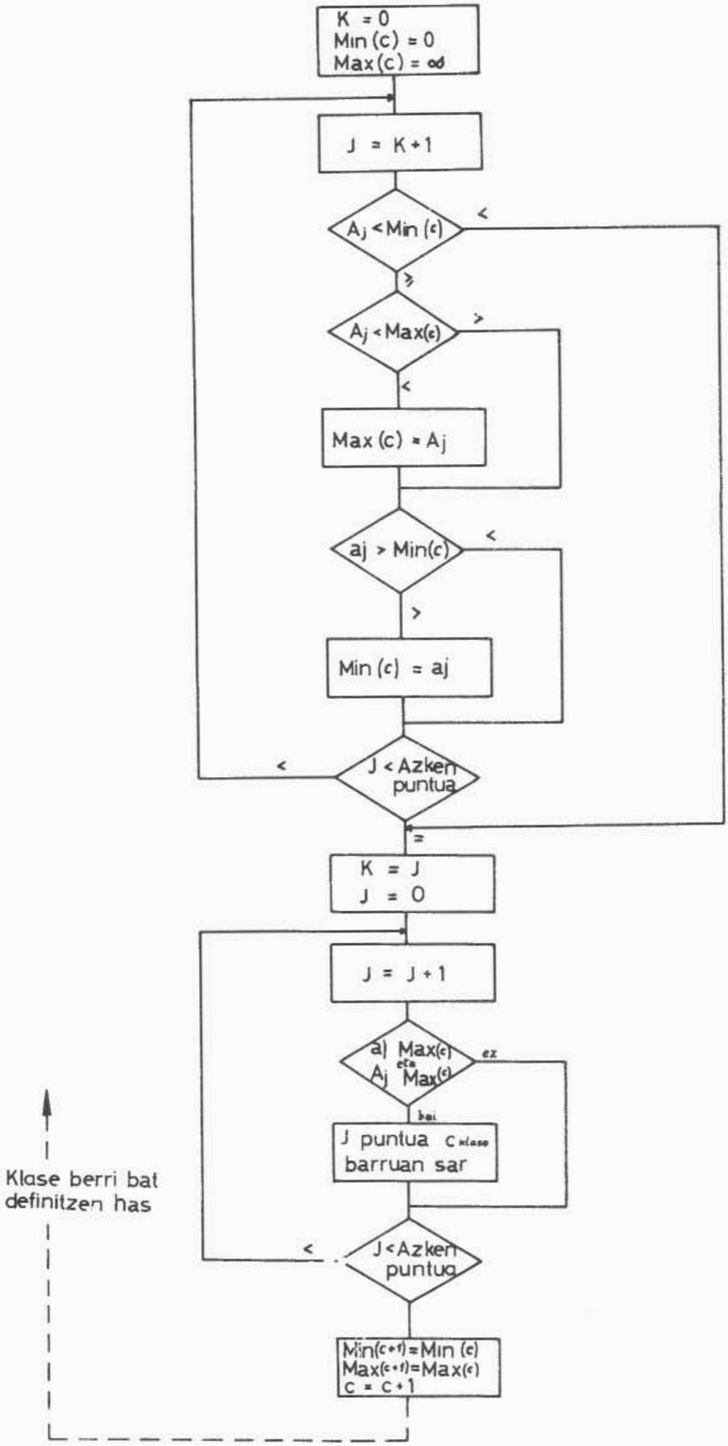
Elkarregaz konbina daitezken puntuak klase bat osatzen dute. Klase bakoitza MIN (c) eta MAX (c) bikoteak definitzen du, muga horien artean tolerantzia daukaten puntuak sartzen direnez gero.

Esate baterako, 1 klasea 1, 2 eta 3 puntuek osatzen dute
2 " 3, 4, 5 eta 6, etc...

Lan horretan zehar gauza bera adierazteko hiru kontzeptu erabili ditugu: "Klasea", "Bortizki loturiko subgraphoa" (BLSG) eta "Submatriz koadratu maximala" (SKM).

Ondoren ematen dugu organigrama baten bitartez klaseak definitzeko algoritmoa edo berdin denez BLSG-k edo berdin denez SKM-k.

Problema modu honetan formalizatuz gero, erabat errazten da programaketa lana eta bide zihur bat eskaintzen soluziobide guziak ez bada ere interesanteenak aztertu ahal izateko eta azken finean hoberena ez bada ere soluzio on baten eskuratzeko.


K = 0
Min (c) = 0
Max (c) = ∞
J = K + 1
Aj < Min (c)
Aj < Max (c)
Max (c) = Aj
aj > Min (c)
Min (c) = aj
J < Azken puntua
K = J
J = 0
J = J + 1
ez
bai
J puntua C klase barruan sar
J < Azken puntua
Min (c+1) = Min (c)
Max (c+1) = Max (c)
C = C + 1
Klase berri bat definitzen has