

20 63

U N I V E R Z I T E T U S A R A J E V U
P R I R O D N O - M A T E M A T I Č K I F A K U L T E T

Univerzitet u Beogradu
Prirodno-matematički fakultet
MATEMATIČKI FAKULTET
BIBLIOTEKA
Dokt. 233 Datum 24. 01. 1990.
Broj

J O V A N L O N Č A R

ALGORITMI ZA ODREĐIVANJE OPTIMALNOG RJEŠENJA PROBLEMA
SMJEŠTANJA FIGURA DATOG OBLIKA U ZADANU FIGURU

D i s e r t a c i j a

Sarajevo 1982.

Ova je radnja predana na ocjenu Prirodno-matematičkom fakultetu Univerziteta u Sarajevu u svrhu stjecanja znanstvenog stupnja doktora prirodnih znanosti iz područja matematike.

U Sarajevu, juli 1982.

Ideja za ovaj rad rezultat je mog dugogodišnjeg iskustva u oblasti linearne optimizacije. Dug je put dok sam uspio prijeći od linearnih na nelinearne probleme.

Na tom putu mnogo su mi pomogle ideje i savjeti prof.dra Djure Kurepe, prof.dra Milorada Zeca i prof.dra Veselina Perića, kojima se iskreno zahvaljujem.

P R E D G O V O R

Prije nekih 20 godina pročitao sam monografiju L.F.Tot [51] i počeo se više interesirati za probleme optimalnog pakovanja i pokrivanja. Kasnije sam se upoznao s radovima C.A.Rogers [42] i monografijom i radovima L.V.Kantorovića [23], [24], [25]. Neugodno me iznenadilo kada sam spoznao da se navedeni rezultati, do tog vremena, kod nas u praksi uglavnom ne primjenjuju, mada bi primjene imale za posljedicu velike uštede materijala. Ispitivanja koja sam izvršio dovela su me do zaključka da se krojenje nelinearnih figura vrši gotovo isključivo na bazi intuicije. Rezultat toga su "brdašca" otpadnog materijala pored mnogih tvornica. Program za računar, za optimalno krojenje ploča iverica i drugih u drvnoj industriji koga sam izradio s doc. dr Egonom Zakrajšek dao je velike uštede a meni je bio jedan od podstreka da se intenzivno nastavim baviti problemima krojenja i drugih još skupljih materijala.

Ispitujući mnoge krojne liste iz raznih oblasti gdje se radi o složenim figurama primjetio sam da svi bez izuzetka poligone figure smještaju u pravokutne ploče tako da je jedna od stranica figure paralelna stranici pravokutnika. Nadalje sam ustanovio da se bez obzira na oblik figura smatra da se kod istih površina uvijek bolje iskorištenje postiže kod kvadratnih nego kod pravokutnih ploča.

Prikupljajući publicirane materijale vidio sam da se samo ponekad navode i algoritmi ali samo načelno. Medjutim, velik dio tih algoritama je neprimjenljiv na računarima jer postoje poteškoće kod obrade nekih geometrijskih pojmova na računaru (problem je što računar nema oči). L.V.Kantorović u spomenutoj monografiji konstatira da u objavljenim radovima do 1970. nitko uz rezultate ne navodi kompjuterske programe. Nažalost do istih rezultata došao sam i ja za radove sve do današnjih dana. Naime, nisam uspio pronaći bar jedan rad da se daje i program za računar.

Upravo zbog toga razvijao sam samo one algoritme koje sam mogao pretočiti u kompjuterske programe, ali takve da se mogu realizirati na računarima u ekonomski opravdanom vremenu. Pored toga vodio sam računa da se programi mogu realizirati i na su-

vremenim "mini" računarima. Svi programi su napisani u FORTRANU ili PASCALU. Upoznavši se s radom Ju.G.Stojana [52] ustanovio sam da ću moći primijeniti jedan paket programa koji sam izradio u periodu 1966-1972. (sastoji se od 21 programa) za rješavanje problema koje obradjujem u glavi VI, mada su programi imali drugu namjenu. U razvoju ovih programa pomogli su mi na razne načine dr.E.Zakrajšek, Franc Mandlec, Janez Lesjak, Nance Nadrah na čemu im se ovom prilikom zahvaljujem. Posebno se želim zahvaliti mr.p.pukovniku Mihajlu Bilbiji na suradnji u oblasti kompjutorskih znanosti jer mi je ta suradnja bila nezamjenljiva.

Želim napomenuti da trajanje izvodjenja za bilo koji problem iz glave III, IV i V nije prelazio jednu minutu na sistemu UNIVAC 1100, pa te programe smatram definitivnim. Međutim, što se tiče problema iz glave VI to još nisam potpuno zadovoljan jer kod smještaja 30-50 objekata složena oblika računi traju dosta dugo, već prema složenosti objekata, mada u granicama ekonomski opravdanim. Zbog toga ove programe je poželjno i dalje poboljšavati.

Napominjem da mi je velike poteškoće stvaralo to što u našoj Zemlji nema nekog afirmiranog naučnog radnika upravo iz ove problematike, što će se svakako odraziti u ovoj disertaciji.

Na kraju želim se zahvaliti radnim ljudima Više zrakoplovne škole Sveučilišta u Zagrebu što su mi omogućili da disertaciju privedem kraju.

S A D R Ž A J

	Strana
U V O D -----	I
G L A V A I	
UVODNI POJMOVI, DEFINICIJE, FORMALIZACIJA PROBLEMA I R-RUNKCIJE	
1. Uvodni pojmovi i definicije -----	1
2. R-funkcija -----	6
3. Opća matematička formulacija problema smještaja geometrijskih objekata -----	14
G L A V A II	
KONTAKT FUNKCIJA I NJEN HODOGRAF	
1. Svojstva kontakt funkcija i njenog hodografa -----	19
2. Kontakt funkcija konveksnih poligona i njen hodograf	24
3. Svojstva HKF konveksnih poligona -----	29
4. Hodograf kontakt funkcije, konkavnih poligona -----	34
5. Hodograf kontakt funkcije objekata ograničenih po dijelovima glatkim krivuljama -----	40
6. Analitičko nalaženje HKF objekata -----	42
7. Problem jednoznačnosti kontakt funkcije -----	49
8. Potporna funkcija i konstrukcija ekvidistante -----	52
9. Aproksimacija granice zadanog objekta -----	56
PROGRAMI UZ GLAVU II -----	71
G L A V A III	
JEDNOREDNI PERIODIČNI RASPOREDI	
1. Formulacija problema i smještanje u beskonačnu traku	95
2. Smještaj objekata u traku zadane širine -----	100
3. Optimalni smještaj objekata u traku konačne duljine	107
4. Optimalan smještaj u pravokutnik -----	111
5. Problemi smještaja kompleta u pravokutnik -----	119
PROGRAMI UZ GLAVU III -----	127
G L A V A IV	
REŠETKASTI RASPORED OBJEKATA	
1. Formulacija problema i smještanje u ravnini -----	149
2. Rešetkasto smještanje u pravokutnik -----	157
PROGRAMI UZ GLAVU IV -----	171

GLAVA V

PERIODIČNO SMJEŠTANJE U n-REDOVA

1. Opis i definicija problema -----	175
2. Formalizacija problema n-rednog smještaja -----	179
3. Uvjeti nepresjecanja simetrija, suženja granica kut- nih parametara -----	193
4. Formuliranje tehnoloških ograničenja pomoću KF, PF i R-funkcija -----	204
5. Formulacija problema smještanja pomoću KF, PF i R- -funkcija specifičnosti i realizacija pomoću kompj- uterskih programa -----	207
PROGRAMI UZ GLAVU V -----	235

GLAVA VI

NEREGULARNO SMJEŠTANJE ORIJENTIRANIH OBJEKATA U RAVNINI

1. Formulacija problema, uvjeti medjusobnog nepresjeca- nja objekata i uvjeti smještanja objekata u oblast --	259
2. Formalizacija funkcije cilja -----	264
3. Neke karakteristike neregularnog smještaja objekata u oblasti s pomičnim granicama i svojstva funkcija cilja -----	272
4. Sukcesivni pojedinačni smještaj i njegova realizacija	276
5. Separabilne funkcije i njihova svojstva -----	283
6. Induciranje diskretne strukture kod problema neregu- larnog smještanja geometrijskih objekata -----	307

LITERATURA -----	319
------------------	-----

AUTOBIOGRAFIJA -----	323
----------------------	-----

POPIS ZNANSTVENIH I STRUČNIH RADOVA -----	325
---	-----

PRILOG: PROGRAMI UZ GLAVU VI	
------------------------------	--

U V O D

Problemi smještaja geometrijskih objekata pojavljuju se u mnogim područjima kao što su:

- problemi optimalnog krojenja u metalnoj, tekstilnoj, drvnoj, plastičnoj, kožarskoj, staklarskoj i drugim industrijama
- problemi mikrominijaturizacija
- problemi u kristalografiji
- problemi kompleksnog pristupa projektiranju
- problemi modeliranja raznih ekonomskih, fizikalnih i kemijskih procesa
- problemi slaganja u štamparskoj industriji itd.

Prvi pokušaj iz područja krojenja učinio je veliki ruski matematičar P.L.Čejbišev još 1878. [8]. E.C.Fedorov izučava u radu [10] i [11] probleme smještaja u kristalografiji. H.Heesch izučava figure koje bez rupa i preklapanja pokrivaju R^2 , tzv. probleme parketiranja [21] [22].

U [5] se korištenjem rezultata Heescha pokušava riješiti problem smještaja kongruentnih figura u ravnini.

Problemima optimalnog iskorištenja materijala bavi se decenijama i dobitnik Nobelove nagrade L.V.Kantorović [23], [24], [25]. U Monografiji ovog autora [25] navodi se 750 radova iz područja krojenja. Međutim, u spomenutim radovima koji se odnose na figure složenog oblika, pretpostavlja se da krojne liste postoje, a onda se metodama linearnog i dinamičkog programiranja vrši izbor boljih rješenja.

U šezdesetim i sedamdesetim godinama publiciran je veći broj radova iz krojenja jednodimenzionalnih i kvadratnih materijala, s time da su i figure bile kvadrati [7], [7*], [14], [15], [20]. U [29], za dovidmenzionalno krojenje uz uvažavanje "giljotinskih" rezova, razvijen je program OPTIMA koji se od 1974. uspješno koristi u drvnoj industriji.

F.L.Tot u monografiji [51], kao i C.A.Rogers u [42], metodama diskretne geometrije tretiraju probleme pakovanja i pokrivanja,

ali za figure kao što su krugovi, elipse, kugle i elipsoidi. V.A.Zalgaller u [53] metodom analogije s nekim problemima u mehanici definira jedan nužan uvjet da objekti budu u kontaktu. Pojavom ERS jako se puno eksperimentira [4], [29], [37], [45], [46], [52], [54]. Ovo se čini zbog toga što još nije bila razvijena adekvatna teorija.

U periodu 1962-1966 razvijena je teorija R-funkcija [43], [44] čime je riješen onaj dio problema koji je vezan za analitički opis figura proizvoljnog oblika. Koristeći aparat R-funkcija u više radova formuliraju se problemi smještaja složenijih geometrijskih figura. Značajniji radovi te vrste su [6], [9], [13], [16], [47], [49].

Naime, konture figura su dijelovi pravaca i lukovi kružnica. Pomoću R-funkcija pripadnost točke figuri može se zapisati jednom nejednadžbom. Specijalni izbor R-funkcija osigurava njihovu diferencijabilnost, što se koristi kod gradientnih metoda. S figurom, osim njene konture, veže se izvjestan broj kružnica smještenih u figuru, koje razbijaju figuru na dijelove koji nisu previše veliki, a dio kružnica se "ugura" u "uglove" figure. Uvjet nepresjecanja figura formulira se preko uvjeta nepresjecanja kružnica. Optimalno rješenje se traži tako da se generira velik broj slučajnih položaja gdje se figure ne presjecaju, a zatim se gradientnim metodama prevode u položaje kod kojih se postižu lokalni minimumi. Opsežni eksperimenti koje sam izvršio pokazuju da računari s brzinama od 5×10^5 do 10^6 operacija u sekundi troše od 4 do 5 sati rada, koristeći navedene metode, tj. metode su za praksu gotovo neprihvatljive.

U [39] navode se asimptotska svojstva jedne klase funkcija od puno varijabla. Koristeći navedena svojstva u [48] i [50] uvodi se pojam r-separabilnih funkcija, i predlaže vjerojatnosna metoda za nalaženje lokalnih i globalnih ekstrema takvih funkcija. Metoda za nalaženje lokalnih ekstrema je u stvari adaptirana Gauss-Seidlova metoda po-koordinatne optimizacije.

Budući da su uvjeti nepresjecanja definirani pomoću R-funkcija vrlo glomazni, tražene su druge metode. Prema navodima u [13], u doktorskoj disertaciji Ju.G.Stojana [49] prvi puta se uvodi

pojam kontakt funkcije pomoću koje je definiran problem smještaja krugova i pravokutnika, kao i figura koje nastaju sastavljanjem pravokutnika. (Kao što je poznato doktorske i kandidatske disertacije se ne mogu dobiti iz SSSR-a, već samo autoreferati, ali za navedenu disertaciju [49] ni to nije bilo moguće.)

U [12] i [13] se, čini se, prvi puta uvodi pojam hodografa kontakt funkcije HKF, gdje se samo izriče bez dokaza jedan broj teorema u vezi s HKF, koje navodimo u Glavi II, od kojih neke dokazujemo. Za sve radove publicirane do 1971. godine, koji se odnose na smještaj geometrijskih objekata složenog oblika L.V.Kantorović u monografiji [25] konstatira da se navode samo opisi algoritama, ali ne i algoritmi, dok se kompjuterski programi nigdje ne navode. Govoreći o sastavljanju krojnih lista za proizvodljne objekte kaže da je to teško sagledivo, te da mu nisu poznati radovi čak ni za jednostavnije figure.

Navedena je tvrdnja i bila impuls da od 1971. godine radim na razradi algoritama, a zatim kompjuterskih programa koji su prikazani u ovoj disertaciji.

Iz rada [2] se vidi da je 1969. odbranjena disertacija na New Yorku University [1]. Međutim, tu disertaciju nisam uspio dobiti. Iz kasnijeg rada [3] proizlazi da do 1977. godine problem smještaja proizvoljnih figura nije riješen. Naime, u radovima do 1977. (kasniji mi nisu poznati) navedeni autori nepravilne figure prethodno smještaju u kvadrate minimalne površine, a zatim najbolji smještaj kvadrata uzimaju kao najbolji smještaj zadanih figura, što očito nije točno. Čak što više, oni u [2] zaključuju "... ALGORITHM THAT SEEKS FOR AN OPTIMAL LAYOUT WORKING DIREKTLY WITH THE IRREGULAR SHAPES SOUNDS INFEASIBLE ..."

U 1980. godini publiciran je zbornik [28]. Na osnovi rada N.J.A. Sloane, str.219-249, i opaski redaktora, dobiva se zaključak o problematici zaključno s 1979. god. Naime, ne vidim neku noviju teoriju koja bi omogućila rješavanje problema o kojima je riječ u ovoj disertaciji na uspješniji način od onog kojeg sam primijenio. U ovoj disertaciji na osnovu HKF, KF, PF i R-funkcija, te njihovih svojstava izradjeno je nekoliko kompleksa takvih kompjuterskih programa koji omogućavaju da čak i na suvremenim mini računarima, za prihvatljivo vrijeme, možemo praktički riješiti vrlo

široku klasu problema koji se svakodnevno javljaju na jako puno mjesta. Naime, problemi o kojima je riječ u disertaciji danas se kod nas rješavaju intuitivno. Računari se tu i tamo koriste samo za brže ispitivanje intuitivnih rješenja.

Svi programi su testirani na velikom broju problema iz prakse. Pri tom se pokazuje da su pomoću navedenih programa dobiveni rezultati koji su za 5 do 10% bolji od onih koji se koriste u praksi. Čak je nađen ne mali postotak slučajeva gdje su rezultati bolji za 25 do 32%. Prema tome, primjenom rezultata ove disertacije omogućavaju se ogromne uštede oskudnih i dragocijenih resursa.

Disertacija sadrži: predgovor, uvod, šest glava i popis literature.

U disertaciji se daju kompleti programa u FORTRANU i PASCALU iz kojih se s jedne strane vide sadržaji primijenjenih algoritama a s druge strane zainteresiranim se pruža mogućnost da specificiranjem ulaza i izlaza brzo sastave programe za svoje računare.

GLAVA I

Glava I se sastoji od tri člana. Daju se uvodni pojmovi i definicije. Navode se neka svojstva R-funkcija. Daje se opća matematička formulacija problema smještanja.

GLAVA II

Ova glava ima devet članova. Obradjuju se hodograf kontakt funkcija (HKF), kontakt funkcija (KF), potporna funkcija (PF). Izriču se određeni teoremi i tvrdnje u vezi sa HKF, KF i PF. Obradjuje se problem jednoznačnosti KF. Daje se postupak za aproksimaciju konture proizvoljnog objekta pomoću poligona s unaprijed zadanom točnošću.

Na osnovi pojma HKF daje se postupak za konstruiranje ekvidis-tante.

Daju se kosturi programa:

- 1 - za konstrukciju HKF za proizvoljne poligone
- 2 - za izračunavanje vrijednosti KF

- 3 - za razbijanje nekonveksnog objekta na konveksne objekte
- 4 - za nalaženje konveksne ljuske nekonveksnog poligona
- 5 - za aproksimaciju proizvoljne konture objekta sa poligonom s proizvoljnom točnosti
- 6 - za konstrukciju ekvidistante na bazi HKF
- 7 - za izračunavanje vrijednosti PF
- 8 - za nalaženje zvjezdaste oblasti (točke) u području omeđenom HKF
- 9 - za razne operacije sa skupovima

Rad programa se demonstrira na konkretnim primjerima a rješenja su predstavljena slikama, grafikonima i tabelama.

GLAVA III

Glava III se sastoji od pet članova. U njoj se, na bazi KF i PF formuliraju i rješavaju problemi jednorodnih periodičnih smještaja.

Daju se kosturi paketa programa pomoću kojih su riješeni slijedeći problemi:

1. optimalno smještanje u beskonačnu traku "JERED"
2. smještanje u traku konačne duljine "KONTRA"
3. smještanje u traku zadane širine "VIRED"
4. smještanje u pravokutnik "PRAVOK"
5. izbor optimalne širine trake iz skupa raspoloživih traka za smještanje zadanog objekta "VIRED I"
6. izbor optimalne širine trake za koju je u prosjeku najmanji otpad "VIRED II"
7. izbor optimalne dimenzije pravokutnika iz skupa raspoloživih dimenzija, za koji je iskorištenje maksimalno "PRAVOK I"
8. izbor pravokutnika za koji je prosječno iskorištenje maksimalno "PRAVOK II"
9. optimalni smještaj kompleta u pravokutnik
10. izbor optimalne dimenzije pravokutnika za zadani komplet
11. izbor najbolje dimenzije pravokutnika za skup kompleta.

Rezultati rada za konkretne primjere prikazani su slikama, tabelama i grafovima.

GLAVA IV

Navedena glava se sastoji od dva paragrafa. Pomoću HKF, KF i P formulira se problem rešetkastog smještanja objekata u ravnini i pravokutniku.

Razradjeni su paketi programa:

1. za rešetkasto smještanje figura u ravnini
2. za rešetkasto smještanje figura u pravokutniku.

Rezultati rada programa za konkretne primjere prikazani su pomoću slika, tabela i grafova.

GLAVA V.

Glava se sastoji od šest paragrafa. Pomoću HKF, KF, PF i R-funkcija definiran je problem n -rednog periodičnog smještanja. KF, PF i R-funkcijom definiraju se razna tehnološka ograničenja. Daju se kosturi paketa programa pomoću kojih su riješeni slijedeći problemi n -rednog smještanja:

1. n -redno smještanje u beskonačnu traku
2. n -redno smještanje u rolnu zadane širine
3. određivanje broja redova kod smještanja jedne vrste objekata u rolnu zadane širine
4. n -redno smještanje u pravokutnik
5. izbor one širine rolne iz skupa zadanih za koju je otpad minimalan
6. izbor one rolne iz skupa zadanih za koju je otpad u prosjeku minimalan
7. izbor pravokutnika iz skupa raspoloživih pravokutnika za koji je otpad za zadane objekte minimalan
8. izbor pravokutnika iz skupa raspoloživih pravokutnika za koji je u prosjeku za zadane objekte otpad minimalan.

Rad paketa programa demonstrira se na konkretnim primjerima a rezultati su prikazani pomoću slika, tabela i grafova.

GLAVA VI

Materijal ove glave obradjuje šest članova problema neregularnog smještanja međusobno orijentiranih objekata.

Formulirani su uvjeti medjusobnog nepresjecanja objekata i uvjeti smještaja objekata u zadanu oblast. Formulirana je funkcija cilja. Prikazane su neke karakteristike oblasti s pomičnim granicama i ukazuje se na svojstva funkcije cilja, koja proizlaze iz nelinearnosti, mnogoekstremalnosti i višestruke suvislosti oblasti definicije. Ukazano je na neperspektivnost determinističkih metoda, nalaženja globalnog ekstrema. Izvršena je realizacija metode SPS (sukcesivni pojedinačni smještaj) pomoću HKF. Pokazano je da metoda SPS kod rješavanja problema neregularnog smještaja proizvoljnih geometrijskih objekata inducira slijedeće:

- a) da se lokalni ekstremi postižu na granici oblasti dopustivih rješenja
- b) da se kontinualni problemi svode na diskretne tj. na probleme minimalizacije funkcionala na skupu permutacija
- c) da se optimalna rješenja koja se navode u radovima [2] i [3] mogu koristiti kao prva aproksimacija u danoj metodi.

Uočavanje ovih činjenica omogućilo je da iskoristim jedan kompleks programa (sastoji se od 21 programa) koji sam izradio u periodu 1966-1972 za vrijeme rada u VISOKOJ TEHNIČKOJ ŠKOLI JNA u Zagrebu, za nalaženje minimuma funkcionala definiranih na skupu permutacija [30] - [36]. Eksperimentalno sam utvrdio da se najbolji rezultati postižu primjenom alfabetske metrike.

Demonstracija paketa programa vrši se na konkretnom primjeru smještanja 20 objekata složena oblika u oblast s pomičnim granicama. Rezultati su predstavljeni slikama i grafovima.

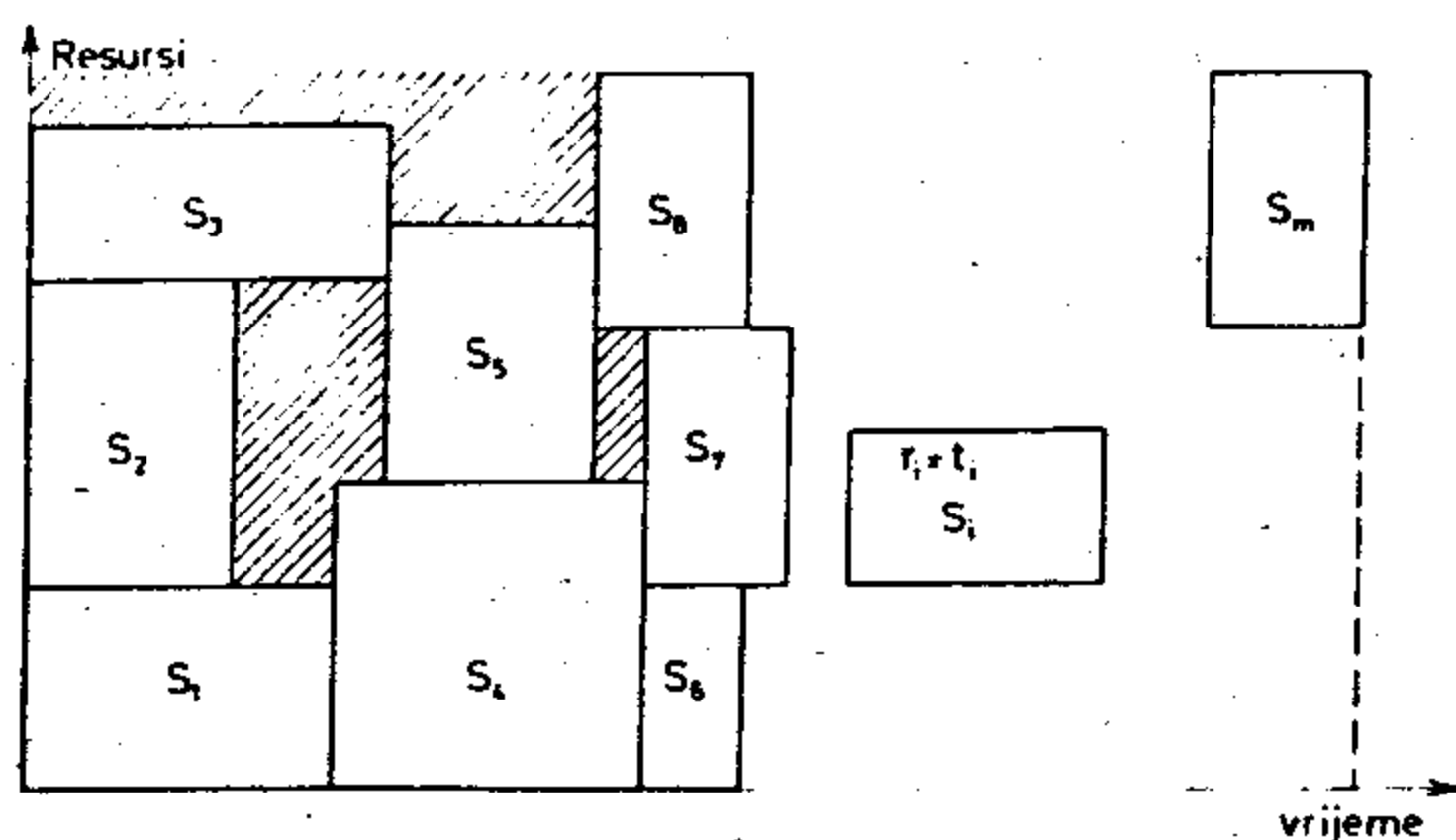
G L A V A I

UVODNI POJMOVI, DEFINICIJE, FORMALIZACIJA PROBLEMA I R-FUNKCIJE

1. Uvodni pojmovi i definicije

Pod geometrijskim objektom (ili jednostavno objektom) podrazumijevat ćemo zatvoren skup S , točaka prostora R^n koje pripadaju oblasti koju zauzima objekt. U našim razmatranjima to će biti razne metalne i drvene ploče, razni formati tkanina, plastike, gume, kože, razni alati, aparature smiještene na avionu, brodu i drugdje, te kontejneri, molekule, tj. sve ono što predstavlja likove u ravnini ili tijela u prostoru.

U širem smislu to su i rokovi izrade nekog predmeta, rashod energije po vremenu, raspoloživi resursi i planovi utroška resursa itd. Dakle, to je sve ono što se može geometrijski interpretirati. Tako na primjer kod proizvodnje, recimo, m - različitih predmeta koji troše iste resurse (raspoloživo vrijeme strugova) čiji je iznos E , s tim što se za svaki predmet troši određena količina resursa r_i kroz određeno vrijeme t_i . Raspoloživi resursi se mogu predstaviti polubeskonačnom trakom, a predmeti kao kvadrati $r_i \times t_i$, sl.1.1.



Sl.1.1.

Optimalno organizirati proizvodnju znači, u polubeskonačnu traku širine E , smjestiti kvadrate zadanih dimenzija $r_i \times t_i = S_i$ tako da svi predmeti S_i budu gotovi za minimalno vrijeme.

Optimalni plan se sastoji u tome da se $\{S_i\}_n$ smjesti u polubeskonačnu traku zadane širine E tako da potrošena duljina trake bude minimalna. Prema tome, kada u buduće budemo govorili o geometrijskim figurama ili objektima, onda to treba shvatiti u širem smislu, tj. da su područja primjene onog o čemu je u ovom radu riječ daleko šira nego se to u prvi trenutak može učiniti.

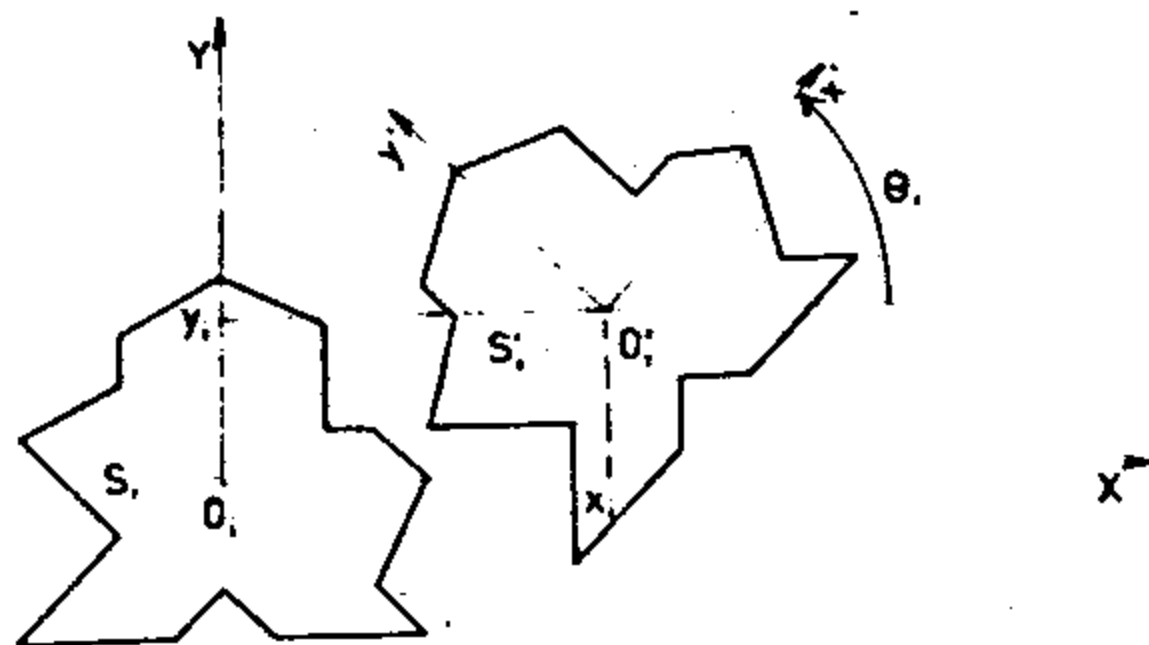
Ako promatramo figure u ravnini R^2 onda uz figure vežemo pomični koordinatni sistem $X'O'Y'$ a uz R^2 , po kojoj se figure premještaju fiksni sistem XOY .

Pretpostavimo da se u početnom položaju objekta S_1 zadanog u $X'O'_1X'$ koordinatni sistemi XOY i $X'O'_1Y'$ podudaraju.

Definicija 1. Ishodište O_1 sistem $X'O_1Y$ zove se pol objekta S_1 [17].

Mada se pol O_1 - objekta S_1 u principu, može birati bilo gdje, mi ćemo uvijek uzimati (iz razloga koji će biti jasni kasnije) da je $O_1 \in S_1$. Isto tako ako figure budu imale neke osi simetrije onda će polovi biti birani na tim osima simetrije, odnosno na presjecima osi simetrije, ako oni postoje.

Kao što vidimo položaj objekta S_i u R^2 potpuno je određen parametrima (x_i, y_i, θ_i) gdje (x_i, y_i) koordinate pola O_i u XO_1Y , a θ_i -kut zarotiranosti objekta.



Sl.1.2.

Likovi kao što su krugovi zadani su sa dva parametra. U slučaju da je područje smještaja neizotropno ili objekti moraju biti međusobno orjentirani, onda je položaj određen s manjim brojem parametara. Mi ćemo se jako puno koristiti upravo zahtjevom da su objekti međusobno orjentirani. Promatrat ćemo samo takve objekte čiji je oblik konfiguracije uvijek isti, tj. zadan je oblikom u početnom položaju. Oblast koju objekt S_i zauzima označimo sa S_i^0 a zadan je jednačinom $f(x,y) \geq 0$, tj. ako je $(x,y) \in S_i$, onda je $f(x,y) \geq 0$, inače je $f(x,y) < 0$. Jednačinom $f(x,y) = 0$ definiramo konturu S_i , i zovemo je kanonska jednačina objekta S_i .

$$\text{Ako je } f_1(x',y') = 0 \quad (1.1.1)$$

onda, obzirom da je

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = (x-x_1; y-y_1) \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix}, \quad (1.1.2)$$

imamo tzv. jednačinu općeg položaja objekta S .

$$f_1[(x-x_1)\cos\theta_1 + (y_1-y_1)\sin\theta_1; -(x-x_1)\sin\theta_1 + (y-y_1)\cos\theta_1] = 0 \quad (1.1.3)$$

Objekti koje ćemo promatrati mogu nastajati na razne načine:

$$\text{a) } S_1 = \bigcup_{i=1}^n S_i; \quad \bigcap S_i \neq \emptyset$$

$$\text{b) } S_1 = \bigcup_{i=1}^n S_i; \quad S_i \cap S_j \neq \emptyset; \quad S_i \setminus S_i \cap S_j \setminus S_j = \emptyset \quad i \neq j \quad (1.1.4)$$

$$\text{c) } S_1 = \bigcup_{i=1}^n S_i; \quad S_i \cap S_j = \emptyset \quad \theta_i - \text{konstantan}$$

Za sva naša razmatranja bitna je podjela na konveksne i nekonveksne objekte. Pokazaće se da ćemo do određenih zaključaka o nekonveksnim objektima dolaziti tako da ih predhodno razbijemo na konveksne dijelove i tako dodjemo do potrebnih pokazatelja.

Definicija 1.2. Pod degeneriranim objektom u R^2 podrazumjevamo

svaki pravac, dužinu, luk, a u R^3 plohu, krivulju ili njihov dio.

Definicija 1.3. Kažemo da je objekt neograničen ako se ne može smjestiti u sferu konačnog radiusa.

Definicija 1.4. Kažemo da je objekt povezan ako za bilo koje dvije točke $x, y \in S$ postoji neprekidna linija iz jedne u drugu koja potpuno leži u S .

Ako u ravnini R^2 povučemo proizvoljni pravac p , onda se R^2 raspada na dvije poluravnine od kojih je ona koja sadrži p - zatvorena.

Definicija 1.5. Zatvorena poluravnina je potporna objektu S ako sa S ima bar jednu dodirnu točku, a sve unutarnje točke S se nalaze s jedne strane p . Pod udaljenošću točke $M \in S_i$ i objekta S koji je zatvoren, podrazumijevamo $\min \rho(M, S_i); S_i \in S$.

Ako su zadana dva zatvorena objekta S_1 i S_2 onda se pod njihovom udaljenošću podrazumjeva $\min \rho_{12}(S_i, S_j) (S_i \in S_1; S_j \in S_2)$.

Ako su ispunjeni uvjeti:

$$S_1 \setminus S_1 \cap S_2 \setminus S_2 \neq \emptyset \text{ uzimamo da je } \rho_{12}(S_1, S_2) < 0$$

$$S_1 \cap S_2 = S_1 \cap S_2 \neq \emptyset \quad " \quad \rho_{12}(S_1, S_2) = 0 \quad (1.1.5)$$

$$S_1 \cap S_2 = \emptyset \quad " \quad \rho_{12}(S_1, S_2) > 0$$

Ako je zadana oblast Ω i objekti $S_i (i=1, 2, \dots, n)$, onda kažemo da

su S_i smješteni u Ω ($\bigcup_{i=1}^n S_i \subset \Omega$)

ako su ispunjeni uvjeti:

$$\bigcup_{i=1}^n S_i \cap \Omega = \bigcup_{i=1}^n S_i \quad (1.1.6)$$

$$S_i \cap S_j = \beta_i \quad (1.1.7)$$

(β_i - prazan ili se sastoji od rubnih točaka).
Uvjete (1.1.6) možemo zapisati i na slijedeći način

$$\rho_i(S_i, R^2 \setminus \Omega) \geq 0 \text{ za } \forall_i \quad i = 1, n, \quad (1.1.8)$$

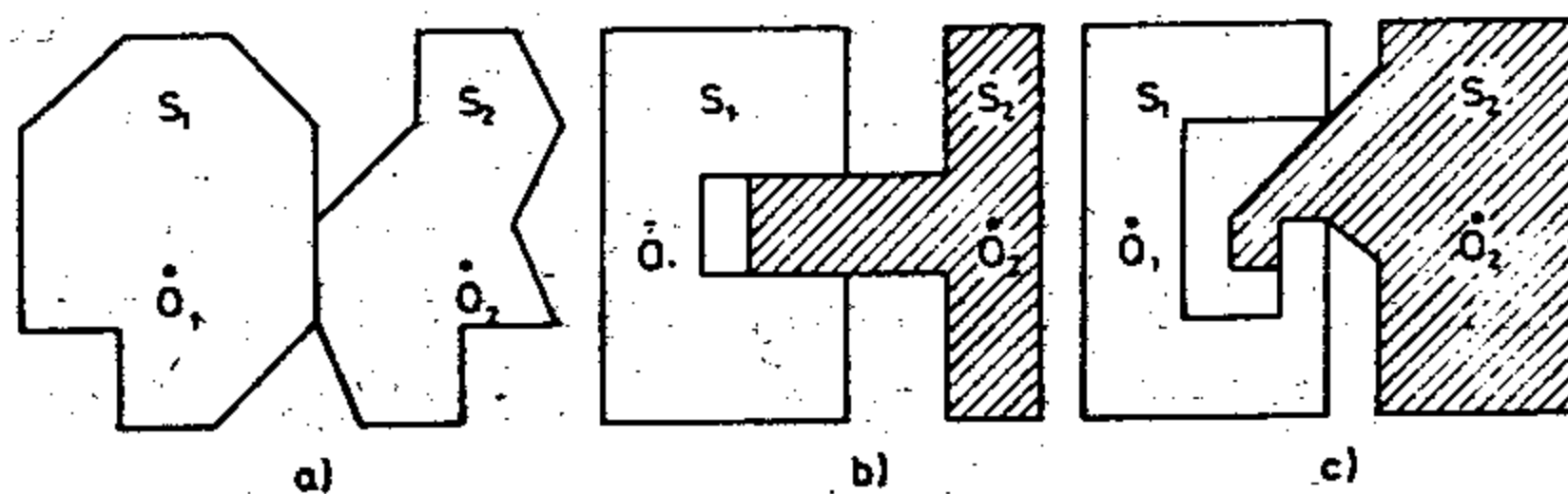
Pod dijametrom zatvorenog objekta S podrazumijevamo:

$$d(S) = \max_p [a(x,y), b(x,y)] \quad a, b \in S.$$

Neka su zadani objekti S_1 i S_2 s parametrima smještaja (x_1, y_1, θ_1) i (x_2, y_2, θ_2) . Kroz polove $O_1(x_1, y_1)$ i $O_2(x_2, y_2)$ povučemo pravac p . Uzmemo da je smjer definiran sa vektorom \vec{p}_{12} .

Definicija 1.6. Kažemo da su objekti u kontaktu ako je $\rho_{12}(S_1, S_2) = 0$, odnosno $h_{12}(S_1, S_2) = l_{12}$ ako je l_{12} - zadana minimalna udaljenost medju objektima S_1 i S_2 .

Definicija 1.7. Za objekte S_1 i S_2 kažemo da su razdjeljivi ako postoji takav položaj polova O_1 i O_2 da kod premještanja iz kontakt rasporeda S_1 ili S_2 u bilo kojem od smjerova \vec{p}_{12} (\vec{p}_{21}), vrijedi uvjet $\rho_{12}(S_1, S_2) > 0$; $\rho_{21}(S_1, S_2) > 0$. Znači da kod premještanja nema ni diranja ni presjecanja. Na sl.1.3a vidimo razdjeljive objekte, a na sl.1.3b nerazdjeljive s točnim ulazom - diranjem, i na sl.1.3c nerazdjeljive.



Sl.1.3.

Tvrđnja 1.1 Konveksni ograničeni objekti su razdjeljivi.

Definicija 1.8. Kažemo da je objekt S_i zvjezdast [18] s obzirom na pol O_i ako svaka spojnica O_i sa proizvoljnom rubnom točkom potpuno leži u S_i . Ako postoji kontinualni skup Ω takvih točaka, onda se Ω - naziva zvjezdasta oblast.

Tvrđnja 1.2 Zvjezdasta oblast je konveksna.

Tvrđnja 1.3 Ograničeni zvjezdasti objekti su razdjeljivi.

Tvrđnja 1.4 Ako jedan od objekata S_1 ili S_2 nije ograničen onda su takvi objekti nerazdjeljivi.

2. R- funkcije

Ovim funkcijama često ćemo se koristiti, pa zato o njima dajemo elementarne informacije [43].

Kada vršimo neke operacije nad realnim varijablama, trebamo voditi računa o njihovom predznaku. No ima dosta primjera gdje je predznak rezultata operacije potpuno određen predznakom veličina nad kojima se operacija vrši. Tako predznak produkta ne zavisi od modula x i y već samo od njihovih predznaka. Shvatimo predznake kao binarne varijable, te neka plusu odgovara 1 a minusu 0. Tada imamo u mnogim slučajevima da operacijama nad realnim veličinama, odgovaraju neke operacije nad binarnim veličinama dakle Bulove funkcije. Iz skupa funkcija realnih argumenata izdvojimo klasu funkcija tzv. R-funkcije, koje imaju navedena svojstva.

Definicija 2.1. Za točku $M(x_1, x_2, \dots, x_n)$ kažemo da je degenerirana ako joj je bar jedna koordinata jednaka nuli.

Degenerirana točka u kojoj je $x_k = 0$ leži na koordinatnoj hiperravnini $x_k = 0$. Ako su $x_i = 0$ i $x_j = 0$, onda se točka nalazi na presjeku koordinatnih hiperravnina $x_i = 0$, $x_j = 0$.

Skup svih degeneriranih točaka iz R^n predstavlja uniju od n - hiperravnina $x_i = 0$ ($i = 1, n$), koje možemo promatrati kao jedinstvenu hiperravninu (H). Očito da H razbija R^n na 2^n oblasti H_j ($j = 1, 2^n$). Svakoј nedegeneriranoј točki $M(x_1, x_2, \dots, x_n) \in R^n$ odgovara određen niz binarnih varijabli X_1, X_2, \dots, X_n , definiranih formulama.

$$X_i = S(x_i) \quad (1.2.1)$$

gdje je S dvoznačni predikat koji x_i - pridružuje 0 ili 1. Tih nizova binarnih veličina X_1, X_2, \dots, X_n može biti 2^n , pa se skup svih nedegeneriranih točaka razbija na 2^n podskupova. Bilo kojoj točki jednog takvog podskupa odgovara jedan te isti niz binarnih veličina X_1, \dots, X_n .

U R^2 imamo tako četiri kvadranta H_j ($j = 1, 2, 3, 4$).

Definicija 2.2. Neka je $y = f(x_1, x_2, \dots, x_n)$ funkcija definirana svuda u R^n . Za navedenu funkciju kažemo da je R-funkcija ako u svakoj oblasti H_j ($j = 1, 2^n$) ima isti predznak tj.

$$S[f(x_1, x_2, \dots, x_n)] = F_i = \text{const} \quad (1.2.2)$$

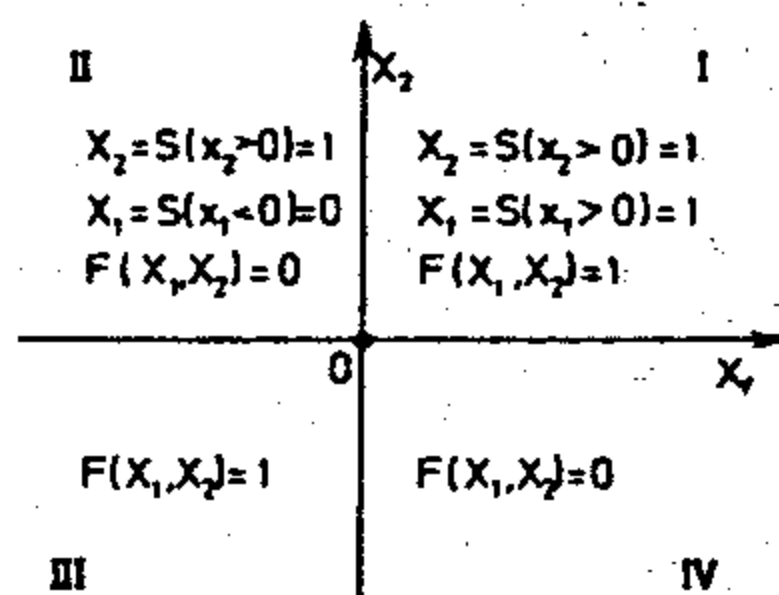
gdje je F_i - binarna veličina ista za sve točke H_j iz $(j=1, 2^n)$.
 Budući da svakom nizu binarnih veličina $X_i = S(x_i)$ odgovara od-
 redjena oblast H_j , znači i veličina F_i , dobijemo neku Bulovu
 funkciju $F(X_1, X_2, \dots, X_n)$, takvu da vrijedi jednažba

$$S[f(x_1, x_2, \dots, x_n)] = F[S(x_1), S(x_2), \dots, S(x_n)]. \quad (1.2.3)$$

Izraz (2.3) predstavlja nužan i dovoljan uvjet da funkcija
 $y = f(x_1, x_2, \dots, x_n)$ bude R-funkcija.

Tako je na primjer $y = x_1 \cdot x_2$ R-funkcija. Naime, predznak produk-
 ta je konstanta u svakom od četiri kvadranta H_j .

Pripadna Bulova funkcija $Y = F(X_1, X_2)$ jednaka je 1 u I i III
 kvadrantu, a nula u II i IV kvadrantu sl.1.4.



Sl.1.4.

Prema tome dobili smo $Y(X_1, X_2) = X_1 \sim X_2$ tj. da produktu $x_1 \cdot x_2$
 odgovara Bulova operacija ekvivalencije.

Tako je funkcija $Y = x_1^2 + x_2^2 + (x_2 - x_1) \sqrt{x_1^2 + x_2^2} \in R$ i njoj
 odgovara operacija implikacije $Y = X_1 \Rightarrow X_2$.

Može se pokazati da vrijede slijedeći izrazi:

R-negacija

$\overline{\phi(x, y)} = -\phi(x, y)$ odgovara joj Bulova funkcija \bar{X} . (1.2.4)

R-konjunkcija (zbog kratkoće ispuštamo varijable)

$$\phi_1 \wedge_{\alpha} \phi_2 = \frac{1}{2} [\phi_1 + \phi_2 - \sqrt{\phi_1^2 + \phi_2^2 - 2\alpha \phi_1 \phi_2}] \quad -1 < \alpha \leq 1 \quad (1.2.5)$$

odgovara joj Bulova funkcija $X_1 \wedge X_2$.

R-disjunkcija

$$\phi_1 \vee_{\alpha} \phi_2 = \frac{1}{2} \left[\phi_1 + \phi_2 + \sqrt{\phi_1^2 + \phi_2^2 - 2\alpha\phi_1\phi_2} \right] \quad -1 < \alpha \leq 1 \quad (1.2.6)$$

odgovara joj Bulova funkcija $x_1 \vee x_2$.

Formula (2.3) svakoj R-funkciji $y = f(x_1, x_2, \dots, x_n)$ pridružuje određenu Bulovu funkciju $Y = F(X_1, X_2, \dots, X_n)$. No postoji beskonačno mnogo R-funkcija kojima pripada ista Bulova funkcija. Različitih Bulovih funkcija od n - argumenata ima svega $g = 2^{2^n}$ - pa se skup svih R-funkcija razbija na 2^{2^n} podskupa ($i = 1, 2^{2^n}$) koje nazivamo granama klase \mathcal{R}_K . R-funkcije koje pripadaju istoj grani odgovara jedna te ista Bulova funkcija.

Ako je zadana neka R-funkcija, onda možemo naći njoj pripadnu Bulovu funkciju tako da u svakoj H_j ($j = 1, 2^n$) uzmemo jednu točku i odredimo predznak iste R-funkcije u toj točki. To uradimo u svakoj H_j .

Tako na primjer: $y = x_1 \cdot x_2 + x_3 - (x_1^2 x_2^2 + x_3^2 - x_1 x_2 x_3)^{1/2}$ (*) je složena R-funkcija, nastala od funkcija

$$Y = u + v - (u^2 + v^2 - u \cdot v)^{1/2}$$

$$u = x_1 \cdot x_2$$

$$v = x_3.$$

Da odredimo predznak za izraz (*) uzmemo točke s koordinatama:

1. Kada bismo sastavili tablicu vidjeli bi da je $Y = 1$ za nizove $(0, 0, 1)$ i $(1, 1, 1)$ pa je pripadna Bulova funkcija oblika

$$Y = (\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

Ako je zadana Bulova funkcija $F(X_1, X_2, \dots, X_n)$, onda je formalno uzevši:

$y = f(x_1, x_2, \dots, x_n) = 2F[S(x_1), S(x_2), \dots, S(x_n)] - 1$ (**) pripadna R-funkcija iz grane \mathcal{R}_K ($K = 1, 2^{2^n}$). Naime, $F[S(x_1), \dots, S(x_n)]$ je jednaka 0 ili 1. Ako je jednaka 0, onda je (**) jednaka (-1) ako je jednak 1, onda je (**) jednak 1, slijedi

$$S[f(x_1, x_2, \dots, x_n)] = S\{2F[S(x_1), \dots, S(x_n)] - 1\} = F[S(x_1), \dots, S(x_n)]$$

Potpuni sistemi

Može se pokazati da: $\phi_1 \wedge \phi_2; \phi_1 \vee \phi_2; \bar{\phi}_1$ ili

$$\phi_1 | \phi_2 = \sqrt{\phi_1^2 + \phi_2^2} - \phi_1 - \phi_2 \quad (\text{Šeferova R-funkcija})$$

čine potpune sisteme, tj. da pomoću njih možemo napraviti R-funkciju iz bilo koje unaprijed zadane grane R-funkcija.

Budući da ćemo mi promatrati smještaj objekata u ravnini, navedimo neka svojstva funkcija R_2 .

1. $\phi_1 \wedge_{\alpha} \phi_2 > 0$ ako je $\phi_1 > 0, \phi_2 > 0$
2. $\phi_1 \wedge_{\alpha} \phi_2 = 0$ ako je $\phi_1 \geq 0; \phi_2 = 0$ ili $\phi_1 = 0; \phi_2 \geq 0$
3. $\phi_1 \vee_{\alpha} \phi_2 > 0$ ako je $\phi_1 > 0; \phi_2 > 0; \phi_1 \geq 0; \phi_2 < 0$
 $\phi_1 < 0; \phi_2 \geq 0$
4. $\phi_1 \vee_{\alpha} \phi_2 = 0$ ako je $\phi_1 = 0; \phi_2 = 0; \phi_1 < 0; \phi_2 = 0$
5. $\phi_1 \wedge_1 \phi_2 = \frac{1}{2}[\phi_1 + \phi_2 - |\phi_1 - \phi_2|] = \min_{x, y \in R^2} \{ \phi_1, \phi_2 \}$
6. $\phi_1 \vee_1 \phi_2 = \frac{1}{2}[\phi_1 + \phi_2 + |\phi_1 - \phi_2|] = \max_{x, y \in R^2} \{ \phi_1, \phi_2 \}$
7. $(\phi_1 \wedge_{\alpha} \phi_2) + (\phi_1 \vee \phi_2) = \phi_1 + \phi_2$
8. $(\phi_1 \wedge_{\alpha} \phi_2) \cdot (\phi_1 \vee \phi_2) = \frac{1+\alpha}{2} \phi_1 \cdot \phi_2$
9. $\overline{\phi_1 \wedge_{\alpha} \phi_2} = \bar{\phi}_1 \vee_{\alpha} \bar{\phi}_2$
10. $\overline{\phi_1 \vee \phi_2} = \bar{\phi}_1 \wedge_{\alpha} \bar{\phi}_2$

Tvrđnja 2.1 Ako neprekidna funkcija $y = f(x_1, x_2, \dots, x_n)$ nema nula u otvorenim oblastima $H_j (j = 1, 2^n)$, onda je ona R-funkcija.

Tvrđnja 2.2 Ako su $f_1, f_2 \in R \rightarrow f_1 \cdot f_2 \in R$

Tvrđnja 2.3 Ako su $y = f(u_1, u_2, \dots, u_m)$ i $u_k = \phi_k(x_1, \dots, x_n) \in R$, onda je

$$y = f[\phi_1(x_1, x_2, \dots, x_n), \dots, \phi_m(x_1, x_2, \dots, x_n)] \in R$$

Tvrđnja 2.4 Ako je $y = f(U_1, U_2, \dots, U_m)$ a $U_K = \phi_K(x_1, x_2, \dots, x_n)$ konstantnog predznaka, onda je $y = f[\phi_1(x_1, x_2, \dots, x_n), \dots, \phi_m(x_1, x_2, \dots, x_n)]$ konstantnog predznaka.

Tvrđnja 2.5 Ako je $y = f(x_1, x_2, \dots, x_n) \in R$ integrabilna po varijabli x_1 , onda je funkcija

$$y = \phi(x_1, \dots, x_n) = \int_0^{x_1} f(x_1, \dots, x_n) dx_1 \in R$$

U navedenoj monografiji [43] navedena su i dokazana mnoga druga svojstva R-funkcija na čemu se nećemo zadržavati.

Opisivanje oblasti koristeći predikate, Bulova algebra i R-funkcije

Neka je nejednadžbom $Z = f(x, y) \geq 0$ u ravnini XOY definirana oblast (D). [$f(x, y)$ - svuda definirana i neprekidna].

Uvedimo binarnu varijablu D pomoću predikata

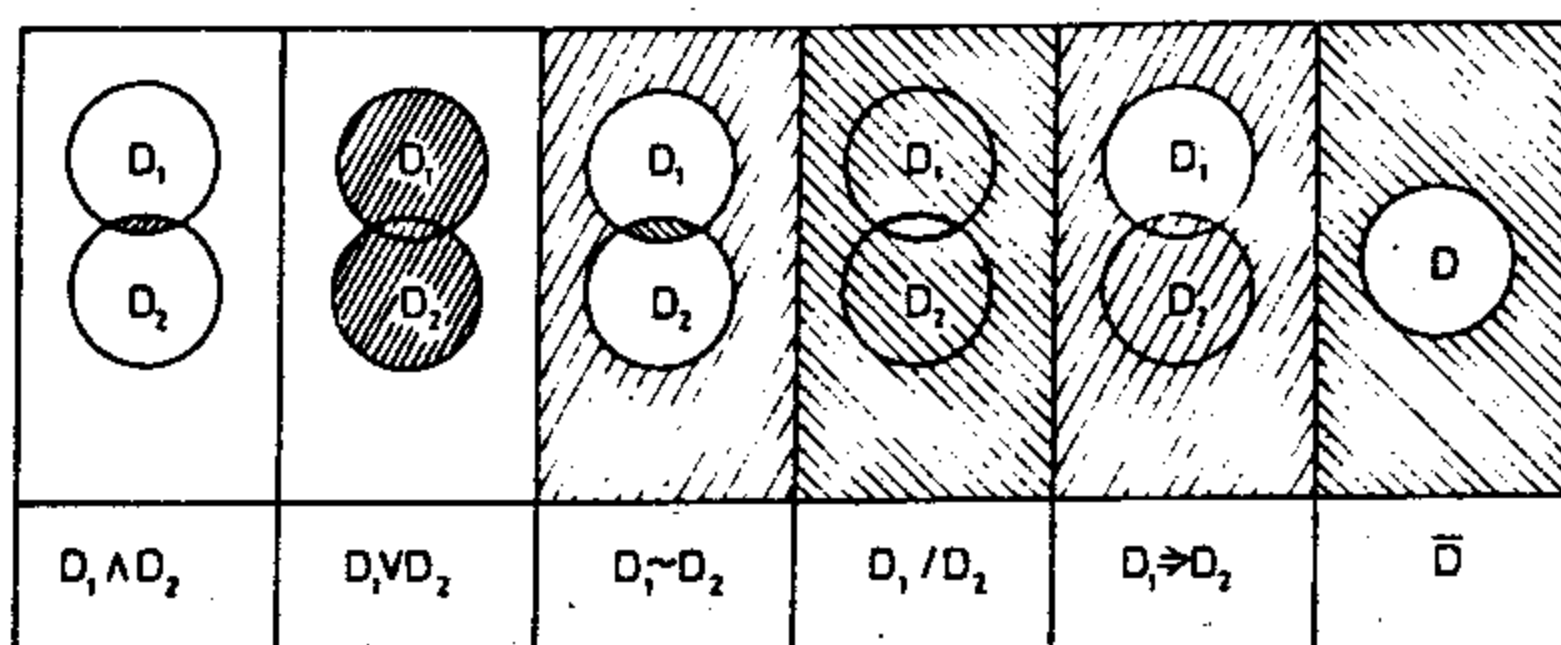
$$D = S[f(x, y)] = [f(x, y) \geq 0].$$

Znači da je $D = 1$ tj. istina ako $M(x, y) \in (D)$ i neistina ako $M(x, y) \notin (D)$.

Neka su (D_1) i (D_2) određene sa $f_1(x, y) \geq 0$ i $f_2(x, y) \geq 0$.

Iz binarnih varijabli D_1 i D_2 sastavimo neku Bulovu funkciju $F(D_1, D_2)$. Već prema tome šta predstavlja $F(D_1, D_2)$ imamo razne slučajeve prikazane na sl.1.5.

Ako je $F(D_1, D_2) = D_1 \wedge D_2$	imamo situaciju prikazanu na a)
" $D_1 \vee D_2$	" b)
" $D_1 \sim D_2$	" c)
" D_1 / D_2	" d)
" $D_1 \Rightarrow D_2$	" e)
" \bar{D}	" f)



Sl.15.

Ako je $Z = \phi_1(u, v) \in R$ kojoj odgovara Bulova funkcija $U \wedge V$,
onda oblast (D) definirana nejednadžbom

$$\psi(x, y) \equiv \phi_1[f_1(x, y); \phi_2(x, y)] \geq 0$$

predstavlja presjek oblasti (D_1) i (D_2) .

Primjer:

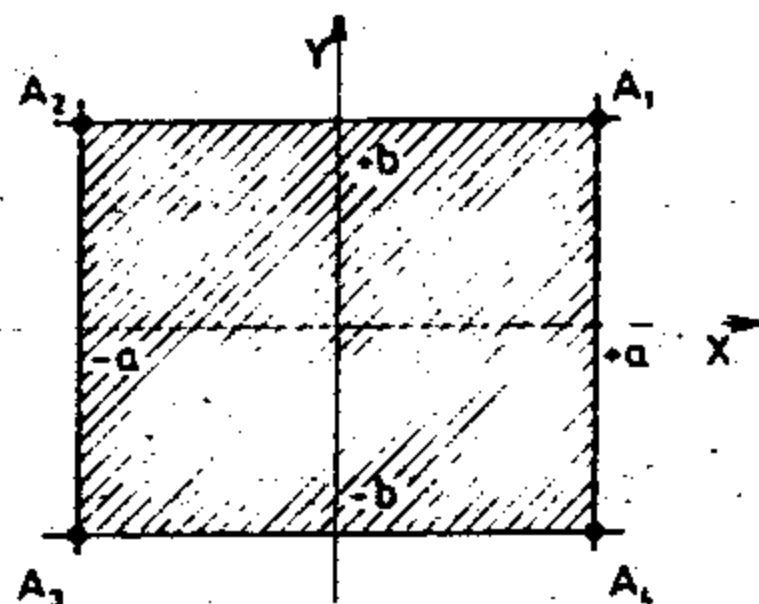
Neka je (D_1) pojas definiran sa $f_1 \equiv a^2 - x^2 \geq 0$.

Neka je (D_2) pojas definiran sa $f_2 \equiv b^2 - y^2 \geq 0$.

Uzmemo li R-konjunkciju $f_1 \wedge_a f_2$, onda nejednadžba

$$(a^2 - x^2) \wedge_a (b^2 - y^2) \geq 0 \quad (1.2.7)$$

definira pravokutnik A_1, A_2, A_3, A_4 , sl.1.6.



Sl.16.

Nejednadžbu (1.2.7) u razvijenom obliku možemo zapisati:

$$\frac{1}{2} [a^2 - x^2 + b^2 - y^2 - \sqrt{(a^2 - x^2)^2 + (b^2 - y^2)^2 - 2\alpha(a^2 - x^2)(b^2 - y^2)}] \geq 0$$

$$-1 < \alpha \leq 1 \quad (1.2.8)$$

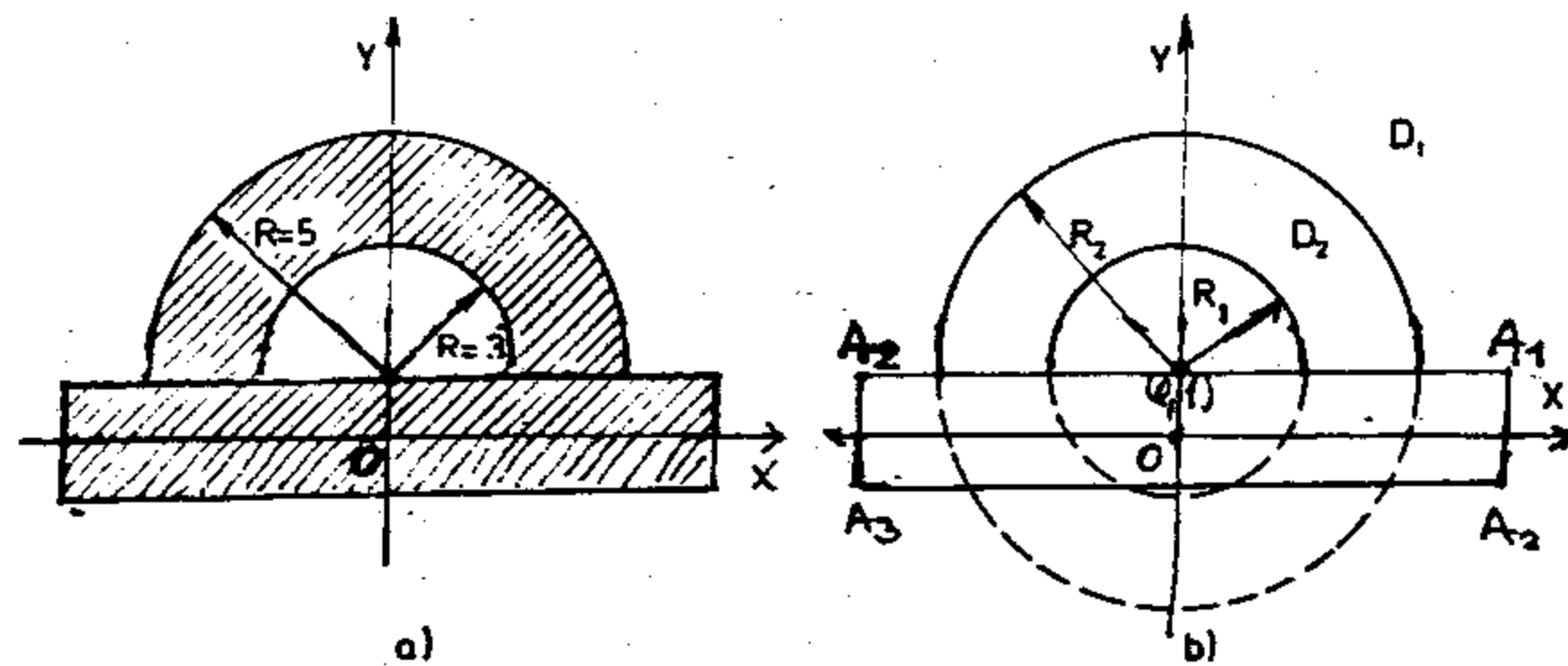
za $\alpha = 1$ imamo:

$$\frac{1}{2} [a^2 - x^2 + b^2 - y^2 - |a^2 - b^2 - x^2 + y^2|] \geq 0 \quad (1.2.9)$$

Kako jednakost vrijedi samo na rubu područja, znači da je jednadžba pravokutnika dana jednim analitičkim izrazom:

$$a^2 + b^2 - x^2 - y^2 - |a^2 - b^2 - x^2 + y^2| = 0. \quad (1.2.10)$$

Recimo da treba napisati jednadžbu granice (D) oblasti koja predstavlja kutomjer, sl.1.7a.



Sl.17.

Oblast D možemo izgraditi pomoću slijedećih oblasti:

- D_1 - zadane sa $f_1 \equiv y - 1 \geq 0$ (poluravnina),
- D_2 - prsten s nutarnjim radiusom $R_1 = 3$ i vanjskim $R_2 = 5$,
- D_3 - pravokutnika A_1, A_2, A_3, A_4 .

Logika formiranja (D) dana je formulom

$$D = (D_1 \wedge D_2) \vee D_3. \quad (1.2.11)$$

$$f_2 \equiv [(9 - x^2 - (y-1)^2)] \cdot [x^2 + (y-1)^2 - 25] \geq 0 \quad (1.2.12)$$

$$f_3 \equiv 37 - x^2 - y^2 - |35 - x^2 + y^2| \quad (1.2.13)$$

Prema tome oblast (D) je dana:

$$f \equiv (f_1 \wedge_1 f_2) \vee_1 f_3 \geq 0. \quad (1.2.14)$$

Ako (1.2.14) napišemo u razvijenom obliku, dobijemo jednadžbu granice oblasti kutomjera:

$$\begin{aligned} & -119+32x^2+26y^2-63y-x^4-2x^2(y-1)^2-y^4+4y^3 - \\ & -|191-34x^2-28y^2+65y+x^4+2x^2(y-1)^2+y^4-4y^3| - \\ & -2|35-x^2+y^2|+|-267+36x^2+30y^2-63y-x^4-2x^2(y-1)^2 - \\ & -y^4+4y^3-|191-34x^2-28y^2+65y+x^4+2x^2(y-1)^2+y^4-4y^3| \\ & +2|35-x^2+y^2|| = 0. \end{aligned}$$

Iz navedenog primjera uočavamo da nam R-funkcije omogućavaju da jednim analitičkim izrazom opišemo vrlo složene konture (granice) raznih objekata. Vidimo da su izrazi vrlo glomazni, ali zato se urade programi za računar koji generiraju potrebne izraze.

3. Opća matematička formulacija problema smještaja geometrijskih objekata

U ovoj radnji će se promatrati dvije grupe problema:

- I. Regularni smještaj objekata
- II. Neregularni smještaj objekata

Definicija 3.1

Kažemo da su objekti regularno smješteni ako kod parametara smještaja postoji neka zakonitost, a kod neregularnih te zakonitosti nema.

Od regularnih promatrat ćemo:

- I-1 PERIODIČNE RASPOREDE
- I-2 REŠETKASTE RASPOREDE

Naime, ako je $\{S_i\}_n$ skup objekata u R^n , onda pod periodičnim smještajem podrazumjevamo rasporede kod kojih bar jedan parametar objekta S_i ($i = 1, n$), mijenja svoju vrijednost od jednog do drugog položaja za istu veličinu.

Periodične rasporede dijelimo u dvije grupe:

- I-1-A. Jednoredne periodične rasporede
- I-1-B. N-redne periodične rasporede.

Kod jednorednih periodičnih smještaja pojavljuju se u praksi problemi:

- I-1-A- a) - Smještanje u beskonačnu traku
- b) - Smještanje u traku zadane širine
- c) - Smještanje u traku zadane duljine
- d) - Smještanje u pravokutnu oblast.

Kod N-rednog ista je podjela tako da se na jednoj liniji - pravcu nalaze objekti samo jednog tipa, raznim objektima odgovaraju različiti periodi a pravci su kongruentni. Medjutim, što se tiče promjene parametara θ , promatraju se takvi slučajevi da se vrši promjena parametara θ ^{ZA ISTU VRIJEDNOST} u svim redovima.

Kod svih problema krojenja funkcija cilja je uvijek maksimalno iskorištenje. Uvijek se uzima da su udaljenosti između objekata

ta, objekata i oblasti te objekata i oblasti zabrane veće ili jednake nuli.

Kod rešetkastih rasporeda promatraju se dvije grupe problema:

I-2-A - raspored u ravnini

I-2-B - raspored u pravokutniku.

Bez obzira što se pokazuje da su periodični rasporedi specijalni slučaj rešetkastih rasporeda, metode rješavanja nisu iste pa se zato posebno promatraju.

Neregularni smještaji

Kod ovih problema važno je objekte S_i ($i=1, n$) smjestiti u neku oblast Ω a da se postigne optimalna vrijednost funkcije cilja.

Tu se pojavljuju slijedeći slučajevi:

- II-1 Oblast Ω u koju se smještaju objekti $\{S_i\}$ ima zadan oblik i dimenzije.
- II-2 Ω - ima zadani oblik a dimenzije se određuju homotetskim transformacijama Ω .
- III-3 I oblik i dimenzija Ω su promjenljivi tj. najopćenitiji problem koji prethodna dva obuhvaća kao specijalne slučajeve.

U vezi sa II-1 obično se postavljaju slijedeći problemi:

- a) Iz zadanog skupa objekata - elemenata $\{S_i\}_n$ izabrati onaj koji se najbolje smješta u dani Ω .
- b) Imamo različite skupove $\{S_i\}_n \in \{\{S_i\}_n\}_m$ i treba najveći mogući broj tih objekata smjestiti u Ω . Ovakvi se problemi susreću u lakoj i teškoj industriji, mostogradnji, brodogradnji, zrakoplovnoj industriji, drvnoj, tekstilnoj, kožarskoj itd. industrijama.
- c) Treba u Ω , koja općenito uzevši može biti i višestruka, smjestiti $\{S_i\}_n$ tako da zahtjevi na smještaj objekata budu najbolje zadovoljeni. Recimo:

- da odstupanje centra težišta objekata $\{S_i\}$ smještenih u Ω od zadane točke $C(X_0, Y_0)$ bude minimalno
- da izvori topline budu optimalno raspoređeni
- da duljina mreže koja povezuje objekte $\{S_i\}_n$, bude minimalna (električne, telefonske, plinske i putne)
- da objekti na ploči budu tako raspoređeni da imamo minimalni progib itd.

U vezi s oblastima s pomičnim granicama obično se postavljaju slijedeći problemi:

- u oblasti Ω zadanog oblika, ali ne i veličine, treba smjestiti $\{S_i\}_n$ tako da dimenzije Ω na najbolji način zadovoljavaju zadane uvjete. Najjednostavniji primjer je problem kako zadani objekt S_i smjestiti u pravokutnik minimalne površine.

Kada za Ω - nije dan ni oblik ni veličina, onda se obično postavljaju problemi kao na primjer:

treba konstruisati Ω pomoću linije $f_i(x,y) \geq 0$ ($i=1,k$) tako da se objekti $\{S_i\}_n$ smjeste u Ω a da su na najbolji način zadovoljeni postavljeni uvjeti.

U svim navedenim problemima na parametre smještaja objekata često se postavljaju razni uvjeti kao što su:

- minimalne udaljenosti između objekata;
- minimalne udaljenosti između objekata i granice oblasti Ω ;
- minimalne udaljenosti između objekata i područja zabrane.

Navedeni uvjeti su uglavnom rezultat protupožarnih, tehnoloških, procesnih, transportnih, neizotopnih uvjeta itd.

Medjutim svi navedeni problemi imaju nešto zajedničko:

- a) - problemi su takvi da se radi o smještaju objekata proizvoljna oblika u oblasti proizvoljna oblika,
- b) - svaki od navedenih problema formulira se, bar formalno, kao problem matematičkog programiranja. Naime, problemi su izraženi u obliku nekih uvjeta koji predstavljaju ograničenja na parametre smještaja poštivanjem kojih treba postići optimalno rješenje koje zavisi od parametara smještaja.

Ti odnosi su definirani funkcijom cilja. Vidimo da je struktura problema smještanja objekata takva da njihova formulacija (analitički opis) u sebe obavezno uključuje:

- A) - uvjete da se objekti uzajamno ne presjecaju, ili da se nalaze na zadanoj udaljenosti l_{ij} (UMN);
- B) - uvjete da su objekti smješteni u Ω s tim da se poštuju zadane udaljenosti između S_i i S_j , objekta S_i i granice oblasti Ω , objekta i oblasti zabrane K_t ;
- C) - formalno predstavljanje funkcije cilja pomoću matematičkih simbola i zavisnosti.

Naime, recimo da imamo ograničene objekte $\{S_i\}_n$ i zatvorenu oblast Ω u koju treba smjestiti $\{S_i\}_n$ na najbolji način s obzirom na zadane kriterije.

Tada uvjete A i B možemo formalno zapisati

$$S_i \setminus S_i \cap S_j \setminus S_j = \emptyset; \quad i, j = 1, n; \quad i \neq j \quad (1.3.1)$$

odnosno uvjete B

$$\bigcup_{i=1}^n S_i \cap \Omega = \bigcup_{i=1}^n S_i \quad (1.3.2)$$

ili pomoću udaljenosti

$$\rho_{ij}(S_i, S_j) \geq 0 \quad i, j = (1, n); \quad i \neq j \quad (1.3.3)$$

odnosno

$$\rho_i(R^2 \setminus \Omega, S_i) \geq 0; \quad i = (1, n) \quad (1.3.4)$$

R^2 - XOY.

Za slučaj kada su zadane minimalne udaljenosti l_{ij} između S_i i S_j ili l_i - između S_i i granice Ω , onda uvjeti (1.3.1.)-(1.3.4) imaju oblik:

$$\rho_{ij}(S_i, S_j) \geq l_{ij} \quad i, j = 1, n; \quad i \neq j \quad (1.3.5)$$

$$\rho_i(R^2 \setminus \Omega, S_i) \geq l_i \quad i = (1, n) \quad (1.3.6)$$

Prema tome iz (1.3.1.)-(1.3.6) vidimo da UMN treba da su predstavljeni pomoću nekakvih nejednadžbi oblika:

$$\phi_{ij}(x_i, y_i, \theta_i; x_j, y_j, \theta_j) \geq l_{ij} \quad i \neq j; \quad i, j = (1, n) \quad (1.3.7)$$

$$\phi_i(x_i, y_i, \theta_i) \geq 0 \quad (1.3.8)$$

Ove nejednadžbe s jedne strane međusobno "povezuju" parametre objekata, a s druge strane na te parametre stavljaju ograničenja, tj. ograničenja na slobodu smještaja objekata. Na taj način mi dolazimo do sistema od $M = \frac{1}{2}n(n-1) + n$ nejednadžbi, koje u R^2 - općenito povezuju $3n$ parametara (x_i, y_i, θ_i) objekata $\{S_i\}_n$ - koje smještamo (n - broj objekata). Funkcija cilja ima vrlo različite oblike za razne probleme, ali uvijek zavisi od parametara smještaja tj. oblika su:

$$K(x, y, \theta) = K(x_1 x_2 \dots x_n, y_1 y_2 \dots y_n, \theta_1 \theta_2 \dots \theta_n). \quad (1.3.9)$$

Prema tome nameću se slijedeći zaključci:

- struktura formalnog opisa problema smještanja, pomoću matematičkih simbola, sastoji se iz sistema nejednadžbi, i funkcije ^{cilja} čija nas vrijednost interesira, ali samo na oblastima definiranim tim sistemom nejednadžbi (OGRAIČENJA).
- rješavanje bilo kojeg od navedenih problema svodi se na optimiziranje funkcije cilja (1.3.9) na oblastima definiranim sa (1.3.7) i (1.3.9).

Ukoliko rješenje postoji, onda to trebaju biti parametri smještaja (x_i, y_i, θ_i) , a budući da znamo jednadžbe kontura objekata (mi promatramo kongruentne transformacije) imamo fizikalni model smještaja, tj. riješen problem.

Uvjeti nepresjecanja (1.3.7) i (1.3.8) su opći za sve probleme, dok funkcija cilja ima specifičnosti od problema do problema pa se zato za svaki problem posebno definira.

G L A V A I I

KONTAKT FUNKCIJA I NJEN HODOGRAF

1. Svojstva kontakt funkcija i njenog hodografa

Neka su zadana dva objekta S_1 i S_2 čiji su parametri smještaja (x_1, y_1, θ_1) i (x_2, y_2, θ_2) . Neka su O_1 (O_2) polovi objekata S_1 (S_2).

Definicija 2.1. Kažemo da su objekti u kontaktu ako su im zajedničke samo granične točke tj. ako je:

$$\begin{aligned} S_1 \setminus s_1 \cap S_2 \setminus s_2 &= \emptyset \\ S_1 \cap S_2 &= \emptyset \end{aligned} \quad (2.1.1)$$

Definicija 2.2. Ako je medju objektima zadana minimalna udaljenost l_{12} koju oni moraju zadovoljiti, onda kažemo da su S_1 i S_2 u kontaktu ako vrijedi $\rho_{12}(S_1, S_2) = l_{12}$. Ovaj slučaj svodi se na prvi jednostavnim pomakom granice objekta S_1 ili S_2 za l_{12} tj.

$$\begin{aligned} S_1 \setminus s_1 \cap S_2^{l_{12}} \setminus s_2 &= \emptyset \\ S_1 \cap S_2 &\neq \emptyset \end{aligned} \quad (2.1.2)$$

ili

$$\begin{aligned} S_1^{l_{12}} \setminus s_1 \cap S_2 \setminus s_2 &= \emptyset \\ S_1 \cap S_2 &= \emptyset \end{aligned} \quad (2.1.2)$$

Neka je S_1 zadana u $X'O_1Y'$ a S_2 u $X''O_2Y''$. Sa ravninom R^2 po kojoj se S_1 i S_2 premještaju neka je vezan fiksni sistem XOY. Smjestimo S_1 tako da se XOY i $X'O_1Y'$ podudare. Uzmemo da je S_1 fiksiran a da se S_2 može premještati u R^2 , ali tako da je uvijek u kontaktu sa S_2 . Povučemo li se vektor \vec{p}_{12} iz pola O_1 s vrhom u O_2 , onda se duljina vektora \vec{p}_{12} mijenja kad se on zarotira za neki kut β . U ovom slučaju parametri smještaja su (x_2, y_2, θ_2) jer su S_1 i S_2 u kontaktu. Skup navedenih točaka u R^2 definira neku

plohu zadanu jednađbom $F_{12}(x_2, y_2, \theta_2) = 0$.

Medjutim, ako S_1 nije fiksna, onda svakom kontakt rasporedu S_1 i S_2 odgovara 6 parametara $(x_1, y_1, \theta_1, x_2, y_2, \theta_2)$. Skup svih tih točaka čini određenu plohu u R^6 , iskazanu izrazom:

$F_{12}(x_1, y_1, \theta_1, x_2, y_2, \theta_2) = 0$. Takva je ploha jedinstvena za zadani par objekata.

Može se pokazati da vrijedi:

$$F_{12}(x_1, y_1, \theta_1, x_2, y_2, \theta_2) = Z_{12}(x_1 - x_2; y_1 - y_2; \theta_1 - \theta_2) = 0 \quad (2.1.4)$$

ili

$$F_{21}(x_1, y_1, \theta_1, x_2, y_2, \theta_2) = Z_{21}(x_2 - x_1; y_2 - y_1; \theta_2 - \theta_1) = 0, \quad (2.1.5)$$

tj. da udaljenost medju polovima O_1 i O_2 objekata u kontaktu ne zavisi od njihovog položaja u ravnini.

Stavimo li $x = x_1 - x_2$ ($x_2 - x_1$); $y = y_1 - y_2$ ($y_2 - y_1$); $\theta = \theta_1 - \theta_2$ ($\theta_2 - \theta_1$),

(2.1.6)

tada iz (2.1.4) i (2.1.5) dobijemo

$$\begin{aligned} F_{21}(x_1, y_1, \theta_1, x_2, y_2, \theta_2) &= Z_{21}(x, y, \theta) \\ F_{12}(x_1, y_1, \theta_1, x_2, y_2, \theta_2) &= Z_{12}(x, y, \theta) \end{aligned} \quad (2.1.7)$$

U polarnom obliku (2.1.7) dana je

$$Z_{12}(x, y, \theta) = Z_{12}^*(\rho_{12}, \theta, \beta), \quad (2.1.8)$$

gdje je: $\beta = \arctg \frac{y}{x}$; $\rho_{12} = (x^2 + y^2)^{1/2}$.

Pretpostavimo da je moguće eksplicite izaziti ρ_{12} ne samo za krugove već i za proizvoljne objekte, tj. da je

$$\rho_{12} = K(\theta, \beta). \quad (2.1.9)$$

Koristeći (2.1.6) iz (2.1.9) dobijemo

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 - K_{12}^2(\theta_1 - \theta_2, \beta) = 0; \quad \beta = \arctg \frac{y_1 - y_2}{x_1 - x_2}. \quad (2.1.10)$$

Na osnovi toga dolazimo do definicije:

Definicija 2.3. Kontakt funkcija (KF) objekata S_1 i S_2 predstavlja zavisnost udaljenosti medju polovima O_1 i O_2 objekata u zavisnosti od njihovog medjusobnog položaja. Znači da KF predstavlja duljinu vektora $\vec{\rho}_{12}$ danog u (2.1.9) [47]

Definicija 2.4. Hodograf vektora ρ_{12} nazivamo hodografom vektora kontakt funkcije HKF [12] [13].

Mi ćemo se ograničiti na slučaj kada su objekti medjusobno orijentirani tj. kada je $\theta_1 - \theta_2 = \theta^1 = \text{const}$ ($\theta_2 - \theta_1 = \theta^2 = \text{const}$).

U takvom slučaju HKF u R^2 predstavlja nekakvu krivulju.

Ako S_1 fiksiramo a S_2 oko njega obilazi, s tim da su stalno u kontaktu, onda HKF u stvari predstavlja geometrijsko mjesto točaka položaja pola O_2 . Ta krivulja predstavlja granicu neke oblasti G_{12} s polom $O_{12} = O_1$ a ima slijedeća svojstva: ako se pol O_2 smjesti unutar G_{12} objekti se sijeku, ako se nalazi na granici objekti su u kontaktu, i ako je O_2 van G_{12} objekti se ne presjecaju i ne diraju.

Vidimo da vrijednost kontakt funkcije dobijemo tako da svakoj točki HKF pridružimo njenu udaljenost do pola $O_{12} = O_1$.

Navedene tvrdnje i definicije bit će stalno korištene kod definiranja uvjeta medjusobnog nepresjecanja (UMN) objekata. Ovi uvjeti se javljaju u svim problemima koje rješavamo u ovoj disertaciji.

Navedimo neka svojstva KF i HKF

$$\begin{aligned} K_{12}(\theta, \beta) &= K_{12}(\theta + 2k\pi, \beta + 2t\pi) \\ Z_{12}(x, y, \theta) &= Z_{12}(x, y, \theta + 2t\pi) \end{aligned} \quad (2.1.11)$$

Teorem 2.1.

Ako su S_1 i S_2 medjusobno orijentirani, onda je

$$\begin{aligned} Z_{12}(x, y, \theta) &= H_{12}(x, y) = 0 \\ Z_{21}(x, y, \theta) &= H_{21}(x, y) = 0. \end{aligned} \quad (2.1.12)$$

Prema tome HKF $H_{12}(x, y) = 0$ ($H_{21}(x, y) = 0$) predstavlja krivulje koje obuhvaćaju S_1 (S_2) i čine geometrijsko mjesto polova O_2 (O_1) kada su S_1 i S_2 u kontaktu.

Teorem 2.2.

Vrijede slijedeće jednačbe [13]

$$Z_{12}(x, y, \theta) = Z_{21}(-x, -y, -\theta) = 0 \quad (2.1.13)$$

$$H_{12}(x, y, \theta) = H_{21}(-x, -y, -\theta) = 0. \quad (2.1.14)$$

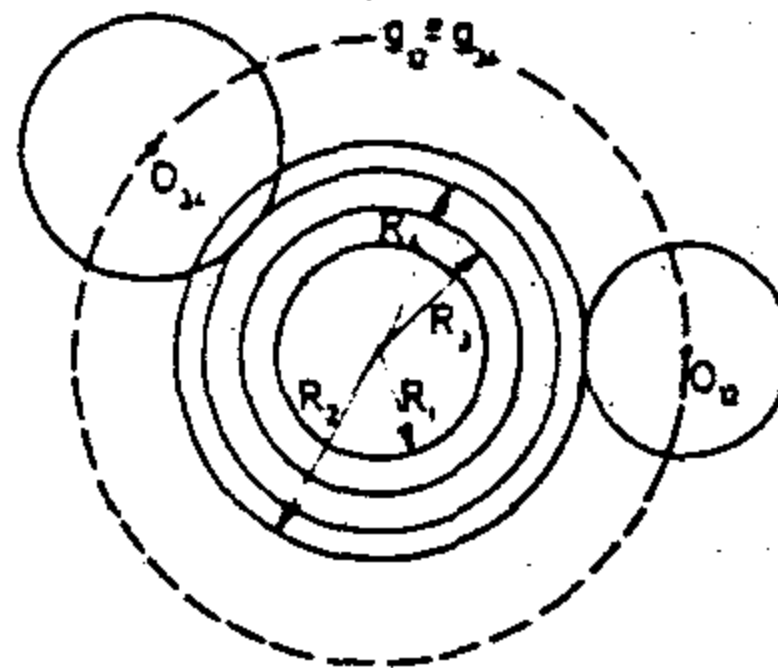
To znači da se Z_{21} i H_{21} jednostavno dobiju iz Z_{12} i H_{12} .
 Kod nekih problema korištenje prednjeg ima za posljedicu znatne uštede kompjutorskog vremena.

Tvrđnja 2.1.

Ako je bar jedan od objekata neograničen, onda HKF predstavlja granicu neograničene oblasti.

Tvrđnja 2.2.

Hodograf kontakt funkcije objekata S_1 i S_2 je jedinstven. Obrat općenito ne vrijedi. Tako npr. neka su zadana dva kruga radijusa R_1 i R_2 sl.2.1.



Sl.2.1.

Ovim krugovima pripada HKF - g_{12} . Međutim vidimo da je g_{12} također hodograf krugova čiji su radijusi R_3 i R_4 .

Znači da jednom g_{12} može pripadati beskonačno mnogo parova objekata, tj. za sve slučajeve gdje vrijedi $R_i + R_j = R_1 + R_2$.

Teorem 2.3.

Ako je $Z_{12}(x_1 - x_2, y_1 - y_2, \theta_1 - \theta_2) < 0$, (2.1.16)
 onda je udaljenost $\rho(S_1, S_2) < l_{12}$.

Ako je $z_{12}(x_1-x_2, y_1-y_2, \theta_1-\theta_2) = 0,$ (2.1.17)
 onda je $\rho(S_1, S_2) = \ell_{12}.$

Ako je $z_{12}(x_1-x_2, y_1-y_2, \theta_1-\theta_2) > 0,$ (2.1.18)
 onda je $\rho(S_1, S_2) > \ell_{12}.$

Znači da su uvjeti medjusobnog nepresjecanja dani izrazima

$$z_{12}(x_1-x_2, y_1-y_2, \theta_1-\theta_2) \geq 0 \quad (2.1.19)$$

$$z_{21}(x_2-x_1, y_2-y_1, \theta_2-\theta_1) \geq 0. \quad (2.1.20)$$

2. Kontakt funkcija konveksnih poligona i njen hodograf

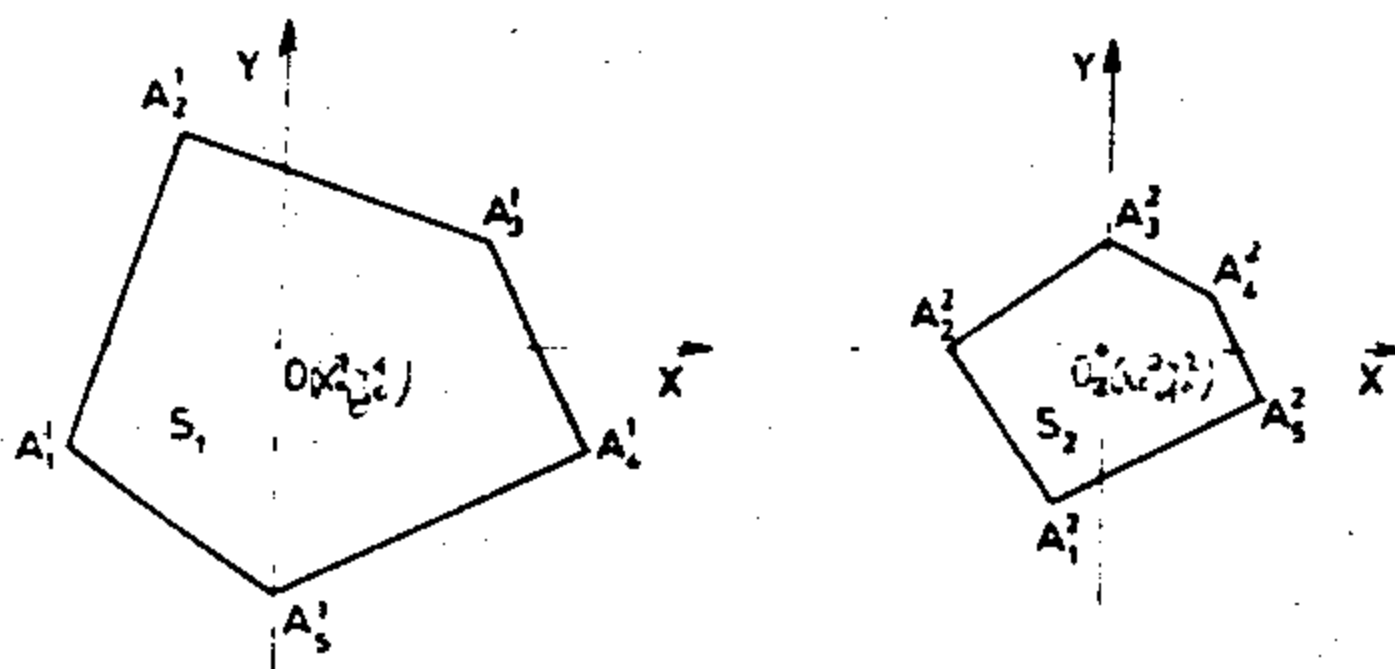
Neka su zadana dva konveksna međusobno orijentirana poligona S_1 i S_2 [9].

Znamo da konveksni poligon S_1 možemo predstaviti kao presjek poluravnina P_r^1 ($r = 1, k$), odnosno S_2 pomoću P_s^2 ($s = 1, q$). Neka su jednačbe pravca p_r (p_s) koji određuju P_r^1 (P_s^2) dane izrazima:

$$f_r^1(x, y) = a_r^1 x + b_r^1 y + c_r^1 = 0 \quad (r = 1, k) \quad (2.2.1)$$

$$f_s^2(x, y) = a_s^2 x + b_s^2 y + c_s^2 = 0 \quad (s = 1, q). \quad (2.2.2)$$

Znači da su stranice poligona djelovi pripadanih pravaca sl.2.2



Sl.2.2.

Jednostavnosti radi uzmimo da su polovi $O_1(x^01, y^01) \in S_1$ ($O_2(x^02, y^02) \in S_2$) i da su vrhovi zadani koordinatama u fiksnom koordinatnom sistemu XOY i to $A_r^1(x^1r, y^1r)$, $A_s^2(x^2s, y^2s)$. Orijetiramo pravce p_r^1 i p_s^2 tako da bude:

$$a_r^1 x^01 + b_r^1 y^01 + c_r^1 < 0 \quad (r = 1, k)$$

$$a_s^2 x^02 + b_s^2 y^02 + c_s^2 < 0 \quad (s = 1, q)$$

Ako polóvi O_1 i O_2 leže unutar S_1 (S_2), onda je njihov otklon od pravca (otklon \neq udaljenost) uvijek negativan, tj. imamo

$$f_r^1(x^01, y^01) < 0, \quad f_s^2(x^02, y^02) < 0.$$

Smjestimo objekte tako da im se polovi poklope sl.2.2. Uočimo onaj vrh $A_s^2(x^2s, y^2s)$ objekta S_2 koji je najviše udaljen od pravca p_r^1 , a koji se nalazi s iste strane s koje i pol $O_1 = O_2$.

Označimo njegovu udaljenost sa d_{1r} . Isto tako uočimo vrh $A_r^1(x^{1r}, y^{1r}) \in S_1$ koji je najviše udaljen od pravca p_s^2 , a leži s iste strane s koje i pol O_2 .

Prema tome imamo da je

$$d_{1r} = \max \frac{|a_r^1 x^{2s} + b_r^1 y^{2s} + c_r^1|}{[(a_r^1)^2 + (b_r^1)^2]^{1/2}} \quad (r = 1, k) \quad (2.2.3)$$

za one s za koje vrijedi

$$a_r^1 x^{2s} + b_r^1 y^{2s} + c_r^1 < 0 \quad r=1, k; s=1, q, \quad (2.2.4)$$

odnosno

$$d_{2s} = \max \frac{|a_s^2 x^{1r} + b_s^2 y^{1r} + c_s^2|}{[(a_s^2)^2 + (b_s^2)^2]^{1/2}} \quad s = (1, q) \quad (2.2.5)$$

za one r za koje vrijedi

$$a_s^2 x^{1r} + b_s^2 y^{1r} + c_s^2 < 0 \quad s=1, q; r=1, k. \quad (2.2.6)$$

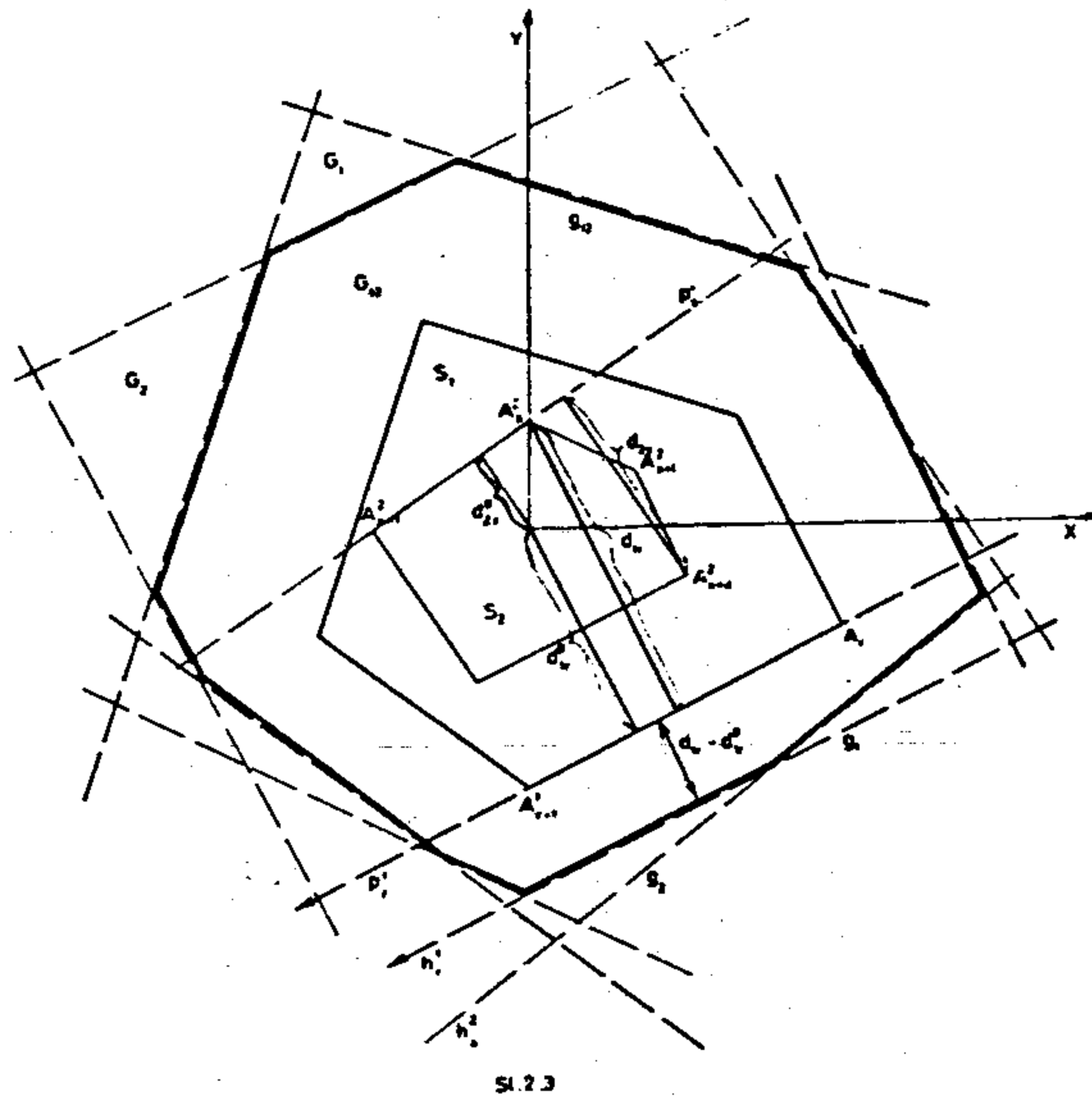
Udaljenost od p_r^1 ($r=1, k$) do $O_1(x^{01}, y^{01})$ prikazana je izrazom:

$$d_{1r}^0 = \frac{|a_r^1 x^{01} + b_r^1 y^{01} + c_r^1|}{[(a_r^1)^2 + (b_r^1)^2]^{1/2}} \quad r = 1, k \quad (2.2.7)$$

a od p_s^2 ($s=1, q$) do $O_2(x^{02}, y^{02})$:

$$d_{20}^0 = \frac{|a_s^2 x^{02} + b_s^2 y^{02} + c_s^2|}{[(a_s^2)^2 + (b_s^2)^2]^{1/2}} \quad (s = 1, q) \quad (2.2.8)$$

Paralelno stranicama p_r^1 povučemo pravce h_r^1 na udaljenosti d_{1r} od pola O_1 tako da pol bude s iste strane p_r^1 i h_r^1 . Isto tako povučemo pravce h_s^2 paralelno s p_s^2 na udaljenosti d_{2s} od pola O_2 , s tim da O_2 leži na istoj strani od p_s^2 i h_s^2 , sl.2.3.



Napomena: Pravci h_2^1 mogu se dobiti pomakom pravca p_{r2}^1 u smjeru vanjske normale, na udaljenost $d_{1r} - d_{1r}^0$, a pravci h_s^2 pomakom u smjeru unutarnje normale za $d_{2s} + d_{2s}^0$.

Neka je orijentacija h_r^1 jednaka orijentaciji p_r^1 a orijentacija h_s^2 suprotna orijentaciji p_s^2 . Znači da sve unutarnje točke $(x, y) \in S_1$ imaju pozitivan otklon od pravca h_s^2 ($s=1, q$).

Označimo sa H_r^1 poluravnine koje se sastoje od onih točaka ravnine čiji je otklon od h_r^1 negativan. Presjek tih poluravnina generira konveksnu oblast G_1 sa granicom g_1 . Isto tako imamo H_1^2 koje čine točke s negativnim otklonom od h^2 , čiji presjek generira oblast G_2 s granicom g_2 .

Postavlja se pitanje šta predstavlja oblast $G_{12} = G_1 \cap G_2$.

Teorem 2.4.

Ako su S_1 i S_2 konveksni i uzajamno orijentirani poligoni, onda granica g_{12} oblasti G_{12} predstavlja HKF dan jednačbom $Z_{12}(x, y) = 0$.

Dokaz

Iz konstrukcije je očito da se g_{12} sastoji od točaka g_1 granice G_1 i točaka g_2 granice G_2 .

Recimo da je $\{\alpha_{12}\}$ skup točaka koje zadovoljavaju $Z_{12}(x,y)=0$.

Dokažimo da je $\alpha_{12} = g_{12}$.

Izaberimo smjer \vec{p}_{12} u kom promještamo S_2 dok ne bude u kontaktu sa S_1 . Pol O_2 tada padne u točku $(x^k, y^k) \in \alpha_{12}$.

To slijedi iz definicije HKF- g_{12} .

Neka vrh $A_k^2(x^{2k}, y^{2k})$ poligona S_2 dira stranicu S_1 koja leži na pravcu p_k^1 . No S_1 i S_2 su konveksni pa kada je $O_1 = O_2$ onda je vrh $A_k^2(x^{2k}, y^{2k})$ najudaljeniji od p_k^1 , tj. za njega vrijedi

$$d_{1k} = \frac{|a^1 x^{2k} + b^1 y^{2k} + c^1|}{[(a^1)^2 + (b^1)^2]^{1/2}} = \max_s \frac{|a^1 x^{2s} + b^1 y^{2s} + c^1|}{[(a^1)^2 + (b^1)^2]^{1/2}} \quad (2.2.9)$$

za one "s" za koje vrijedi

$$a_k^1 x^{2s} + b_k^1 y^{2s} + c_k^1 < 0. \quad (2.2.10)$$

Kroz $(x^k, y^k) \in g_k$ povučemo pravac m_k^1 paralelno s p_k^1 .

Jednadžba od m_k^1 je dana

$$(a_k^1 x + b_k^1 y + c_k^1) - (a_k^1 x^k + b_k^1 y^k + c_k^1) = 0. \quad (2.2.11)$$

Kako $A_k^2(x^{2k}, y^{2k})$ leži na p_k^1 pa je s obzirom na (2.2.7), (2.2.9) i (2.2.11) udaljenost

$$\rho(p_k^1, m_k^1) = \frac{(a_k^1 x^{2k} + b_k^1 y^{2k} + c_k^1) - (a_k^1 x^k + b_k^1 y^k + c_k^1)}{[(a_k^1)^2 + (b_k^1)^2]^{1/2}} = \frac{|a_k^1 x^{2k} + b_k^1 y^{2k} + c_k^1|}{[(a_k^1)^2 + (b_k^1)^2]^{1/2}} - \frac{(a_k^1 x^k + b_k^1 y^k + c_k^1)}{[(a_k^1)^2 + (b_k^1)^2]^{1/2}} = d_{1k} - d_{1k}^0 \quad (x^k, y^k) \in g_1. \quad (2.2.12)$$

Iz konveksnosti S_1 i S_2 slijedi konveksnost G_2 .

Da je $(x^k, y^k) \in G_2$ slijedi iz konstrukcije G_2 i konveksnosti. Naime, iz pretpostavke da $(x^k, y^k) \notin G_2$ slijedilo bi da postoji pravac h_k^2 čija je udaljenost d_{2k} od pola O_1 manja nego udalje-

nost od O_1 do (x^k, y^k) što je u suprotnosti sa konstrukcijom G_2 .

Pošto je $(x_k, y_k) \in g_1$ i $(x^k, y^k) \in G_2$, slijedi $(x^k, y^k) \in g_{12}$.

Znači, ako je $(x^k, y^k) \in \alpha_{12}$, onda je

$$(x^k, y^k) \in g_{12}. \quad (2.2.13)$$

Neka je $(x^i, y^i) \in g_{12}$ i, jednostavnosti radi, neka (x^i, y^i) leži na pravcu m_t^1 (vidi 2.2.11) slijedi $(x^i, y^i) \in g_1$ i $(y^i, y^i) \in G_2$. No udaljenost između m_t^1 i p_t^1 je $d_{1t} - d_{it}^0$ i jednaka je udaljenosti točke (x^i, y^i) do p_t^1 .

Neka u slučaju kada se polovi podudaraju koordinate vrhova $A_s^2(x^{2s}, y^{2s})$ zadovoljavaju (2.2.3) i (2.2.4), tada je udaljenost od $A_s^2(x^{2s}, y^{2s})$ do pravca kroz $O_1 = O_2$, a koji je paralelan m_t^1 , dana su $d_{1t} - d_{it}^0$, slijedi A_s^2 leži na p_t^1 .

Slijedi da su S_1 i S_2 u kontaktu ako $(x^i, y^i) \in g_{12}$, znači da je točka $(x^i, y^i) \in \alpha_{12}$. U [13] se prednji teorem izriče kao lema bez dokaza.

3. Svojstva HKF konveksnih poligona

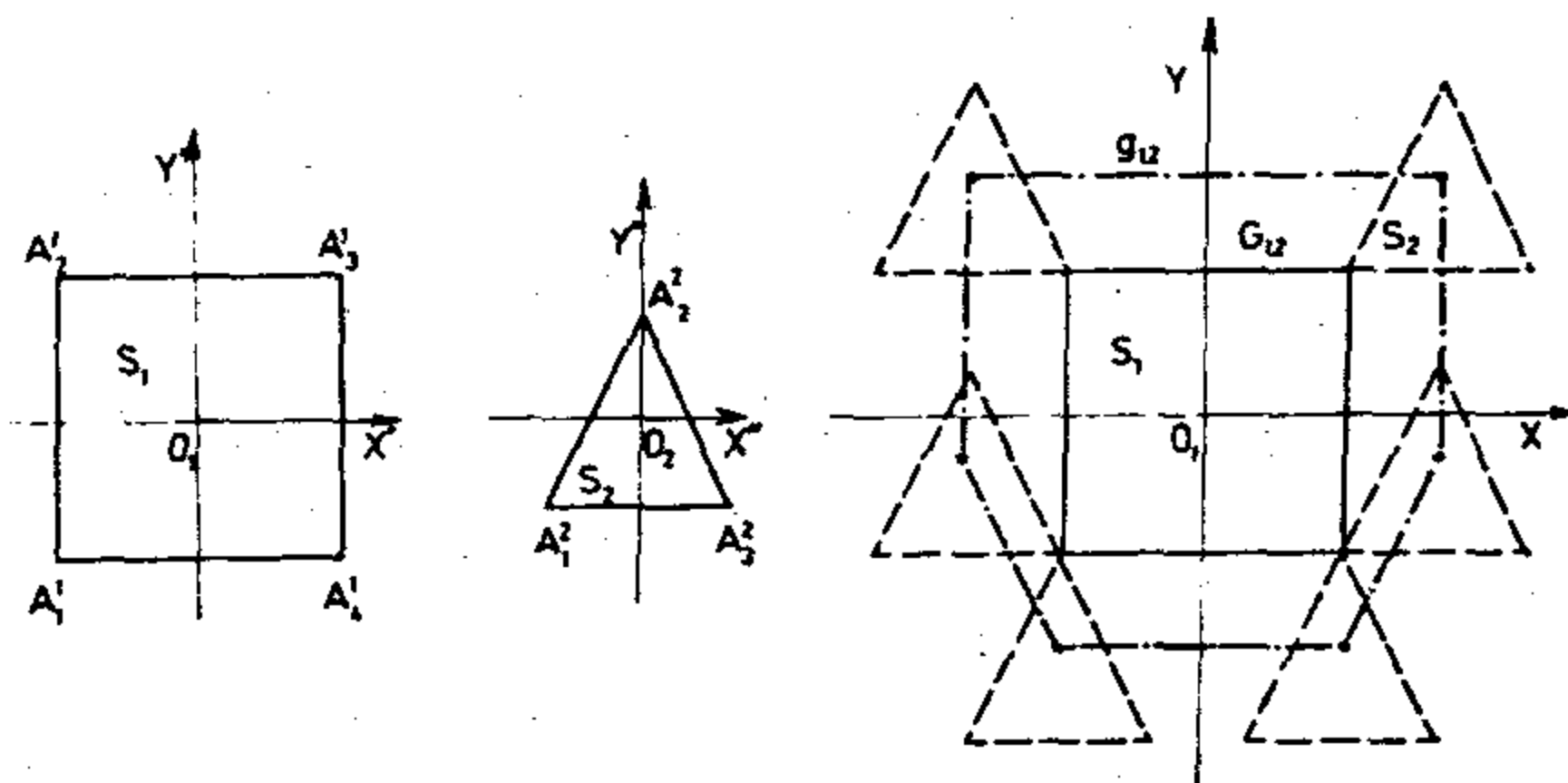
Teorem 2.5.

Ako su S_1 i S_2 konveksni, zatvoreni i medjusobno orijentirani poligoni, onda je njihov HKF $Z_{12}(x,y) = 0$ granica konveksnog poligona [13] sl.2.4.

Dokaz

Da je G_1 konveksna oblast slijedi iz njene konstrukcije. To isto vrijedi i za G_2 . No tada je i $G_{12} = G_1 \cap G_2$ takodjer konveksna, a granica od G_{12} predstavlja HKF objekata S_1 i S_2 , zadana sa $Z_{12}(x,y) = 0$.

Posljedica: HKF navedenih objekata može biti zadana bilo koordinatama vrhova bilo jednađzbama pravaca na kojima se nalaze stranice HKF.



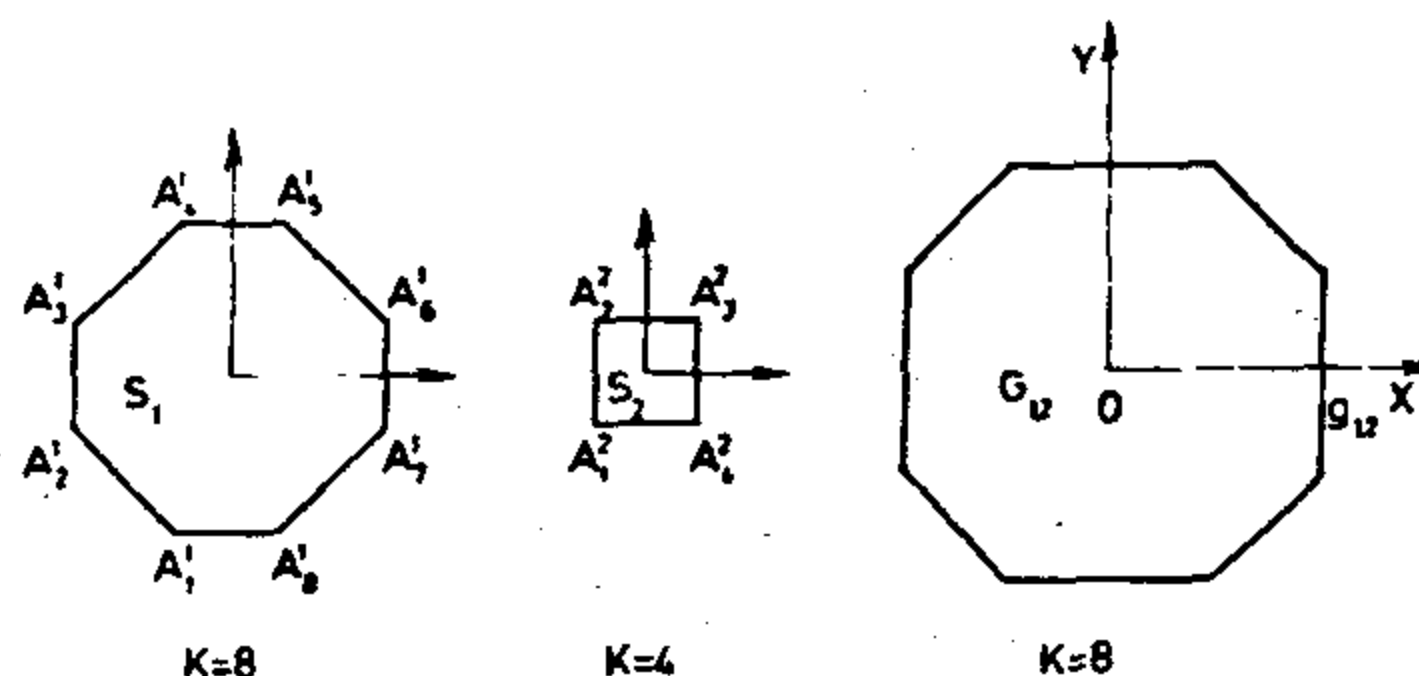
Sl.2.4.

Teorem 2.6.

Ako su S_1 i S_2 konveksni poligoni i imaju K_1 odnosno K_2 stranica onda HKF nema više od $K_1 + K_2$ stranica. [13]

Dokaz:

Granice G_1 odnosno G_2 su nastale translacijom stranica S_1 , odnosno S_2 a njih ima K_1 (K_2). Znači da $G_{12} = G_1 \cap G_2$ nema više od $K_1 + K_2$ stranica. Sl.2.5.



Sl. 2.5.

Teorem 2.7.

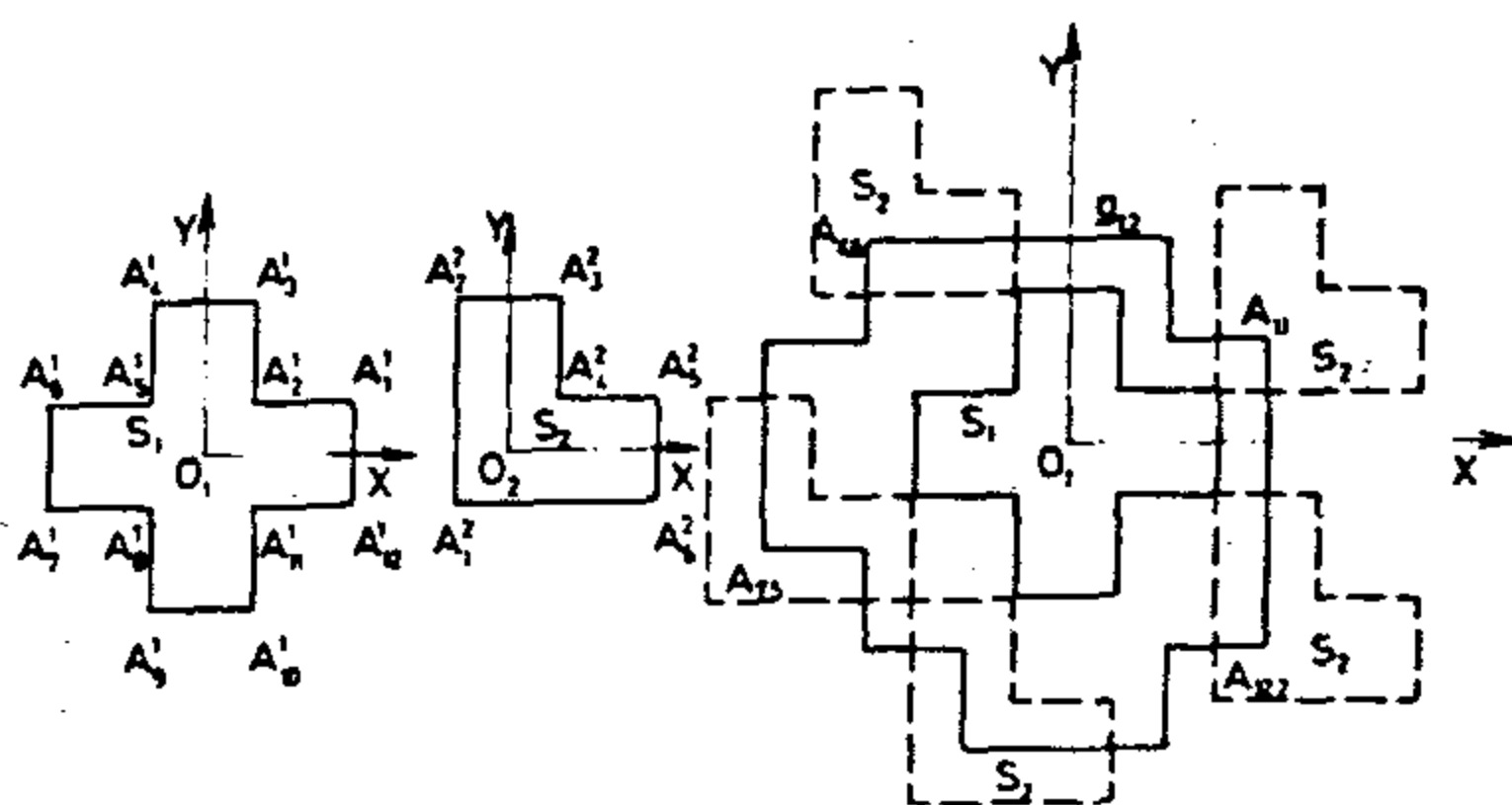
Svakom vrhu $A_{rs}^{rs}(x^{rs}, y^{rs})$ HKF - g_{12} odgovara podudaranje nekog vrha A_r^1 objekta S_1 , s nekim vrhom A_s^2 objekta S_2 , kada su objekti S_1 i S_2 u kontaktu.

To drugim riječima znači da u slučaju kad pol O_2 objekta S_2 smjestimo u neki vrh A_{rs}^{rs} HKF- g_{12} tada se neki vrh objekta S_2 podudara s nekim vrhom objekta S_1 .

Dokaz

Neka su zadani konveksni objekti S_1 i S_2 koordinatama svojih vrhova $A_r^1(A_s^2)$.

Svaki vrh $A_{rs}^{rs}(x^{rs}, y^{rs})$ HKF- g_{12} predstavlja presjek dvaju pravaca. No iz konstrukcije g_{12} slijedi da neki vrh A_r^2 objekta S_2 mora istovremeno da dira dvije stranice od S_1 , a to znači neki vrh od S_1 . Sl.2.6.



Sl. 2.6

Teorem 2.8.

Neka su koordinate vrhova $A_r(x^{1r}, y^{1r}) \in S_1$ zadane u fiksnom koordinatnom sistemu XOY, a koordinate vrhova objekta

$S_2 : A_s^2(x^{2s}, y^{2s})$ u pomičnom sustavu $\tilde{X}O_2\tilde{Y}$. Neka su S_1 i S_2 u kontaktu i neka se podudaraju vrhovi A_r^1 i A_s^2 . Koordinate odgovarajućih vrhova HKF- g_{12} u sistemu XOY tada su dane izrazom

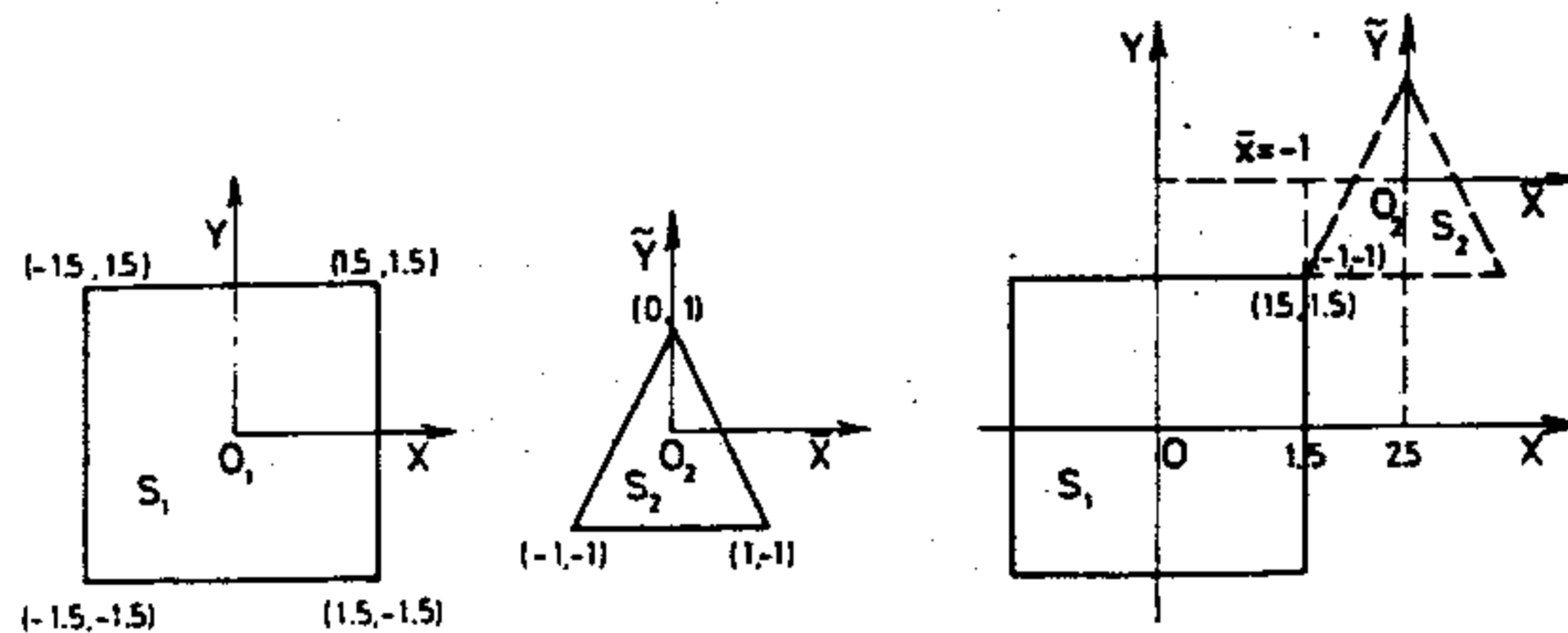
$$\begin{aligned} x^{rs} &= x^{1r} - x^{2s} \\ y^{rs} &= y^{1r} - y^{2s}. \end{aligned} \quad (2.3.1)$$

Dokaz

Pomični koordinatni sistem $\tilde{X}O_2\tilde{Y}$ u kome su zadane koordinate vrhova objekta S_2 smjestimo tako da se pol O_2 podudara sa polom O_1 (znači $x_1 = x_2; y_1 = y_2$). Zatim translaticamo S_2 tako da se podudara $A_r^1(x^{1r}, y^{1r})$ i $A_s^2(x^{2s}, y^{2s})$. Budući da su S_1 i S_2 međusobno orijentirani, koordinate pola objekta S_2 dane su izrazom

$$\begin{aligned} x_2 &= x^{1r} - x^{2s} \\ y_2 &= y^{1r} - y^{2s}, \end{aligned}$$

a odavde na osnovi teorema 2.7. slijedi tačnost formule (2.3.1) sl.2.7.



Sl.2.7

Teorem 2.9.

Ako konveksni objekti S_1 i S_2 imaju koordinatne osi simetrije, onda je i njihov HKF- g_{12} simetričan s obzirom na te osi [13]

Teorem 2.10.

Izbor mjesta polova objekata $S_1(S_2)$ ne utječe na oblik i dimenzije oblasti čija je granica HKF- g_{12} . [13]

Dokaz

Ako se ne promjeni udaljenost medju proizvoljnim vrhovima poligona, neće se promijeniti ni oblik ni dimenzije tog poligona.

$B_{pt}(x^{pt}, y^{pt})$ koji su dobiveni podudaranjem

$B_r^1(x^{lr}, y^{lr}) \in S_1$ i $B_s^2(\tilde{x}^{2s}, \tilde{y}^{2s}) \in S$ te $B_p^1(x^{lp}, y^{lp})$ sa

$B_t^2(x^{2t}, y^{2t}) \in S_2$, tada su koordinate vrhova g_{12} dane izrazom:

$$\begin{aligned} x^{rs} &= x^{lr} = \tilde{x}^{2s} & x^{pt} &= x^{lp} - \tilde{x}^{2t} \\ y^{rs} &= y^{lr} - \tilde{y}^{2s} & y^{pt} &= y^{lp} - \tilde{y}^{2t} \end{aligned} \quad (2.3.3)$$

$$\text{Udaljenost } \rho(B_{rs}, B_{pt}) = [(x^{rs} - x^{pt})^2 + (y^{rs} - y^{pt})^2]^{1/2}$$

Pomaknemo polove $O_1(O_2)$ u odnosu na $S_1(S_2)$ za vektor $\vec{d}_1(\vec{d}_2)$,

tada dobijemo nove koordinate vrhova: $B_r^1(x^{lr} - x_\alpha^1, y^{lr} - y_\alpha^1)$;
 $B_s^2(\tilde{x}^{2s} - x_\alpha^2, \tilde{y}^{2s} - y_\alpha^2)$; $B_p^1(x^{lp} - x_\alpha^1, y^{lp} - y_\alpha^1)$; $B_t^2(\tilde{x}^{2t} - x_\alpha^2, \tilde{y}^{2t} - y_\alpha^2)$.

Nove koordinatne vrhove \bar{B}_{rs} i \bar{B}_{pt} HKF nakon pomaka odrede se pomoću izraza:

$$\bar{x}^{rs} = x^{lr} - x_\alpha^1 - (\tilde{x}^{2s} - x_\alpha^2)$$

$$\bar{y}^{rs} = y^{lr} - y_\alpha^1 - (\tilde{y}^{2s} - y_\alpha^2)$$

$$\bar{x}^{pt} = x^{lp} - x_\alpha^1 - (\tilde{x}^{2t} - x_\alpha^2)$$

$$\bar{y}^{pt} = y^{lp} - y_\alpha^1 - (\tilde{y}^{2t} - y_\alpha^2)$$

$$\rho^1(B_{rs}, B_{pt}) = [(\bar{x}^{rs} - \bar{x}^{pt})^2 + (\bar{y}^{rs} - \bar{y}^{pt})^2]^{1/2} =$$

$$= [x^{lr} - \tilde{x}^{2s} - x^{lp} + \tilde{x}^{2t}]^2 + [y^{lr} - \tilde{y}^{2s} - y^{lp} + \tilde{y}^{2t}]^2]^{1/2} = \rho$$

(2.3.4)

Teorem 2.11.

Pomaknemo li pol O_1 objekta S_1 proizvoljno za vektor \vec{d}_1 a pol O_2 objekta S_2 za proizvoljni vektor \vec{d}_2 , tada se pol $O_{12}^{HKF-g_{12}}$ u odnosu na svoj pol pomakne za $\vec{d} = \vec{d}_1 - \vec{d}_2$. [13]

Teorem 2.12.

Neka su T_1 i T_2 pravokutnici opisani oko konveksnih objekata S_1 odnosno S_2 i imaju uzajamno paralelne stranice. Neka su polovi $O_1(O_2)$ izabrani u točkama presjeka dijagonala $T_1(T_2)$.

Pol O_{12} HKF- g_{12} podudara se tada s presjekom dijagonala pravokutnika T_{12} opisanog oko oblasti G_{12} a stranice su mu paralelne stranicama $T_1(T_2)$. [13]

4. Hodograf kontakt funkcije, konkavnih poligona

Pretpostavimo da je S_1 konkavan povezan poligon i da je prikazan kao unija konačnog broja konveksnih poligona, tj. $S_1 = \bigcup_{r=1}^p S_{1r}$. Neka je S_2 konveksan. Tada vrijedi slijedeća lema:

Lema 2.1.

Ako su $S_1 = \bigcup_{r=1}^p S_{1r}$ i S_2 međusobno orijentirani onda njihov HKF- g_{12} može biti predstavljen kao granica oblasti

$G_{12} = \bigcup_{r=1}^p G_{12}^r$, gdje je G_{12}^r oblast čija je granica g_{12}^r HKF konveksnog dijela S_{1r} i S_2 .

Dokaz

Pretpostavimo da svi konveksni poligoni S_{1r} imaju isti pol, tada na osnovi teorema 2.4 i teorema 2.11 možemo konstruirati sve HKF- g_{12} ($r=1, p$) konveksnih objekata S_{1r} i S_2 , tako da oni imaju isti pol. Oblasti G_{12}^r su svakako konveksne i općenito međusobno se presjecaju.

Granica oblasti $G_{12} = \bigcup_1^n G_{12}^r$ je g_{12} i ona se sastoji od slijedećih točaka:

Skupa točaka $(x, y) \in N_1$ za koje vrijedi

$$(x, y) \in g_{12}^r, (x, y) \notin G_{12}^s, \quad r, s = 1, 2, \dots, p; \quad r \neq s$$

Skupa točaka $(x, y) \in N_2$ za koje vrijedi:

$$(x, y) \in \bigcap_{k=1}^2 g_{12}^{r_k}; \quad (x, y) \notin G_{12}^s \quad r_k, s = 1, 2, \dots, p; \quad r_k \neq s$$

=====

Skupa točaka $(x, y) \in N_2$ za koje vrijedi:

$$(x, y) \in \bigcap_{k=1}^z g_{12}^{r_k}; \quad (x, y) \notin G_{12}^s \quad r_k, s = 1, 2, \dots, p \\ r_k \neq s; \quad z = 3, 4, \dots, t \leq p$$

t- je broj stranica koje imaju isti vrh.

Skup točaka N_r dobiven na prednji način predstavlja HKF objekta $S_1 = \bigcup S_{1r}$ i S_2 .

Vrijedi $\bigcap_{r=1}^z N_r = \emptyset$.

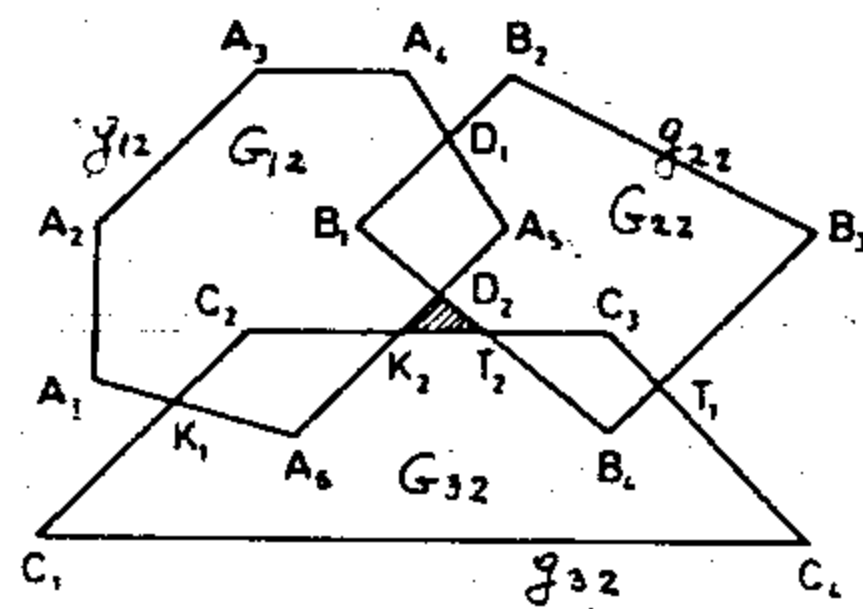
Recimo da je $(x,y) \in N_1$. To znači, kad pol O_2 smjestimo u (x,y) , da će S_2 biti u dodiru s nekim konveksnim dijelom S_{1r} objekta S_1 (to slijedi iz T 2.4). Budući da je $(x,y) \notin G_{12}^s (s=1,p,r \neq s)$, to znači da S_2 nema zajedničkih točaka ni sa jednim $S_{1r} (r=1,p; r \neq s)$, tj. S_2 dira S_1 . Dakle iz $(x,y) \in N_1$ slijedi $(x,y) \in g_{12}$.

Neka je $(x,y) \in N_2$ znači da (x,y) istovremeno pripada dvama hodografima $g_{12}^{r_1}$ i $g_{12}^{r_2}$ ali ne pripada oblastima čije su granice ostali hodografi $g_{12}^s (s=1,2,\dots,p, s \neq r_1 \text{ i } r_2)$.

Znači, ako se pol O_2 smjesti u $(x,y) \in N_2$, onda S_2 istovremeno dira S_{1r_1} i S_{1r_2} i nema zajedničkih točaka sa ostalim

$S_{1j} (j=1,p, j \neq r_1, r_2)$, tj. S_2 dira S_1 . Znači, ako je $(x,y) \in N_2$, slijedi $(x,y) \in g_{13}$. To se dokazuje za sve $(x,y) \in N_r (r=3,4,\dots,t)$.

Vidimo da su G_{12}, G_{22}, G_{32} konveksni ali je unija jednostruko suvisla. Sl.28.



Sl.28

$S_1 = S_{11} \cup S_{12} \cup S_{13}$; S_{11} i S_2 daju g_{12} ; S_{12} i S_2 daju g_{22} ; S_{13} i S_2 daju g_{32} ;

$(x,y) \in N_2$ su: D_1 i D_2 jer istovremeno pripadaju g_{12} i g_{22} a nisu u G_{32} .

T_1 i T_2 - jer istovremeno pripadaju g_{22} i g_{32} .

Granica $G = G_{12} \cup G_{22} \cup G_{32}$ je: $A_1 A_2, A_3, A_4, D_1, B_2, B_3, T_1, C_4, C_1, K_1$ (vanjska granica)

Nutarnja granica $G: T_2, K_2, D_2$

Iz $(x, y) \in g_{12}$ slijedi $(x, y) \in \bigcup_{r=1}^z N_z$.

Smjestimo pol O_2 u (x, y) , onda S_2 dira S_1 .

Pošto je $S_1 = \bigcup_{r=1}^p S_{1r}$, S_2 dira S_1 u točki koja pripada općoj stranici ili općem vrhu nekih od S_{1r_1}, S_{1r_2} slijedi $(x, y) \in N_z$.

Ako su S_1 i S_2 konkavni, povezani zatvoreni poligoni i dađu se prikazati kao unija konveksnih, tj. $S_1 = \bigcup_{r=1}^p S_{1r}$, $S_2 = \bigcup_{s=1}^p S_{2s}$,

tada vrijedi slijedeći teorem:

Teorem 2.13.

Ako su $S_1 = \bigcup_{r=1}^p S_{1r}$, $S_2 = \bigcup_{s=1}^q S_{2s}$ međusobno orijentirani,

onda njihov hodograf g_{12} predstavlja granicu oblasti

$$G_{12} = \bigcup_{r=1}^p \bigcup_{s=1}^q G_{12}^{rs} \quad [13]$$

Ovdje je G_{12}^{rs} oblast čija je granica HKF - g_{12}^{rs} konveksnih djelova S_{1r} i S_{2s} .

Dokaz slijedi neposredno iz leme 2.1.

Navedimo neka svojstva HKF konkavnih, povezanih, ograničenih zatvorenih i međusobno orijentiranih poligona.

Teorem 2.14.

Hodograf g_{12} objekata $S_1 = \bigcup_{r=1}^p S_{1r}$ i $S_2 = \bigcup_{s=1}^q S_{2s}$ je poligon [13]

Teorem 2.15.

Svakom vrhu A_{rs} HKF- g_{12} objekata S_1 i S_2 odgovara podudaranje vrhova A_r^1 i A_s^2 objekata S_1 i S_2 .

Teorem 2.16.

Koordinate vrhova HKF- g_{12} $A_{rs}(x^{rs}, y^{rs})$ dobiju se kao razlike koordinata podudarnih vrhova $A_r^1(x^r, y^r)$ i $A_s^2(x^s, y^s)$ s tim da su $A_r^1(x^r, y^r)$ zadane u fiksnom, a $A_s^2(x^{1s}, y^{2s})$ u pomičnom koordinatnom sistemu $\tilde{X}\tilde{O}\tilde{Y}$, tj.

$$\begin{aligned}x^{rs} &= x^{1r} - \tilde{x}^{2s} \\y^{rs} &= y^{1r} - \tilde{y}^{2s}\end{aligned}\tag{2.4.1}$$

Dokaz

Saglasno teoremu 2.15, kad pol O_2 smjestimo u vrh $A_{rs}(x^{rs}, y^{rs})$, onda to ogovara podudaranju vrha $A_r^1(x^{1r}, y^{1r})$ i $A_s^2(x^{2s}, y^{2s})$.

Ako su $A_r^1(x^{1r}, y^{1r})$ uzeti s obzirom na fiksni koordinatni sistem XOY , a koordinate vrha $A_s^2(x^{2s}, y^{2s})$ u odnosu na pomični $\tilde{X}\tilde{O}\tilde{Y}$, onda vrijedi

$$x^{1r} = x^{rs} - \tilde{x}^{2s}; \quad y^{1r} = y^{rs} - \tilde{y}^{2s},$$

a odavde slijedi tvrdnja u teoremu.

Tvrdnja 2.3.

Ako konkavni poligoni S_1 i S_2 imaju osi simetrije koje su paralelne koordinatama fiksnog sistema XOY , onda je HKF- g_{12} simetričan s obzirom na te iste osi.

Iz toga slijedi da je HKF centralno simetričnih poligona centralno simetričan.

Tvrdnja 2.4.

Izbor položaja polova O_1 i O_2 proizvoljnih poligona ne utječe na oblik i dimenzije pripadnog HKF.

Tvrdnja 2.5.

Ako pol O_1 objekta $S_1 = \bigcup_{r=1}^p S_{1r}$ pomaknemo za vektor \vec{d}_1 , a pol O_2 objekta $S_2 = \bigcup_{s=1}^q S_{2s}$ za vektor \vec{d}_2 , onda se HKF pomakne u odnosu na svoj pol za $\vec{d} = \vec{d}_2 - \vec{d}_1$ (\vec{d}_1, \vec{d}_2 - proizvoljni).

Tvrđnja 2.6.

Ako su $P_1(P_2)$ pravokutnici opisani oko konkavnih objekata $S_1(S_2)$ a odgovarajuće stranice od $P_1(P_2)$ su međusobno paralelne, pa ako polove $O_1(O_2)$ izaberemo u presjeku dijagonala pravokutnika, tada pol HKF padne u točku presjeka dijagonala pravokutnika P opisanog oko HKF, a stranice su mu paralelne odgovarajućim stranicama $P_1(P_2)$. Znamo da razbijanje konkavnog poligona na konveksne, čija je unija jednaka njemu samom, nije jednoznačno određena, tj. može se sprovesti na beskonačno načina. Zato se prirodno postavlja pitanje, da li postupak razbijanja utječe na oblik, dimenziju i položaj oblasti G_{12} čija granica predstavlja HKF. Odgovor daje slijedeći teorem.

Teorem 2.17.

Način razbijanja S_1 na konveksne dijelove S_{1r} ($r=1,p$) i S_2 na konveksne dijelove S_{2s} ($s=1,q$) ne utječe na položaj, oblik i dimenziju HKF- g_{12} . Također ne zavisi ni od broja p odnosno q .

[13]

Dokaz

Konveksni dijelovi S_{1r} ($r=1,p$) i S_{2s} ($s=1,q$) potpuno iscrpljuju poligone S_1 i S_2 pa karakter razbijanja kao i $p(q)$ nemaju nikakvog utjecaja na vrhove $S_1(S_2)$, a vrhovi g_{12} su određeni vrhovima S_1 i S_2 . To znači da je g_{12} jednoznačno određen vrhovima S_1 i S_2 . Slijedi da način razbijanja i broj elemenata ne utječe na položaj, oblik i dimenziju g_{12} .

Napomena:

Iz prednjeg se ne smije izvući zaključak da za praksu nije važno kako se vrši razbijanje i na koliko i kakvih dijelova. Naime, razbijanje treba vršiti na najmanji mogući broj konveksnih elemenata s tim da elementi budu što jednostavniji (trokut je jednostavniji od četverokuta i općenito poligon T_{n-1} je jednostavniji od T_n).

Tvrđnja 2.7.

Ako se iz S_1 homotetijom dobije objekt $S_1^{k_1}$ a iz S_2 objekt $S_2^{k_2}$, onda se HKF- g_{12}^k dobije iz g_{12} homotetijom za $K = K_1 + K_2$.

Tvrdnja 2.8.

Dva zatvorena i ograničena objekta uvijek imaju HKF- g_{12} i on je jedinstven.

Tvrdnja 2.9.

Ako je među objektima zadana minimalna udaljenost l_{12} koja mora biti poštovana, onda se HKF za takav slučaj dobije iz HKF- g_{12} , gdje je $l_{12} = 0$, tako da se oko g_{12} konstruira ekvidistantna udaljenost l_{12} . Navedeni teoremi i tvrdnje upućuju na način kako da konstruiramo HKF proizvoljnih objekata.

5. Hodograf kontakt funkcije objekata ograničenih po dijelovima glatkim krivuljama

Neka je zadan konveksni objekt $S \subset \mathbb{R}^2$ čija je granica "s" po dijelovima tlatka i dana sa

$$Z(x,y) = 0. \quad (2.5.1)$$

Budući da je granica po dijelovima glatka, slijedi da su dijelovi glatki. Granicu "s" možemo aproksimirati poligonom P_n . Uzmemo li da je $\lim_{\epsilon \rightarrow 0} n_\epsilon \rightarrow \infty$, $\lim_{\epsilon \rightarrow 0} d_\epsilon = 0$ (2.5.2)

gdje je n - broj stranica poligona, d_ϵ - duljina najdulje stranice poligona.

Kad $\epsilon \rightarrow 0$, onda stranice objekta S , koje su uz vrh A_m poligona P_n kad predjemo na limes, leže na tangenti na "s" u točki A_m . Znači da svaku točku $(x,y) \in s$ možemo predstaviti kao vrh poligona, kad predjemo na limes, a tangentu kao stranicu poligona. Sada se i na ovaj slučaj mogu primjeniti razmatranja provedena za poligone.

Neka su S_1 i S_2 konveksni i međusobno orijentirani objekti a granica im je zadana

$$Z_1(x,y) = 0; \quad Z_2(x,y) = 0, \quad (2.5.3)$$

gdje $Z_1, Z_2 \in C^{(1)}$.

Izaberimo polove da bude $O_1(x_{01}, y_{01}) \in S_1$ i $O_2(x_{02}, y_{02}) \in S_2$ i da

bude $Z_1(x_{01}, y_{01}) < 0; \quad Z_2(x_{02}, y_{02}) < 0$.

Smjestimo objekte tako da se polovi podudaraju, tj. $O_1 = O_2$ ($x_{01} = x_{02}; \quad y_{01} = y_{02}$).

Neka je p_{1r} pravac koji dira S_1 u točki (x_{1r}, y_{1r}) a p_{2s} , pravac koji dira S_2 u točki (x_{2s}, y_{2s}) . Sada je udaljenost d_{1r} od pravca p_{1r} do najudaljenije točke $(x_{2s}^t, y_{2s}^t) \in s_2$ dana izrazom

$$d_{1r} = \max_{(x_{2s}, y_{2s}) \in s_2} \frac{|d_{1r}(x_{2s} + x_2) + b_{1r}(y_{2s} + y_2) + c_{1r}|}{[(a_{1r})^2 + (b_{1r})^2]^{1/2}} =$$

$$= \frac{|a_{1r}(x_{2s} + x_2) + b_{1r}y_{2s} + y_2 + c_{1r}|}{[(a_{1r})^2 + (b_{1r})^2]^{1/2}}$$

Promatraju se one točke koje su na istoj strani od p_{1r} na kojoj je i pol $O_1 = O_2$.

Za pravce p_{2s} i najudaljeniju točku $(x_{1r}^*, y_{1r}^*) \in s_1$ imamo

$$d_{2s} = \max_{(x_{1r}, y_{1r}) \in s_1} \frac{|a_{2s}x_{1r} + b_{2s}y_{1r} + c_{1r}|}{[(a_{2s})^2 + (b_{2s})^2]^{1/2}} = \frac{|a_{2s}x_{1r}^* + b_{2s}y_{1r}^* + c_{1r}|}{[(a_{2s})^2 + (b_{2s})^2]^{1/2}}$$

Neka su \vec{n}_{1r} i \vec{n}_{2s} vektori normala u točkama diranja pravaca $p_{1r}(p_{2s})$ s objektima $S_1(S_2)$, tada kad vrijedi

$$-\vec{n}_{1r} = \alpha \vec{n}_{2s} \quad (2.5.4)$$

točka (x_{2s}^*, y_{2s}^*) će biti najudaljenija od p_{1r} , odnosno (x_{1r}^*, y_{1r}^*) najudaljenija od p_{21} . No zbog (2.5.4) u tom slučaju je p_{1r} paralelan p_{2s} , a budući da su objekti konveksni, to je $d_{1r} = d_{2s}$.

Prema tome

$$G_{12} = \lim_{\epsilon \rightarrow 0} G_{12}^\epsilon = \lim_{\epsilon \rightarrow 0} (G_1^\epsilon \cap G_2^\epsilon) = G_1 = G_2.$$

Tvrđnja 2.10.

Svakoј točki HKF- g_{12} objekata S_1 i S_2 koji su ograničeni po dijelovima glatkim linijama odgovara dodir bar po jedne točke granice objekta S_1 i S_2 .

Napomena: Odgovarajući teoremi i tvrdnje izrečeni za konveksne i konkavne objekte prenose se i na prednje objekte.

Tvrđnja 2.11.

- Ako su objekti S_1 i S_2 razdijeljivi, onda njima pripadna oblast G_{12} ima svojstva da je suvisla i zvjezdasta.
- Ako su objekti nerazdijeljivi, onda je oblast G_{12} ili višestruko suvisla ili suvisla bez zvjezdaste oblasti.

Tvrđnja 2.12.

Neka su zadana dva objekta: S_1 nepovezan tj. $S_1 = S_{11} \cup S_{12}$ i S_2 čiji je dijametar jednak d . Tada se, za slučaj ako je $\rho(S_{11}, S_{12}) > d$, g_{12} raspada ^{OBAVEŠKO} na dva nepovezana dijela.

6. Analitičko nalaženje HKF objekata

Neka su zadani objekti S_1 i S_2 s regularnim granicama s_1, s_2 zadanim jednažbama

$$F_1(x,y) = 0 \quad F_2(x,y) = 0 \quad (2.6.1)$$

Neka su parametarske jednažbe oblika

$$\begin{aligned} x &= \phi_1(t_1) & x &= \psi_1(t_2) \\ y &= \phi_2(t_1) & y &= \psi_2(t_2). \end{aligned} \quad (2.6.2)$$

Budući da su $s_1(s_2)$ regularne, znači da u svakoj točki (x,y) možemo povući jedinstvenu tangentu. Znamo da svakoj točki $(x^*,y^*) \in s_1$ pripada točka $(x^{**},y^{**}) \in s_2$. Kada se te točke podudaraju a vrijedi $S_1 \setminus s_1 \cap S_2 \setminus s_2$, tada postoji točka $(x,y) \in g_1$ koja se određuje pomoću izraza:

$$\begin{aligned} x &= x^* - x^{**} \\ y &= y^* - y^{**}, \end{aligned}$$

odnosno parametarski

$$\begin{aligned} x &= \phi_1(t_1) - \psi_1(t_2) \\ y &= \phi_2(t_1) - \psi_2(t_2). \end{aligned} \quad (2.6.3)$$

Kada pol O_2 smjestimo u točku (x,y) , onda kroz $(x^*,y^*) (x^{**},y^{**})$ prolazi jedinstven pravac $p_1 \equiv p_2$ koji istovremeno tangira s_1 i s_2 . Kako je uvjet paralelnosti tangenta p_1 i p_2 izražen preko uvjeta da je vektorski produkt $\vec{A}_1 \times \vec{A}_2 = 0$ $[(A_1, A_2) \perp \text{okomit na } p_1(p_2)]$, to uvjet paralelnosti pravac p_1 i p_2 koji su tangente na $s_1(s_2)$ možemo zapisati

$$F'_{1x}F'_{2y} - F'_{1y}F'_{2x} = 0, \quad (2.6.4)$$

odnosno koristeći parametarski oblik

$$\phi'_{1x}(t_1)\psi'_{2y}(t_2) - \phi'_{2y}(t_1)\psi'_{2x}(t_2) = 0 \quad (2.6.5)$$

Slijedi da parametar t_2 (odnosno t_1) možemo predstaviti kao funkciju od t_1 , tj. $t_2 = v_1(t_1)$ ($t_1 = v_2(t_2)$).

Stavimo li mjesto t_2 izraz $v_1(t_1)$ (2.6.3), dobijemo analitički

izraz za HKF- g_{12} objekata S_1 i S_2 u parametarskom obliku

$$\begin{aligned} x &= \phi_1(t_1) - \psi_1[v_1(t_1)] \\ y &= \phi_2(t_1) - \psi_2[v_1(t_1)]. \end{aligned} \quad (2.6.6)$$

Odgovor na pitanje kada je prednje moguće, tj. koja je to klasa krivulja koja to dozvoljava daje poznati slijedeći teorem [13].

Teorem 2.18.

Ako su granice $s_1(s_2)$ objekata $S_1(S_2)$ regularne i zadane parametarskim jednažbama (2.6.2) gdje je

$$\phi_1(t_1) = \sum_{j=1}^{n_i} A_{ij} t_1^j; \quad \psi_i(t_2) = \sum_{t=1}^{m_i} B_{it} t_2^t \quad i = 1, 2,$$

tada HKF- g_{12} objekata S_1 i S_2 može biti dan parametarski pomoću (2.6.6) ako vrijedi [13]

$$\max_{1 < i < 2} \{n_i, m_i\} \leq 5$$

Analitički izrazi za HKF konveksnih poligona

Neka su zadani konveksni međusobno orijentirani objekti S_1 i S_2 . Neka su $O_1 \in S_1$ i $O_2 \in S_2$ i neka se oni podudaraju $O_1 = O_2$. Neka su objekti definirani sistemom nejednažbi

$$\begin{aligned} F_r(x, y) &= a_{1r}x + b_{1r}y + c_1 \leq 0 & r &= 1, p \\ Z_s(x, y) &= a_{2s}x + b_{2s}y + c_2 \leq 0 & s &= 1, q. \end{aligned} \quad (2.6.8)$$

Neka su zadane koordinate vrhova $A_r^1(x^{1r}, y^{1r})$ i $A_s^2(x^{2s}, y^{2s})$. Znamo da se koeficijenti u (2.6.8) izražavaju pomoću koordinata vrhova kroz koje prolaze pripadni pravci. Koristeći (2.6.8) i način formiranja HKF, jednažbe pravaca koji prolaze stranicama G_1 odnosno G_2 glase

$$V_r(x, y) = a_{1r}x + b_{1r}y + \min_s (a_{1r}x^{2s} + b_{1r}y^{2s} + c_{1r}) \quad r=1, p \quad (2.6.9)$$

za one s za koje vrijedi

$$F_r(x^{2s}, y^{2s}) < 0 \quad s = (1, t_1),$$

odnosno

$$n_s(x,y) = -(a_{2s}x + b_{2s}y) + \min_s(a_{2s}x^{1r} + b_{2s}y^{1r} + c_{2s}) \quad s=1,q \quad (2.6.1)$$

za one r za koje vrijedi $z_s(x^{1r}, y^{1r}) < 0$ ($r=1, t_2$).

znači da je G_1 definirana sa $V_r(x,y) \leq 0$ a G_2 sa $n_s(x,y) \leq 0$.

No kako je $G_{12} = G_1 \cap G_2$, koristeći aparat R_2 - funkcija dobijemo da je HKF- g_{12} , kao granica G_{12} , dan izrazom

$$\left[\bigwedge_{r=1}^p V_r(x,y) \right] \bigwedge_1 \left[\bigwedge_{s=1}^q n_s(x,y) \right] = 0 \quad (2.6.11)$$

Napomena: Kada se radi o poligonima s većim brojem stranica izraz (2.6.11) u razvijenom obliku je vrlo glomazan.

Analitički izrazi za HKF objekata ograničenih krivuljama II reda i dijelovima pravaca

Pored primjera gdje su granice objekata izvorno krivulje II reda ili dijelovi pravaca, imamo i slučajeve da nakon aproksimacije stvarnih granica s ϵ -točnošću dobivamo da su granice krivulje II reda ili dijelovi pravaca.

- I Hodograf kontakt funkcija za: a) elipsa-elipsa
 b) elipsa-krug
 c) krug-krug

a) Neka su parametarski zadane dvije elipse

$$\begin{aligned} x &= a \cos t_1 & 0 < t_1 < 2\pi \\ y &= b \sin t_1 \end{aligned} \quad (2.6.12)$$

$$\begin{aligned} x^1 &= A \cos t_2 & 0 < t_2 < 2\pi \\ y^1 &= B \sin t_2 \end{aligned} \quad (2.6.13)$$

Na osnovi formula transformacije

$$\begin{pmatrix} X \\ Y \end{pmatrix} = (x', y') = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \quad (2.6.14)$$

jednadžba elipse S_2 u XOY je dana sa

$$\begin{aligned}x &= A \cos t_2 \cos \theta - B \sin t_2 \sin \theta \\y &= A \cos t_2 \sin \theta + B \sin t_2 \cos \theta\end{aligned}\quad (2.6.15)$$

Na osnovu (2.6.3), uzevši (2.6.12) i (2.6.15) imamo da je jednadžba HKF dviju elipsa u prameterskom obliku dana izrazom

$$\begin{aligned}x &= a \cos t_1 - A \cos t_2 \cos \theta + B \sin t_2 \sin \theta \\y &= b \sin t_1 - A \cos t_2 \sin \theta - B \sin t_2 \cos \theta \quad 0 \leq \theta \leq 2\pi\end{aligned}\quad (2.6.16)$$

Na osnovi (2.6.4) odnosno (2.6.5) nadjemo $\operatorname{tg} t_2$, odnosno $\cos t_2$ i $\sin t_2$ koje kada uvrstimo u (2.6.16), dobijemo analitičke parametarske jednadžbe za HKF dviju elipsa koje nisu medjusobno orijentirane.

$$\begin{aligned}x &= a \cos t_1 + \frac{A^2 \cos \theta (b \cos \theta \cos t_2 + a \sin \theta \sin t_2) - B^2 \sin \theta (a \cos \theta \sin t_2 - b \sin \theta \cos t_2)}{[A^2 (b \cos \theta \cos t_2 + a \sin \theta \sin t_2)^2 + B^2 (a \cos \theta \sin t_2 - b \sin \theta \cos t_2)^2]^{1/2}} \\y &= b \sin t_1 + \frac{A^2 \sin \theta (b \cos \theta \cos t_2 + a \sin \theta \sin t_2) + B^2 \cos \theta (a \cos \theta \sin t_2 - b \sin \theta \cos t_2)}{[A^2 (b \cos \theta \cos t_2 + a \sin \theta \sin t_2)^2 + B^2 (a \cos \theta \sin t_2 - b \sin \theta \cos t_2)^2]^{1/2}}\end{aligned}\quad (2.6.17)$$

Posebni slučajevi:

a-1 Za medjusobno orijentirane elipse, tj. za $\theta = 0$ imamo HKF

$$\begin{aligned}x &= a \cos t_1 + \frac{A^2 b \cos t_2}{[A^2 b^2 \cos^2 t_2 + B^2 a^2 \sin^2 t_2]^{1/2}} \\y &= b \sin t_1 + \frac{B^2 a \sin t_2}{[A^2 b^2 \cos^2 t_2 + B^2 a^2 \sin^2 t_2]^{1/2}}\end{aligned}$$

a-2 Elipse su kongruentne ($A=a$, $B=b$) i medjusobno orijentirane

$$\begin{aligned}x &= 2a \cos t \\y &= 2b \sin t.\end{aligned}\quad (2.6.18)$$

b) Ako je $a=b=R$ dobijemo HKF kruga i elipse

$$\begin{aligned}x &= R \cos t + \frac{A^2 \cos t}{[A^2 \cos^2 t + B^2 \sin^2 t]^{1/2}} \\y &= R \sin t + \frac{B^2 \sin t}{[A^2 \cos^2 t + B^2 \sin^2 t]^{1/2}}\end{aligned}\quad (2.6.19)$$

c) Za $a=b=R$, $A=B=R$ dobijemo HKF za dva kruga radijusa R_1 i R_2

$$\begin{aligned}x &= (R_2 + R_1) \cos t \\y &= (R_2 + R_1) \sin t.\end{aligned}\quad (2.6.20)$$

Hodograf konktakt funkcije parabole i elipse

Neka granica s_1 objekta S_1 predstavlja parabolu čija je parametarska jednačba oblika:

$$\begin{aligned}x &= t_1 \\y &= at_1^2 + bt_1 + c,\end{aligned}$$

a granica s_2 elipse S_2 je dana jednačbom (2.6.15). Na osnovi (2.6.3) imamo parametarsku jednačbu HKF objekata S_1 i S_2 .

$$\begin{aligned}x &= t_1 - A \cos t_2 \cos \theta + B \sin t_2 \sin \theta \\y &= at_1^2 + bt_1 + c - A \cos t_2 \sin \theta - B \sin t_2 \cos \theta\end{aligned}\quad (2.6.21)$$

Na osnovi (2.6.4), odnosno (2.6.5) dobijemo HKF za S_1 i S_2 kada su međusobno neorijentirani

$$\begin{aligned}x &= t + \frac{A^2 \cos \theta (2at \cos \theta + b \cos \theta + \sin \theta) - B^2 \sin \theta (\cos \theta - 2at \sin \theta - b \sin \theta)}{[A^2 (2at \cos \theta + b \cos \theta + \sin \theta)^2 + B^2 (\cos \theta - 2at \sin \theta - b \sin \theta)^2]^{1/2}} \\y &= at + bt + c + \frac{A^2 \sin \theta (2at \cos \theta + b \cos \theta + \sin \theta) + B^2 \cos \theta (\cos \theta - 2at \sin \theta - b \sin \theta)}{[A^2 (2at \cos \theta + b \cos \theta + \sin \theta)^2 + B^2 (\cos \theta - 2at \sin \theta - b \sin \theta)^2]^{1/2}}\end{aligned}$$

Posebni slučajevi

a) Za $\theta = 0$ dobijemo iz (2.6.22) HKF- g_{12} parabole i elipse čiji su polusi paralelne osima koordinata XOY

$$x = t \mp \frac{A^2(2at+b)}{[A^2(2at+b)^2+B^2]^{1/2}}$$

$$y = at^2+bt+c \pm \frac{B^2}{[A^2(2at+b)^2+B^2]^{1/2}} \quad (2.6.23)$$

b) Za $A=B=R$ iz (6.22) dobijemo HKF kruga radiusa R i parabole

$$x = t \mp \frac{(2at+b)}{[1+(2at+b)^2]^{1/2}}$$

$$y = at^2+bt+c \pm \frac{R}{[1+(2at+b)^2]^{1/2}}$$

Hodograf kontakt funkcije za slučaj s_1 - hiperbola, s_2 - elipsa

Parametarska jednadžba hiperbole je oblika

$$x = a \frac{1}{\cos t_1} \quad (2.6.24)$$

$$y = b \operatorname{tg} t_1.$$

Na osnovi (2.6.3) i (2.6.15) HKF S_1 i S_2 ima oblik

$$x = a \sin t_1 - A \cos t_2 \cos \theta + B \sin t_2 \sin \theta$$

$$y = b \operatorname{tg} t_1 - A \cos t_2 \sin \theta - B \sin t_2 \cos \theta \quad (2.6.25)$$

Koristeći (2.6.4), odnosno (2.6.5) nadjemo $\sin t$ i $\cos t$ koje uvrstimo u (2.6.25) i dobijemo HKF za hiperbolu i elipsu kada su međusobno neorijentirane

$$x = a \sin t \pm \frac{A^2 \cos \theta (b \cos \theta - a \sin \theta \sin t) - B^2 \sin \theta (b \sin \theta + a \cos \theta \sin t)}{[A^2 (b \cos \theta - a \sin \theta \sin t)^2 + B^2 (b \sin \theta + a \cos \theta \sin t)^2]^{1/2}}$$

$$y = b \operatorname{tg} t \pm \frac{A^2 \sin \theta (b \cos \theta - a \sin \theta \sin t) - B^2 \cos \theta (b \sin \theta + a \cos \theta \sin t)}{[A^2 (b \cos \theta - a \sin \theta \sin t)^2 + B^2 (b \sin \theta + a \cos \theta \sin t)^2]^{1/2}}$$

Posebni slučajevi

a) Ako je $A=B=R$ i $\theta = 0$, imamo kanonsku jednadžbu HKF kruga radiusa R i hiperbole

$$x = a \sin t + \frac{Rb}{[b^2 + a^2 \sin^2 t]^{1/2}}$$

$$y = b \tan t + \frac{Ra}{[b^2 + a^2 \sin^2 t]^{1/2}}$$

7. Problem jednoznačnosti kontakt funkcije

Vidjeli smo da između HKF i KF postoji tijesna veza. Izveli smo teorem iz koga slijedi da forma HKF ne zavisi od izbora polova. Međutim oblik KF bitno zavisi od izbora polova. Već kod najjednostavnijih objekata kao što su krugovi, ako su polovi izabrani van centara krugova, dobijemo jasno izražene minimume i maksimume za $\theta \in (0, 2\pi)$.

Može se dogoditi da KF postane čak i višeznačna, pa i kontinualno višeznačna, ako se nezgodno izabere pol, mada po svojoj prirodi nije takva. Uvijek treba polove smjestiti tako da KF bude jednoznačna, kada je to moguće, te da amplituda vrijednosti KF bude minimalna. U slučaju kada oblast G_{12} čija granica predstavlja HKF ima u sebi zvijezdastu oblast (zvijezdasta oblast je uvijek konveksan skup) izbor pola HKF O_{12} iz zvijezdaste oblasti ima za posljedicu da je KF jednoznačna.

Zato orijentirajmo stranice HKF tako da G_{12} uvijek ostaje desno kada se krećemo u smjeru kazaljke na satu, tj. da za jednadžbe pravaca na kojima leže pripadne stranice hodografa vrijedi

$$f_r(x, y) = a_r x + b_r y + c_r < 0 \quad (r=1, k) \quad (2.7.1)$$

gdje je k - broj stranica HKF.

Znači da treba naći, ako postoji, točku $O_{12}(x_{12}, y_{12})$ oblasti G_{12} čija je granica g_{12} u kojoj vrijedi (2.7.1) za sve k .

Radi toga uvodimo pomoćnu varijablu Z i zahtijevamo da vrijedi

$$f_r(x, y) = a_r x + b_r y + c_r < Z \quad (r=1, k). \quad (2.7.2)$$

Tako se problem nalaženja O_{12} sveo na problem linearnog programiranja. Naći minimum linearne forme

$$Z^* = \min Z$$

uz ograničenja

$$f_r(x, y) = a_r x + b_r y + c_r < Z \quad r = 1, k$$

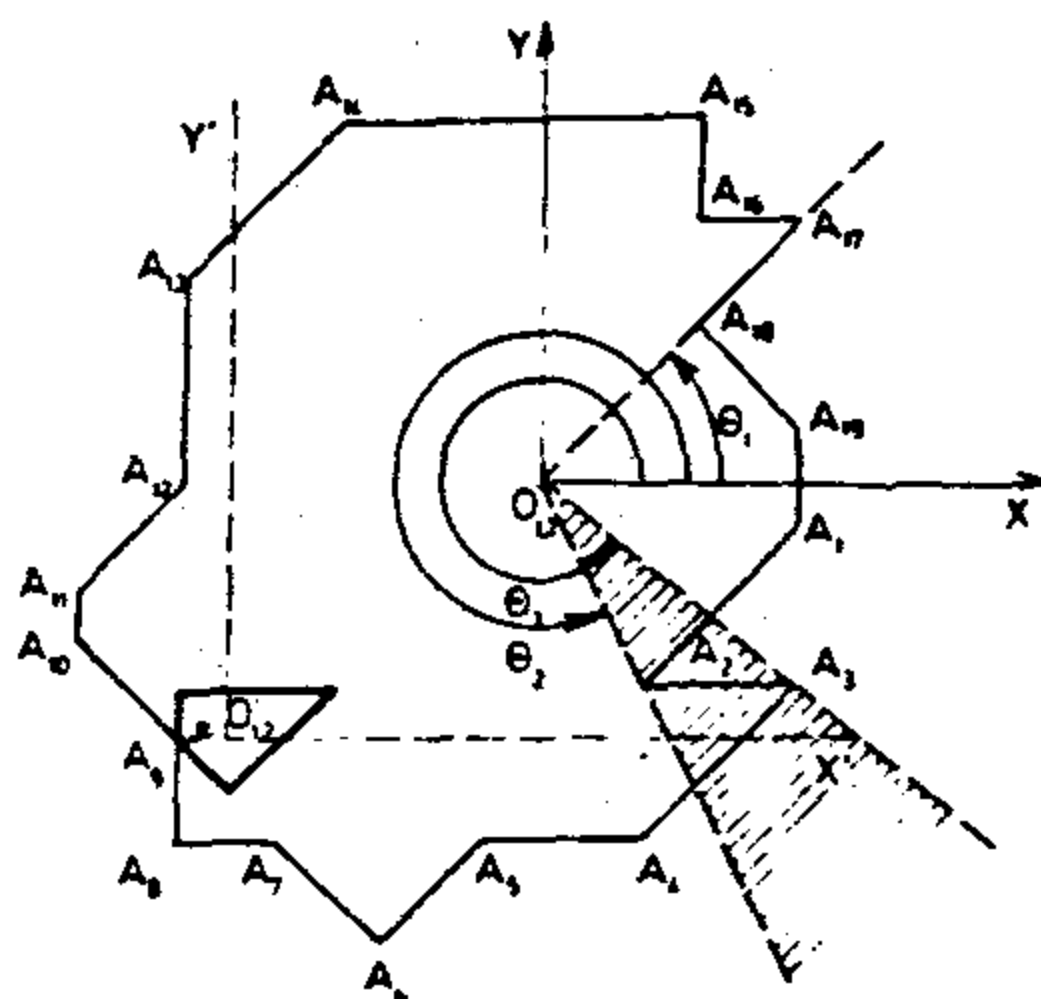
Pokazuje se da imamo slijedeće mogućnosti

- a) $Z^* < 0$ - znači da O_{12} leži u zvijezdastoj oblasti što povlači da je KF jednoznačna

- b) $Z^* = 0$ - znači da su objekti nerazdjeljivi a nadjena točka (x^*, y^*) obavezno se nalazi u jednom od vrhova hodografa.
- c) $Z^* > 0$ - znači da je KF višeznačna bez obzira koju točku izaberemo za pol, a objekti kojima taj HKF pripada međusobno su nerazdjeljivi.

Na slici 2.9 vidimo hodograf kome za $\theta = 45^\circ$ pripada KF koja je kontinualno višeznačna, a za $297 < \theta < 321$ kontakt funkcija je troznačna. Međutim izaberemo li pol O_{12} hodografa u točki $(-3.0; -2.5)$ koja se nalazi u zvijezdastoj oblasti Ω , KF je jednoznačna.

Na slici 2.26. vidimo pripadne grafove kontakt funkcija za navedena dva izbora pola.



Sl.2.9.

Ako su S_1 i S_2 krugovi radijusa R_1, R_2 i ako se polovi izaberu u centrima, onda je KF dana izrazom

$$K(\theta) = R_1 + R_2 = \text{const.}$$

Ako su S_1 i S_2 dvije kongruentne elipse, onda je KF dana sa

$$K(\theta) = 2[a^2 \cos^2 \theta + b^2 \sin^2 \theta]^{1/2}.$$

Ako su S_1 i S_2 poligoni, onda njima pripadni HKF predstavlja također poligon, a KF se računa pomoću izraza

$$K(\theta) = \frac{d_r}{|\cos(\theta_r - \theta)|} \quad \alpha_r < \theta < \alpha_{r+1}; \quad r = 1, k$$

gdje je d_r - udaljenost od pola do r -te stranice HKF; k - broj vrhova, $\alpha_r = \arctg \frac{y_r}{x_r}$; $\alpha_{r+1} = \arctg \frac{y_{r+1}}{x_{r+1}}$

Pomoću koordinata vrhova HKF $(x_r, y_r, x_{r+1}, y_{r+1})$ formula za računanje kontakt funkcije glasi:

$$K(\theta) = \frac{y_r - m_r x_r}{\sin\theta - m_r \cos\theta}; \quad m_r = \frac{y_{r+1}y_r}{x_{r+1}x_r}; \quad \alpha_r \leq \theta \leq \alpha_{r+1}.$$

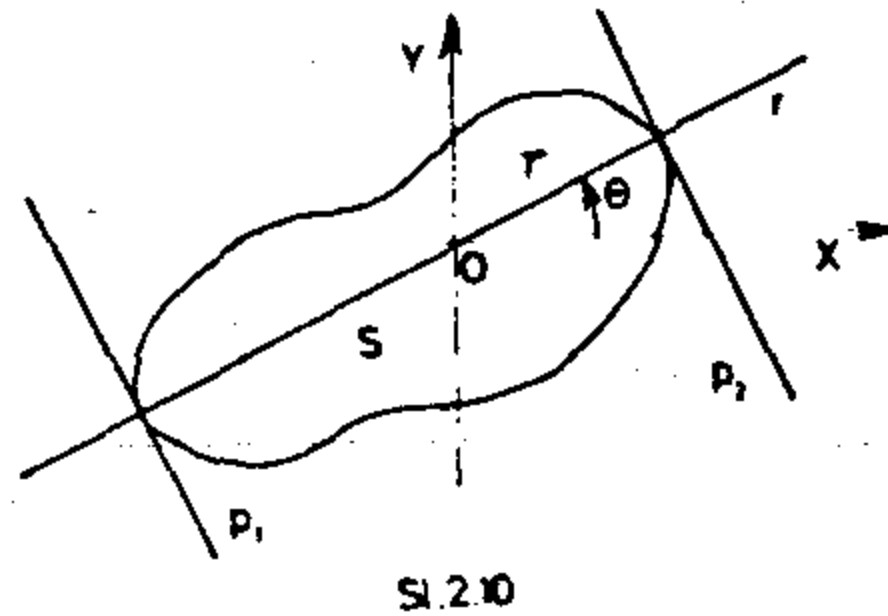
Iz toga vidimo neka važna svojstva kontakt funkcije:

- $\max K(\theta)$ se uvijek postiže u vrhovima HKF
- $\min K(\theta)$ se ne postiže u vrhovima HKF, ako su S_1 i S_2 konveksni
- ako S_1 i S_2 nisu konveksni, onda općenito uzevši njima pripada HKF koji također nije konveksan. $\min K(\theta)$ se tada može postizati i u vrhovima HKF, tj. minimum se postiže u konkavnim vrhovima HKF. (Kažemo da je vrh konkavan ako mu je pripadajući unutarnji kut veći od π .)

8. Potporna funkcija i konstrukcija ekvidistante

Neka je zadan zatvoren nedegeriran objekt $S \subset \mathbb{R}^2$.

Znamo da svakom pravcu $p \subset \mathbb{R}^2$ odgovaraju dva različita potpor-
na pravca $p_1 \parallel p_2$ na objekt S . Sl.2.10.



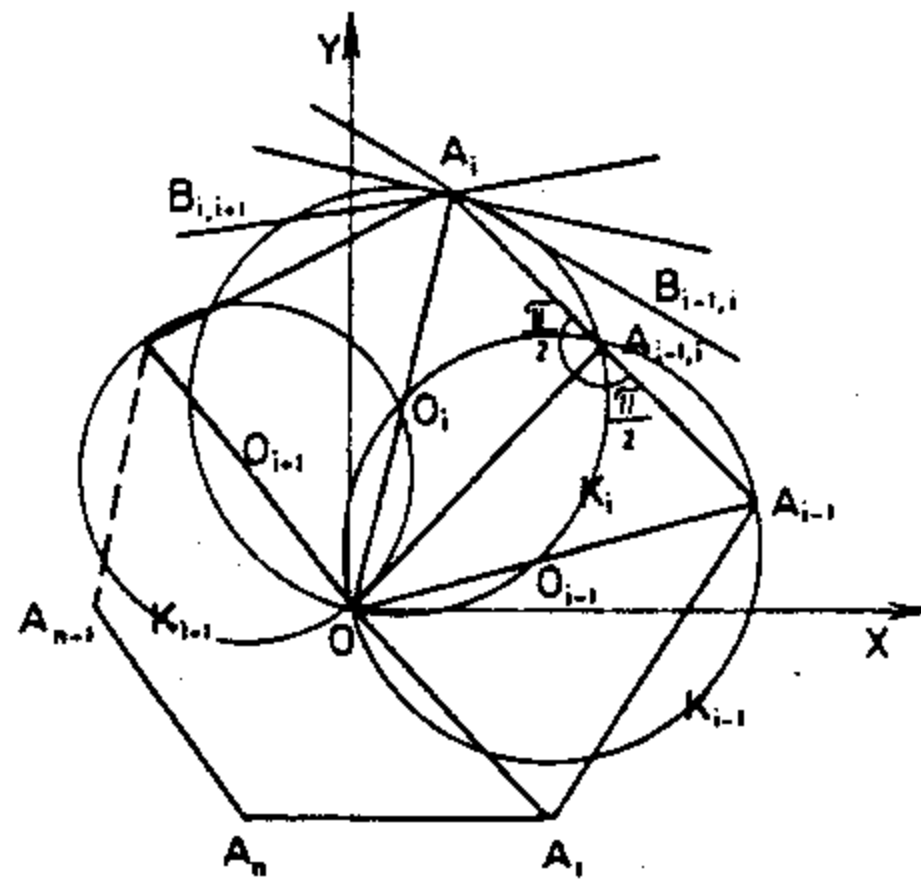
Na potporne pravce spustimo okomiti pravac r koji prolazi kroz pol O objekta S . Udaljenost od ishodišta do potpornog pravca označimo s p ($p > 0$). Pripadni kut vektora \vec{r} sa osi x označimo sa θ . Funkcija $p=p(\theta)$ prema Minkovskom zove se potporna funkcija. Neka svojstva potporne funkcije:

1. Potporna funkcija proizvoljnog objekta podudara se s potpor-
nom funkcijom njegove konveksne ljuske.
2. Potporna funkcija je jednoznačna.
3. Potporna funkcija je periodična, općenito je $p(\theta)+p(\theta+2k\pi)$
 $K \in \mathbb{N}$.
4. Ako objekt ima m - osi simetrije, onda potporna funkcija ima
period

$$T = \frac{2\pi}{m} .$$
5. Potporna funkcija je neprekidna.
6. Ako objekt S nije ograničen onda potporna funkcija nije defi-
nirana na čitavom segmentu $[0, 2\pi]$.

Vidimo, na osnovu svojstva 1, da je za konstrukciju potporne funkcije proizvoljnog objekta dovoljno znati konstrukciju potporne funkcije konveksnog poligona. Recimo da je zadan poligon kao na sl.2.11. Zadani poligon predstavlja konveksnu ljusku pa se potporni pravci konstruišu na slijedeći način:

Spojimo vrhove konveksne ljuske sa polom O i uzmemo da je OA_i dijametar na kome su konstruirane kružnice K_i radijusa $R_i = \frac{1}{2}OA_i$.



Sl. 2.11

Označimo točke presjeka kružnica K_i i K_{i+1} sa $A_{i,i+1}$ dok je druga točka presjeka u polu O . Očito da je $\sphericalangle OA_{i,i+1}$

$$A_i = \sphericalangle OA_{i,i+1} A_{i+1} = \frac{\pi}{2}.$$

Točka presjeka $A_{i,i+1}$ uvijek leži na pravcu koji prolazi kroz vrhove A_i, A_{i+1} .

Kroz vrh A_i povučemo potporni pravac p_i koji može sjeći kružnicu K_i u točkama A_i i $B_{i,i+1}$ ili A_i i $B_{i-1,i}$ ili tangirati kružnicu K_i u točki A_i .

Ako p_i tangira K_i u A_i onda $\rho(O, A_i)$ predstavlja vrijednost potporne funkcije. Granični slučaj potpornog pravca p_i je onaj kada se p_i podudara sa stanicom $\overline{A_i A_{i-1}}$, odnosno $\overline{A_i A_{i+1}}$. Vidimo da je $\rho(O, B_{i,i+1})$ vrijednost potporne funkcije za $\theta = \sphericalangle xOB_{i,i+1}$.

Označimo $\sphericalangle A_n OA_i$ sa θ_i , a $\sphericalangle A_n OA_{i,i+1}$ sa $\beta_{i,i+1}$ i pretpostavimo da se os Ox podudara sa OA_n . U tom slučaju je potporna funkcija $p(\theta)$ konveksnog poligona dana izrazom

$$p(\theta) = \begin{cases} OA_1 \cos(\theta_1 - \theta) & \text{za } \beta_{n1} \leq \theta \leq \beta_{12} \\ OA_2 \cos(\theta_2 - \theta) & \text{za } \beta_{12} \leq \theta \leq \beta_{23} \\ \dots & \dots \\ OA_n \cos(\theta_n - \theta) & \text{za } \beta_{n-1,n} \leq \theta \leq \beta_{n1} \end{cases} \quad (2.8.1)$$

Analitički izrazi za potporna funkciju nekih karakterističnih objekata

a) Potporna funkcija za elipsu $p(\theta) = [a^2 \sin^2 \theta + b^2 \cos^2 \theta]^{1/2}$ (2.8.2)

b) Potporna funkcija krugova glasi $p(\theta) = R = \text{const}$ (2.8.3)

c) Potporna funkcija hiperbole

$$p(\theta) = x_1 \sin \theta - y_1 \cos \theta + (a^2 \sin^2 \theta - b^2 \cos^2 \theta)^{1/2} \quad (2.8.4)$$

Potporna funkcija za hiperbolu nije definirana za

$$\theta \in \left[\arctg\left(-\frac{b}{a}\right), \arctg\frac{b}{a} \right]$$

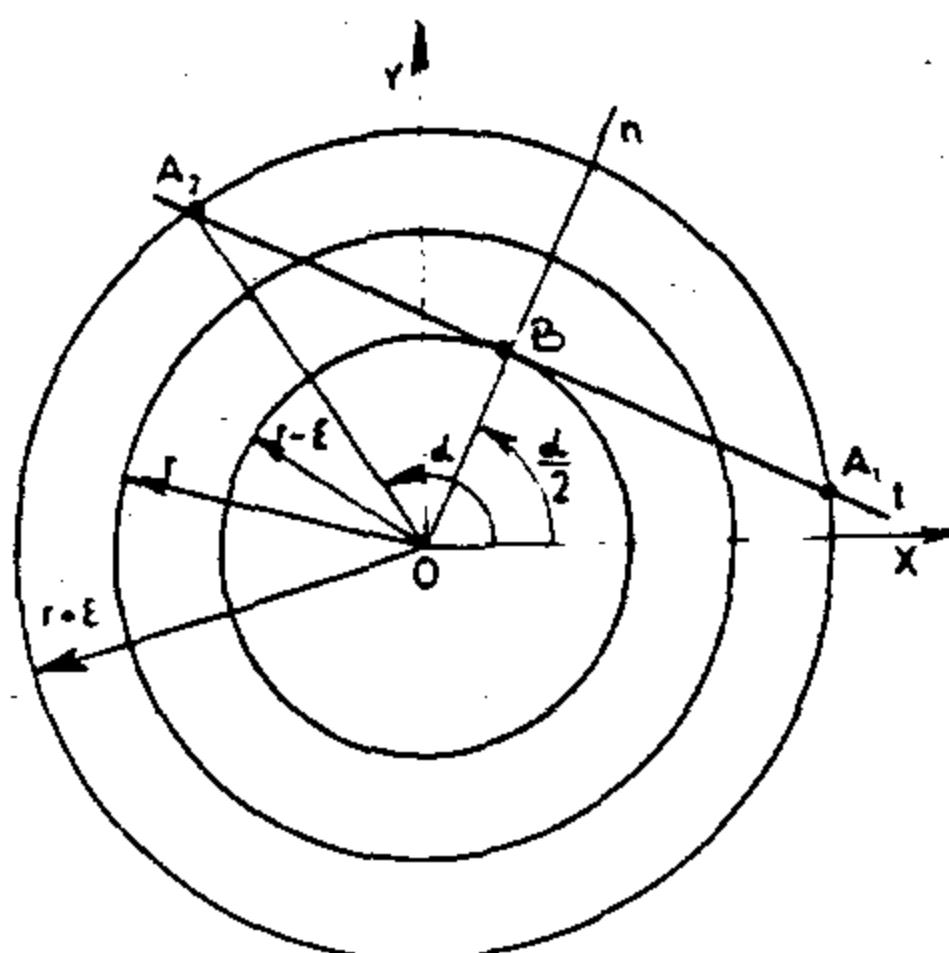
d) Potporna funkcija neograničenog objekta čija je kontura parabola $y^2 = 2px$ je dana sa

$$p(\theta) = \left| \frac{p}{2} \cdot \frac{\cos^2 \theta}{\sin \theta} + x \sin \theta - y_1 \cos \theta \right|$$

Vidimo da za $\theta = 0$ i $\theta = \pi$ prednja funkcija ima prekide II reda.

Konstrukcija ekvidistante

U praksi se vrlo često postavljaju uvjeti da udaljenost medju objektima ne smije biti manja od unaprijed zadane veličine. Ovo se onda reflektira na dimenzije ^{u blizini} HKF. Recimo da nam je zadan objekt S_1 i da oko njega treba konstruirati ekvidistantu na udaljenosti r . Sada iskoristimo ideju o HKF [13]. Naime, uzmemo krug radijusa r i njega aproksimiramo poligonom S_2 sa ϵ točnošću. Zatim konstruiramo HKF S_1 i S_2 što sada predstavlja ekvidistantu oko objekta S_1 na udaljenosti r sa tošnošću ϵ . Naime neka su zadana tri kruga s radijusima $r-\epsilon$, r ; $r+\epsilon$. Sl.2.12.



Kroz točku B na kružnici $r-\varepsilon$ povučemo tangentu t i normalu n . Tangenta t na kružnicu radijusa $r-\varepsilon$ u točki B siječe kružnicu radijusa $r+\varepsilon$ u točkama A_1 i A_2 . Svaka točka dužine $\overline{A_1 A_2}$ aproksimira luk kružnice $A_1 A_2$ radijusa r s točnošću $\pm\varepsilon$.

Vidimo da je

$$\alpha = 2 \arccos \left(\frac{r-\varepsilon}{r+\varepsilon} \right). \quad (2.8.6)$$

Broj vrhova aproksimirajućeg poligona je dan izrazom:

$$n = N \left[\frac{\pi}{2 \arccos \left(\frac{r-\varepsilon}{r+\varepsilon} \right)} + 1 \right] \quad (2.8.7)$$

Uzmemo li da je aproksimirajući poligon pravilni n -trokut aproksimacija neće biti lošija ako za veličinu luka $A_1 A_2$ uzmemo da je

$$\phi = \frac{2\pi}{K}. \quad (2.8.8)$$

Koordinate vrhova pravilnog n -terokuta dane su izrazima

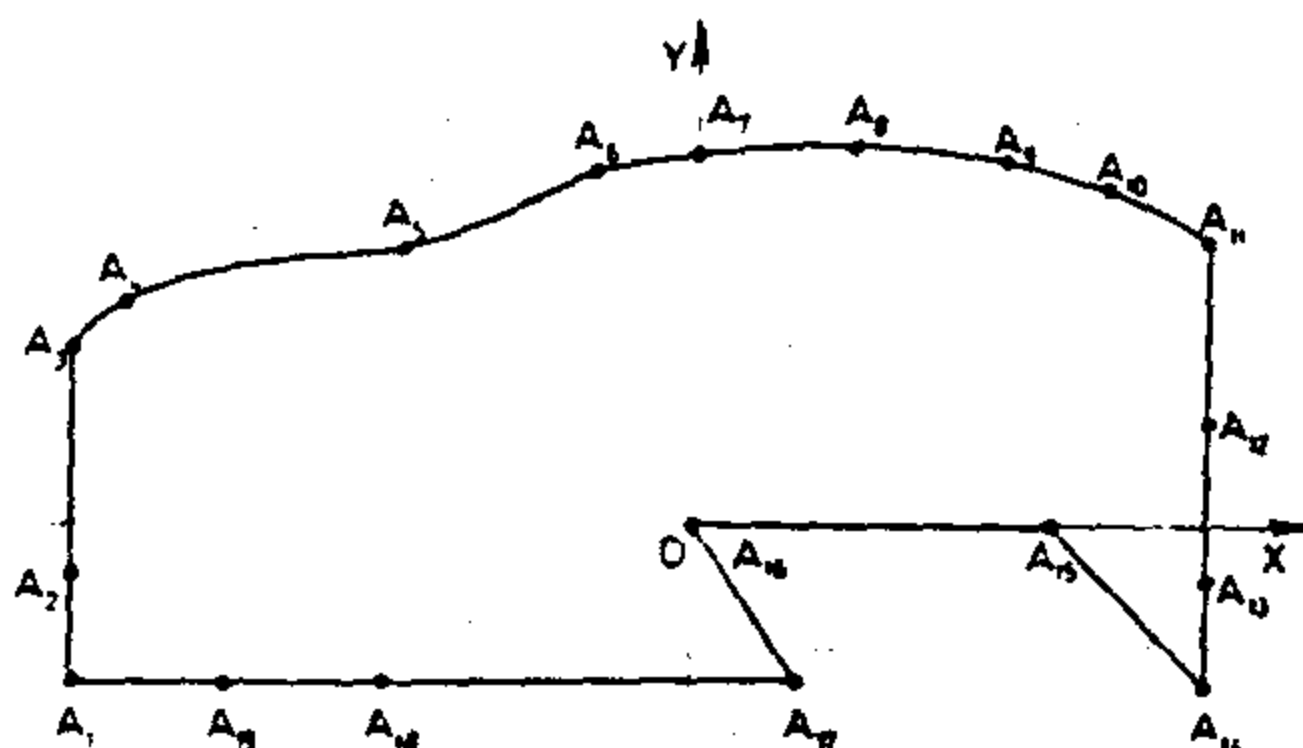
$$\begin{aligned} x_a &= (r+\varepsilon) \cos \left[\frac{2\pi(a-1)}{n} \right] \\ y_a &= (r+\varepsilon) \sin \left[\frac{2\pi(a-1)}{n} \right] \quad a=1, 2, \dots, n \end{aligned} \quad (2.8.9)$$

9. Aproksimacija granice zadanog objekta

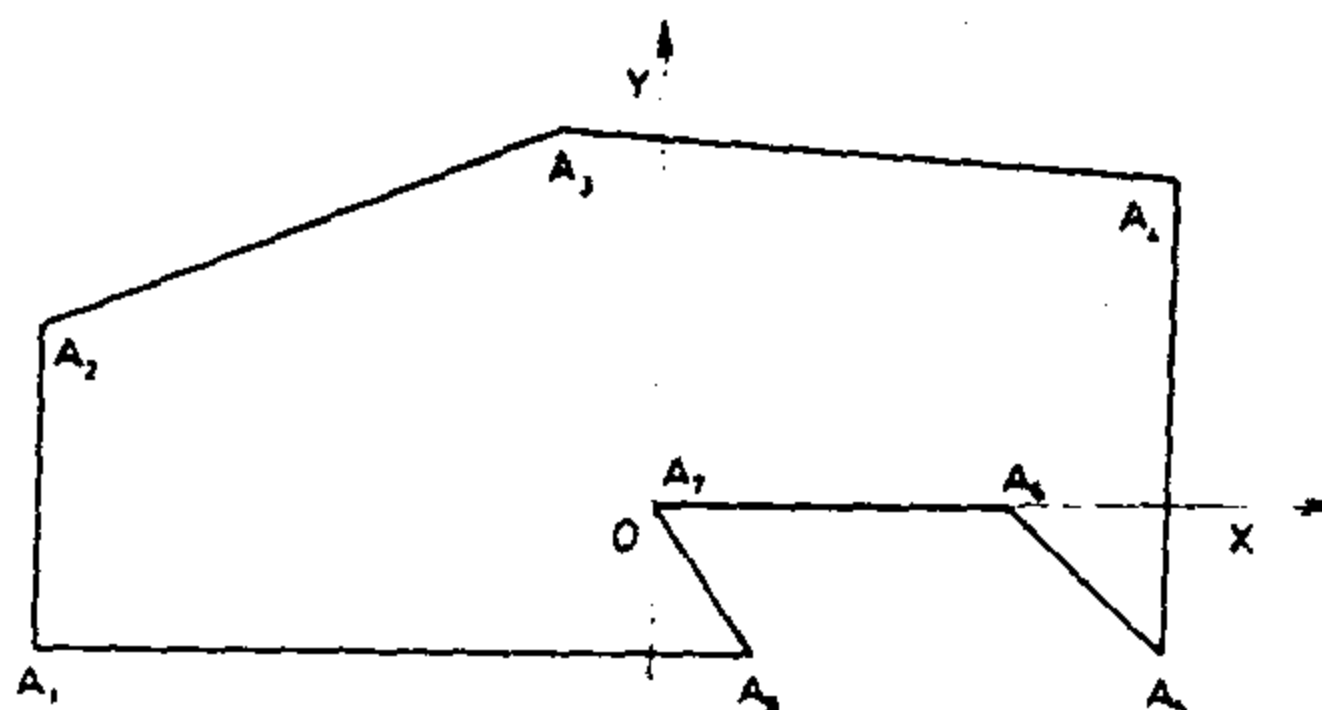
U praksi se susrećemo s raznim načinima dobijanja informacija konturi objekta. Vrlo često imamo zapravo višak informacija jer se kontura opisuje precizno a dalja tehnološka obrada (rezanje piljenje, krojenje itd.) se obavlja sa znatno manjom tačnošću. S druge strane imamo problem da stvarnu konturu objekta treba aproksimirati poligonom sa zadanom tačkom.

U ovom radu je obradjen slučaj kada su podaci o konturi zadani u diskretnom skupu tačaka, a konturu želimo aproksimirati poligonom s unapred zadanom tačnošću. Širenje ili sužavanje broja tačaka za koje promatrani pravac vrši traženu aproksimaciju realizirat će se pomoću brojeva Fibonaci [38].

Na sl.2.13. je dana kontura objekta sa 19 vrhova, a na sl.2.14. predstavljena je aproksimacija s poligonom koji ima 8 vrhova s tačnošću $\epsilon = 0,5$. To je dobiveno pomoću kompjutorskog programa APROX.



Sl.2.13



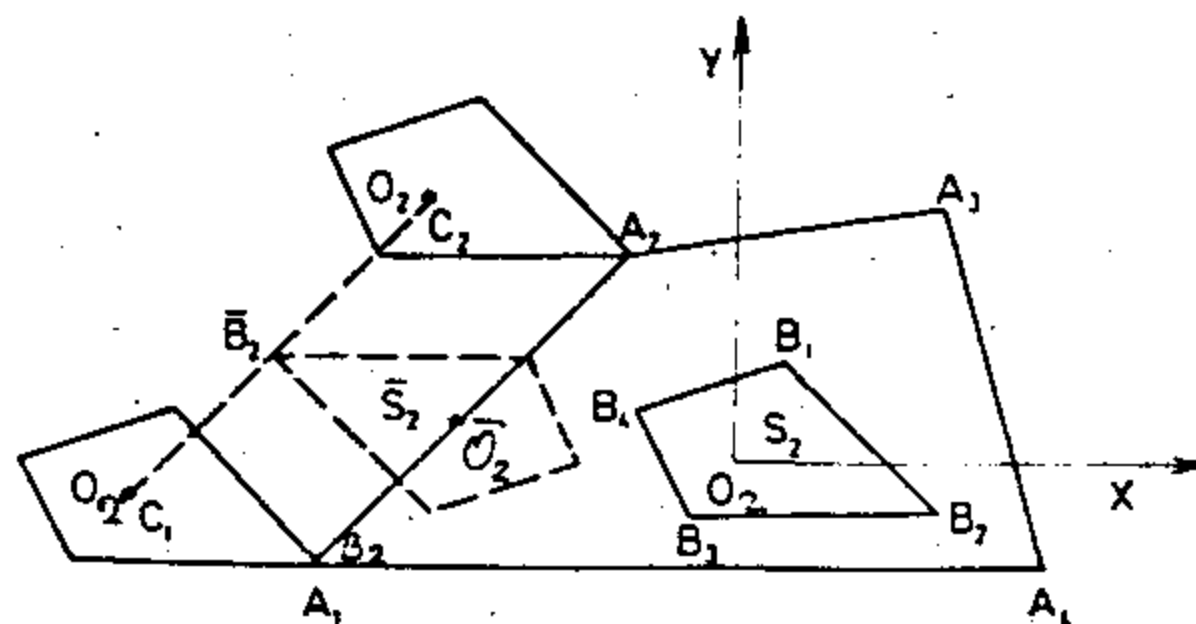
Sl.2.14.

Neke napomene u vezi s glavom II

Vidimo da se izračunavanje vrijednosti KF vrši pomoću HKF dotičnih objekata. Ako su objekti konveksni, onda se odmah vrši konstruiranje HKF. Međutim ako bar jedan od objekata nije konveksan, već je i višestruko suvisao, tada se prethodno izvrši razbijanje na konveksne dijelove, zatim se formiraju parcijalne konveksne oblasti G_{rs} , koje uniramo. Granica unije predstavlja parcijalni hodograf. Pored drugih slučajeva kod uniranja jako dolaze do izražaja problemi vezani za kompjutersku obradu geometrijskih problema. Čini se da su one poteškoće koje su geometrijske prirode prevladane u problemima koji su rješavani u ovoj disertaciji posebno s točke gledišta praktične primjene. Tako npr. geometrijska konstrukcija HKF opisana na početku ove glave i prikazana na sl.2.2. ima velike prednosti zbog jednostavnosti same konstrukcije a i zbog očiglednosti izvjesnih teorema. Međutim kada se navedeni postupak želi realizirati pomoću računara nailazi se na takve poteškoće da se kod poligona sa složenim granicama teško postiže određena točnost za razumno vrijeme. Zato je program HOKOF-II radjen po algoritmu koristeći slijedeći teorem.

Teorem 2.19.

Kod gibanja pola O_2 po granici objekta S_1 vanjska granica "traga" objekta \bar{S}_2 predstavlja HKF konveksnih objekata S_1 i S_2 . Objekt \bar{S}_2 se dobije iz S_2 , tako da ga zarotiramo za Π , što se svodi na promjenu predznaka vrhova koordinata S_2 . Skup položaja objekta \bar{S}_2 , kada se njegov pol kreće po nekoj liniji, nazivamo tragom objekta \bar{S}_2 , sl.2.15.



Sl. 2.15

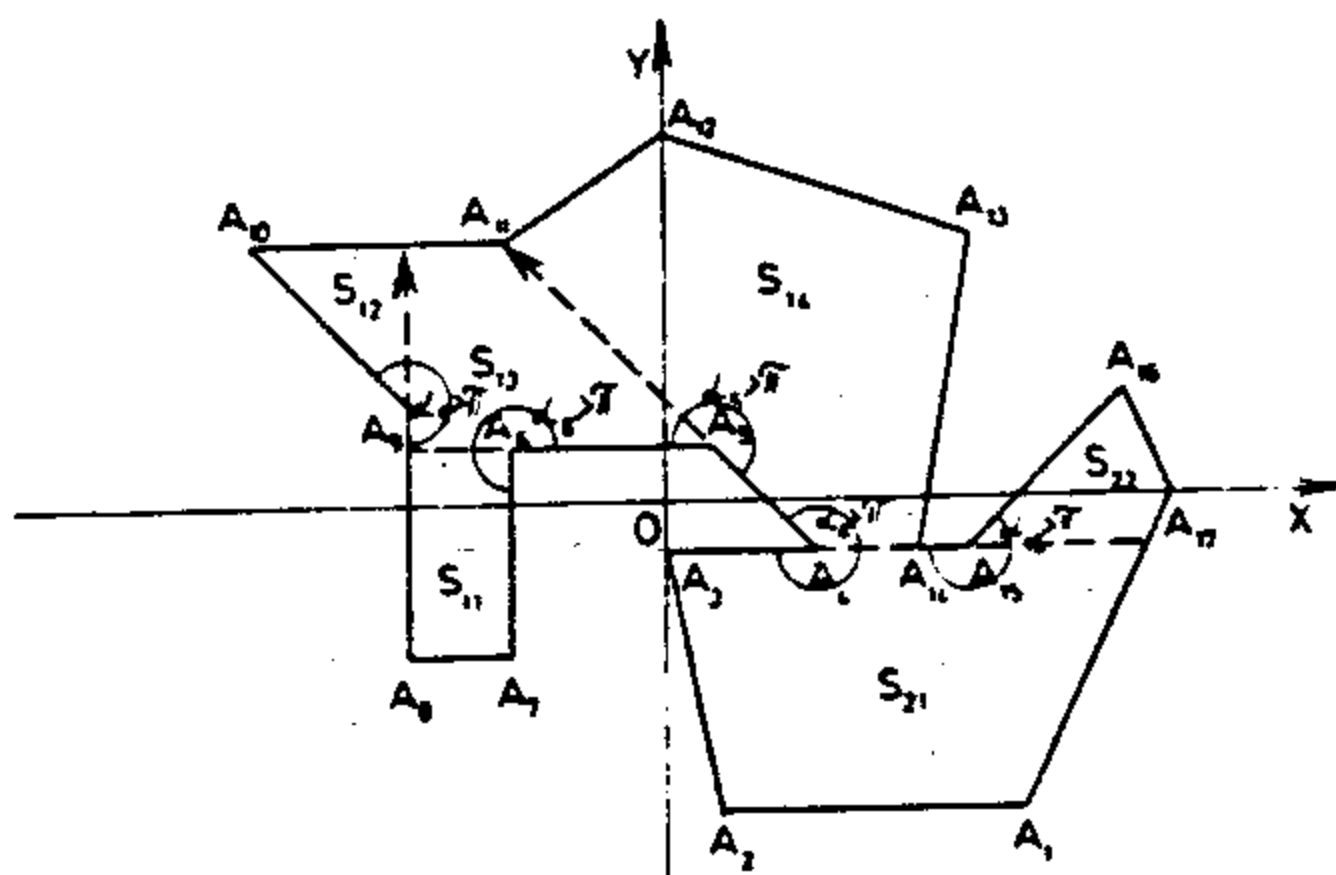
Dokaz

Smjestimo objekte tako da bude $O_1=O_2$. Pomaknemo S_2 tako da se vrh B_2 podudara sa A_1 . Kada se vrh B_2 kreće po stranici $\overline{A_1A_2}$ onda se pol O_2 kreće po liniji $\overline{C_1C_2} \parallel \overline{A_1A_2}$, znači da $\overline{C_1C_2}$ određuje stranicu HKF.

Smjestimo pol objekta \bar{S}_2 na $\overline{A_1A_2}$ pa vidimo da je vrh $\bar{B}_2 \in \bar{S}_2$ najudaljeniji od stranice $\overline{A_1A_2}$ i određuje vanjsku granicu traga objekta \bar{S}_2 , kada se pol O_2 objekta \bar{S}_2 kreće po $\overline{A_1A_2}$. No $O_2A_1 \parallel \bar{B}_2\bar{O}_2$ i jednaki su, pa je duljina $\overline{A_1A_2} = \overline{C_1C_2}$, slijedi $\overline{C_1C_2} \subset \text{HKF}$.

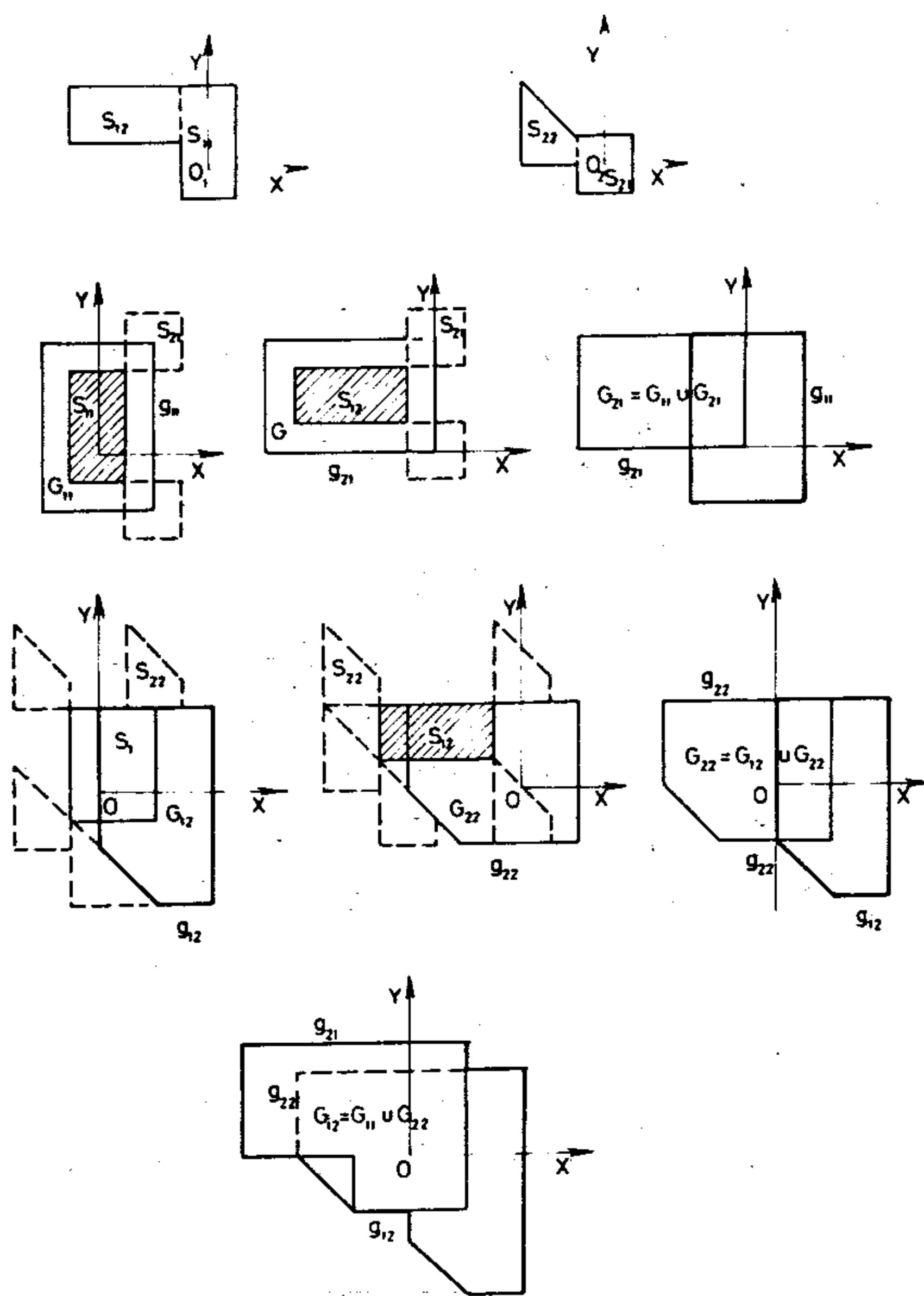
Pokazalo se da nije dobro raditi tako da se nadju sve parcijalne oblasti i onda odjednom izvrši uniranje svih njih a zatim nadje granica te unije koja predstavlja HKF promatranih objekata. Naime, uniranje treba izvršiti odmah kada se koja parcijalna oblast nadje.

Razbijanje objekta na sl.2.16. u konveksne dijelove dobijeno je pomoću programa "RAZBIJANJE".



Sl.2.16

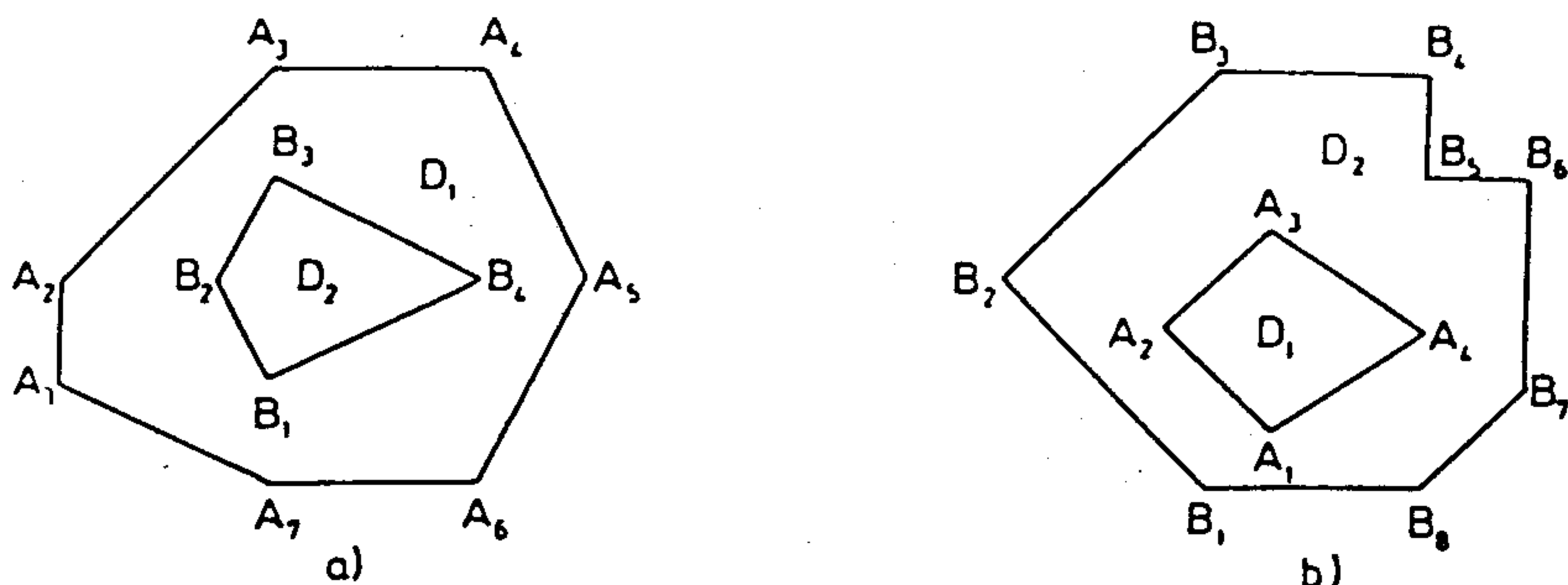
Na sl.2.17. vidimo postupnost formiranja HKF dvaju nekonveksnih objekata S_1 i S_2 dobivenih pomoću programa HKOF, kada su S_1 i S_2 razdjeljivi.



Sl.2.17

Za dobijanje HKF nerazdjeljivih i nekonveksnih objekata bilo je potrebno razviti vrlo opsežan i kompliciran program za rad sa skupovima. Ukazujemo na neke komplikacije koje se općenito javljaju.

1. Ako imamo područja D_1 i D_2 , kao na sl.2.18, onda nema nikakvih komplikacija.

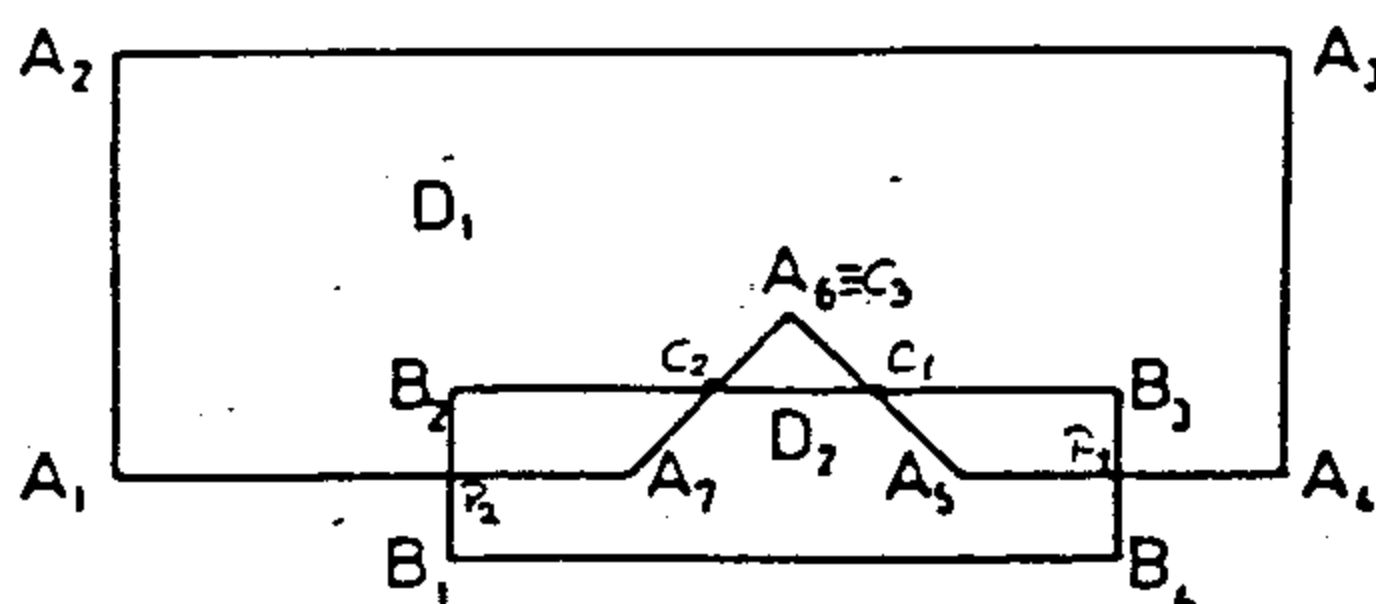


Sl.2.18

Za sve A_i vrijedi $A_i \notin D_2$, za sve B_j vrijedi $B_j \in D_1$ i za sve stranice od D_2 vrijedi $B_j B_{j+1} \subset D_1$, sl.2.18a, zato je $D = D_1 \cup D_2 = D_1$.

Za situaciju na sl.2.18b je $D = D_1 \cup D_2 = D_2$.

2. Na sl.2.19. vidimo slučaj da su D_1 i D_2 suvisle oblasti, dok $D = D_1 \cup D_2$ nije suvisla, pa se HKF raspada u dva zatvorena dijela (vanjski i unutarnji).



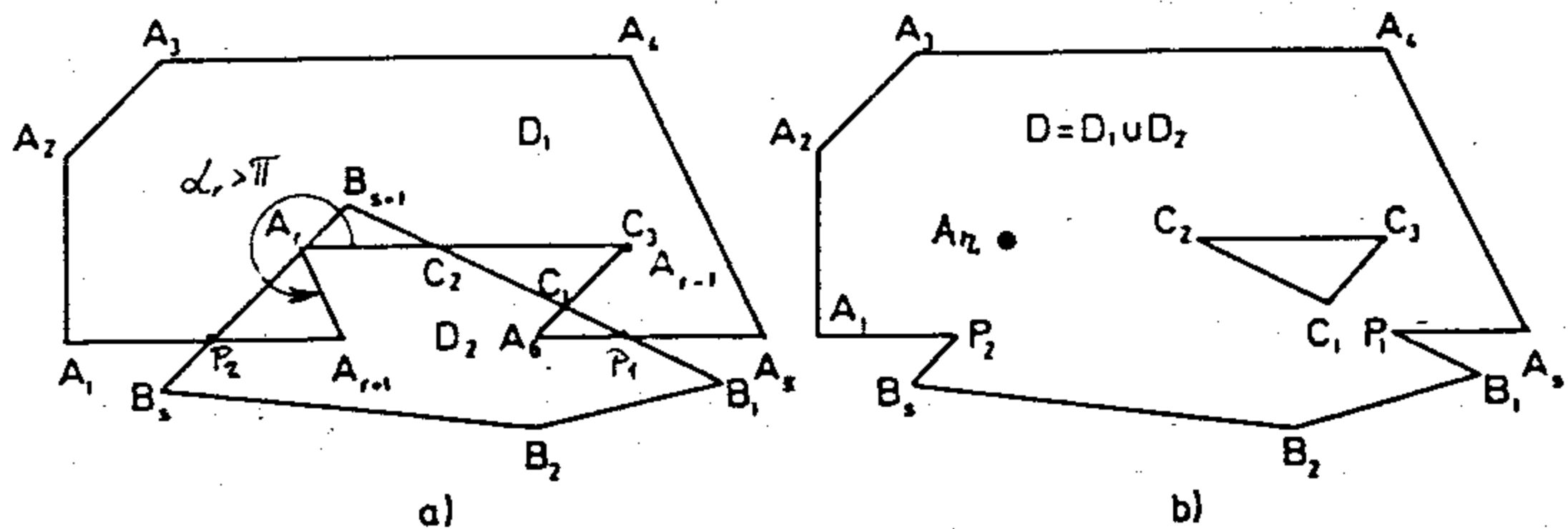
Sl.2.19

Vanjski dio HKF : $A_1 A_2 A_3 A_4 P_1 B_4 B_1 P_2$
 Nutarnji dio HKF: $C_1 C_2 C_3$

3. Problem izolirane točke. Neka su D_1, D_2 suvisla područja. U toku procesa postupnog uniranja uvijek se već postojećoj oblasti koja, općenito, već kod drugog uniranja prestaje biti konveksna, pa čak i suvisla, dodaje oblast D_2 , koja je uvijek konveksna. Recimo da imamo situaciju kao na sl.2.20.

HKF čini: - vanjska kontura $A_1 A_2 A_3 A_4 P_1 B_1 B_2 B_3 P_2$
 - unutarnja kontura $C_1 C_2 A_{r-1}$
 - izolirana točka $A_{r=7}$

Točka A_r pripada HKF kao izolirana točka jer ona ima sljedeća svojstva:



Sl.2.20

Svojstvo A

A_r pripada konkavnom vrhu oblasti D_1 tj. $\angle A_{r-1} A_r A_{r+1} > \pi$, što znači kada u pravac p_{r-1} uvrstimo koordinate točke vrha

A_{r+1} vrijedi $p_{r-1}(x_{r+1}, y_{r+1}) > 0$

gdje je $p_r(x, y) \equiv (y - y_{r-1})(x - x_{r-1}) + (x_{r-1} - x)(y_r - y_{r-1}) = 0$ (a)
 jednačba pravca koja prolazi kroz $A_{r-1} A_r \subset D_1$

Svojstvo B

Vrh A_r se nalazi na granici oblasti D_2 tj. $A_r \in \overline{B_s B_{s+1}}$.

To znači da je:

$$g_s(x_r, y_r) \equiv (y_r - y_s)(x_{s+1} - x_s) + (x_s - x_r)(y_{s+1} - y_s) = 0 \quad (b)$$

$$g_k(x_r, y_r) < 0 \quad \text{za} \quad K : K = 1, 2, \dots, m; \quad K \neq s \quad (c)$$

gdje (x_s, y_s) koordinate vrha B_s (x_{s+1}, y_{s+1}) koordinate vrha B_{s+1} .

Svojstvo C

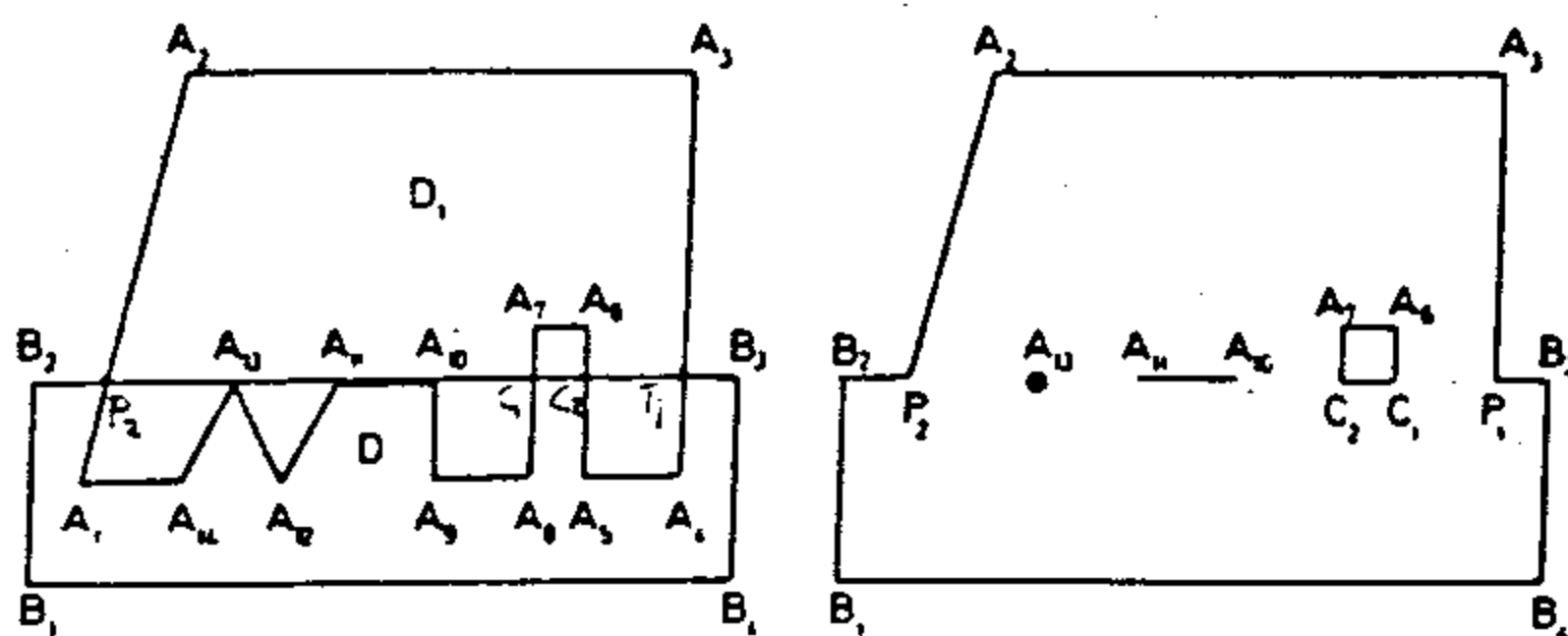
Vrhovi A_{r-1} i A_{r+1} nalaze se desno od pravca koji prolazi kroz $\overline{B_s B_{s+1}}$ tj. vrijedi

$$g_s(x_{r-1}, y_{r-1}) < 0 \quad (d)$$

$$g_s(x_r, y_r) < 0 \quad (e)$$

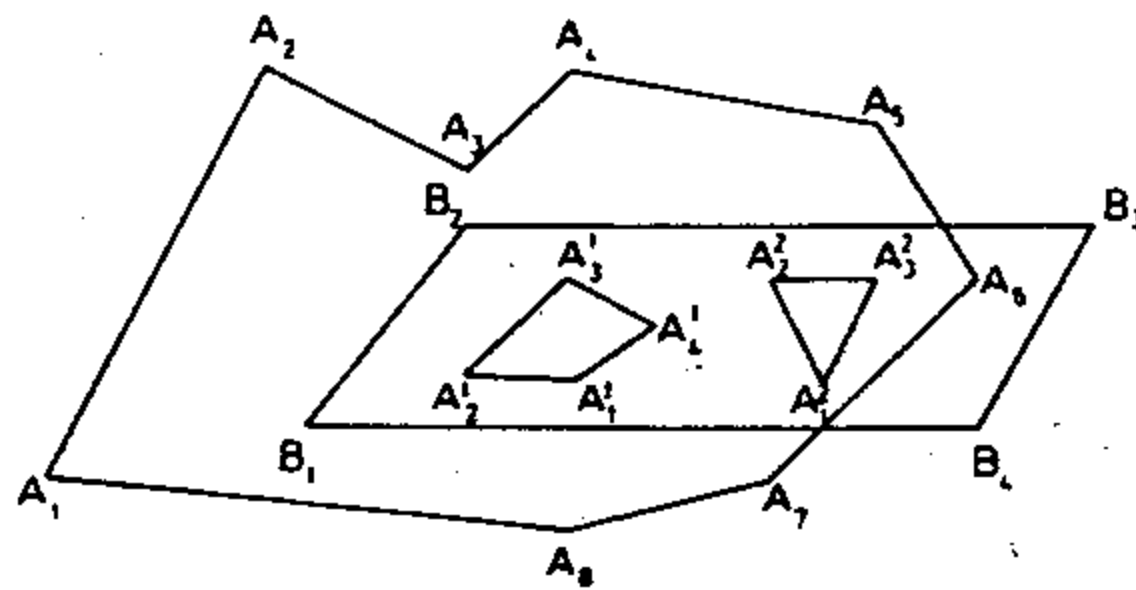
Ako su ispunjena sva tri svojstva uvjeta A, B i C, onda točka A_r pripada HKF kao izolirana točka. Naime, u HKF su ušle točke $A_1 A_2 A_3 A_4$ u vanjsku konturu a $A_{r-1} \equiv C_3$ unutarnju. Točke A_5 i $A_{r+1} \equiv A_8$ su isključene iz razmatranja jer su A_5 i $A_{r+1} \in D_2$. Točka $A_r = A_7 \in D_2$, ali je na granici. Za sve takve točke moraju se provjeriti sva tri uvjeta da ne bi ispustili neku od izoliranih točaka HKF.

4. D_1, D_2 su suvisle a u HKF ulaze pored izoliranih točaka i izolirani intervali sl.2.21.



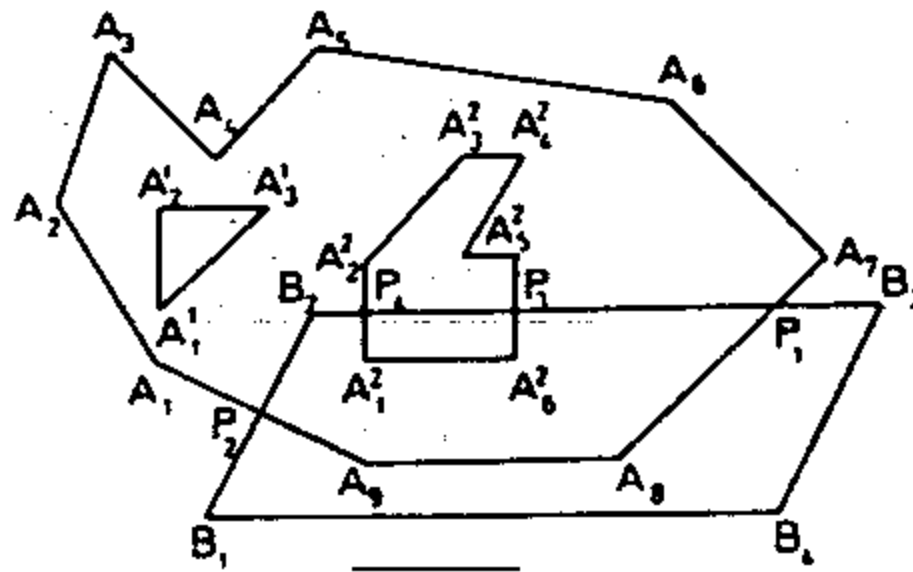
- HKF - vanjska kontura je $B_1 B_2 P_2 A_2 A_3 P_1 B_3 B_4$
 - unutarnja kontura je $C_1 C_2 A_6 A_7$
 - interval $\overline{A_{10} A_{11}}$
 - izolirana točka A_{13} .

5. Može se desiti da D_1 bude višestruko suvisla pa da poslije unije sa D_2 postane suvisla jer D_2 prekrije rupe. Sl.2.22.



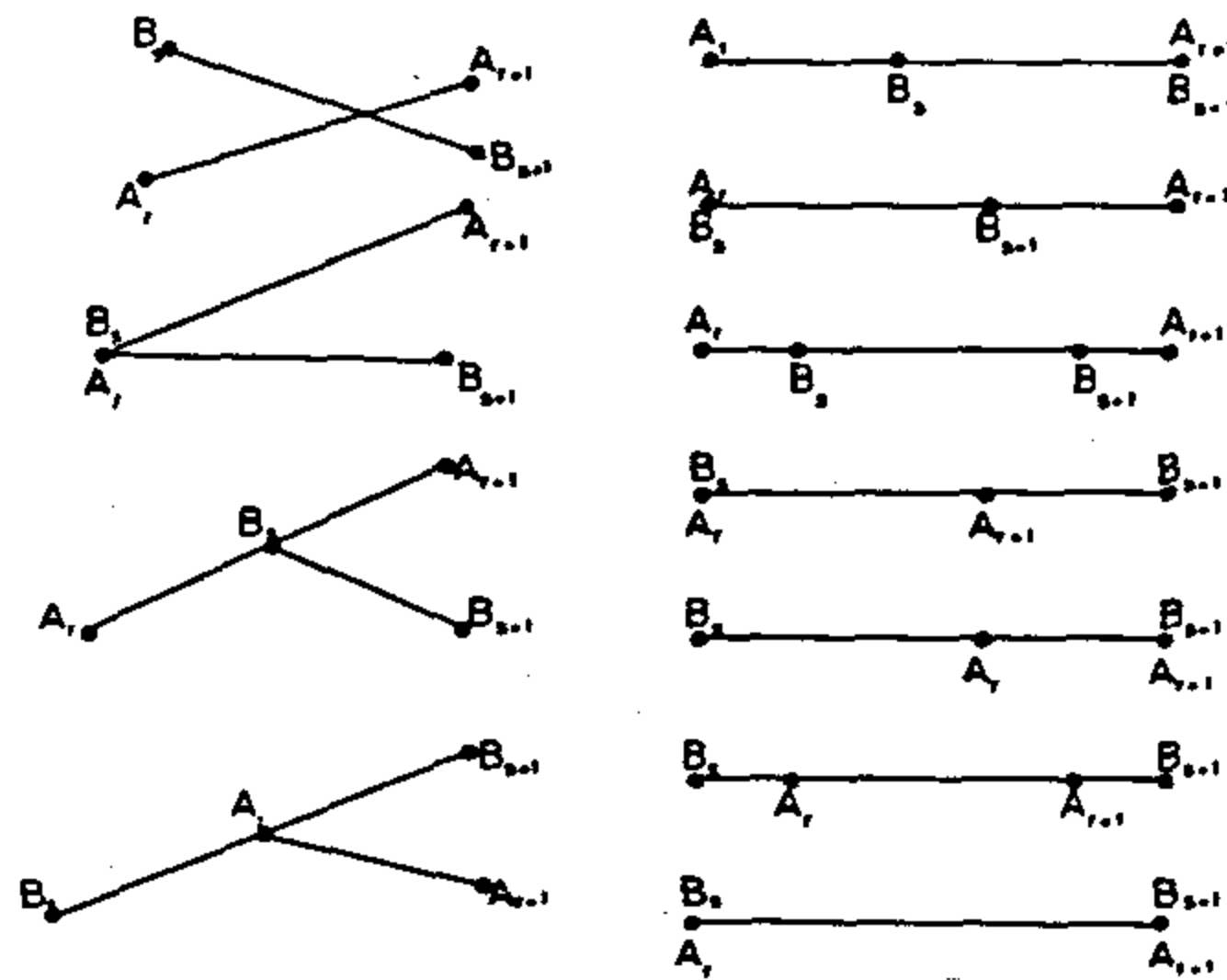
Sl.2.22

6. D_1 - višestruko suvisla, D_2 konveksno. $D = D_1 \cup D_2$ je višestruko suvisla - Sl.2.23.



Sl.2.23

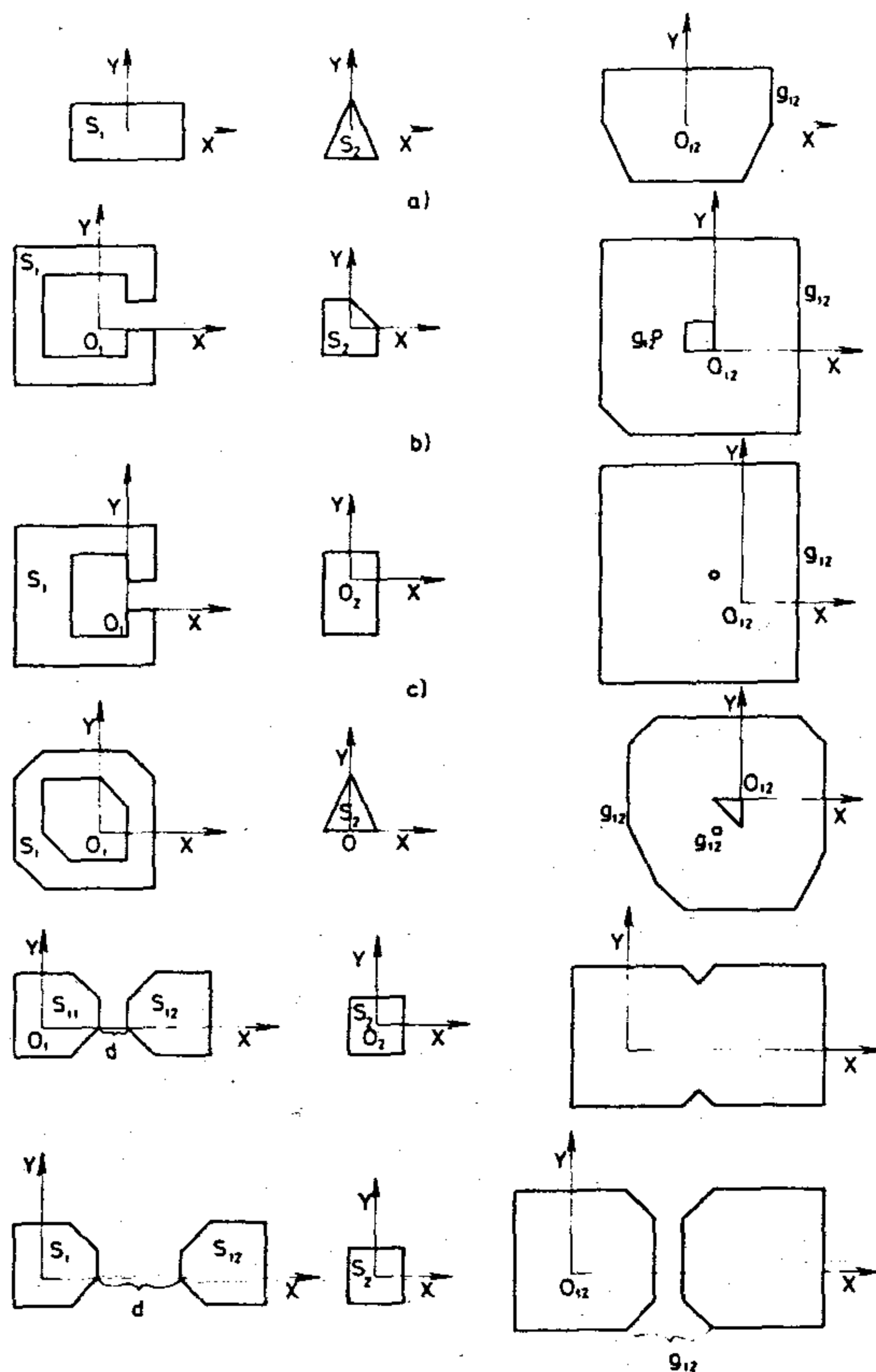
Što se tiče odnosa stranica odnosno vrhova oblasti D_1 i D_2 , očito je da su mogući slijedeći međusobni položaji. Sl.2.24.



Sl.2.24

Na sl.2.2 5. vidimo nekoliko karakterističnih HKF za različite vrste objekata:

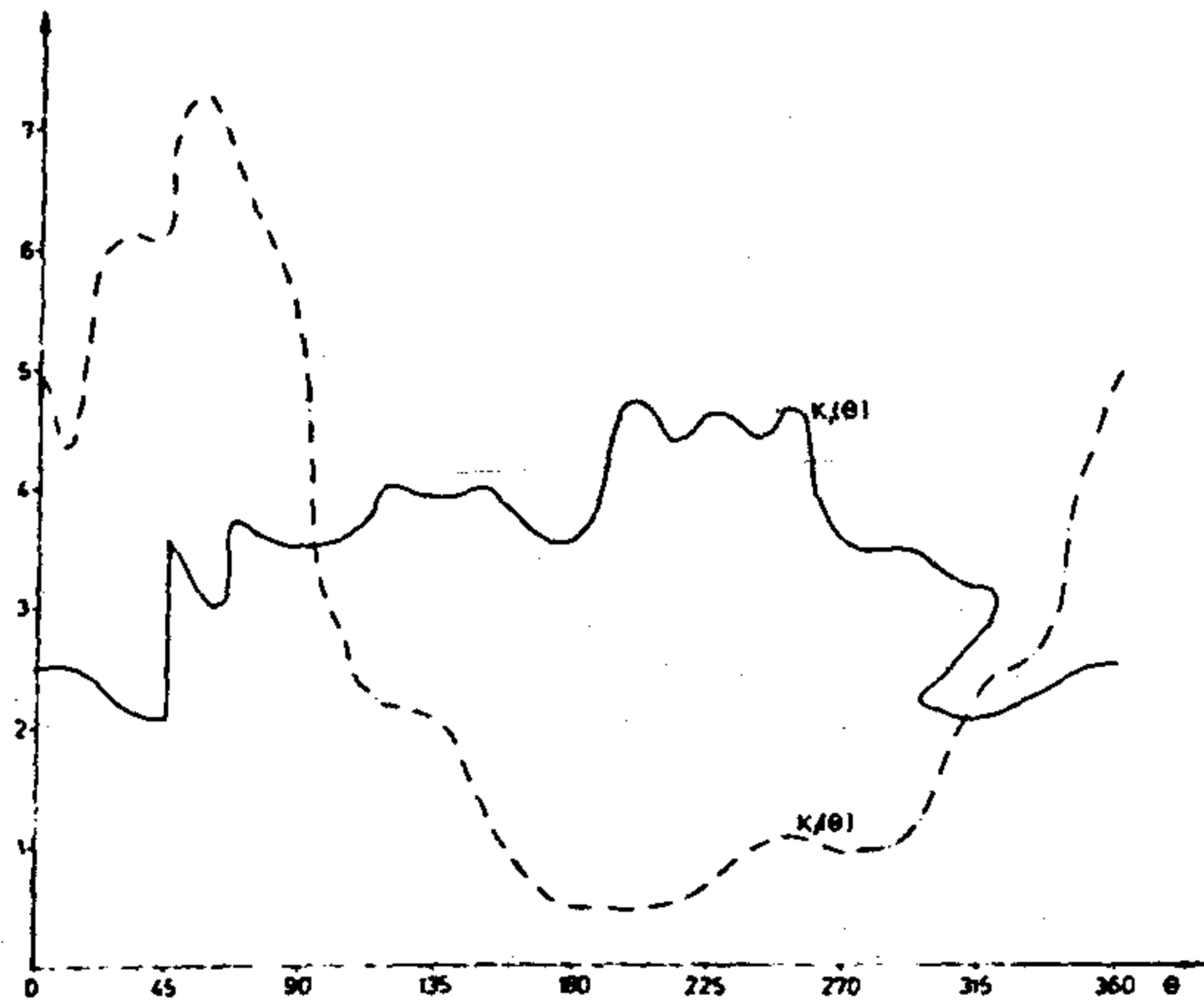
- a) S_1 i S_2 su konveksni
- b) nerazdjeljivi
- c) s točnim ulazom
- d) višestruko suvisli
- e) nepovezani s preuskim prolazom
- f) nepovezani s dovoljnim prolazom



S(2.25)

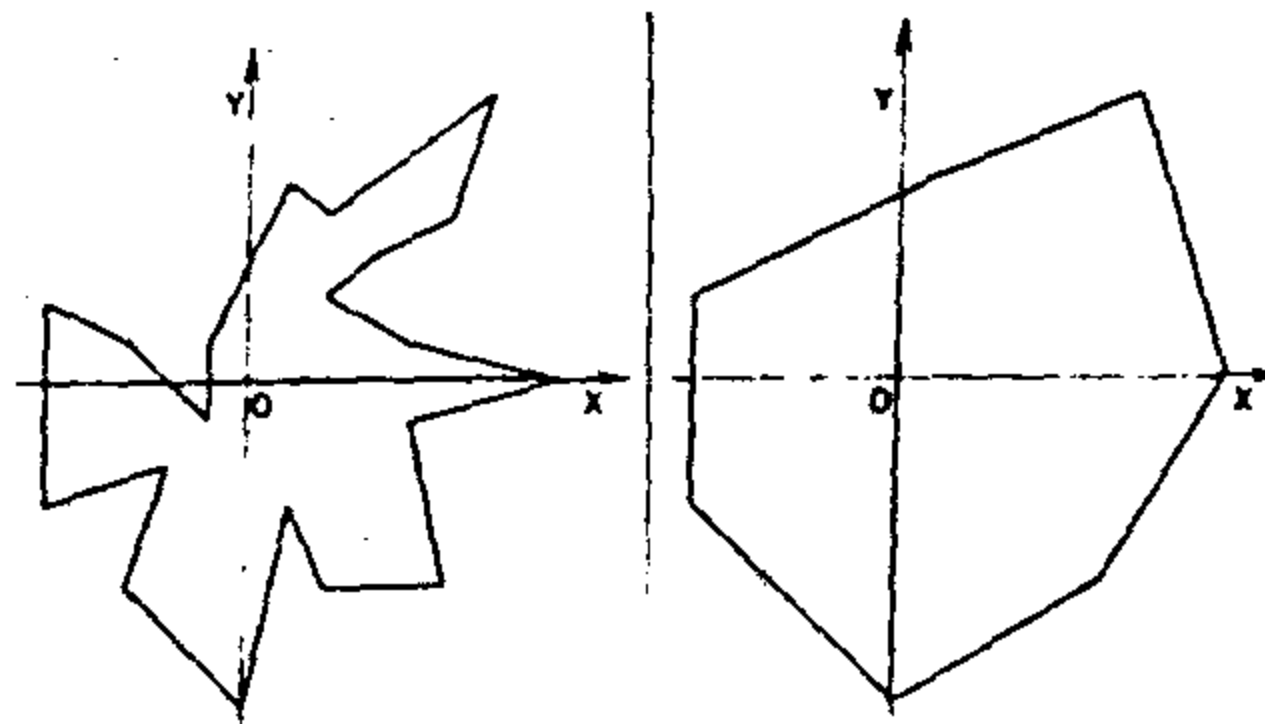
Kao što se vidi iz oblika grafa, može se izvesti izvjestan sud o vrsti ovjekta.

Za HKF dan na sl.2.9. pomoću programa KONFU izračunate su vrijednosti kontakt funkcije i to $K_1(\theta)$ kada je pol smješten u ishodište $O_{12}(0,0)$ te $K_2(\theta)$ kada je pol smješten u točki $O_{12}(-3,0,-2,5)$, tj. u točki iz zvijezdaste oblasti Ω . Sl.2.26.



Sl 2.26

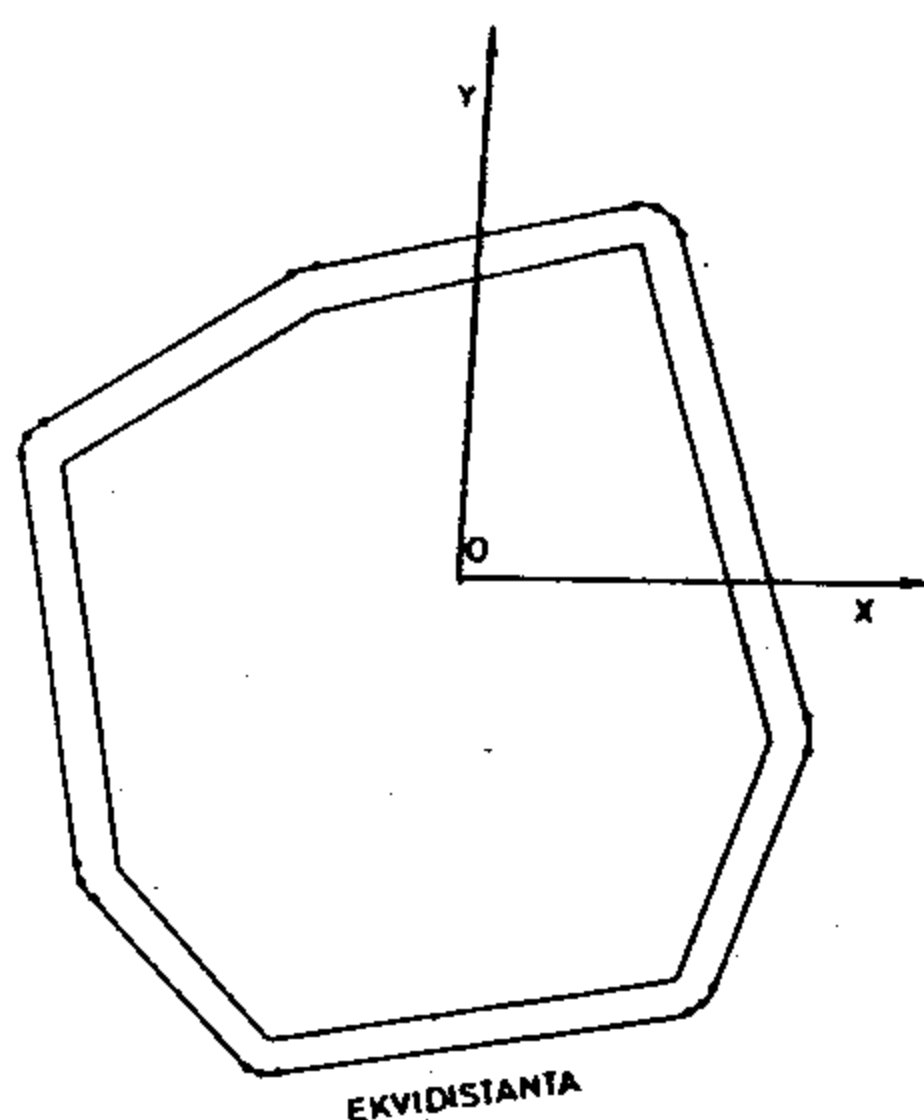
Da bi za zadani objekt našli vrijednost potporne funkcije prvo se pomoću programa "LJUSKA" nađe konveksna ljuska za zadani objekt na sl.2.27. a zatim se nađu vrijednosti potporne funkcije za ljusku sl.2.28.



Sl 2.27

Sl 2.28

Na bazi HKF za objekt sl.2.29. dobivena je ekvidistanta $\epsilon=0,5$, pomoću programa EQVID, koju vidimo na slici.



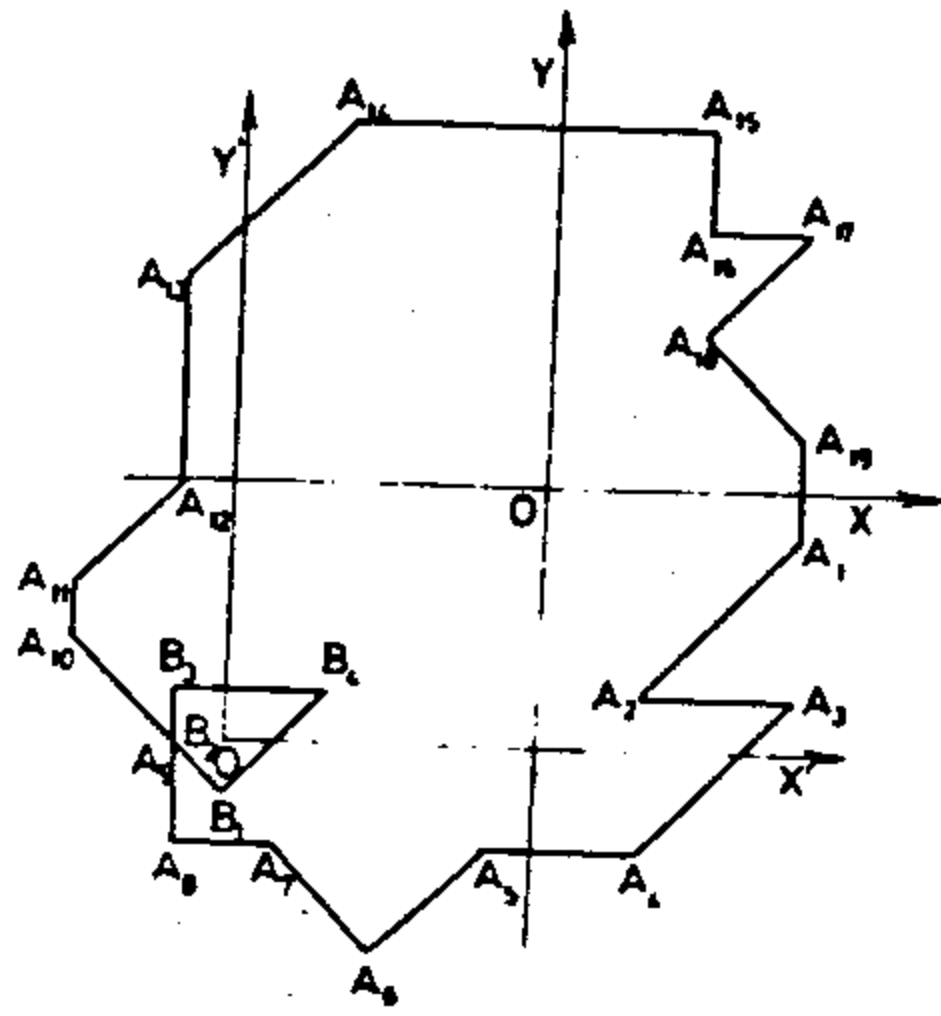
Sl. 2.29

Pomoću programa ZVIJEZDA nalazimo, ako postoji, zvijezdastu oblast Ω u koju smještamo pol hodografa kako bi kontakt funkcija bila jednoznačna. Za hodograf prikazan na sl.2.9. nadjena je konveksna oblast Ω sa vrhovima: $B_1(-3.0, -3.0)$; $B_2(-3.5, -2.5)$; $B_3(-3.5, -2.0)$; $B_4(-2.0, -2.0)$. Sl.2.30.

Navedenih devet kompjutorskih programa od kojih svaki u sebi sadrži razne podprograme, javljaju se skoro u svim problemima koje sam rješavao u ovoj disertaciji i oni čine temeljne programe.

Kod neregularnog smještanja objekata, pojedini programi se izvršavaju i po nekoliko desetina tisuća puta što me je prisililo da strogo vodim računa o optimalizaciji vremena izvodjenja takvih programa.

Naime, pokazalo se da su neki teoretski dobri algoritmi zapravo neprihvatljivi upravo zbog toga što kompjutorska realizacija zahtijeva utrošak vremena koje nije ekonomski opravdano.



SI. 2.30

PROGRAMI UZ GLAVU II

.....

• NASTAVNA: ODREĐIVANJE HODOGRAFA KONTAKT FUNKCIJE (HKF)

.....

```
COMMON /APOL/ XA(50), YA(50), N
      /BEPO/ XB(50), YB(50), M
      /VHKF/ XC(90), YC(90), T
```

```
DIMENSION RI(50)
INTEGER T, R, TZ
```

```
I = 0
J = 0
T = 0
```

```
DO 10 J = 1, N
  XB(J) = -XB(J)
  YB(J) = -YB(J)
10 CONTINUE
```

```
NZ = N + 1
NZ = 1 + 1
```

```
XA(NZ) = XA(1)
YA(NZ) = YA(1)
XB(NZ) = XB(1)
YB(NZ) = YB(1)
```

```
I = I + 1
T = T + 1
```

```
XRAZ = XA(I + 1) - XA(I)
YRAZ = YA(I + 1) - YA(I)
```

```
DO 25 J = 1, M
```

```
RI(J) = (YB(J) - YA(I)) * XRAZ -
      (XB(J) - XA(I)) * YRAZ
```

```
25 CONTINUE
```

```
RMAX = RI(1)
```

```
K=1
```

```
K1 = 1
```

```
DO 30 J = 2, M
```

```
IF(RI(J) .LE. RMAX) GO TO 30
```

```
RMAX = RI(J)
```

```
K = J
```

```
K1 = J
```

```
30 CONTINUE
```

```
XC(T) = XA(I) + XB(K)
```

```
YC(T) = YA(I) + YB(K)
```

```
XC(T+1) = XA(I+1) + XB(K)
```

```
YC(T+1) = YA(I+1) + YB(K)
```

```
50 I = T + 2
```

```
60 I = I + 1
```

```
IF(NZ+1-I) 80, 170, 60
```

```

80 XPAZ = XA (I + 1) - XA(I)
   YPAZ = YA (I + 1) - YA(I)

```

```

C
  DO 85 J = 1, N

```

```

    RI(J) = (YE(J) - YA(I)) * XPAZ -
    * (ZE(J) - ZA(I)) * YPAZ

```

```

85 CONTINUE

```

```

C
  R = 1

```

```

  RMAX = RI(1)

```

```

  DO 90 J = 2, N

```

```

    IF(RI(J), LE, RMAX) GO TO 90

```

```

    RMAX = RI(J)

```

```

    R = J

```

```

90 CONTINUE

```

```

C
C
  IF(R = 1) 130, 110, 130

```

```

C
110 XC(T) = XA(I+1) + ZE(T)

```

```

   YC(T) = YA(I+1) + YE(T)

```

```

   T = T + 1

```

```

   GO TO 60

```

```

C
130 R = R + 1

```

```

   IF(R, EQ, NZ) 1 = 1

```

```

   IF(R, EQ, N1) GO TO 170

```

```

140 XC(T) = XA(I) + ZE(T)

```

```

   YC(T) = YA(I) + YE(T)

```

```

   IF(R, EQ, R) GO TO 160

```

```

   T = T + 1

```

```

   GO TO 130

```

```

160 XC(T+1) = YA(I+1) + ZE(T)

```

```

   YC(T+1) = YA(I+1) + YE(T)

```

```

   GO TO 50

```

```

C
170 TZ = T

```

```

C
  DO 175 J = 1, N

```

```

    XE(J) = - XE(J)

```

```

    YE(J) = - YE(J)

```

```

175 CONTINUE

```

```

C
  XC(TZ) = XC(1)

```

```

  YC(TZ) = YC(1)

```

```

  XC(TZ+1) = XC(2)

```

```

  YC(TZ+1) = YC(2)

```

```

  T = 0

```

```

  L = 0

```

```

  R = 0

```

```

180 IF(TZ, EQ, T+1) GO TO 250

```

```

  T = T + 1

```

```

  L = T + 1

```

```

  R = L + 1

```

```

210 RT = (YC(L) - YC(T)) * (XC(R) - XC(T)) -

```

```

  * (XC(L) - XC(T)) * (YC(R) - YC(T))

```

```

C
  IF(RT, NE, 0.0) GO TO 190

```

```

      LL = L
      IF (L.EQ.TZ) LL=1
      DO 220 J = LL, TZ
        XC(J) = XC(J+1)
        YC(J) = YC(J+1)
220 CONTINUE
      TZ = TZ - 1
      IF (L.LT.TZ) GO TO 215
      R E T U R N
      END

```

SUBROUTINE INSPCT

• NARJENA: OZNAČAVANJE VPIROVA POLIGONA KOJI IMAJU KONKAVNE RUTEVE

```

      COMMON /WORK/ X(50), Y(50), NN
      • /ROKO/ R(50)
      • /KAV/ KAV
      • /EFI/ E
      X(NN+1) = X(1)
      Y(NN+1) = Y(1)
      KAV = 0
      DO 15 I = 2, NN
        R(I) = (Y(I+1)-Y(I-1)) * (X(I)-X(I-1)) -
        • (X(I+1)-X(I-1)) * (Y(I)-Y(I-1))
15 CONTINUE
      R(1) = (Y(2)-Y(NN)) * (X(1)-X(NN)) -
      • (X(2)-X(NN)) * (Y(1)-Y(NN))

```

••• TEST KONVEKSNOŠTI POLIGONA

```

      DO 18 I = 1, NN
        IF (R(I).LE.0.) GO TO 18
        KAV = KAV + 1
        IF (KAV.EQ.1) I = I
18 CONTINUE
      RETURN
      END

```

SUBROUTINE PART 01

```

C .....
C * NAMJENA: ODREĐIVANJE PARAMETARA KONTAKT FUNKCIJE (NF) I TO
C * ALFA, FI, RO, PSI
C .....
COMMON /VHFF/ XH(90), YH(90), N
* /ALFA/ ALFA(90)
* /FIKU/ FI(90)
* /GKON/ RO(90)
* /PSIK/ PSI(90)
* /MENA/ CH(360), HF(360), FM12(1000), K, N
* /LELA/ CL(20), FL1(20), FL2(20), L
C
IF(DX.NE.0.0) GO TO 10
XF = XH(J)
YF = 0.0
RO(I) = ABS(XH(J))
GO TO 15
10 IF(DY.NE.0.0) GO TO 12
XF = 0.0
YF = YH(J)
RO(I) = ABS(YH(J))
GO TO 15
12 A = DY/DX
B = -1
C = YH(J) - A*XH(J)
RO(I) = ABS(C) / SQRT(A*A + B*B)
XF = -C/(A+1./A)
YF = -XF/A
15 PSI(I) = ATAN2(YH(J+1),XH(J+1))
FI(I) = 0.0
IF(XF.NE.0.0.GR.YF.NE.0.0) FI(I) = ATAN2(YF,XF)
IF(PSI(I).LT.0.0) PSI(I) = PSI(I) + 2.0 * PI
XH(N+1) = YH(1)
YH(N+1) = YH(1)
PI = 3.14159265
PRINT 5
5 FORMAT(//9X,'ALFA',16X,'FI',18X,'RO',17X,'PSI'/)
DO 20 I = 1, N
J = I+1-I
DX = XH(J) - XH(J+1)
DY = YH(J) - YH(J+1)
ALFA(I) = ATAN2(DY,DX)
IF(ALFA(I).LT.0.0) ALFA(I) = ALFA(I) + 2.0 * PI
C
IF(FI(I).LT.0.0) FI(I) = FI(I) + 2.0 * PI
PRINT 11, ALFA(I), FI(I), RO(I), PSI(I)
11 FORMAT(1X,4F18.5)
20 CONTINUE
RETURN
END

```

SUBROUTINE COMPKT

• • • • •
 • NAMJENA: RAČUNANJE KONTAKT FUNKCIJE (KF) JEDNOSTRUKO SUVISLOG
 • KONKAVNOG POLIGONA
 • • • • •

```

COMMON /VHPF/ X(90), Y(90), N
•       /ALFA/ ALFA(90)
•       /NEMA/ ON(360), NF(360), FF12(1800), K, H
•       /LELA/ QL(20), FL1(20), FL2(20), L
•       /FIKU/ FI(90)
•       /OKOM/ RO(90)
•       /PSIK/ PSI(90)
•       /KORAK/ DQ5
•       /IERR/ IERR
  
```

PI = 3.14159265

H = 1

L = 0

PSI(N+1) = PSI(1)

DO 5 II = 1, 360

5 NF(II) = 0

DQ = DQ5 * PI / 180.0

QLDQ = DQ

Q = DQ

DO 8 I = 1, N

IF (PSI(I), NE, PSI(I+1)) GO TO 8

L = L + 1

J = N + 1 - I

FL2(L) = SQRT(X(J)**2 + Y(J)**2)

FL1(L) = SQRT(X(J+1)**2 + Y(J+1)**2)

QL(L) = PSI(I) * 180. / PI

PRINT 20, L, QL(L), FL1(L), FL2(L)

20 FORMAT(5X, 'FORMAT 20', 15, 3F15, 5)

8 CONTINUE

I = 1

10 IF (PSI(I), EQ, PSI(I+1)) GO TO 20

IF (PSI(I), GT, PSI(I+1)) GO TO 150

IF (PSI(I), NE, Q) GO TO 90

70 IF (Q, EQ, QLDQ) GO TO 71

QLDQ = Q

H = H + 1

IF (H, LE, 360) GO TO 71

PRINT 74, H

74 FORMAT (' H = ', 15, ' IBERNS H JE VECI OD 360')

IERR = 1

RETURN

71 QL(H) = Q * 180. / 3.14159265

NF(H) = NF(N) + 1

K = 0

DO 75 II = 1, H

75 K = K + NF(II)

```

C      IF (K .LE. 1000) GO TO 78
      PRINT 77, I
77     FORMAT (' K = ', I6, ' INDEKS K JE VECI OD 1000')
      IERR = 1
      RETURN

C
C 78 CONTINUE
C
      F1N0 = F1(I) - Q
      F12(N) = RO(I) / AFS(COS(F1N0))
80     I = I + 1
      IF (I .LE. N) GO TO 10
      Q = Q + DQ
      I = 1
      IF (Q .LE. 2.0 * PI) GO TO 10
      RETURN
90     IF (PSI(I+1) .EQ. Q) GO TO 70
      IF (PSI(I) .GE. PSI(I+1) - PI) GO TO 130
      IF (PSI(I+1) .LT. Q) GO TO 70
      IF (Q .LT. PSI(I)) GO TO 70
      GO TO 80
130    IF (PSI(I) .GT. Q) GO TO 80
      IF (Q .LT. PSI(I+1)) GO TO 70
      GO TO 80
150    IF (PSI(I) .EQ. Q) GO TO 70
      IF (PSI(I+1) .EQ. Q) GO TO 70
170    IF (PSI(I+1) .GE. PSI(I) - PI) GO TO 200
      IF (Q .GT. PSI(I)) GO TO 70
      IF (Q .LT. PSI(I+1)) GO TO 70
      GO TO 80
200    IF (PSI(I+1) .GE. Q) GO TO 80
      IF (Q .LT. PSI(I)) GO TO 70
      GO TO 80
      END

```

— * —

SUBROUTINE KONTAK

.....
 * NAMJENA RACUNANJE KONTAKT FUNKCIJE (KF)
 *

COMMON /VHKF/ XH(90), YH(90), N
 * /MENA/ QH(360), NF(360), FM12(1000), K, M
 * /LELA/ QL(20), FL1(20), FL2(20), L
 * /KORAK/ DQS

CALL PARKOF

10 PRINT 10
 FORMAT (' POZIV PUTPROGRAMU C O M P K F')

CALL COMPKF

LST = 1
 IF (LST .EQ. 1) RETURN

40 PRINT 40
 40 FORMAT (/5X, 'KONTAKT FUNKCIJA'/5X, I6, 7X,
 * 'QH', 10X, 'NF', 10X, 'FM12(1)'/)

I1 = 0

I2 = 0

DO 41 I = 1, N

PRINT 42, I, QH(I), NF(I)

I1 = I2 + 1

I2 = I2 + NF(I)

PRINT 43, (FM12(I1), II = I1, I2)

41 CONTINUE

42 FORMAT(2X, I6, F10, 1, 4X, I5)

43 FORMAT(1H+, 25X, 5F10, 5, 5(26X, 5F10, 5))

PRINT 45

45 FORMAT(/5X, 'QL', 6X, 'FL1', 6X, 'FL2')

PRINT 46, (QL(I), FL1(I), FL2(I), I=1..L)

46 FORMAT(2X, 3F10, 5)

RETURN

END

—————*—————

SUBROUTINE POTPOF

```

C .....
C .. NAMJENA: RACUNANJE POTPORNE FUNKCIJE POLIGONA
C .....
COMMON /LUSK/ X(50), Y(50), K
      /ALFA/ ALFA(90)
      /OVRH/ OV(50), OA(50)
      /POTP/ H(360), NH
      /EXEP/ XF(90), YF(90), H
      /KORAF/ DG
PRINT 51, DG
C .....
C ... RACUNANJE ELEMANATA POTPORNE FUNKCIJE
C .....
CALL POTPOF
C .....
      IERR = 0
      PI = 3,14159265
      DGR = DG * PI / 180.
      G = DGR
      II = 1
      NP1 = K + 1
      DO 30 JJ = 1, NP1
10      IF(D.GT,6,2831853) GO TO 40
          IF(D.GT,ALFA(JJ)) GO TO 30
          H(II) = OA(JJ) * COS(OV(JJ)-G)
          II = II + 1
          IF(II.EQ,361) GO TO 35
          G = G + DGR
          GO TO 10
30      CONTINUE
35      PRINT 56
C .....
40      NH = II - 1
      PRINT 60, NH, DG, (H(JJ),JJ=1,NH)
      RETURN
50      FORMAT(15,F10.5)
51      FORMAT(5X,'KORAK VARIJABLE G - DG =',F10.5)
56      FORMAT(/5X,'PROVJERITE DA LI JE IZRACUNATA PF NA INTERVALU G-2I
      /5X,'AKO NIJE POVECATI KORAK VARIJABLE G - VARIJ, DG')
60      FORMAT(1H1,5X,'POTPORNA FUNKCIJA U',15,' TACAKA'
      /5X,'SA KORAKOM VARIJABLE G - DG =',F10.5,' STEPENI'
      /5X,'VRIJEDNOSTI POTPORNE FUNKCIJE',
      10(/5F15.5))
LHD

```



SUBROUTINE POTPOF

C * * * * *
 C * * * * * NAMJENA: FORMIRANJE POTPORNE FUNKCIJE POLIGONA STO ODGOVARA
 C * * * * * POTPORNOJ FUNKCIJI NJEGOVE KONVEKSNE LJUSKE
 C * * * * *

COMMON /LUSK/ X(50), Y(50), K
 /ALFA/ ALFA(90)
 COMMON /QVRH/ QV(50), OA(50)
 PI = 3.14159265

C
 C POZIVA SE PODPROGRAM LJUSKA
 C

CALL LJUSKA

PRINT 10, (ALFA(II), II=1,K)
 10 FORMAT(5X, 'KUTEVI STRANICA LJUSKE', 10(/5X, 5F15.5))
 IF(ALFA(1), EQ, 0, 0) ALFA(K+1) = 2, 0 * PI
 IF(ALFA(1), GT, 0, 0) ALFA(K+1) = ALFA(1) + 2, 0 * PI

C
 C
 DO 75 II = 1, K
 OA(II) = SQRT(X(II)**2 + Y(II)**2)
 IF(X(II), NE, 0, 0) GO TO 52
 QV(II) = PI/2, 0
 IF(Y(II), LT, 0, 0) QV(II) = 3, 0 * PI/2, 0
 GO TO 75
 52 QV(II) = ATAN(Y(II)/X(II))
 IF(X(II), GE, 0, 0, AND, Y(II), GE, 0, 0) GO TO 75
 IF(X(II), LT, 0, 0) GO TO 54
 QV(II) = QV(II) + 2, 0 * PI
 GO TO 75
 54 QV(II) = QV(II) + PI
 75 CONTINUE

C
 C
 DO 95 II = 1, K
 QV(II) = QV(II) + PI/2, 0
 IF(QV(II), LE, 2, 0 * PI) GO TO 95
 QV(II) = QV(II) - 2, 0 * PI
 95 CONTINUE
 OA(K+1) = OA(1)
 QV(K+1) = QV(1)
 PRINT 96, (OA(II), II=1,K), (QV(II), II = 1,K)
 96 FORMAT(/5X, 'FORMAT 96' /20(/5F15,5))
 RETURN
 END

— * —

SUBROUTINE LJUSKA

C
 C * NAIJENA: PODPROGRAM ODREĐUJE KONVEKSNU LJUSKU POLIGONA
 C

C
 COMMON /EXEP/ XP(50), YP(50), N
 * /LUSK/ X(50), Y(50), K
 * /ALFA/ ALFA(90)
 INTEGER PSI

C
 C
 PI = 3,14159265
 YMIN = YP(1)
 XMIN = XP(1)
 LL = 1
 DO 10 II = 2, N
 IF(YP(II).GT.YMIN) GO TO 10
 IF(YP(II).LT.YMIN) GO TO 5
 IF(XP(II).GE.XMIN) GO TO 10
 5 YMIN = YP(II)
 XMIN = XP(II)
 LL = II
 10 CONTINUE
 PRINT 11, LL
 11 FORMAT(' LL = ', I5)

C
 C
 DO 20 II = 1, N
 J = LL + II - 1
 IF(J.GT.N) J = J - N
 X(II) = X(J)
 Y(II) = Y(J)
 20 CONTINUE
 PRINT 21, (X(II), Y(II), II=1, N)
 21 FORMAT(/5X, 'KOORDINATE PREDUMERIRANOG POLIGONA', 50(/4F15,5))

C
 K = N
 NPI = N + 1
 X(NPI) = X(1)
 Y(NPI) = Y(1)

C
 DO 25 II = 1, N
 H = X(II+1) - X(II)
 V = Y(II+1) - Y(II)
 IF(H.EQ,0,0) GO TO 22
 ALFA(II) = PI/2,0
 IF(V.LT,0,0) ALFA(II) = 3,0 * PI/2,0

GO TO 25
 22 ALFA(II) = ATAN(V/H)
 IF(H.GE,0,0.AND.V.GE,0,0) GO TO 25
 IF(H.LT,0,0) GO TO 24
 ALFA(II) = ALFA(II) + 2,0 * PI
 GO TO 25
 24 ALFA(II) = ALFA(II) + PI
 25 CONTINUE
 PRINT 26, (ALFA(II), II=1, N)
 26 FORMAT(5X, 'ALFA', 10(/5X, SF15,5))

C

```

      I = 1
30  PSI = 0
40  RAF = ALFA(I+1) - ALFA(I)
      IF(RAF,GT,0,0) GO TO 60
      IF(ABS(RAF),LT,PI) GO TO 70
      GO TO 120
60  IF(ABS(RAF),LT,PI) GO TO 120

70  H = X(I+2) - X(I)
      V = Y(I+2) - Y(I)
      IF(H,NE,0,0) GO TO 71
      ALFA(I) = PI/2,0
      IF(V,LT,0,0) ALFA(I) = 3,0 * PI/2,0
      GO TO 75
71  ALFA(I) = ATAN(V/H)
      IF(H,GE,0,0,AND,V,GE,0,0) GO TO 75
      IF(H,LT,0,0) GO TO 74
      ALFA(I) = ALFA(I) + 2,0 * PI
      GO TO 75
74  ALFA(I) = ALFA(I) + PI
75  PSI = 1
      K = K - 1
      IF1 = I + 1
      PRINT 61, I, ALFA(I), K
61  FORMAT(IX,15,F15,5,15)

      DO 85 II = IF1, K
          X(II) = X(II+1)
          Y(II) = Y(II+1)
          ALFA(II) = ALFA(II+1)
85  CONTINUE
      IF(I,EQ,1) GO TO 30
      I = I-1
      GO TO 40
120  I = I+1
      IF(I,LE,K-1) GO TO 40
      IF(PSI,EQ,0) RETURN

      KNI = K-1
      DO 155 II=1,KNI
          IF(ALFA(II),LT,ALFA(II+1)) GO TO 155
          I = II
          GO TO 70
155  CONTINUE
      RETURN
      END

```

———— * ————

FUNCTION SP(X,Y,N)

```

C .....
C * NAMJENA: IZRACUNAVANJE PLOŠTINE KONVEKSNOG POLIGONA
C .....
C .....
C
  DIMENSION X(50), Y(50)
  SP = 0.0
  X(N+1) = X(1)
  Y(N+1) = Y(1)
C
  DO 20 I = 1, N
    DX = X(I+1) - X(I)
    DY = Y(I+1) - Y(I)
    IF(DX.NE.C.0) GO TO 10
    RD = ABS(X(I))
    GO TO 15
  10 IF(DY.NE.C.0) GO TO 12
    RD = ABS(Y(I))
    GO TO 15
C
  12 A = DY/DX
    B = -1
    C = Y(I) - A * X(I)
    RD = ABS(C) / SQRT(A*A + B*B)
C
C    *** DUZINA STRANICE A(I) DO A(I+1) ***
  15 DUZ = SQRT(DX*DX + DY*DY)
C
    SP = SP + RD*DUZ/2
  20 CONTINUE
  RETURN
  END

```

PROGRAM BAZEL

PROGRAM PRIBAVLJA ULAZNE PODATKE ZA RAD SUBROUTINE KONVEX
POZIVA JE I ISPISUJE DOBIVENE REZULTATE

```

DO 15 I=1,IVEX
  PRINT 56, I
  NV = NVEX(I)
  PRINT 57, (XVEX(I,J),YVEX(I,J),J=1,NV)
15 CONTINUE
STOP
20 PRINT 60, IDIM
40 FORMAT(1H1,'BROJ KONKAVNIH POLIGONA :',I5)
41 FORMAT(//1X,I3,' POLIGON OD',I4,' VRHOVA')
50 FORMAT(I5)
51 FORMAT(4(2F10.5))
55 FORMAT(1H1,4X,'KONVEKSNI POLIGONI')
56 FORMAT(//5X,'KOORDINATE',I5,' POLIGONA')
57 FORMAT(2(2X,'( ',F11.5,' ; ',F11.5,')',3X))
60 FORMAT(5X,'*** GRESKA *** PROGRAM DIMENZIONIRAN ZA POLIGONE DO',
  15,' VRHOVA. POVECATI DIMENZIJE.')
STOP
END

```

SUBROUTINE KONVEX

.....
 * NAJNA: RAZDIJANJE KONKAVNOG POLIGONA NA KONVEKSNE PODPOLIGONE *

* FUNKCIJE I PODPROGRAMI

* INSPCT : ANALIZA KONVEKSNOСТИ POLIGONA
 * DIPOL : DIJELJENJE KONKAVNOG POLIGONA

COMMON /WORK/ X(50), Y(50), NI

* /KAVN/ XKAV(50,50), YKAV(50,50), NKAV(50), IKAV
 * /KVEX/ XVEX(50,50), YVEX(50,50), NVEX(50), IVEX
 * /SECT/ XP(50), YP(50), IP(50), NI, ISEC(50)
 * /DIMZ/ IDIM
 * /ERR/ IERR
 * /REFI/ M
 * /ROKO/ R(50)
 * /KKAV/ KKAV

DIMENSION DIS(50), MIS(2)

IVEX = 0

DO 10 I = 1, IDIM

NVEX(I) = 0

DO 10 N = 1, IDIM

XVEX(I,N) = 0.

YVEX(I,N) = 0.

10 CONTINUE

... KOPIRANJE SADRŽINE VEKTORA XKAV I YKAV SA NAJVIŠIM INDEKSON U
 RADNE VEKTORE X I Y RESPEKTIVNO

18 IF(IKAV.EQ.0) RETURN

NN = NKAV(IKAV)

DO 20 I = 1, NN

X(I) = XKAV(IKAV,I)

Y(I) = YKAV(IKAV,I)

```
20 CONTINUE
   IKAV = IKAV - 1
```

```
C
C ... ANALIZA KONVEKSNOSTI POLIGONA CIJE SU KOORDINATE VRHOVA U
C      VEKTORIMA X I Y.
```

```
CALL INSECT
PRINT 25, (R(II), II=1, NN)
25 FORMAT(5(3X, F10, 2))
IF (KKAV.NE.0) GO TO 50
```

```
C
C
   IVEX = IVEX + 1
   IF (IVEX.LE.IDIM) GO TO 35
   PRINT 33, IDIM
33  FORMAT(5X, '*** GRESKA U KONVEKSNOSTI POLIGONA SA VISE OD', IS,
   *      ' VRHOVA')
   IERR = 1
   RETURN
35  DO 38 I = 1, NN
      XVEX(IVEX, I) = X(I)
      YVEX(IVEX, I) = Y(I)
38  CONTINUE
   NVEX(IVEX) = NN
   GO TO 18
```

```
C
C
50  MOD = NN - 2
   R(N) = 0.0
   NN1 = N - 1
   IF (N.EQ.1) NN1 = NN
   R(NN1) = 0.0
   DO 52 II = 1, MOD
      I = N + II
      IF (I.GT.NN) I = I - NN
      R(I) = (Y(I)-Y(NN1)) * (X(N)-X(NN1)) -
   *      (X(I)-X(NN1)) * (Y(N)-Y(NN1))
52  CONTINUE
   PRINT 25, (R(II), II=1, NN)
```

```
C
C
   DO 55 I=1, NN
55  ISEC(I) = 0
   MOD = MOD - 1
   DO 56 II=1, MOD
      I = N + II
      IF (I.GT.NN) I = I - NN
      IF (R(I).EQ.0.) GO TO 56
      IP1 = I + 1
      IF (IP1.GT.NN) IP1 = IP1 - NN
      IF (R(IP1).EQ.0.) GO TO 56
      ABSUM = ABS(R(I)+R(IP1))
      SUMAB = ABS(R(I)) + ABS(R(IP1))
      IF (ABSUM.EQ.SUMAB) GO TO 56
56  ISEC(I) = I
58  CONTINUE
   PRINT 59, (ISEC(II), II=1, NN)
59  FORMAT(5(3X, I2))
```

```
C
C
   I = 0
   DO 68 II=1, MOD
```

```

I = I+1
IF(I.GT.NN) I=I-NN
IF(ISEC(I).EQ.0) GO TO 68
IS = ISEC(I)
IF(R(IS).EQ.0.) GO TO 65
IP1 = I + 1
IF(IP1.GT.NN) IP1 = IP1-NN
K = K + 1
IF(X(M).NE.X(MM1)) GO TO 60
XP(K) = X(M)

IF(X(IP1).NE.X(I)) GO TO 57
PRINT 61
61 FORMAT(5X,'PRAVCI SE NE SIJEKU VEC MINOILAZE')
IERR = 1
RETURN

57 A2 = (Y(IP1)-Y(I)) / (X(IP1)-X(I))
B2 = Y(IP1) - A2 * X(IP1)
YP(K) = A2*XP(K) + B2
GO TO 66

60 IF(X(I).NE.X(IP1)) GO TO 63
XP(K) = X(IS)
A1 = (Y(I)-Y(MM1)) / (X(I)-X(MM1))
B1 = Y(I) - A1 * X(I)
YP(K) = A1*XP(K) + B1
GO TO 66

63 A1 = (Y(I)-Y(MM1)) / (X(I)-X(MM1))
A2 = (Y(IP1)-Y(I)) / (X(IP1)-X(I))
B1 = Y(I) - A1 * X(I)
B2 = Y(IP1) - A2 * X(IP1)
XP(K) = (B2-B1) / (A1-A2)
YP(K) = A1*XP(K) + B1
GO TO 66

65 K = K + 1
XP(K) = X(I)
YP(K) = Y(I)

66 IP(K) = I
68 CONTINUE
PRINT 59, (IP(II), II=1,K)
PRINT 25, (XP(II), YP(II), II=1,K)

```

```

DO 75 I=1,K
70 DIS1 = SQRT((XP(I)-X(M))**2 + (YP(I)-Y(I))**2)
DIS2 = SQRT((XP(I)-X(MM1))**2 + (YP(I)-Y(MM1))**2)
IF(DIS1.LT.DIS2) GO TO 74
K = K - 1
DO 72 L = I, K
XP(L) = XP(L+1)
YP(L) = YP(L+1)
IP(L) = IP(L+1)
72 CONTINUE
IF(K.GT.I) GO TO 70
GO TO 75
74 DIS(I) = DIS1
75 CONTINUE

PRINT 25, (XP(II), YP(II), II=1,K)

```



```

C
  NI = 1
  IF(K.EQ.1) GO TO 100
  DO 85 I = 1,2
    HIS(I) = 1
    IPI = I+1
    DO 83 L = IPI,K
      IF(DIS(I).LE.DIS(L)) GO TO 83
      DETEM = DIS(I)
      DIS(I) = DIS(L)
      DIS(L) = DETEM
      HIS(I) = L
83  CONTINUE
    IF(K.EQ.2) GO TO 90
85  CONTINUE
90  HI1 = HIS(1)
    HI2 = HIS(2)
    I1 = IP(HI1)
    I2 = IP(HI2)
    PRINT 59, (HIS(I1), I1=1,2)
    IF(R(I1).EQ.0.0.AND.R(I2).EQ.0.0) NI = I2

```

```

C
100 CALL DIPOL
    PRINT 101
101 FORMAT(' VRATIO SE IZ PODPROGRAMA  D I P O L ')

```

```

C
  IF(IERR.EQ.0) GO TO 18
  RETURN
  END

```

———— * ————

SUBROUTINE DIPOL

```

C .....
C .....
C *
C *  NARJENA: DIJELJENJE KONKAVNOG POLIGONA NA DVA PODPOLIGONA
C *  STRANICOM KOJA SPAJA TACKE A(N-1), A(N), A(N1)
C *
C .....
C .....
C *
C *  COMMON /LDFK/ X(50), Y(50), NN
C *  /KAVN/ XKAV(50,50), YKAV(50,50), NKAV(50), IKAV
C *  /SECT/ XP(50), YP(50), IP(50), NI, ISEC(50)
C *  /REFI/ N
C *  /ERR/ IERR
C *  /DIBZ/ IDIM
C
  IERR = 0
  NZ = XP(N1)
  YZ = YP(N1)
  NZ = IP(N1)
C
  PRINT 59, NI
59  FORMAT(3X, 'NI=', I5)

```

```

IKAV = IKAV + 1
IF (IKAV.GT.IDIM) GO TO 220
N1 = NN-NZ+M
IF (NZ.LT.M-1) N1 = M-NZ
XKAV(IKAV,1) = XZ
YKAV(IKAV,1) = YZ
MM = 1
IF (N1.EQ.1) GO TO 105
IF (IP(1).LT.IP(2)) GO TO 105
N1 = N1-1
XKAV(IKAV,1) = X(NZ+1)
YKAV(IKAV,1) = Y(NZ+1)
MM = 0
105 NKAV(IKAV) = N1
DO 110 II = 2,N1
    I = NZ+II-MM
    IF (I.GT.NN) I = I-NN
    XKAV(IKAV,II) = X(I)
    YKAV(IKAV,II) = Y(I)
110 CONTINUE
PRINT 300, (XKAV(IKAV,II),YKAV(IKAV,II),II=1,N1)
300 FORMAT(3(1X,2F10,2))
IKAV = IKAV + 1
IF (IKAV.GT.IDIM) GO TO 220
N2 = NZ-M+2
IF (NZ.LT.M-1) N2 = NN+N2
IF (N1.EQ.2) GO TO 201
IF (XZ.EQ.X(NZ).AND.YZ.EQ.Y(NZ)) N2 = N2 - 1
XKAV(IKAV,N2) = XZ
YKAV(IKAV,N2) = YZ
GO TO 205
201 N2 = N2-1
IF (IP(1).LT.IP(2)) GO TO 203
XKAV(IKAV,N2) = X(NZ)
YKAV(IKAV,N2) = Y(NZ)
GO TO 205
203 N2 = N2 - 1
XKAV(IKAV,N2) = X(NZ-1)
YKAV(IKAV,N2) = Y(NZ-1)
205 NKAV(IKAV) = N2
N3 = N2-1
DO 210 II=1,N3
    I = M-1+II
    IF (I.GT.NN) I = I-NN
    XKAV(IKAV,II) = X(I)
    YKAV(IKAV,II) = Y(I)
210 CONTINUE
PRINT 300, (XKAV(IKAV,II),YKAV(IKAV,II),II=1,N2)
RETURN
C
220 IERR = 1
PRINT 225, IDIM
225 FORMAT(5X,'POLIGON MOZE BITI PODJELJEN I.A',IS,' PODPOLIGONA')
RETURN
END

```

PROGRAM APPROX

C ... PROGRAM POZIVA U IZVRŠAVANJE PODPROGRAME ZA APROKSIMACIJU
C VIŠESTRUKO SUVISLIH KONTURA POLIGONIMA
C

```

INTEGER R
COMMON /RINX/ R
COMMON /UGAO/ XM(10,50), YM(10,50), NVP(10)
• /KVEX/ XVS(10,250), YVS(10,250)
• /NOKO/ NO(10)
• /MEGA/ LOMEGA
• /EPIN/ ED
• /ERIK/ E
• /LINX/ L
READ 15, LOMEGA
DO 5 I=1,LOMEGA
  READ 10, E, ED
  PRINT 11, LOMEGA, E, ED
  READ 15, NO(I)
15  FORMAT(15)
  PRINT 16, NO(I)
  NK = NO(I)
  READ 20, (XVS(I,J),YVS(I,J),J=1,NK)
  PRINT 21, (XVS(I,J),YVS(I,J),J=1,NK)
  L = I

```

```

C
CALL APPROX
C
R = NVP(I)
PRINT 21, (XM(I,J),YM(I,J),J=1,R)
5 CONTINUE
10 FORMAT(5X,2F10,5)
11 FORMAT(11F1,5X,'LOMEGA=',15,2F15,7)
16 FORMAT(5X,'NI=',16)
20 FORMAT(8F10,5)
21 FORMAT(3(2(3X,E13,3)))
STOP
END

```

SUBROUTINE APPROX

C
C * NAMJENA: APROKSIMACIJA PROIZVOLJNIH RAVNIH LIKOVA POLIGONIMA

```

INTEGER R
LOGICAL LOG, ERROR

C
COMMON /EPIN/ ED /ERIK/ E
COMMON /UGAO/ XM(10,50), YM(10,50), NVP(10)
• /KVEX/ XVS(10,250), YVS(10,250)
• /NORD/ X(250),Y(250)
• /NOKO/ NO(10)
• /NOST/ N(10)
• /MEGA/ LOMEGA
• /JINX/ J
• /LINX/ L
• /RINX/ R
• /FINX/ F
• /INIX/ I
• /FOEF/ A(3), L(3)
• /XVFP/ XV(3)
• /LOCC/ LOG(3)
• /LEER/ ERROR
• /ECI/ LI, I2
• /NODR/ XFT, YFT
• /LAST/ XF, YF
• /ILFA/ IF

```

```

I = 1
J = 1
R = 0
ND1 = ND(L)
DO 2 II=1,ND1
  X(II) = XVS(L,II)
  Y(II) = YVS(L,II)
2 CONTINUE
X(ND1 + 1) = X(1)
Y(ND1 + 1) = Y(1)
DO 5 II = 1, 3
  LOG(II) = .FALSE.
  A(II) = 0,0
  B(II) = 0,0
  XV(II) = 0,0
5 CONTINUE
DX = X(I+1) - X(I)

... TEST ZA IZBOR USLOVA ZA SEPARACIJU

IF(ABS(DX),GE,ED) GO TO 120
R = 1
50 I = I + 1
IF(I,GT,ND(L)) GO TO 80
DX = X(I+1) - X(I)
IF(ABS(DX),LT,ED) GO TO 50
80 N(J) = I
IP = 1
IF(J,GT,1) IP=N(J-1)
IK = N(J)
XMIN = X(IP)

-----
XMAX = X(IP)
IPP = IP + 1

-----
DO 95 II = IPP,IK
  IF(X(II),GT,XMAX) XMAX = X(II)
  IF(X(II),LT,XMIN) XMIN = X(II)
95 CONTINUE
XS = (XMIN + XMAX) / 2.
PRINT 96, R,XS
96 FORMAT(5X,'FORMAT 96',5X,15,5X,F10,5)

-----
XK = X(IK)
YK = Y(IK)

-----
IF(R,NE,1) GO TO 110
XV(1) = XS
XV(2) = XS
LOG(1) = .TRUE.
LOG(2) = .TRUE.
GO TO 118

110 L1 = 2
L2 = 3
XV(3) = XS
LOG(3) = .TRUE.

```

```

CALL P R E S T A
IF ( ERROR ) R E T U R N
C
A(2) = A(3)
B(2) = B(3)
A(3) = 0.0
b(3) = 0.0
XN (L, R) = XPT
YM (L, R) = YPT
XV (2) = XV (3)
LOG(2) = LOG(3)
PRINT 115, L, R, XN(L,R), YM(L,R)
115 FORMAT( ' F115:', 2I6, 2F15.5)
118 J = J + 1
120 IF(I,GT,NO(L)) RETURN
DX = X(I+1) - X(I)
IF(DX,LE,E) GO TO 220
150 I = I + 1
IF(I,GT,NO(L)) GO TO 180
DX = X(I+1) - X(I)
IF(DX,GT,E) GO TO 150
180 N(J) = I
PRINT 181, J, N(J)
181 FORMAT(5X, 'FORMAT181', 2(5X, 15))
CALL A P L I N E
IF(ERROR) RETURN
J = J + 1
IF(I,GT,NO(L)) GO TO 300
DX = X(I+1) - X(I)
220 IF(ABS(DX),GE,EO) GO TO 230
R = R + 1
GO TO 50
230 I = I + 1
IF(I,GT,NO(L)) GO TO 260
DX = X(I+1) - X(I)
IF(DX,LT,-E) GO TO 230
260 N(J) = I
PRINT 182, J, N(J)
182 FORMAT(5X, 'FORMAT182', 2(5X, 15))
CALL A P L I N E
IF(ERROR) RETURN
J = J + 1
IF(I,GT,NO(L)) GO TO 300
DX = X(I+1) - X(I)
IF(DX,LE,E) GO TO 265
I = I + 1
GO TO 120
265 R = R + 1
GO TO 50
300 L1 = 1
L2 = 2
CALL P R E S T A
XN(L,1) = XPT
YM(L,1) = YPT
PRINT 301, L, XN(L,1), YM(L,1)
301 FORMAT(5X, 'FORMAT 301', 5X, 15, 2F15.5)
WVP(L) = R
R E T U R N
END

```

SUBROUTINE PRESTA

• NAMJENA : ODREĐIVANJE MEDJUSOBNOG POLOŽAJA DVA PRAVCA KOJA SU
 • PRILIKOM POZIVA POKAZANA

```

LOGICAL LOG, ERROR
COMMON /KDEF/ A(3), B(3)
•      /KOOR/ XPT, YPT
•      /XVER/ XV(3)
•      /LOGC/ LOG(3)
•      /LEPR/ ERROR
•      /PECI/ I, J

ERROR = .FALSE.
PRINT 3, I, J, A(I), B(I), LOG(I), A(J), B(J), LOG(J)
3  FORMAT(15X, 'P R E S T A' / 5X, 2I10, 3(2F10, 5, L3))
IF (LOG(I)) GO TO 20
IF (LOG(J)) GO TO 10
IF (A(I).EQ.A(J)) GO TO 5
XPT = (B(J)-B(I)) / (A(I)-A(J))
YPT = A(I) * XPT + B(I)
RETURN

5  ERROR = .TRUE.
IF (B(I).EQ.B(J)) GO TO 7
PRINT 60
RETURN

7  PRINT 61
RETURN

10 XPT = XV(J)
YPT = A(I) * XPT + B(I)
RETURN

20 IF (LOG(J)) GO TO 30
XPT = XV(I)
YPT = A(J) * XPT + B(J)
RETURN

30 ERROR = .TRUE.
IF (XV(I).EQ.XV(J)) GO TO 35
PRINT 60
RETURN

35 PRINT 61
60 FORMAT(5X, 'PRAVCI SE MINOILAZE')
61 FORMAT(5X, 'PRAVCI SE POKLAPAJU')
RETURN
END

```

————— * —————

SUBROUTINE APLINE

```

C .....
C
C *   NAHJENA: APROKSIMACIJA IZDVOJENOG DIJELA ZATVORENE KONTURE
C *   PRAVCINA KORISTENJEM METODE NAJMANJIH KVADRATA.
C .....
C
C
C   DIMENSION MR(3,50) , FR(50) , FNI(50)
C   R E A L   NASUM
C   INTEGER R
C   LOGICAL LOG, ERROR
C
C   COMMON /ERIK/ E
C   COMMON /UGAD/ XM(10,50), YM(10,50), NVP(10)
C   *   /KORD/ X(250), Y(250)
C   *   /NOST/ N(10)
C   *   /JINX/ J
C   *   /LINX/ L
C   *   /RINX/ R
C   *   /INIX/ I
C   *   /KOEK/ A(3), B(3)
C   *   /XVER/ XV(3)
C   *   /LOGC/ LOG(3)
C   *   /PECI/ L1, L2
C   *   /LERR/ ERROR
C   *   /KORR/ XPT, YPT
C   *   /LAST/ XM, YM
C   *   /INXA/ IF
C
C
C   NBR = N(J)
C   IXJ = 1
C   IF (J.EQ.1) GO TO B
C   IXJ = N(J-1)
C   NBR = N(J)-N(J-1) + 1
C   R = R + 1
C   IR = R
C
C   B IF(R.GT.1) GO TO 10
C   XK = X(1)
C   YK = Y(1)
C   IK = 0
C
C   10 C O N T I N U E
C
C   K = R - IR
C   ND ( 1, R ) = 0
C   ND ( 2, R ) = 1
C   ND ( 3, R ) = 1

```

```

12 DO 14 II = 1, 2
    MR(II,R) = MR(II+1,R)
14 CONTINUE
MR(3,R) = MR(2,R) + MR(1,R)
IK = IK + 1
NAS = 0
DO 15 II = IR, R
    NAS = NAS + MR(3, II)
15 CONTINUE
ITEST = NBR - NAS + K
IF(ITEST,GE,0) GO TO 40
MR(3,R) = N(J) - IXJ + 1
40 CONTINUE
PRINT 39, ITEST, NBR, R, MR(3,R)
39 FORMAT(5X, 'ITEST =', 15, ' NBR =', 15, ' R =', 15, ' MR(3,R) =', 15)
BRSUM = 0
NASUM = 0

IKR = MR(3,R) - 1
DO 50 II = 1, IKR
    IX = IXJ + II
    BRSUM = BRSUM + (X(IX)-XK) * (Y(IX)-YK)
    NASUM = NASUM + (X(IX)-XK) ** 2
50 CONTINUE
AR = BRSUM/NASUM
BR = YK - AR * XK
DO 60 II = 1, IKR
    IX = II + IXJ
    FR(II) = BR + AR * X(IX)
    FNI(II) = ABS(FR(II)-Y(IX))
60 CONTINUE
PRINT 61, AR, BR, (FR(II),FNI(II),II=1,IKR)
61 FORMAT(/5X,2(F10.5,2X),8(/2X,6(F10.5,2X)))
FNI MAX = FNI(1)
DO 65 II=1,IKR
    IF(FNI(II).GT.FNI MAX) FNI MAX = FNI(II)
65 CONTINUE
IF(FNI MAX,LE,E) GO TO 70
IF(MR(3,R),EQ,2) GO TO 100
NAS = NAS - MR(3,R) + MR(2,R)
MR(3,R) = MR(2,R)
IF(R,EQ,1) GO TO 60
L1 = 2
L2 = 3
CALL P R E S T A
IF(ERROR) RETURN
A(2) = A(3)
B(2) = B(3)
LOG(2) = LOG(3)
XP(L,R) = XPT
YP(L,R) = YPT

```



```

IXJ = N(J-1) + MAS - R + IR - 1
XK = X(IXJ)
YK = Y(IXJ)
PRINT 67, L, R, IXJ, XM(L,R), YM(L,R), X(IXJ), Y(IXJ)
67 FORMAT(5X, 'FORMAT 67', 3I10/14X, 4F10, 5)
-----
68 R = R + 1
GO TO 10
70 A(3) = AR
B(3) = BR
LOG(3) = .FALSE.
IF(R.GT.1) GO TO 75
DO 73 II = 1,2
  A(II) = A(3)
  B(II) = B(3)
  LOG(II) = LOG(3)
73 CONTINUE
75 IF(ITEST.GT.0) GO TO 12
L1 = 2
L2 = 3
CALL P R E S T A
IF(ERROR) RETURN
A(2) = A(3)
B(2) = B(3)
LOG(2) = LOG(3)
XM(L,R) = XPT
YM(L,R) = YPT
IXJ = N(J)
XK = X(IXJ)
YK = Y(IXJ)
PRINT 77, L, R, IXJ, XM(L,R), YM(L,R), X(IXJ), Y(IXJ)
77 FORMAT(5X, 'FORMAT 77', 3I10/14X, 4F10, 5)
RETURN
100 PRINT 101
101 FORMAT(3X, 'METODA NAJMANJIH KVADRATA NE FUNKCIONIRA')
ERROR = .TRUE.
RETURN
END

```

————— * —————

G L A V A I I I

JEDNOREDNI PERIODIČNI RASPOREDI

1. Formulacija problema i smještanje u beskonačnu traku

Neka je zadan skup kongruentnih objekata $S \in \{S\}$ proizvoljna oblika koji su smješteni jednoredno i periodično, što znači da za parametre objekata smještenih u redosljedu:

S_{r-1}, S_r, S_{r+1} vrijedi:

$$x_{r+1} = x_r + T_x; \quad y_{r+1} = y_r + T_y; \quad \theta_{r+1} = \theta_r + T_\theta. \quad (3.1.1)$$

pored toga parametri su međusobno vezani nekim izrazom

$$\phi(x_r, y_r, \theta_r) = 0. \quad (3.1.2)$$

Takav raspored objekata se naziva JEDNOREDNI PERIODIČNI RASPORED. Ovakve probleme vrlo često susrećemo u metalnoj industriji gdje su poznati pod imenom jednoredno štampanje [17] [18].

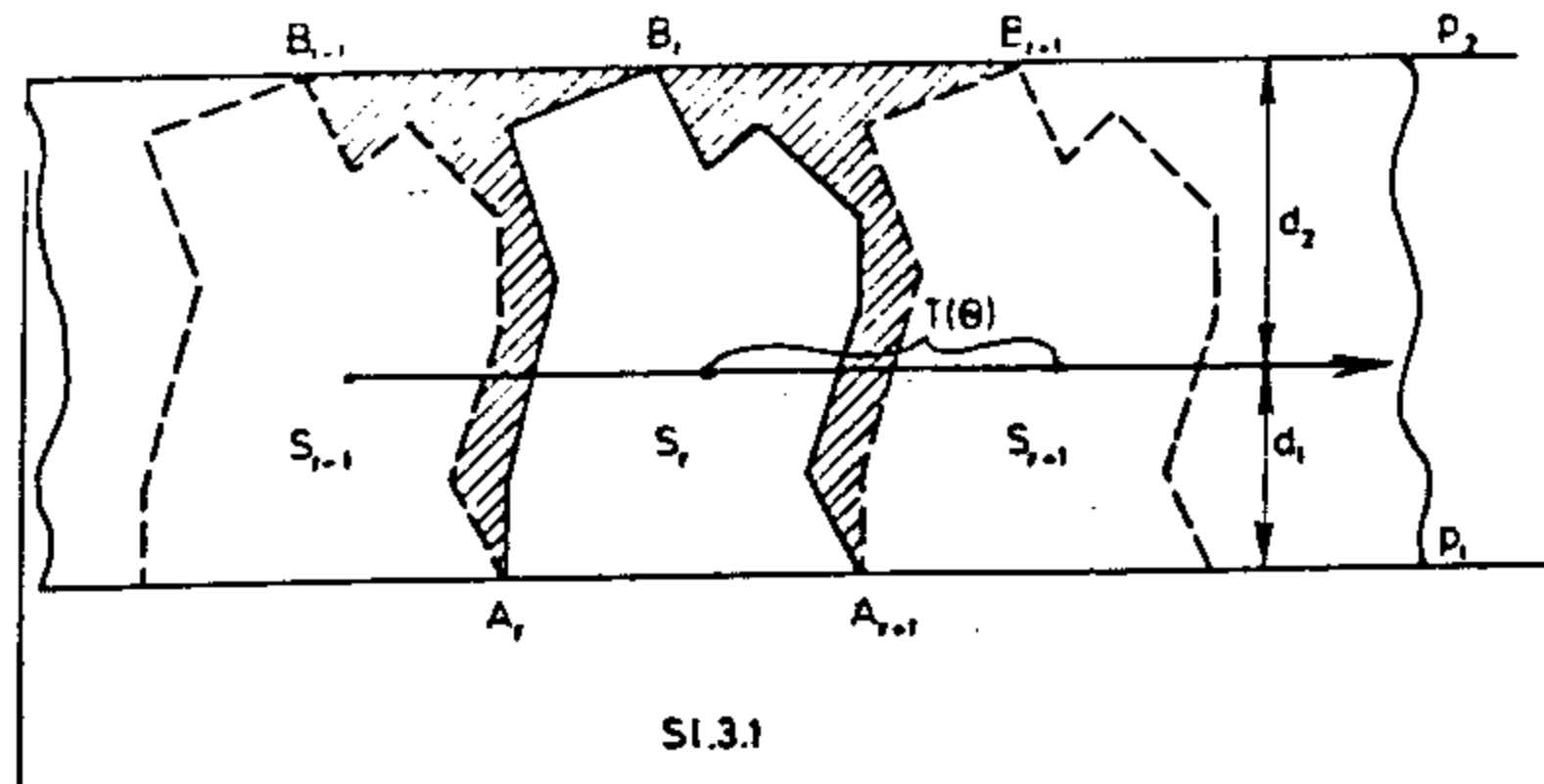
Funkcija $\phi(x, y, \theta) = 0$ može općenito biti proizvoljna. Međutim, u praksi se gotovo isključivo javljaju problemi gdje je $\theta = \text{const.}$ tj.

$$\phi(x, y, \theta) = \phi(x, y) = ax + by + c = 0.$$

Smještaj objekata u pojas beskonačne duljine

Pojam "beskonačna duljina" očito treba uvjetno shvatiti. Praktično se uvijek radi o konačnoj duljini. Međutim duljina trake je toliko velika da je otpad koji se javlja na početku i na kraju zanemarivo malen prema ostalom otpadu.

Neka su zadani $S_i \in \{S\}$ proizvoljna oblika. Zadan je koordinatni siste XOY i $\phi(x, y, \theta) = y = 0$. Za zadani $\theta = \theta_1$ objekti se smještaju tako da je $S_r \cap S_s = \emptyset$, za $\forall_{r, s} \in \mathbb{N}; \quad r \neq s$. Sl.3.1.



Budući da su polovi objekata na istom pravcu i da je $\theta = \theta_1 =$ konstanta, to svim objektima $S_r \in \{S\}$ odgovaraju isti potporni pravci.

p_1 - donji i p_2 - gornji, koji su paralelni sa $y = 0$. Zahtijevamo da bude $S_r \setminus s_r \cap S_j \setminus s_j = \emptyset$ za $\forall r, j \in \mathbb{N}; r \neq j$. tj. objektivna su zajedničke samo rubne točke. Vidimo da je period T_x definiran zapravo pomoću KF tj.

$$T_x = K(\theta_1) \quad (3.1.3)$$

a iz definicije potporne funkcije imamo:

$$d_1 + d_2 = p_1(\theta_1) + p_2(\theta + \pi). \quad (3.1.4)$$

Problem se sastoji u tome da nađemo takav θ , zapravo širinu beskonačnog pojasa, u koji su smješteni jednoreдно i periodično $S \in \{S\}$, da koeficijent iskorištenja bude maksimalan. Linijom B_r, A_r odjeljen je objekt S_r od S_{r-1} , a linijom B_{r+1}, A_{r+1} objekt S_r od S_{r+1} . U kvazi paralelogram [16] [17] $V_r = A_r, B_r, B_{r+1}, A_{r+1}$ površine $P(V_r)$ smješten je samo jedan objekt S_r površine $P(S_r)$. Kvazi paralelogrami V_r bez rupa i preklapanja pokrivaju pojas L , a $P(S_r)$ je konstanta, pa je koeficijent iskorištenja definiran:

$$\mu = \frac{P(S_r)}{P(V_r)} \quad (3.1.5)$$

Na osnovu definicije KF i PF te periodičnosti smještaja objekta, u jednom redu, imamo:

$$P[V(\theta)] = [p(\theta) + p(\theta + \pi)] \cdot K(\theta) \quad (3.3.6)$$

gdje: $p(\theta)$ - PF, a $K(\theta)$ - KF objekata S_r i S_{r+1} .

Prema tome problem je sveden na određivanje

$$\min_{\theta \in [0, \Pi]} P[V(\theta)] = \min_{\theta \in [0, \Pi]} [p(\theta) + p(\theta + \Pi)] \cdot K(\theta) \quad (3.1.7)$$

Općenito za taj problem je karakteristično:

- s obzirom na svojstva KF i PF, $P[V(\theta)] : \theta \in [0, \Pi]$ je općenito mnogoekstremalna,
- $P[V(\theta)] \in C_{[0, \Pi]}$
- ako je $P[V(\theta)] \in C_{[0, \Pi]}$ a objekti konveksni, $P[V(\theta)]$ je unimodalna jer $p(\theta) + p(\theta + \Pi)$ i $K(\theta)$ postižu ekstreme za iste vrijednosti θ .

Prema tome ako je $P[V(\theta)] \in C_{[0, \Pi]}^{(1)}$, onda se problem svodi na to da nadjemo sve korjene jednačbe,

$$p'(\theta) + p'(\theta + \Pi) \cdot K(\theta) + [p(\theta) + p(\theta + \Pi)] \cdot K'(\theta) = 0 \quad (3.1.8)$$

a zatim izračunamo vrijednosti $P[V(\theta)]$ i nadjemo one θ_i za koje se postiže $\min_{\theta_i} P[V(\theta_i)]$.

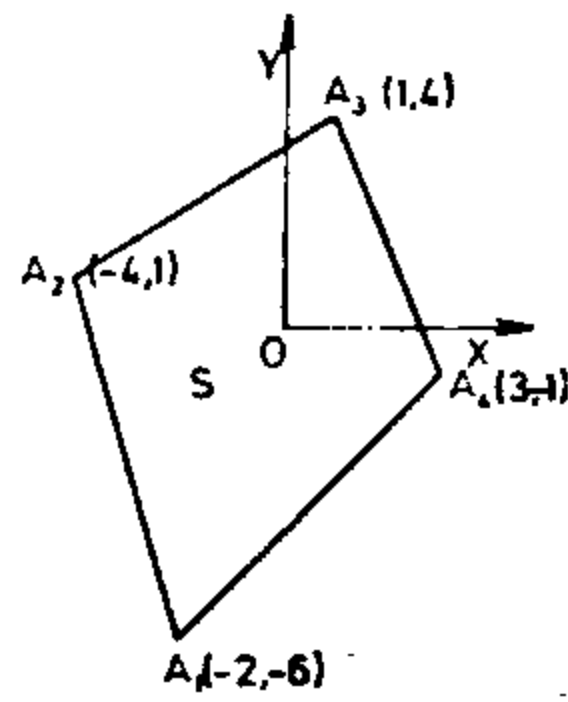
Ako je $P[V(\theta)] \in C_{[0, \Pi]}$ onda se nalaze svi korjeni (3.1.8) osim u točkama gdje $P[V(\theta)]'$ ne postoji. U tim točkama se računa $P[V(\theta)]$, i nadje onaj θ_i - za koji se postiže $\min P[V(\theta)]$. S obzirom na činjenicu da je uvijek $K(\theta) > 0$ i $p(\theta) > 0$, slijedi da $P[V(\theta)]$ može postizati ekstreme na intervalima gdje imamo jedan od slijedeća tri slučaja:

$$\begin{array}{ll} K'(\theta) = 0 & p'(\theta) + p'(\theta + \Pi) = 0 \\ K'(\theta) < 0 & p'(\theta) + p'(\theta + \Pi) > 0 \\ K'(\theta) > 0 & p'(\theta) + p'(\theta + \Pi) < 0. \end{array} \quad (3.1.9)$$

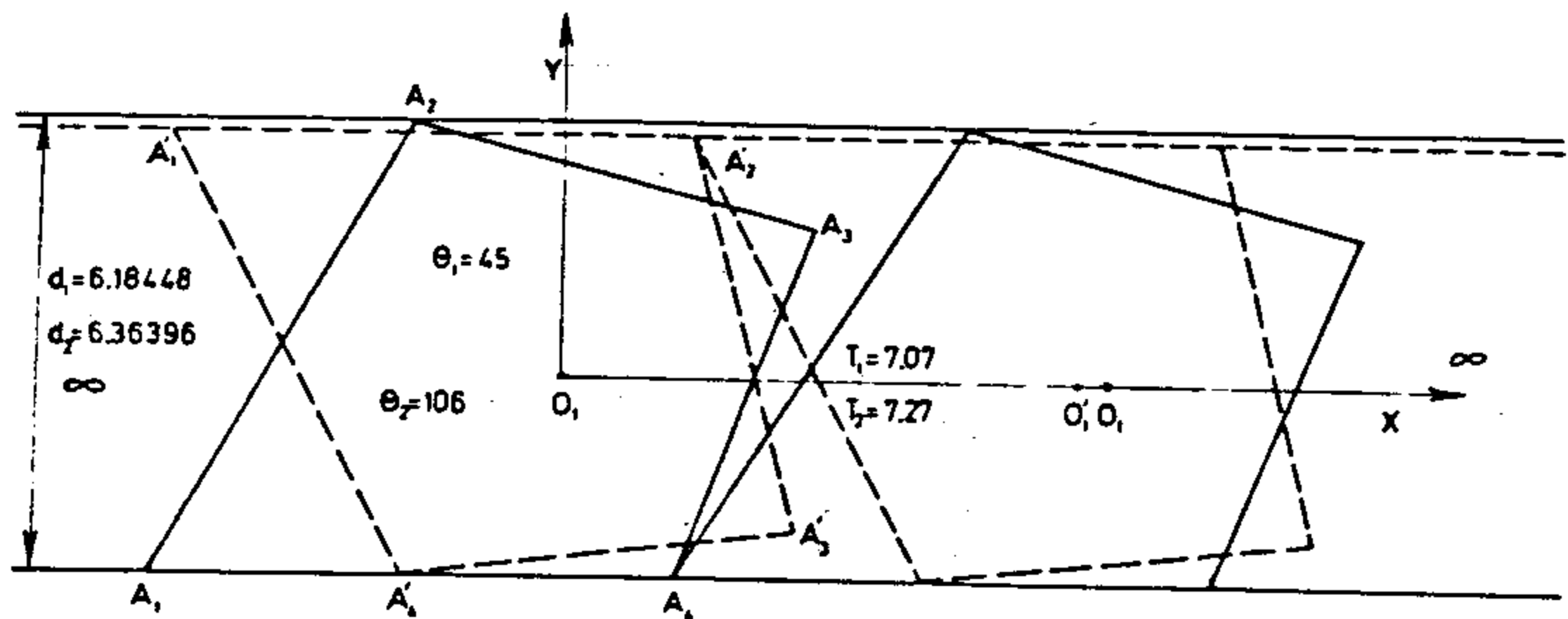
Na osnovi toga zaključujemo da se kod smještaja poligona problema svodi na traženje $P[V(\theta)]$ u vrhovima gdje derivacija ne postoji (u glatkim dijelovima funkcija $P[V(\theta)]'$ je monotona).

Prednje vrijednosti se jednostavno dobiju kada prethodno imamo izračunate vrijednosti kontakt i potpune funkcije.

Za poligon na sl.3.2. prikazan je optimalni raspored na sl.3.3.



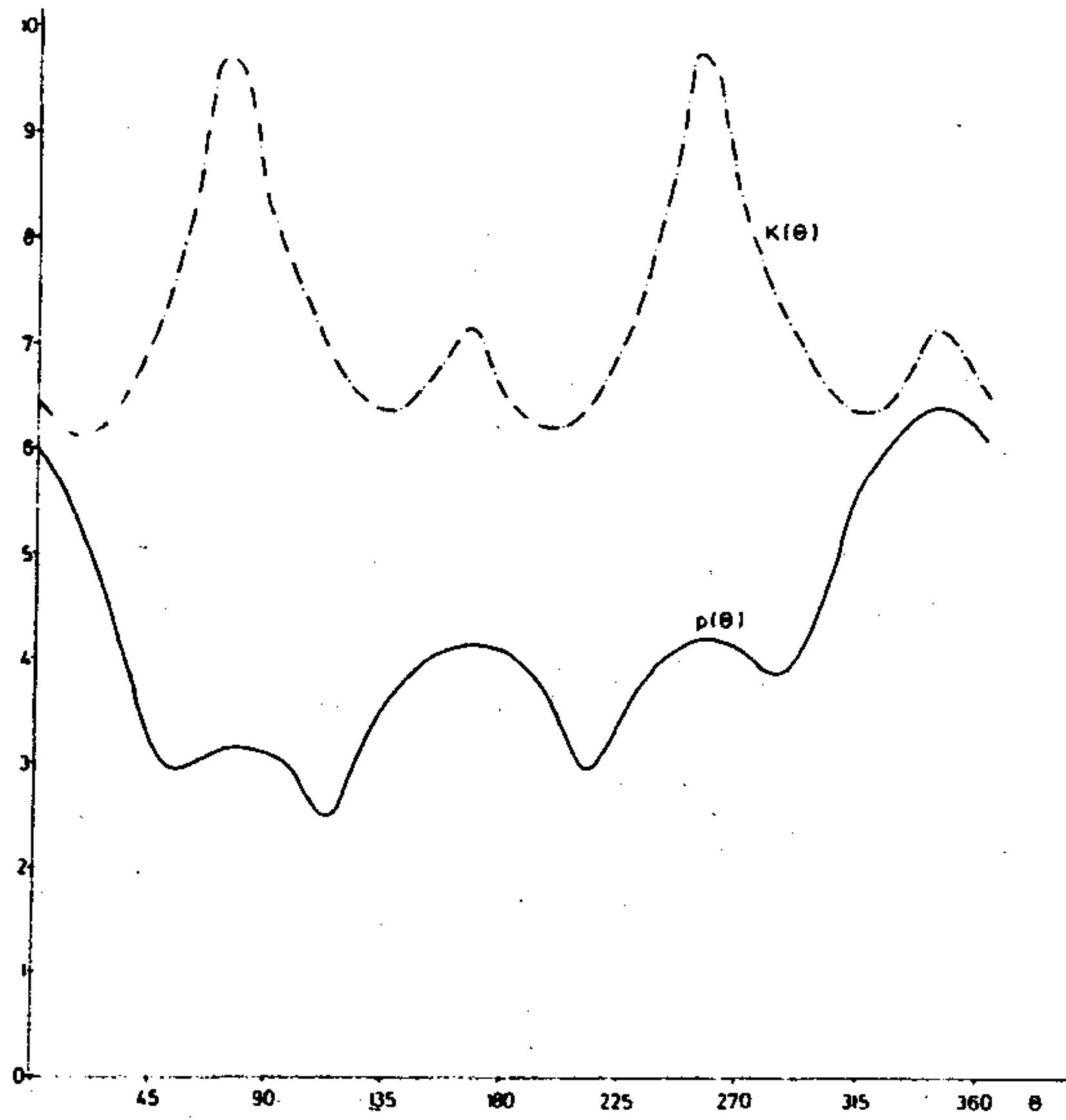
SI.3.2



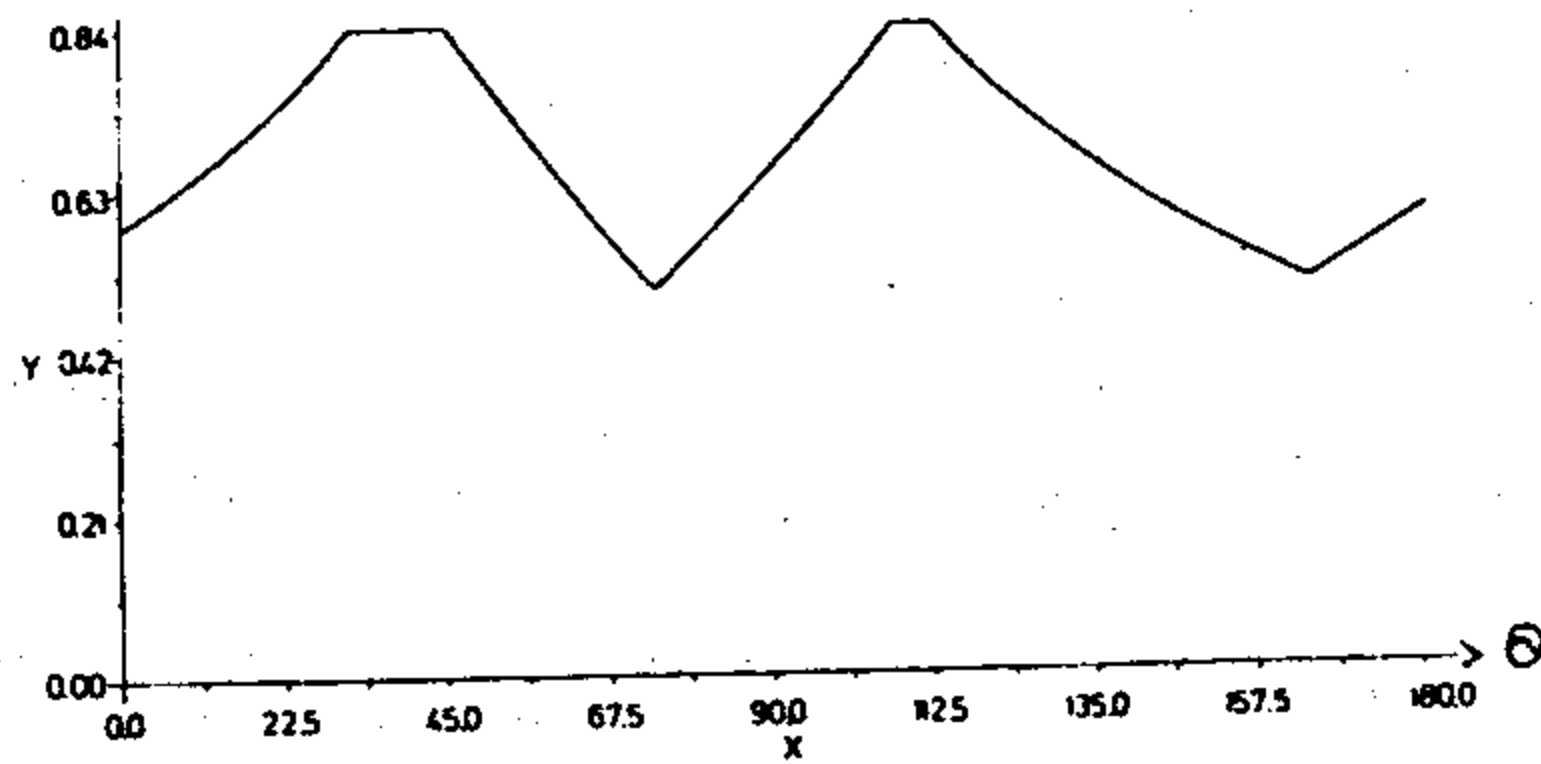
SI.3.3

Maksimalno iskorištavanje se postiže za $31^\circ < \theta < 45^\circ$ i $\theta_2 = 106^\circ$
 $\nu(\theta_1) = \nu(\theta_2) = 0,84445$. Širina trake $d_1(\theta_1) = 6,36396$ a
 $d_2(\theta_2) = 6.18448$. Period $T_1 = 7.07$ i $T_2 = 7.27$. Na sl.3.4.
 vidimo grafove kontakt i potporne funkcije. Na sl.3.5. vidimo
 kako se mijenja koeficijent iskorištenja u zavisnosti od prom-
 jene kuta θ , tj. vidimo da se koeficijent iskorištenja kreće
 u granicama od 0,50499 do 0,84445.

Rezultati su dobiveni korištenjem paketa programa JERED.



SI.34

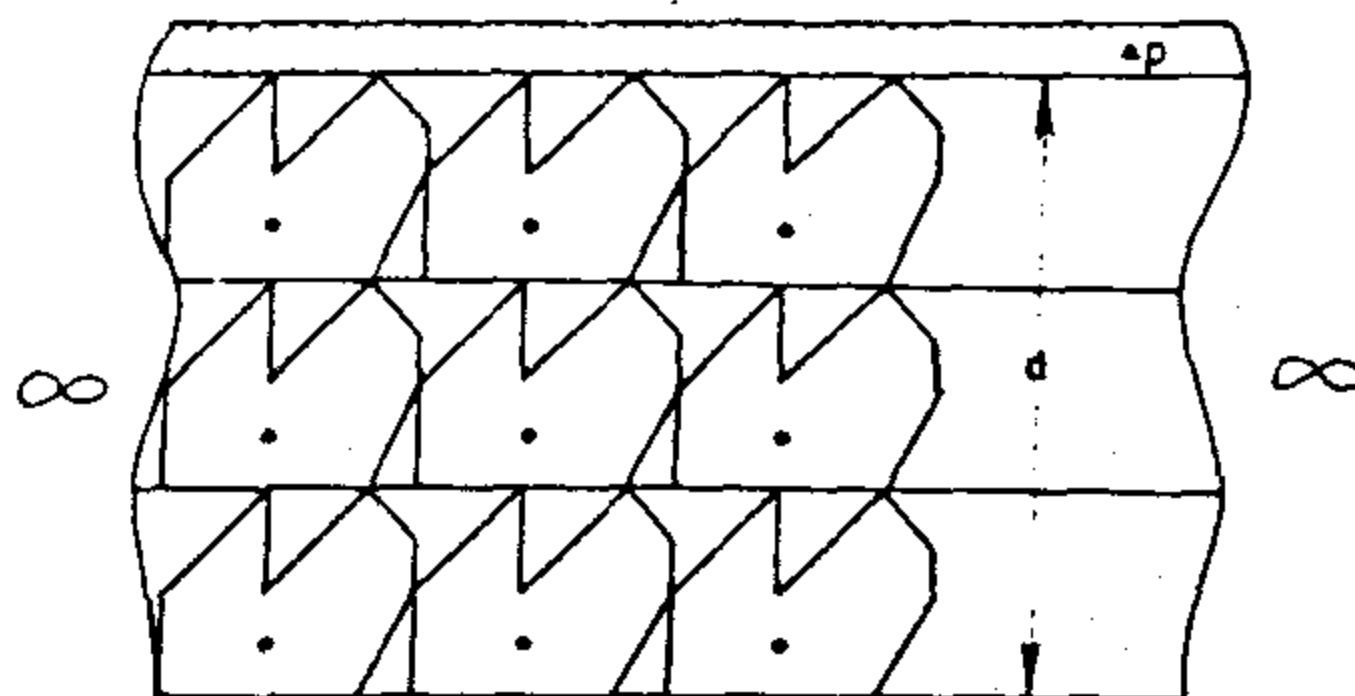


SI.35

2. Smještaj objekata u traku zadane širine

Često se javlja slučaj da industrija proizvodi "beskonačne" trake standardne širine. Korisnici, s druge strane, raspolažu takvom tehnologijom koja omogućava samo jednoređno štampanje. Prema tome, zavisno od veličine lika koji se kroji, imamo situaciju da se u standardnu tvorničku traku može smjestiti više pojaseva u svakom od kojih se nalaze jednoređno smješteni kongruentni objekti $S \in \{S\}$.

Recimo da je zadana beskonačna traka T fiksne širine $d > \max[p(\theta) + p(\theta + \pi)]$. Ako traku T prekrijemo pojasevima T_i i to bez rupa i preklapanja, onda kao pravilo jedan dio trake Δp ostane nepokriven. Sl.3.6.



Sl.3.6

Problem

Treba naći onu vrijednost θ , da koeficijent iskorištenja bude maksimalan kada traku T prekrijemo pojasevima T_i , u kojima su smješteni jednoređno i periodično kongruentni objekti $S \in \{S\}$. Očito da je $\Delta p < p(\theta) + p(\theta + \pi)$. Sada zbog Δp , nemožemo površinu kvazi paralelograma koristiti za ocjenu koeficijenta iskorištenja. Zato promatramo paralelogram R koji se sastoji od n -paralelograma i dijela $K(\theta) \cdot \Delta p$, tj.

$$P(R) = nP(V) + K(\theta) \cdot \Delta p$$

Koeficijent iskorištenja sada se definira u obliku

$$\mu(\theta) = \frac{nP(S)}{P(R)} = \frac{n \cdot P(S)}{nP(V) + K(\theta) \Delta p} \quad (3.2.1)$$

odnosno

$$\mu(\theta) = \frac{nP(S)}{K(\theta)\{n[p(\theta)+p(\theta+\pi)+\Delta p]\}}. \quad (3.2.2)$$

No $P(S)$ - je konstanta, $n[(p(\theta)+p(\theta+\pi)+\Delta p)] = d = \text{const.}$
pa ostaje da je jedino promjenljiv broj objekata koji očitito zavisi od kuta θ . On se može izraziti pomoću izraza:

$$n(\theta) = N\left[\frac{d}{p(\theta)+p(\theta+\pi)}\right]. \quad (3.2.3)$$

$N[\]$ - sada i ubuduće označava cijeli dio izraza u zagradama.
Prema tome problem je sveden na određivanje

$$\mu^*(\theta) = \max_{\theta \in [0, \pi]} \mu(\theta) = \frac{P(S)}{d} \max_{\theta \in [0, \pi]} \left\{ N\left[\frac{d}{p(\theta)+p(\theta+\pi)}\right] : K(\theta) \right\} \quad (3.2.4)$$

Vidimo da su svojstva prednjeg problema sadržana u svojstvima potporne i kontakt funkcije.

- $\mu^*(\theta)$ - je općenito mnogoznačna.
- $\mu^*(\theta) \in C_{[0, \pi]}$.
- Ako je $\mu^*(\theta) \in C_{[0, \pi]}$ a objekt konveksan, onda je $\mu^*(\theta)$ unimodalna, jer $p(\theta)+p(\theta+\pi)$ i $K(\theta)$ postižu ekstreme za iste vrijednosti θ .
- $\mu^*(\theta)$ je na $[0, \pi]$ po dijelovima neprekidna, tj. u konačnom broju točka ima prekide I reda.
- Broj prekida je potpuno određen potpornom funkcijom $p(\theta): \theta \in [0, \pi]$.
- Ako je $p(\theta)+p(\theta+\pi)$ unimodalna, onda $\mu^*(\theta)$ ima samo jedan prekid za $\theta \in [0, \pi]$.

Vidimo da za rješenje prednjeg problema moramo naći one θ za koje po dijelovima konstantna funkcija

$n(\theta) = N\left[\frac{d}{p(\theta)+p(\theta+\pi)}\right]$ ima prekide. To se svodi na to da na $[0, \pi]$ nadjemo sve korjene jednadžbe

$$d - 2[p(\theta)+p(\theta+\pi)] = 0 \quad i = z_1, z_1+1, \dots, z_2 \quad (3.2.5)$$

gdje

$$z_1 = N \left[\frac{d}{\max_{\theta \in [0, \pi]} [p(\theta) + p(\theta + \pi)]} \right]; \quad z_2 = N \left[\frac{d}{\min_{\theta \in [0, \pi]} [p(\theta) + p(\theta + \pi)]} \right],$$

jer su upravo to točke prekida funkcije $p(\theta)$.

Analitički ili numerički odredimo θ_i , u kojima se postiže $\min_{\theta \in [\theta_i, \theta_{i+1}]} K(\theta)$, $i = (1, z-1)$.

Svakako je

$$\max_{\mu}(\theta) = \frac{P(S)}{d} \max \left\{ N \left[\frac{d}{p(\theta_i^*) + p(\theta_i^* + \pi)} \right] : K(\theta_i^*) \right\} \quad (3.2.6)$$

Da ne bi kod numeričkog računanja bio propušten ni jedan prekid a da s druge strane ne bi trošili više vremena nego je potrebno treba odabrati adekvatan korak $\Delta\theta$. Pokazuje se da ukoliko je

$$\Delta\theta < \frac{1}{d} \frac{\{\min_{\theta \in [0, \pi]} [p(\theta) + p(\theta + \pi)]\}}{\max_{\theta \in [0, \pi]} p'(\theta)} \quad (3.2.7)$$

Izborom $\Delta\theta^* < \Delta\theta$ neće biti propušten nijedan prekid.

Napomena:

Može se desiti da za $i=1$ jednačba (3.2.5) nema rješenja a to geometrijski znači da se objekt koga smještamo u pojas može smjestiti kod proizvoljne orijentacije.

Ovdje je posebno interesantno da recimo kod prednjeg smještaja trokuta ne možemo postići iskorištenje veće od 50%, a optimalna rješenja se postižu kod raznih kuteva zarotiranosti.

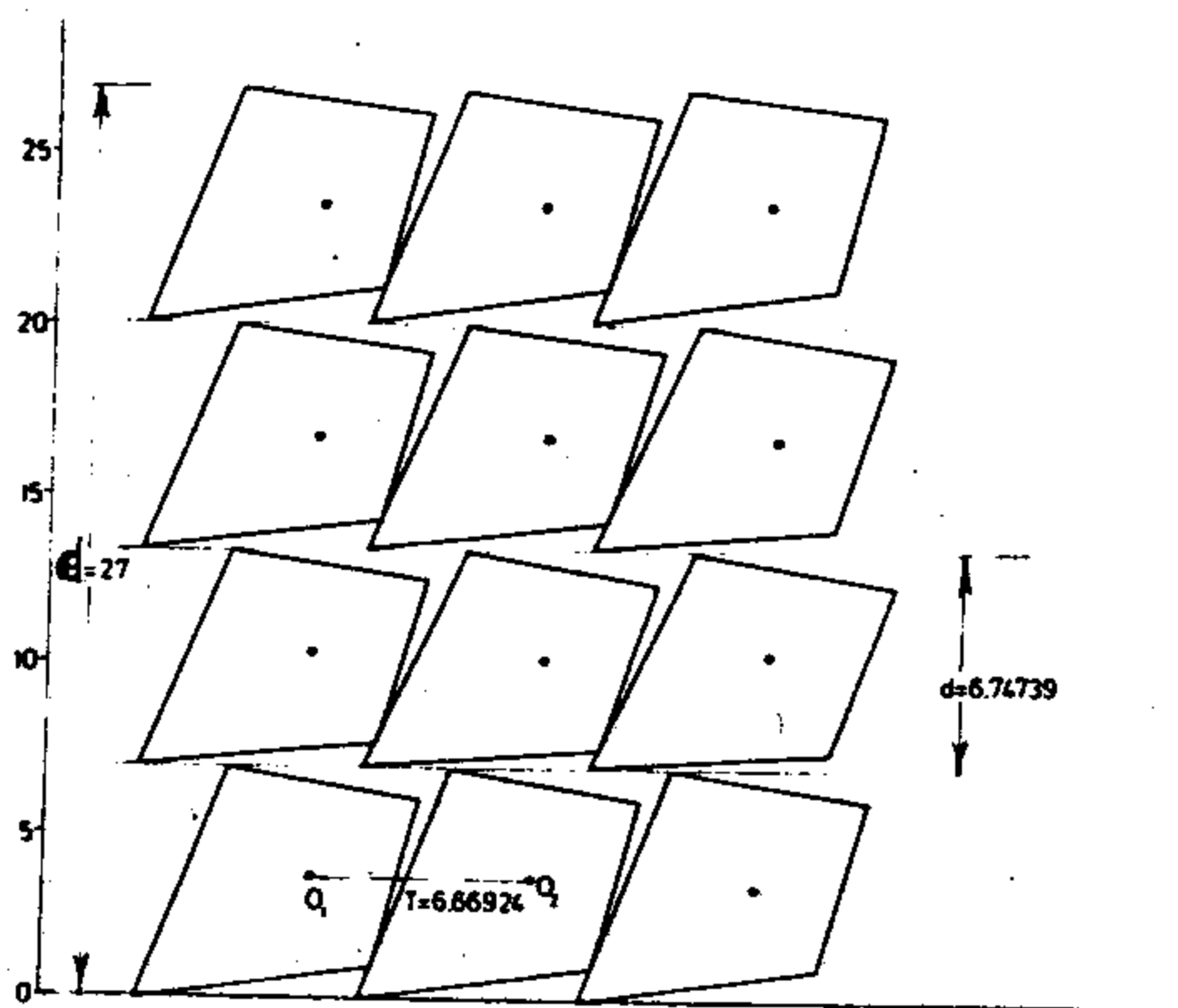
Na sl.3.7. vidimo optimalni smještaj objekta S: $A_1(-2,6)$; $A_2(-4,1)$; $A_3(1,4)$; $A_4(3,-1)$.

Na sl.3.8. vidimo kako se mijenja broj objekata $n(\theta)$ u zavisnosti od θ .

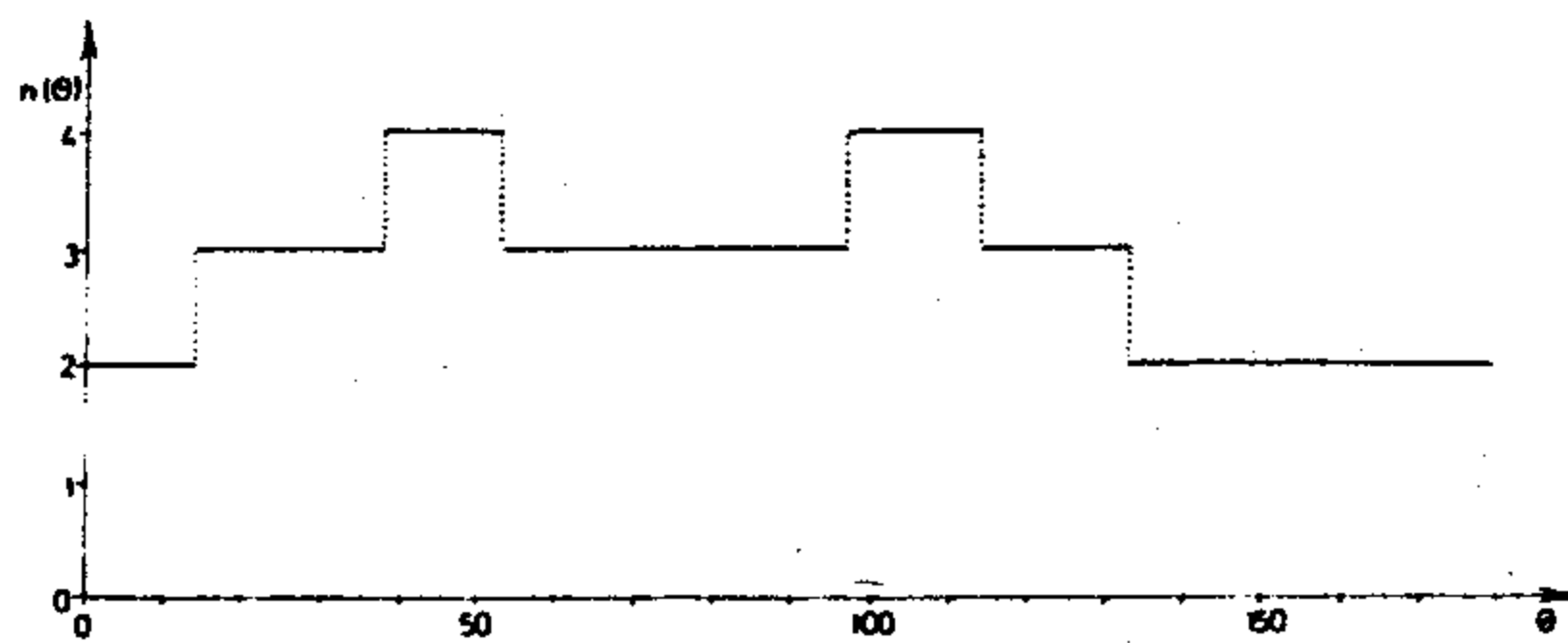
Na sl.3.9. vidimo promjenu iskorištenja u zavisnosti od θ , iz koga slijedi da je $\Delta\mu(\theta) = 40,18\%$.

Optimalni raspored za traku širine $d = 27$ se postiže za $\theta_1 = 38^\circ$.

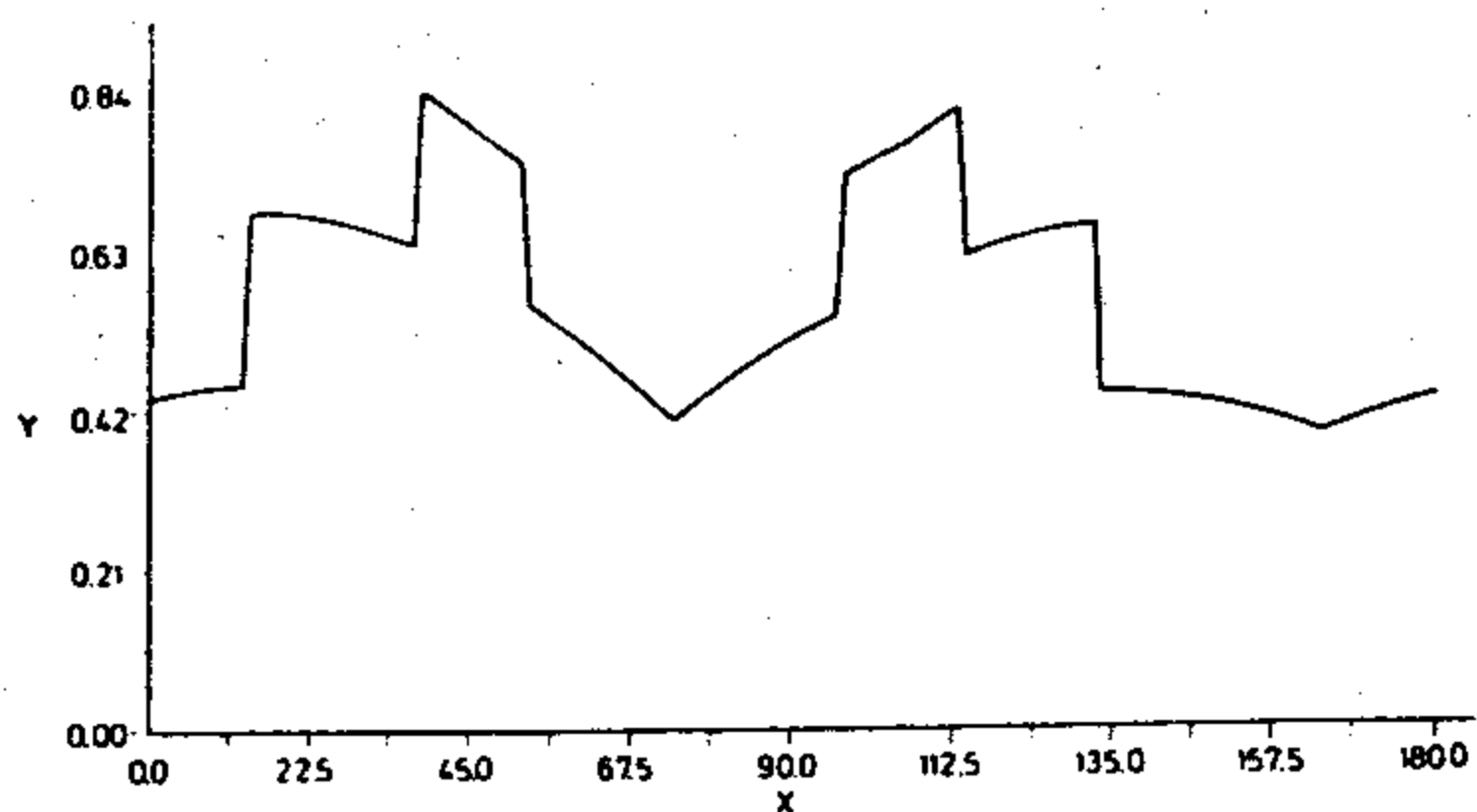
$\mu(38^\circ) = 0,84452$; $n(\theta) = 4$; $d = 6,74739$; period $T = 6,66924$.



Sl.37



Sl.3.8



Sl.3.9

Poznato je da se beskonačne trake proizvode u raznim širinama. U vezi s tim, kada korisnik nabavlja trake nije svejedno koja se koristi, jer se pokazuje da se optimalne iskorištenosti mogu razlikovati i po nekoliko desetina procenata. U vezi s tim promatra se slijedeći problem.

Neka je zadan skup beskonačnih traka E_i ($i=1,2,\dots,\ell$) čije širine zadovoljavaju uvjete $d_i \geq \max_{\theta \in [0, \pi]} [p_i(\theta) + p_i(\theta + \pi)]$

Trake se prekrivaju pojasevima u kojima su smješteni kongruentni objekti $S \in \{S\}$ jednoredno i periodično. Treba izabrati onu traku E_i za koju je koeficijent iskorištenja maksimalan. Očito da se ovaj problem svodi na određivanje

$$\max_{i=1,\ell} \{ \max_{\theta \in [0, \pi]} \nu_i(\theta) \}. \quad (3.2.8)$$

Prema tome, problem se svodi na rješavanje ℓ problema smještaja u beskonačnu traku.

Međutim, neki korisnici ne kroje samo jedan tip objekata iz traka, već više njih. Zato je interesantno odabrati traku takve širine da prosječni koeficijent iskorištenja bude maksimalan. Dakle, radi se o slijedećem problemu.

Zadan je skup različitih objekata $\{S_i\}$ $i=(1,n)$ i skup traka E_r ($r=1,s$). U pojedinu traku se smješta samo jedna vrsta objekata. Treba naći onu traku koja kada je popunimo samo sa $S \in \{S_1\}$ s koeficijentom iskorištenja ν_{1r} , samo sa $S \in \{S_2\}$ s koeficijentom iskorištenja ν_{2r} itd., ima srednji koeficijent iskorištenja

$$\frac{1}{n} \sum_{i=1}^n \overline{\nu_{ir}} \text{ maksimalan.}$$

Prema tome problem se svodi na određivanje

$$\max_{r \in \{1,s\}} \max_{\theta \in [0, \pi]} \frac{1}{n} \sum_{i=1}^n \nu_{ir}(\theta). \quad (3.2.9)$$

Vidimo da funkcija cilja zavisi od θ - parametara a svaki ima oblast definicije $[0, \pi]$.

Budući da je $\sum_{i=1}^n \mu_{ir}(\theta)$ separabilna, to je

$$\frac{1}{n} \max_{r \in [1, s]} \max_{\theta \in [0, \pi]} \sum_{i=1}^n \mu_{ir}(\theta) = \frac{1}{n} \max_{r \in [1, s]} \left[\sum_{i=1}^n \max_{\theta \in [0, \pi]} \mu_{ir}(\theta) \right]$$

U tabeli 3.1 navedeni su rezultati dobiveni pomoću paketa programa VIRED za tri objekta S_1, S_2, S_3 i pet različitih širina traka: 27, 37, 47, 57, 67.

S_1 : $A_1(-4, -4)$; $A_2(-3, 4)$; $A_3(3, 3)$; $A_4(4, -1)$

S_2 : $A_1(0, -5)$; $A_2(-5, 2)$; $A_3(-1, 3)$; $A_4(6, -1)$

S_3 : $A_1(-2, -6)$; $A_2(-4, 1)$; $A_3(1, 4)$; $A_4(3, -1)$

Tabela 3.1.

d_i	S_1			S_2			S_3			Prosjek za S_1, S_2, S_3									
	$n(0)$	$\mu(0)$ maks.	$n(0)$	$n(0)$	$\mu(0)$ maks.	$n(0)$	$n(0)$	$\mu(0)$ maks.	$n(0)$										
27	2	111	0.43575	3	110	0.65361	2	65	0.48809	4	145	0.68224	2	135	0.44231	4	38	0.84452	0.72659
37	3	114	0.47555	5	17	0.68462	4	179	0.51874	5	124	0.67008	3	136	0.48407	5	29	0.80929	0.72133
47	5	111	0.62581	6	86	0.68841	5	180	0.51047	7	145	0.68110	4	138	0.50748	7	39	0.84248	0.73733
57	5	119	0.51044	7	91	0.68486	4	74	0.44917	8	132	0.69462	5	139	0.52251	9	107	0.84005	0.73652
67	6	121	0.51809	8	99	0.68818	5	64	0.49069	8	22	0.68453	6	140	0.53269	10	39	0.84424	0.73898
Prosjek maks.iskor. za trake			0.67993			0.68051			0.83611										

106

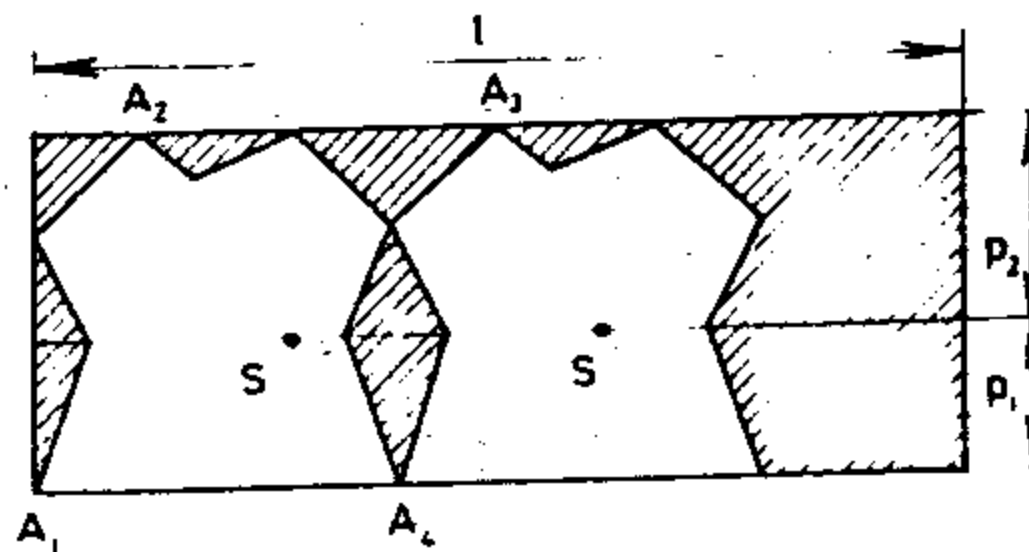
1. Vidimo da je prosječno iskorištenje za navedena tri objekta gotovo jednako za svih zadanih pet širina. Međutim, ovo nije općeniti slučaj.
2. Razlika između minimalnog i maksimalnog iskorištenja kod svih zadanih traka je jako velika.
3. Maksimalni iskorištenja za pojedine objekte vrlo su stabilni s obzirom na promjenu širine. Međutim, to za minimume ne vrijedi.
4. Za pojedine objekte maksimalna iskorištenja se postižu za različite širine (S_1 - za $d = 47$; S_2 - za $d = 57$; S_3 za $d = 27$). Traka $d = 67$ je u prosjeku za S_1, S_2, S_3 najbolja.
5. Prosjek za trake: za $S_1 - 0,67993$; $S_2 - 0,68051$; $S_3 - 0,83611$.

3. Optimalni smještaj objekata u traku konačne duljine

Vrlo često, posebno u metalnoj industriji imamo situaciju da je traka iz koje se kroje objekti konačne duljine, pa se treba naći onu širinu trake kod koje je iskorištenost maksimalna.

Problem:

Odrediti takav parametar θ objekata $S \in \{S\}$ kod koga je koeficijent iskorištenja $\mu(\theta)$ trake E konačne duljine maksimalan. Očito da ovdje ne možemo koristiti površine kvazi paralelograma $A_1A_2A_3A_4$ kao kriterij iskorištenosti, jer se ne uzimaju u obzir otpadi na rubovima, koji mogu biti jako veliki, kada su objekti i traka bliskih dimenzija, sl.3.10.



Sl.3.10

Za ovaj problem koeficijent iskorištenja je dan izrazom

$$\mu(\theta) = \frac{nP(S)}{|P_1+P_2|} \quad (3.3.1)$$

gdje: n - broj objekata S u E .

Pošto su $P(S)$ i duljina l - konstante, a broj objekata zavisi od parametra θ tj.

$$n(\theta) = N \left[\frac{l - p(\theta + \frac{\Pi}{2}) - p(\theta - \frac{\Pi}{2}) + K(\theta)}{K(\theta)} \right] \quad (3.3.2)$$

a $p_1 + p_2 = p(\theta) + p(\theta + \Pi)$, slijedi da se problem svodi na nalaženje

$$\max_{\theta \in [0, \Pi]} \mu(\theta) = \frac{P(S)}{l} \max_{\theta \in [0, \Pi]} \frac{N \left[\frac{l - p(\theta + \frac{\Pi}{2}) - p(\theta - \frac{\Pi}{2}) + K(\theta)}{K(\theta)} \right]}{p(\theta) + p(\theta + \Pi)} \quad (3.3.3)$$

Neka svojstva funkcije zadane sa (3.3.3)

- 1) - općenito $\nu(\theta) : \theta \in [0, \Pi]$ ima mnogo ekstrema,
- 2) - ona je po dijelovima neprekidna na $|0, \Pi|$,
- 3) - ako je $p(\theta + \frac{\Pi}{2}) + p(\theta - \frac{\Pi}{2}) : \theta \in [0, \Pi]$ monotona, onda se $\max \nu(\theta)$ postiže u točkama prekida funkcije $n(\theta)$, koja je po dijelovima neprekidna.

Iz prednjih svojstava slijedi da treba naći one θ za koje funkcija $n(\theta)$ - ima prekide, a to se svodi na nalaženje svih korjena na $|0, \Pi|$ jednačbe:

$$L - [p(\theta + \frac{\Pi}{2}) + p(\theta - \frac{\Pi}{2}) - (r-1)K(\theta)] = 0 \quad r = z_1, z_1+1, \dots, z_2 \quad (3.3.4)$$

gdje je

$$z_1 = N \left[\frac{L - \max [p(\theta + \frac{\Pi}{2}) + p(\theta - \frac{\Pi}{2})]}{\max K(\theta)} \right] + 1$$

$$z_2 = N \left[\frac{L - \min [p(\theta + \frac{\Pi}{2}) + p(\theta - \frac{\Pi}{2})]}{\min K(\theta)} \right] + 1 \quad (3.3.5)$$

Ako za $r = z_1$ jednačba (3.3.4) nema rješenje, onda za taj isti broj objekata $n(\theta)$, funkcija $\nu(\theta) : \theta \in [0, \Pi]$ ima maksimum tamo gdje je $p(\theta) + p(\theta + \Pi)$ minimalna. Ako za $r = z_2$ (3.3.4) nema rješenja, onda to znači da se objekti u traci E ne mogu smjestiti bez preklapanja.

Našavši korjene θ_r , ($i=1, m$), koje poredamo rastući, odredimo onaj θ_i^* za koji se postiže $\min [p(\theta) + p(\theta + \Pi)]$, a zatim odredimo

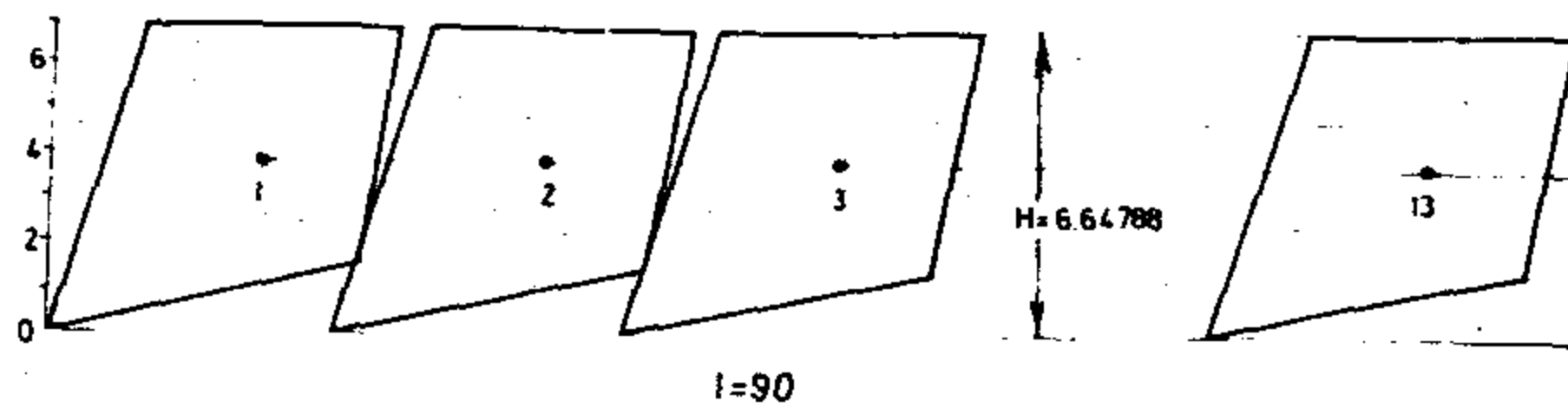
$$\max \nu(\theta) = \frac{P(S)}{L} \max_{r \in [1, m-1]} \frac{n(\theta_r^*)}{p(\theta_r^*) + p(\theta_r^* + \Pi)} \quad (3.3.6)$$

Kod numeričkog računanja da ne bi došlo do propuštanja nekog korijena s jedne strane, te da ne bi radili s većom tačnošću nego je to potrebno, duljinu intervala treba odabrati tako da vrijedi izraz

$$\Delta\theta < \frac{[\min K(\theta)]^2}{\ell - \min_{\theta \in [0, \pi]} [p(\theta) + p(\theta + \pi)] \cdot \max_{\theta \in [0, \pi]} K'(\theta)}$$

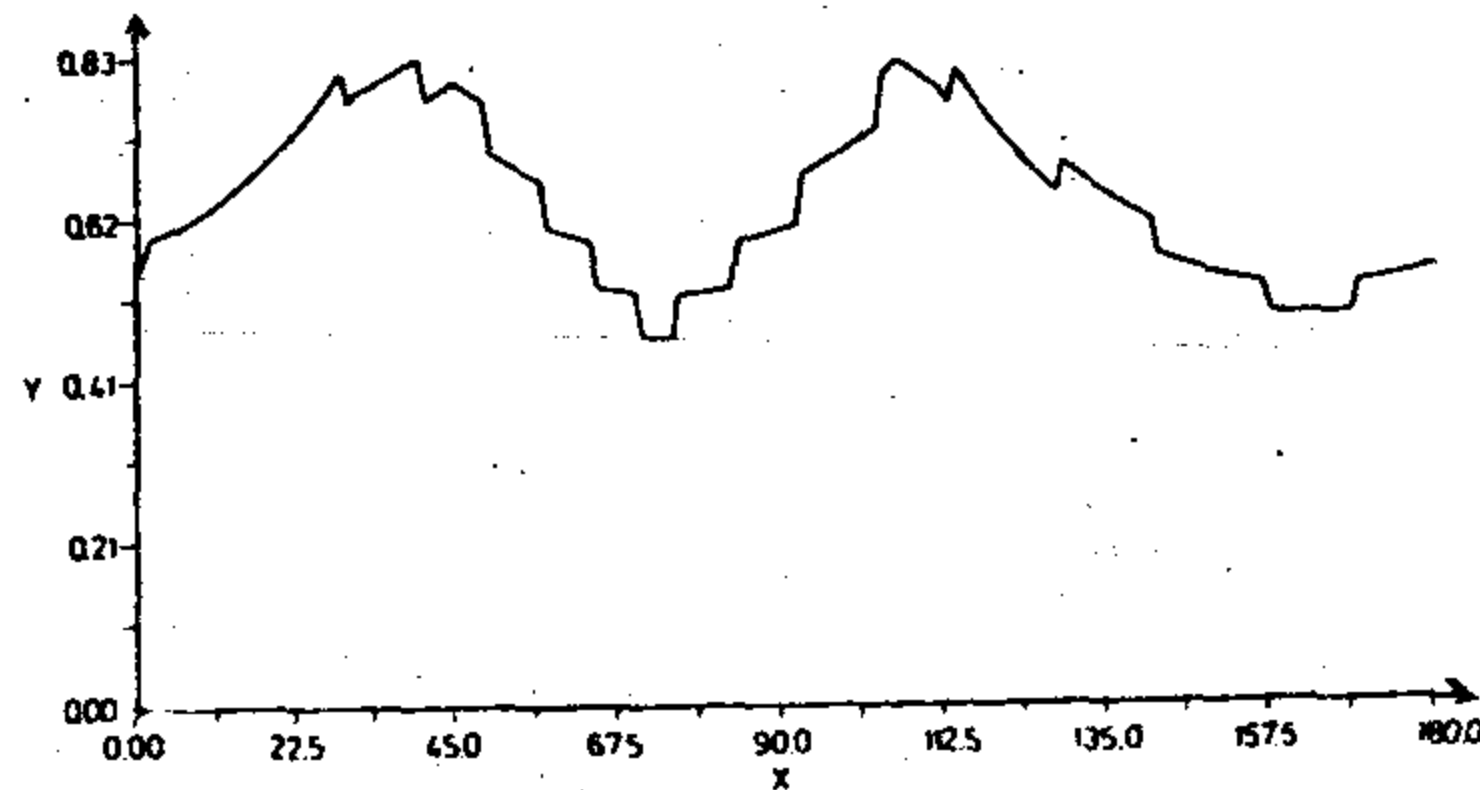
Na sl.3.11. vidimo optimalni raspored objekta S:

$A_1(-2, -6)$; $A_2(-4, 1)$; $A_3(1, 4)$; $A_4(3, -1)$ u traku duljine $\ell = 90$.
 $\text{Max } \mu(40^\circ) = 0,82566$; $\text{Min } \mu(74^\circ) = 0,46397$; $d(40^\circ) = 6,64788$;
 $T(40^\circ) = 6,76907$. Podaci su dobiveni pomoću paketa programa KONTRA.



Sl.3.11

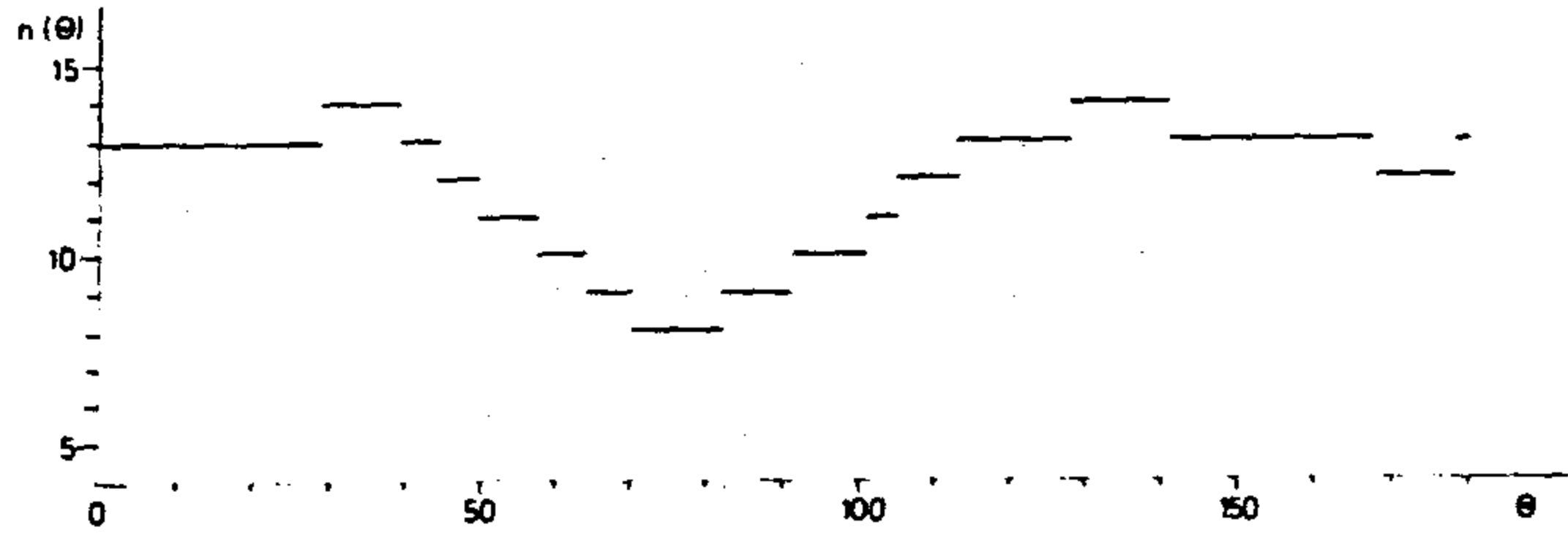
Na sl.3.12. dan je graf promjena koeficijenta iskorištenja u zavisnosti od θ za $\ell = 90$.



Sl.3.12

Kao što se vidi iz navedenog primjera razlike izmedju minimalnih i maksimalnih iskorištenja mogu biti drastične.

Na sl.3.15. dan je graf funkcije $n(\theta)$ za $l = 90$.



Sl.3.13

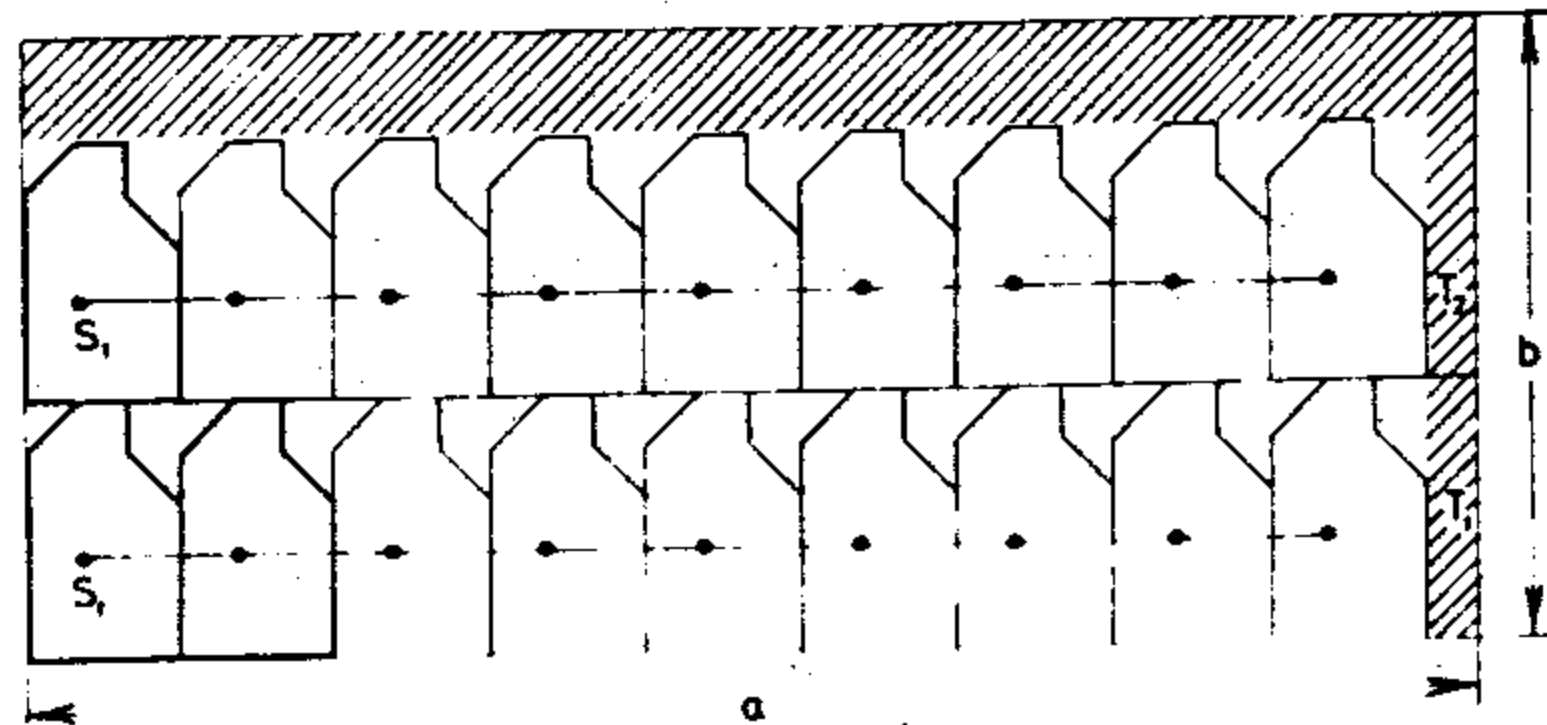
Tabela 3.2.

Duljina trake	MINIMUM			MAKSIMUM			Ampli- tuda $\mu(\theta)$
	$n(\theta)$	θ	$\mu(\theta)$	$n(\theta)$	θ	$\mu(\theta)$	
60	5	71	0.43559	8	106 ^o	0.81927	0.38367
70	6	71	0.44804	10	40	0.81659	0.36855
80	8	164	0.45500	12	34	0.82350	0.36800
90	8	73	0.46405	13	40	0.82566	0.36169
100	9	74	0.46397	14	110	0.83014	0.36006

Iz Tabele 3.2. vidimo da minimum, odnosno maksimum, iskorištenja gotovo da ne reagiraju na zadane promjene duljina. Također vidimo da su i amplitude iskorištenja gotovo jednake za navedene duljine. Međutim, razlike između minimalnih i maksimalnih iskorištenja za sve duljine traka su jako velike, tj. kreću se od 36 do 38%.

4. Optimalan smještaj u pravokutnik

U praksi je najčešći slučaj krojenja objekata iz pravokutnih materijala $\Omega = (a \times b)$. Pravokutnik Ω se prekriva pojasevima T_i u kojima se nalaze kongruentni objekti $S \in \{S\}$. Trake mogu biti postavljene paralelno stranici a ili stranici b , već prema tome gdje je bolje iskorištenje. Sl.3.14.



Sl.3.14

Prema tome imamo slijedeći problem:

Treba naći takvu orijentaciju objekata $S \in \{S\}$ tj. parametar θ i smjer pojaseva T_i (paralelno stranici a ili b) da koeficijent iskorištenja bude maksimalan.

Ovdje je očito da je koeficijent iskorištenja dan izrazom

$$\mu(\theta) = \frac{nP(S)}{a \times b} \quad (3.4.1)$$

gdje n - broj objekata u Ω .

Pošto su $P(S)$ i $(a \times b)$ konstante, $\max \mu(\theta)$ zavisi samo od n . No $n = m \times K$ - gdje m - broj objekata u jednom pojaseu; K - broj pojaseva.

Broj objekata u Ω se može iskazati pomoću

$$n_a(\theta) = N_a \left[\frac{a - p(\theta - \frac{\pi}{2}) - p(\theta + \frac{\pi}{2}) + K(\theta)}{K(\theta)} \right] \times N_{2a} \left[\frac{b}{p(\theta) + p(\theta + \pi)} \right] \quad (3.4.2)$$

ako su pojasevi paralelni stranici a .

Na isti način izračunamo $n_b(\theta)$.

Pomoću operacije R - disjunkcije [43], $V_a = V_1$ rješenje navedenog problema se svodi na nalaženje

$$\max_{\theta \in [0, \Pi]} n(\theta) = \max_{\theta \in [0, \Pi]} [n_a(\theta) \vee_1 n_b(\theta)] \quad (3.4.3)$$

S obzirom na svojstva R - disjunkcije V_1 problem se rješava u dva koraka, i to tako da se najdže $\max_{\theta} n_a(\theta) = \bar{n}_a$, a zatim $\max_{\theta} n_b(\theta) = \bar{n}_b$, a onda izabere bolji.

Općenito, funkcija $n(\theta) : \theta \in [0, \Pi]$ je mnogoeksremalna. Iz same prirode problema je jasno da kod promjene orijentacije nećemo imati kontinuiranu promjenu broja objekata, već da će $n(\theta)$ biti po dijelovima konstantna, a kako se radi o smještaju samo cijelog broja objekata, funkcija općenito ima prekide I reda.

Označimo li sa:

$$n_{1a} = N \left[\frac{a - \max_{\theta \in [0, \Pi]} [p(\theta + \frac{\Pi}{2}) + p(\theta - \frac{\Pi}{2})]}{\max_{\theta \in [0, \Pi]} K(\theta)} \right] + 1 \quad (3.4.4)$$

$$\bar{n}_{1a} = N \left[\frac{b}{\max [p(\theta) + p(\theta + \frac{\Pi}{2})]} \right] \quad (3.4.5)$$

$$n_{2a} = N \left[\frac{a - \max_{\theta \in [0, \Pi]} [p(\theta + \frac{\Pi}{2}) + p(\theta - \frac{\Pi}{2})]}{\min_{\theta \in [0, \Pi]} K(\theta)} \right] + 1 \quad (3.4.6)$$

$$\bar{n}_{2a} = N \left[\frac{l}{\min [p(\theta) + p(\theta + \Pi)]} \right] \quad (3.4.7)$$

kao i adekvatne iznose sa n_{1b} , \bar{n}_{1b} , n_{2b} , \bar{n}_{2b} , kada su trake paralelne stranici b , onda se pokazuje da broj prekida funkcije $n(\theta)$ nije veći od n .

$$n = n_{1a} \cdot \bar{n}_{1a} \vee_1 n_{2b} \bar{n}_{2b} \quad (3.4.8)$$

Znači da problem možemo riješiti tako da najdemo korijene jednadžbe:

$$a - [p(\theta) + \frac{\Pi}{2}] + p(\theta - \frac{\Pi}{2}) - (i-1)K(\theta) = 0 \quad i = (n_{1a} \dots n_{2a}) \quad (3.4.9)$$

i korijene jednadžbe:

$$b - i[p(\theta) + p(\theta + \pi)] = 0 \quad i = (\bar{n}_{1a}, \dots, \bar{n}_{2a}) \quad (3.4.10)$$

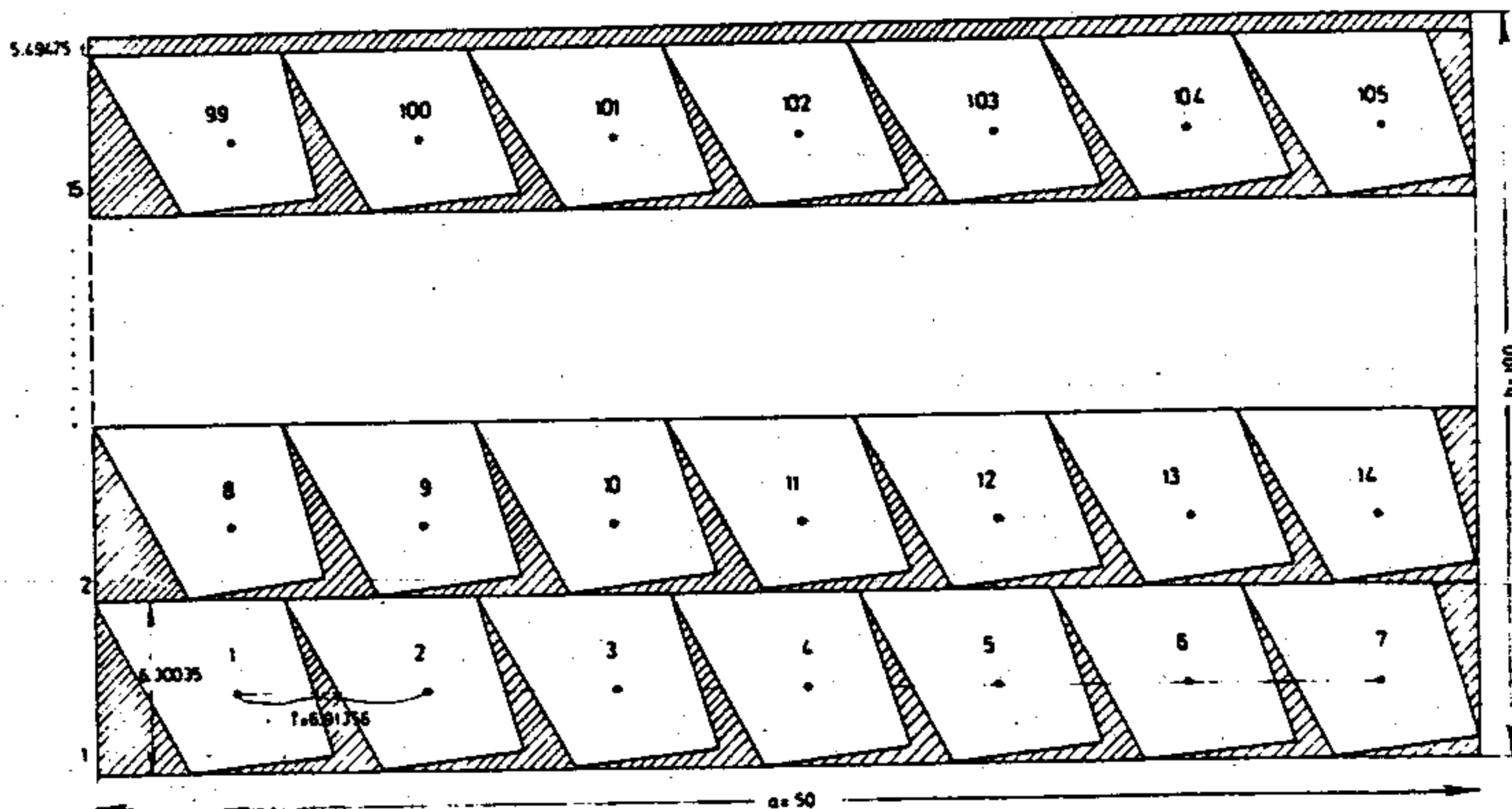
Ako su θ_r ($r=1, 2, \dots, K_1$) korijeni jednadžbe (3.4.9) a θ_{K_1+s} ($s=1, 2, \dots, K_2$) korijeni (3.4.10), onda računamo

$$n_a = \max_{r \in [1, K_1 + K_2]} [n_{1a}(\theta_r) \times \bar{n}_{1a}(\theta_r)] \quad (3.4.11)$$

Napravimo doslovno iste proračuna za slučaj kada su pojasevi paralelni stranici b , tj. nadjemo $n_{2b} \times \bar{n}_{2b}$ u (3.4.8), i nadjemo

$$\max n(\theta) = \max\{\bar{n}_a, \bar{n}_b\}$$

Na sl.3.15. vidimo optimalni smještaj objekata S : $A(-2, -6)$; $A_2(-4, 1)$; $A_3(1, 4)$; $A_4(3, -1)$ u pravokutnik $\Omega = 50 \times 100$.

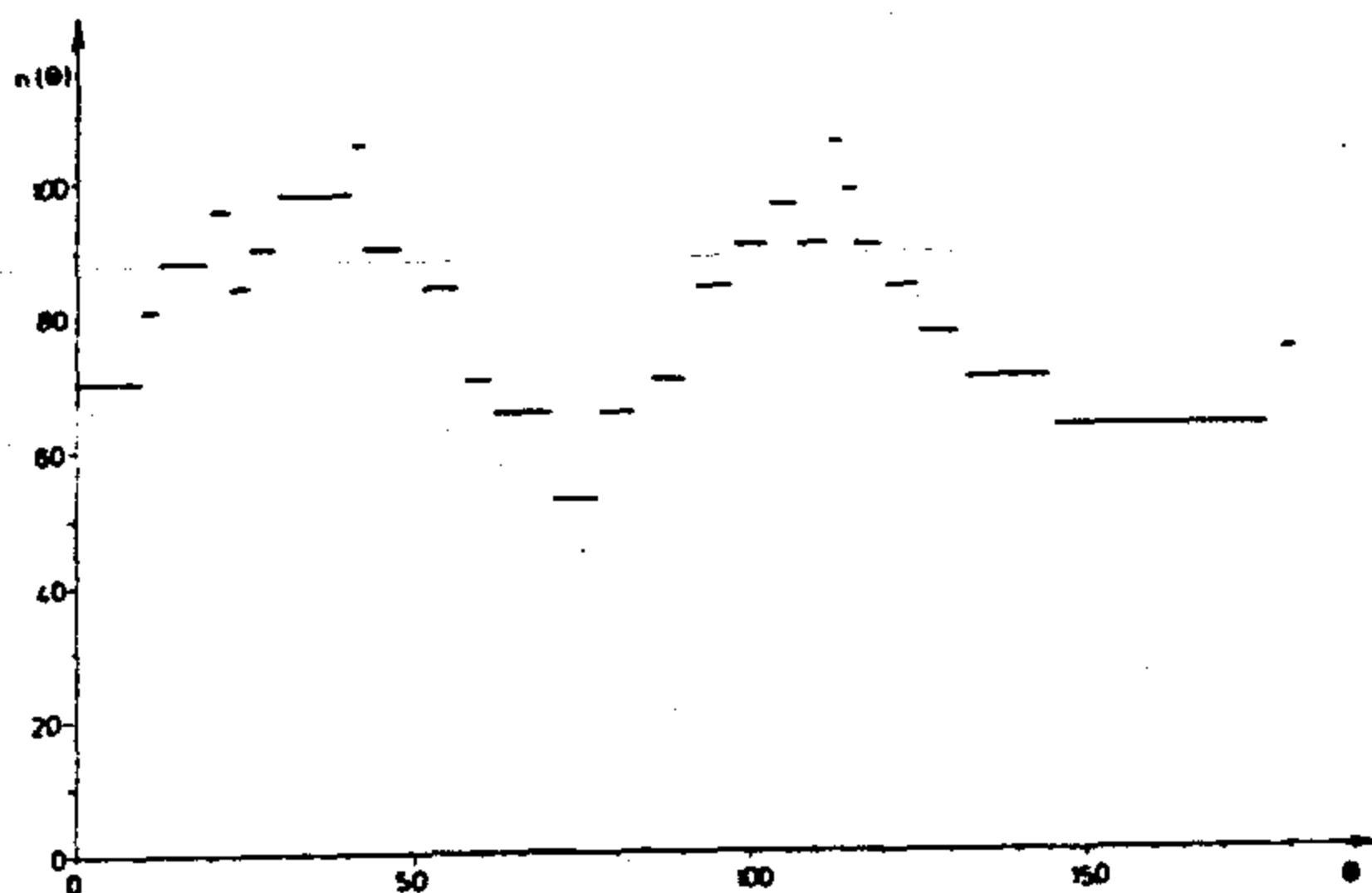


Sl.3.15

Pojasevi su paralelni stranici $a = 50$, a ima ih 15, širina im je $6,30035$. U svaki pojas se smjesti 7 objekata s periodom $T = 6,91356$. U Ω se ukupno smjesti 105 objekata, tako da je maksimalno iskorištenje kod $\theta = 112^\circ$ i iznosi $0,79800$. Isti koeficijent iskorištenja postiže se i kod $40 \leq \theta \leq 41$, broj traka je 15, a širina trake $6,64780$. Minimalno iskorištenje se postiže za $72 \leq \theta \leq 75$, jer se smjesti 52 objekta, pa je koeficijent iskorištenja $0,39520$.

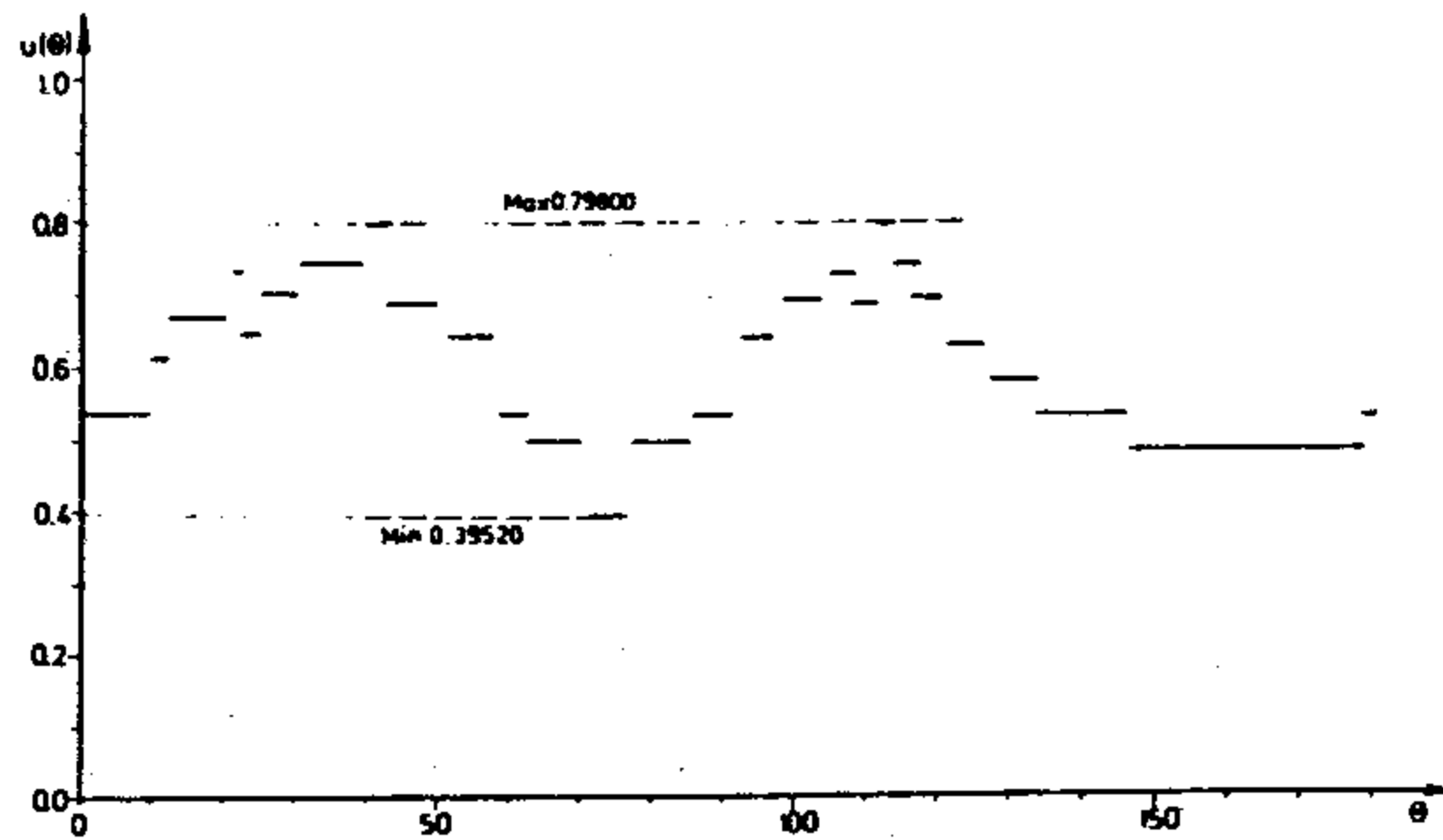
Ako režemo pojaseve paralelno stranici $b = 100$, onda se optimum od 105 smještenih objekata postiže za $31^\circ \leq \theta \leq 35^\circ$. Dobije se 7 pojaseva širine 7,03024. U svaki pojas se smjesti 15 objekata s periodom $T = 6,22838$. Rezultati su dobiveni pomoću paketa programa PRAVOK.

Kako se mijenja broj objekata u zavisnosti od θ , vidimo na sl. 3.16.



Sl. 3.16

Zavisnost koeficijenta iskorištenja od θ vidimo na sl. 3.17. Iz toga slijedi da za naš primjer koeficijent iskorištenja varira za 0,40280. Pošto nema nikakvog razloga da vjerujemo da je to slučaj samo za naš primjer, očito da razlike između minimalnog i maksimalnog iskorištenja mogu biti još i veće. Posebno treba ukazati na to da ima ne mali broj takvih slučajeva gdje se maksimalni broj objekata može smjestiti u Ω za vrlo mali interval θ , ponekad i ispod jednog stupnja, pa je za takve slučajeve vrlo mala vjerojatnost da ih praktičari, kako se to kaže, od oka otkriju. U našem primjeru 105 objekata se smjesti samo kod $\theta \in [40^\circ - 41^\circ]$ i $\theta = 112^\circ$. Već kod $\theta = 39^\circ$ i 113° smjesti se 98 objekata a kod $\theta = 42^\circ$ i 110° samo 90 objekata.



Sl. 3.17

POSEBNI SLUČAJ I:

Kao što je poznato, kod nas i u svijetu, tvornice proizvode mnogo raznih dimenzija pravokutnih materijala. S obzirom na to prirodno se nameću slijedeći problemi:

Iz skupa $\{\Omega_r\}$ ($r=1,s$) mogućih dimenzija izabrati onu kod koje je otpad za zadani objekt kojeg krojimo minimalan.

Znači da trebamo riješiti "s" problema smještaja u pravokutnik i onda naći

$$\mu = \max \left\{ \max_{\substack{r=1,s \\ \theta \in [0, \pi]}} \mu_r(\theta) \right\} \quad (3.4.12)$$

Kada problem rješavamo numerički, onda ga treba rješavati paralelno za sve Ω_r , i to sa $\Delta\theta = \min\{\Delta\theta_1, \Delta\theta_2, \dots, \Delta\theta_s\}$.

Za isti objekt S_3 čiji optimalni smještaj vidimo na sl.3.15 i još dva objekta S_1 : $A_1(-4, -4)$; $A_2(-3, 4)$; $A_3(3, 3)$; $A_4(4, -1)$ i S_2 : $A_1(0, -5)$; $A_2(-5, 2)$; $A_3(-1, 3)$; $A_4(6, -1)$ navedeni su optimalni smještaji za pet različitih dimenzija pravokutnika.

Rezultate vidimo u tabelama 3.3a do 3.3d.

Rješenja su dobivena pomoću adaptiranog paketa programa PRAVOK.

Minimumi kod smještanja paralelno stranici a

TABELA 3.3

Dimenzije pravo- kutnika	S ₁			S ₂			S ₃		
	n	θ	μ(θ)	n	θ	μ(θ)	n	θ	μ(θ)
30x 60	21	1- 14	0.49000	14	166-179	0.33056	24	1- 18 51-179	0.50667
35x 70	24	30-179	0.41143	27	35- 37 82-179	0.46837	27	61- 38	0.41878
40x 80	40	77-179	0.47250	28	97-103	0.37187	30	72- 76	0.35625
47x 90	40	37- 56	0.41481	36	164-165	0.37778	48	64- 84	0.45037
50x100	55	41- 68	0.46200	48	170-179	0.40800	52	71- 76	0.39520

Minimumi kod smještanja paralelno stranici b

TABELA 3.3b

Dimenzije pravo- kutnika	S ₁			S ₂			S ₃		
	n	θ	μ(θ)	n	θ	μ(θ)	n	θ	μ(θ)
30x 60	18	26-179	0.42000	16	47 72-179	0.37778	24	61-179	0.50667
35x 70	24	133-161	0.41143	27	35- 37 82-179	0.46837	24	71-179	0.51224
40x 80	32	33- 56	0.42000	32	1- 2	0.42500	35	71- 75	0.41542
45x 90	40	140-151	0.41481	36	81- 84	0.37778	48	71- 75	0.45037
50x100	50	41- 53	0.42000	52	85-179	0.44200	54	72- 75	0.41040

Maksimumi kod smještanja paralelno stranici a

TABELA 3.3c

Dimenzije pravo- kutnika	S ₁		S ₂		S ₃		S ₁ , S ₂ , S ₃ $\bar{\mu}(0)$			
	n	$\mu(0)$	n	$\mu(0)$	n	$\mu(0)$				
30x 60	24	15-25	0.56000	28	29-30	0.66111	36	40-43	0.76000	0.66037
35x 70	36	9-23	0.61714	36	39-46	0.60714	50	115	0.77551	0.66659
40x 80	50	1-9	0.65623	48	147-150	0.63750	66	30-31	0.78375	0.69249
45x 90	60	15-25	0.62223	60	27-36	0.62963	84	44-46	0.78815	0.68000
50x100	78	11-16	0.65523	77	36-37	0.65450	105	40-41	0.79800	0.70256
			$\Delta \approx 9,5\%$			$\Delta \approx 5,4\%$		112	$\Delta \approx 3,8\%$	$\Delta \approx 4,2\%$

Maksimumi kod smještanja paralelno stranici b

TABELA 3.3d

Dimenzije pravo- kutnika	S ₁		S ₂		S ₃		S ₁ , S ₂ , S ₃ $\mu(0)$			
	n	$\mu(0)$	n	$\mu(0)$	n	$\mu(0)$				
30x 60	28	15-20	0.65333	24	26-46	0.56667	35	28-33	0.76000	0.66000
35x 70	36	1-4	0.61714	36	19-34	0.62449	32	32-40	0.77551	0.67238
40x 80	50	1-10	0.65623	44	32-46	0.58437	66	40-45	0.78375	0.67478
45x 90	60	15-25	0.62222	60	24-36	0.62963	84	28-29	0.78815	0.69000
50x100	78	1	0.65520	78	19-21	0.66300	105	31-35	0.79800	0.70540

POSEBNI SLUČAJ II:

Kao što smo imali kod trake zadane širine, i kod pravokutnika se može postaviti problem izbora onog pravokutnika koji u prosjeku, kada krojimo razne objekte, daje minimalni otpad.

Funkcija cilja sada zavisi od $n \times s$ parametara (n - broja različitih objekata, s - broj različitih pravokutnika).

Problem se svodi na to da se nadje:

$$\frac{1}{n} \max_{j \in [1, s]} \sum_{i=1}^n \max_{ij} \nu_{ij}(\theta) \quad (3.4.13)$$

U tabeli 3.3c i 3.3d vidimo u zadnjem stupcu rezultate za tri objekta i pet različitih dimenzija ploča. Najveće iskorištenje je za $\Omega = 50 \times 100$, što je i logično. Međutim, može se desiti da površina Ω raste ali tako da se jedna stranica povećava a druga smanjuje, koeficijent iskorištenja tada može padati.

Rezultati su dobiveni pomoću modificiranog paketa programa PRAVOK.

5. Problemi smještaja kompleta u pravokutnik

U [20] se promatra problem kompleta. Naime, radi se o slijedećem. Zadan je pravokutnik $\Omega = a \times b$ i objekti S_1, S_2, \dots, S_n . Samo jednu vrstu objekta smještamo jednoređno i periodično u pojaseve $T_i (i=1, n)$. Sa tim pojasevima pokrijemo Ω bez rupa i preklapanja. Objekti smješteni u Ω čine komplet, tj. S_1 ulazi u komplet r_1 puta ... $S_n - r_n -$ puta.

Problem:

U Ω treba smjestiti maksimalno mogući broj kompleta. Objekti se mogu smještati u pojaseve paralelno stranici a ili stranici b . Označimo sa x_1 - dio učešća S_1 smještenih u jednom pojasu u formiranju kompleta, sa x_2 - dio učešća S_2, \dots, x_n - dio učešća S_n . Broj objekata u jednom pojasu, kada je on paralelan stranici a , dan je izrazom

$$n_a(\theta) = N_a \left[\frac{a - [p(\theta + \frac{\pi}{2}) + p(\theta + \frac{3\pi}{2})] + K(\theta)}{K(\theta)} \right], \quad (3.5.1)$$

odnosno, ako su trake paralelne sa stranicom b , izrazom

$$n_b(\theta) = N_b \left[\frac{b - [p(\theta) + p(\theta + \pi)] + K(\theta)}{K(\theta)} \right]. \quad (3.5.2)$$

Pretpostavimo da u prvi red paralelno b stavljamo S_1 . Ostatak pravokutnika Ω , nakon što je smješten jedan red sa S_1 , dan je sa:

$$a_{1b}^I = a - [p(\theta + \frac{\pi}{2}) + p(\theta + \frac{3\pi}{2})] = a - H_{1b}(\theta) \quad (3.5.3)$$

odnosno sa:

$$b_{1a}^I = b - [p_1(\theta) + p_1(\theta + \pi)] = b - H_{1a}(\theta) \quad (3.5.4)$$

Ukupan broj objekata m_{ib} - smještenih Ω dan je sa:

$$m_{ib} = x_i N_{ib} \left[\frac{b - p_i(\theta_i) - p_i(\theta_i + \pi) + K(\theta_i)}{K(\theta_i)} \right] \quad i=1, n \quad (3.5.5)$$

$$1 \cdot \frac{r_i}{N_{ib}} ; 2 \cdot \frac{r_i}{N_{ib}} ; \dots ; v_{ib} \frac{r_i}{N_{ib}} \quad (i = 1, n)$$

$$1 \cdot \frac{r_i}{N_{ia}} ; 2 \cdot \frac{r_i}{N_{ia}} ; \dots ; v_{ia} \frac{r_i}{N_{ia}} \quad (i = 1, n)$$

Ako se vrijednosti $x'_i(x''_i)$ promijene $v_{ib}(v_{ia})$ puta, onda se sve vrijednosti $x'_1 \dots x'_{i-1} x'_{i+1} \dots x'_n (x''_1, x''_2 \dots x''_n)$ takodjer promijene za $v_{ib}(v_{ia})$ puta.

III) Funkcija (3.5.8) je po dijelovima konstantna, i ima konačan broj prekida I reda.

IV) Zbog svojstva III funkcija (3.5.8) postiže ekstremalne vrijednosti ili u jednoj ili u kontinumu točaka, između dviju točaka prekida.

V) Funkcija (3.5.8) je općenito mnogoekstremalna.

Znači da se u stvari treba izračunati:

$$K_b^* = \max_{\substack{\theta' \in [0, \pi] \\ X' \in [0, \infty]}} K_b(\theta', X') \quad (3.5.10)$$

zatim

$$K_a^* = \max_{\substack{\theta'' \in [0, \pi] \\ X'' \in [0, \infty]}} K_a(\theta'', X'') \quad (3.5.11)$$

a onda naći:

$$\max_{\substack{\theta \in [0, \pi] \\ X \in [0, \infty]}} K(\theta', \theta'', X', X'') = \max\{K_b^*, K_a^*\} \quad (3.5.12)$$

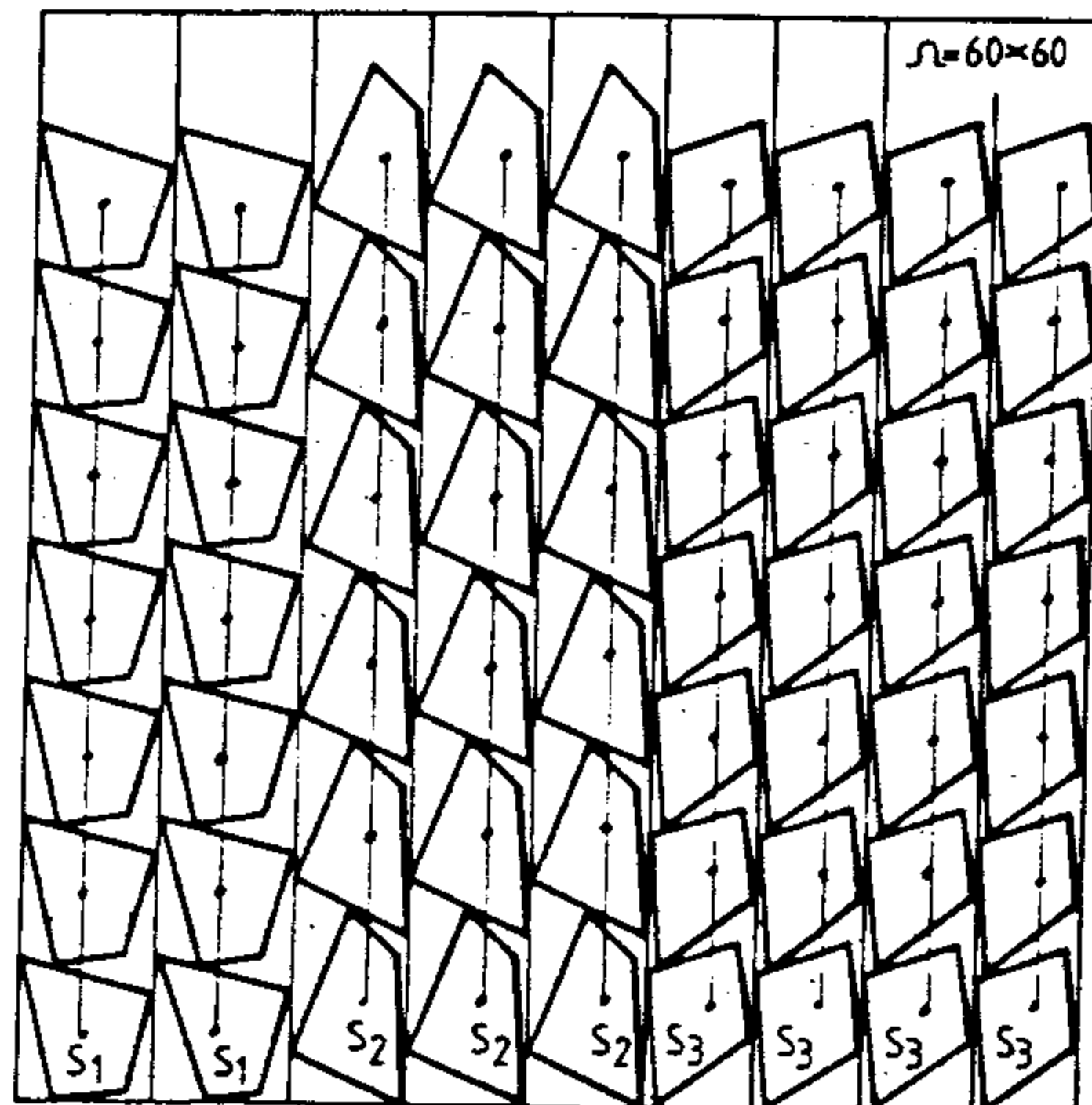
Zbog svojstava III i V, rješavanje (3.5.6) treba da uključuje promatranje (3.5.1) na svakom intervalu između susjednih prekida. Taj put rješavanja bi nas odveo do ogromnog broja operacija.

Problem je riješen pomoću algoritma čija se srž vidi iz priloženog kompjutorskog programa.

Neka su zadana tri objekta sa sl.5.10. koji ulaze u komplet u odnosu 1:2:3. $P(S_1) = 42$; $P(S_2) = 42,5$; $P(S_3) = 38$. Objekte treba smjestiti u $\Omega = 60 \times 60$ tako da se dobije maksimalni broj kompleta.

Na sl.3.18. vidimo optimalni raspored. Parametri su slijedeći: $\theta_1 = 96^\circ$, $\theta_2 = 57^\circ$, $\theta_3 = 12^\circ$. Objekt S_1 - se smješta u 2 pojasa širine 7,74507; S_2 - u 3 pojasa širine 6,46410 i S_3 u 4 pojasa širine 6,24384. Dobiveno je 9 kompleta a koeficijent iskorištenja $\mu = 0,60250$.

Rezultati su dobiveni pomoću paketa programa KOMPLET.



SLIKA 3.18

POSEBNI SLUČAJEVI

- I) Budući da na tržištu postoje različite dimenzije pravokutnika, postavlja se problem izbora one dimenzije za koju je koeficijent iskorištenja najveći za zadani komplet. U tabeli 3.4 vidimo rezultate za pet različitih dimenzija pravokutnika iz kojih se kroji komplet $\bar{S}_1 = \{S_1, S_2, S_3\}$.
- II) U tabeli 3.5 vidimo rezultate za tri različita kompleta $\bar{S}_1 = \{S_1, S_2, S_3\}$; $\bar{S}_2 = \{S_4, S_5, S_6\}$; $\bar{S}_3 = \{S_7, S_8, S_9\}$ i pet različitih dimenzija traka. Za sva tri slučaja $r_1:r_2:r_3$ je 1:2:3.
- III) U poslednjoj koloni tabele 3.5 vidimo prosjek iskorištenja za razne komplete \bar{S}_1, \bar{S}_2 i \bar{S}_3 .
 Iz tabele 3.5 vidimo da se iz iste površine pravokutnika 60 x 60 i 90 x 40 dobiva bitno različit broj kompleta tj. da se koeficijent iskorištenja razlikuje od 6 do 9%. Ova razlika može biti i bitno veća.
 Podaci su dobiveni izvjesnim modifikacijama paketa programa KOMPLET.

TABELA 3.4

Dimenzija pravo- kutnika	PARAMETRI SMJEŠTAJA												Broj kom- pleta	Koeffic. iskori- štenja			
	S ₁				S ₂				S ₃								
	O ₁	H ₁ (0)	n ₁ /TR	BR-TR n(S ₁)	O ₂	H ₂ (0)	n ₂ /TR	BR-TR n(S ₂)	O ₃	H ₃ (0)	n ₃ /TR	BR-TR n(S ₃)					
60x60	96	7.75	7	2	14	57	6.46	6	3	18	12	6.24	7	4	28	9	0.60250
75x75	0	8.15	9	2	18	21	8.83	9	3	27	12	6.24	9	5	45	13	0.55698
70x80	0	8.15	10	2	20	15	9.53	9	3	27	12	6.24	10	4	10	13	0.55946
80x80	0	8.15	10	2	20	57	6.46	8	4	32	12	6.24	10	6	60	16	0.60250
90x40	0	8.15	5	2	10	21	8.83	4	4	16	12	6.24	5	6	30	8	0.53556
40x90	0	8.15	11	1	11	57	6.46	9	2	18	12	6.24	11	3	33	9	0.60250

H_i(0) - širina trake; n_i/TR - broj objekata S_i u traci; BR-TR - broj traka

TABELA 3.5

Dimenzija pravokut- nika	S ₁ = S ₁ , S ₂ , S ₃			S ₂ = S ₄ , S ₅ , S ₆			S ₃ = S ₇ , S ₈ , S ₉			Prosječni koefficient iskorištenja {S ₁ , S ₂ , S ₃ }
	Broj kompleta	Koefficient iskorištenja	Broj kompleta	Koefficient iskorištenja	Broj kompleta	Koefficient iskorištenja	Broj kompleta	Koefficient iskorištenja		
60 x 60 ← P ₁	9	0.60250	24	0.74333	20	0.63889	0.66156			
75 x 75	13	0.55698	37	0.73342	34	0.69511	0.66183			
70 x 80	13	0.55946	38	0.75661	33	0.67768	0.66458			
80 x 80	16	0.60250	45	0.78398	40	0.71875	0.70174			
90 x 40 ← P ₅	8	0.53556	21	0.65042	18	0.57500	0.57032			

P₁ = P₅ ; Δμ = 6,7%

A = P₅ ; Δμ = 9,3%

P₁ = P₅ ; Δμ = 6,4%

Δμ = 9,1%

PROGRAMI UZ GLAVU III

PROGRAM J E R E D

• • • • •
 • NAMJENA: JEDNOREDNO PERIODIČKO SMJESTANJE KONGRUENTNIH OBJEKATA,
 • ZA TRAKU PESKONACNE DULJINE ODREDITI KUT ZAKOŠENOSTI
 • KONGRUENTNIH OBJEKATA (POLIGONA) PRI KOJEMU SE MINIMI-
 • ZIRA OTPAD. IZ OVAKO POSTAVLJENE FUNKCIJE CILJA SLEDI
 • SIRINA TRAKE,
 •

• • • • •
 • COMMON /APOL/ XA(50), YA(50), N
 • /BEPO/ XB(50), YB(50), M
 • /VHKF/ XC(90), YC(90), T
 • COMMON /EXEP/ XP(90), YP(90), NN
 • /ALFA/ ALFA(90)
 • /POTP/ H(360), NH
 • /KORAK/ DQS
 • /MEMA/ GM(360), HT(360), FM12(1000), K1, H1
 • /LELA/ GL(20), FL(20), FL2(20), L1
 • COMMON /FEXT/ F(200), NF
 • /INEXT/ IN(50), K
 • /IERR/ IERR
 • DIMENSION QV(50), V(50), FUM(50)
 • DIMENSION XZ(50), YZ(50)
 • INTEGER T

READ 10, N, DQS
 PRINT 15, N
 READ 20, (XA(I), YA(I), I=1, N)
 PRINT 25, (XA(I), YA(I), I=1, N)

M = N
 DO 5 II = 1, M
 XB(II) = XA(II)
 YB(II) = YA(II)
 5 CONTINUE

CALL HOKOF

PRINT 30, T
 PRINT 25, (XC(I), YC(I), I=1, T)
 10 FORMAT(15, F10.5)
 15 FORMAT(1H1, ' KOORDINATE POLIGONA KOJI SE SMJESA OD', I6, ' VRHOVA', /)
 20 FORMAT(8F10.5)
 25 FORMAT(3(2X, '(, F10.5, ' ;, F10.5, ')', /)
 30 FORMAT(//1X, ' KOORDINATE HODOGRAFA OD', I6, ' VRHOVA', /)

DO 40 II = 1, N
 JJ = N+1-II
 XP(II) = XA(JJ)
 YP(II) = YA(JJ)
 40 CONTINUE
 NN = N

CALL POTFOR
 CALL KONTAF

```

C
C
NF = K1/2
C
DO 50 II = 1, NF
  F(II) = FM12(II)
50 CONTINUE
C
IERR = 0
CALL EXTREM
IF(IERR.NE.0) STOP 222
NF = NH/2
DO 60 II = 1, NF
  F(II) = H(II) + HINF+II)
60 CONTINUE
V(II) = F(II) * FM12(II)
QV(II) = QM(II)

-----
FUN(II) = F(II)
C
KH = K + 1
DO 70 II = 2, KH
  IQPO = IN(II-1)
  QV(II) = QM(IQPO)
  V(II) = F(IQPO) * FM12(IQPO)
  FUN(II) = F(IQPO)
70 CONTINUE
C
IERR = 0
CALL EXTREM
IF(IERR.NE.0) STOP III
C
C
DO 80 II = 1, K
  IQPO = IN(II)
  QV(KH+II) = QM(IQPO)
  V(KH+II) = F(IQPO) * FM12(IQPO)
  FUN(KH+II) = F(IQPO)
80 CONTINUE
C
-----
NZ = N
DO 85 II = 1, NZ
  XZ(II) = XA(II)
  YZ(II) = YA(II)
85 CONTINUE
C
S = SF(XZ, YZ, NZ)
PRINT 86, S
86 FORMAT(//5X, 'POVRšina POLIGONA SP =', 15.5)
C
-----
PRINT 87
87 FORMAT(//5X, 'REZULTATI JEDNOREDNE OPTIMIZACIJE'
  • //5X, 'RED. BR.', 10X, 'KUT', 10X, 'POVRšina' 7X, 'ISKOLISTIVOST',
  • 5X, 'SIRINA TRAKE'//)
V'IN = V(1)
Q'IN = QV(1)
ININ = 1
FUNIN = FUN(1)
VQIS = S/V'IN
PRINT 89, ININ, QMIN, V'IN, VQIS, FUNIN
LOOP = KH + 1

```

```

DO 90 II = 2, LOOP
  VQIS = S/V(II)
  IF(V(II),GE,VMIN) GO TO 88
  FUMIN = FUN(II)
  VMIN = V(II)
  QMIN = QV(II)

```

```

      IMIN = II
88   PRINT 89, II, QV(II), V(II) , VQIS , FUN(II)
89   FORMAT(5X, I5, 10X, F10.1, 3(5X, F10.5))
90 CONTINUE

```

```

      •• ISKORISTIVOST ••
      VQIS = S/VMIN
      PRINT 99, QMIN, VQIS, FUMIN
99   FORMAT(//5X, 'KUT OBJEKTA      =',F15,1
      •       /5X, 'ISKORISTIVOST   =',F15,5
      •       /5X, 'SIRINA TRAKE   =',F15,5)

```

```
CALL FUMIJA(S)
```

```
STOP
END
```

PROGRAM VIRED

•••••

• NAMJENA: VISEREDNO PERIODICKO SMJESTANJE KONGRUENTNIH OBJEKATA
 • TO ZNACI DA ZA TRAKU ZADANE SIRINE AT I BESKONACNE
 • DUZINE TREBA ODREDITI U KOLIKO TRAKA TREBA PERIODICKI
 • SLOZITI ZADANE POLIGONE DA ISKORISTIVOST BUDE MAKIMALNA

•••••

```

COMMON /APOL/ XA(50), YA(50), N
      • /BEPO/ XB(50), YB(50), M
      • /VHYF/ XC(90), YC(90), T
      • /EXEP/ XP(90), YP(90), NN
      • /QVRH/ QV(50), QA(50)
      • /ALFA/ ALFA(90)
      • /POTP/ H(360), NH
      • /KORAK/ DQS
      • /MEHA/ QM(360), NT(360), FN12(1000), K1, M1
      • /LELA/ QL(20), FL1(20), FL2(20), L1
      • /IERR/ IERR

```

```

DIMENSION XZ(50), YZ(50)
DIMENSION ITA(50), INA(50), FUN(50), QUN(50), HUN(50), VIS(50)

```

```
INTEGER T
```

```
READ 10, I, DQS, AT
```

```

PRINT 15, N
READ 20, (XA(I),YA(I),I=1,N)
PRINT 25, (XA(I),YA(I),I=1,N)
N = N
DO 5 II = 1, N
    XB(II) = XA(II)
    YB(II) = YA(II)
5 CONTINUE

```

```

C
CALL HOKOF

```

```

C
PRINT 30, T
PRINT 25, (XC(I),YC(I),I=1,T)
10 FORMAT(15,F10.5,F10.5)
15 FORMAT(1H1,' KOORDINATE POLIGONA KOJI SE SHJESA OD',I6,' VRHOVA'
20 FORMAT(8F10.5)
25 FORMAT(3(2X,'(',F10.5,';',F10.5,')'))
30 FORMAT(//1X,' KOORDINATE HODOGRAFA OD',I6,' VRHOVA'//)

```

```

DO 40 II = 1, N
    JJ = N+1-II
    XP(II) = XA(JJ)
    YP(II) = YA(JJ)
40 CONTINUE
NH = N

```

```

C
CALL POTFOR

```

```

C
CALL MONTAF

```

```

C
NPI = NH/2
ILD = 0
JH = 0
DO 50 II = 1,NPI
    IL = AT/(P(II) + H(NPI + II))
    IF(IL,EQ,ILD) GO TO 50
    JH = JH + 1
    IF(JH,GT,50) GO TO 100
    ITA(JH) = IL
    INA(JH) = II
    ILG = IL
50 CONTINUE

```

```

DO 60 II = 1,JH
    ID1 = IFA(II)
    ID2 = NPI
    IF(II.LT,JH) ID2 = INA(II + 1) - 1
    FUN(II) = FM12(ID1)

```

```

    QUN(II) = QN(ID1)
    HUN(II) = P(ID1) + H(NPI + ID1)
    ID1 = ID1 + 1
    IF(ID1,GT,ID2) GO TO 60
    DO 60 JJ = ID1,ID2
        IF(FM12(JJ),GT,FUN(II)) GO TO 60
        FUN(II) = FM12(JJ)
        QUN(II) = QN(JJ)
        HUN(II) = P(JJ) + H(NPI + JJ)

```

```

60 CONTINUE

```

```

PRINT 61, (INA(II),ITA(II),FUN(II),QUN(II),HUN(II),II=1,JH)
61 FORMAT(/6X,'INDEX BR,POJAS,',7X,'PERIOD',8X,'KUT',7X,'SIR,POJAS
  ' ,2D(/6X,I4,6X,I4,6X,F12.5,F8.1,F12.5))

```

```

NZ = N
DO 70 II = 1,NZ
  XZ(II) = XA(II)
  YZ(II) = YA(II)
70 CONTINUE

```

```

S = SP(XZ,YZ,NZ)
PRINT 77, S
77 FORMAT(5X,'POVRŠINA POLIGONA S =',F15.5)

```

```

PRINT 78
78 FORMAT(/5X,'BR,TRAKA KUT ISKORISTIVOST'//)
DO 80 II = 1, JH
  VIS(II) = S/AT * ITA(II) / FUN(II)
  PRINT 85, ITA(II), QUN(II), VIS(II)
80 CONTINUE
85 FORMAT (7X, I4, F8.1, F12.5)

```

```

QUNX = QUN(1)
ITNX = ITA(1)
VIMX = VIS(1)
DO 90 II = 2, JH
  IF(VIS(II).LE.VIMX) GO TO 90
  VIMX = VIS(II)
  QUNX = QUN(II)
  ITNX = ITA(II)
90 CONTINUE
PRINT 95, ITNX, QUNX, VIMX
95 FORMAT(/10X,'BROJ TRAKA =',I5
  /10X,'KUT OBJEKTA =',F10.1
  /10X,'ISKORISTIVOST =',F10.5)

```

```

C
C
C
CALL V R I S K ( S, AT, INA, ITA, JH )

```

```

C
C
STOP

```

```

100 PRINT 101
101 FORMAT(/5X,'ODBAČEN ZATJEV ZA RADOM SA VIŠE OD 50 TRAKA')
STOP 111
END

```



```

NPI = NH/2
ILO = 0
JH = 0
ND4 = NH/4
DO 50 II = 1, NPI
  JH1 = NH - ND4 + II
  IF ( JH1 .GT. NH ) JH1 = JH1 - NH
  JH2 = ND4 + II
  IL = (AT - (H(JH1)+H(JH2))) / FM12(II) + 1
  IF(IL,EG,ILO) GO TO 50
  JH = JH + 1
  IF(JH,GT,50) GO TO 100
  IOB(JH) = IL
  INA(JH) = II
  ILO = IL

```

50 CONTINUE

```

DO 60 II = 1, JH
  ID1 = INA(II)
  ID2 = NPI
  IF ( II .LT. JH ) ID2 = INA ( II + 1 ) - 1
  HUN(II) = H(ID1) + H(ID1+NPI)
  QUN(II) = QM(ID1)
  FUN(II) = FM12(ID1)
  ID1 = ID1 + 1
  IF(ID1,GT,ID2) GO TO 60
  DO 60 JJ = ID1, ID2
    SUMH = H(JJ) + H(JJ+NPI)
    IF(HUN(II),LE,SUMH) GO TO 60
    HUN(II) = SUMH
    QUN(II) = QM(JJ)
    FUN(II) = FM12(JJ)

```

60 CONTINUE

```

NZ = N
DO 70 II = 1, NZ
  XZ(II) = XA(II)
  YZ(II) = YA(II)
70 CONTINUE
S = SP(XZ,YZ,NZ)
PRINT 77, S
77 FORMAT(5X, ' POUVRŠINA POLIGONA S = ', F15,5)

```

```

PRINT 78, AT
78 FORMAT(//5X, 'REZULTATI RASPOREDA U TRAKU KONST,DULJINE', F10,5,
  * //5X, 'RED,BR. BR.OBJEKATA KUT OBJEKTA', 4X,
  * 'ISKORISTIVOST', 3X, 'ŠIRINA TRAKE', 5X, 'PERIOD')
DO 80 II = 1, JH
  VIS(II) = (S * IOB(II)) / (AT * HUN(II))
  PRINT 85, II, IOB(II), QUN(II), VIS(II), HUN(II), FUN(II)
80 CONTINUE
85 FORMAT(6X, 14, 8X, 14, 4X, F8,1, 4X, 3F16,5)

```

```

QUNX = QUN(1)
IOBX = IOB(1)
HMIN = HUN(1)
VIMX = VIS(1)
DO 90 II = 2, JH
  IF(VIS(II),LE,VIMX) GO TO 90

```


RACUNANJE POUVSINE POLIGONA

C A L L H O K O F

DO 40 II = 1, N
 JJ = N+1-II
 XP(II) = XA(JJ)
 YP(II) = YA(JJ)
 40 CONTINUE
 NN = N

C A L L P O T P O R

C A L L K O N T A F

NZ = N
 DO 50 II = 1, NZ
 XZ(II) = XA(II)
 YZ(II) = YA(II)
 50 CONTINUE
 S = S P (XZ, YZ, NZ)

C A L L N T R A K A (BT, ITA, INA, NT)
 IF (IERR .EQ. 1) STOP
 C A L L N P O L I G (AT, IOB, INO, NO)
 IF (IERR .EQ. 1) STOP
 C A L L M U L F U N (ITA, INA, NT, IOB, INO, NO, MAZ, IMZ, MA)
 IF (IERR .EQ. 1) STOP

... PARAMETRI ISKORISTIVOSTI PRI UZDUZNIH PANTLJIKAMA

C A L L I S K O R (MAZ, IMZ, MA, S, AT, BT)
 C A L L N T R A K A (AT, ITA, INA, NT)
 IF (IERR .EQ. 1) STOP

C A L L N P O L I G (BT, IOB, INO, NO)
 IF (IERR .EQ. 1) STOP

C A L L M U L F U N (ITA, INA, NT, IOB, INO, NO, MAZ, IMZ, MA)
 IF (IERR .EQ. 1) STOP

C A L L I S K O R (MAZ, IMZ, MA, S, BT, AT)

S T O P

E N D

SUBROUTINE I S K O R (MAZ, IMZ, MA, SP, AT, BT)

- NAMJENA: ODREĐIVANJE PARAMETARA ISKORISTIVOSTI PRI SMJESTANJU
- POLIGONA U PRAVOKUTNIK

COMMON /NEMA/ QM(360), NT(360), FN12(1000), KI, NI
 DIMENSION MAZ (100), IMZ (100)

SPRK = AT * BT
 P R I N T 25, AT
 25 FORMAT(5X, 'PRAVOKUTNIK JE RAZDIJELJEN PANTLJIKAMA'
 • /5X, 'PARALELNE STRANICE VELICINE', F10.5)

P R I N T 30
 30 FORMAT(5X, 'RED. BR', 5X, 'OD KUTA', 5X, 'DO KUTA', 5X, 'BR. OBJEKATA',
 • 5X, 'ISKORISTIVOST')

```

DO 60 I = 1, NA
  IX1 = IMZ (I)
  IX2 = IMZ (I + 1) - 1
  IF ( I .EQ. NA ) IX2 = KI / 2 - 1
  VQIS = MAZ ( I ) * SP / SPRK
  PRINT 65, I, QM(IX1), QM(IX2), MAZ( I ), VQIS
60 CONTINUE

```

```

65 FORMAT( 5X, I4, 6X, F9.2, 4X, F8.2, 6X, I6, 11X, F8.5)

```

```

MAXI = MAXIND ( MAZ, NA )
IX1 = IMZ ( MAXI )
IX2 = KI / 2 - 1
IF ( MAXI .LT. NA ) IX2 = IMZ( MAXI + 1 ) - 1
VQIS = MAZ ( MAXI ) * SP / SPRK
PRINT 75, QM(IX1), QM(IX2), MAZ( MAXI ), VQIS
75 FORMAT(//5X, 'ZA KUTEVE OD', F10.2, ' DO', F10.2, ' SHJESTI SE',

```

```

16, ' OBJEKATA SA ISKORISTIVOSTI', F9.5)

```

```

RETURN
END
FUNCTION MAXIND ( ARREA, N )

```

```

INTEGER ARREA ( 100 )
MAXIND = 1
MAXEL = ARREA(1)
DO 10 I = 2, N
  IF ( ARREA(I) .LE. MAXEL ) GO TO 10
  MAXEL = ARREA(I)
  MAXIND = I
10 CONTINUE
PRINT 15, MAXIND
15 FORMAT(//5X, 'INDEX Maksimalnog Elementa JE', I6)
RETURN
END

```

— * —

I N T E G E R T

```

COMMON /APOL/ XA(50), YA(50), N
*       /BEPO/ XB(50), YB(50), M
*       /VHKF/ XC(90), YC(90), T
*       /EXEP/ XP(90), YP(90), NN
*       /POTP/ H(360), NH
*       /MEMA/ QM(360), NT(360), FM12(1000), K1, M1
*       /LELA/ QL(20), FL1(20), FL2(20), L1
*       /KORAK/ DQS
*       /IERR/ IERR
*       /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
*       /POJA/ SIR, DUZ
*       /VIHOD/ XHK(90, 10), YHK(90, 10), NHK(10)
*       /VIFKF/ FKF(1000, 5), NFK(360, 5)
*       /KLELA/ QLK(20, 10), FLK1(20, 10), FLK2(20, 10), LK1(10)
*       /PERIOD/ TE(360)
*       /POFOS/ PTP(360, 5)
*       /NPINT/ NIN

```

```

DIMENSION S(10), IQZ(10), ISL(10), ISN(10), NBP(10), HB(10)
DIMENSION MIB(10), XI(10), KMPL(10)
DIMENSION IQZN(10), IGOP(10)
DIMENSION NKMP(10)

```

```
I E R R = 0
```

```

10 READ 10, NSID, DSIR, DDUZ
   FORMAT( I2, 2F10.5 )

```

```

CALL U L A Z
IF(IERR .EQ. 1) S T O P 111
READ 20, (KMPL(I), I = 1, NOBJ)
20  FORMAT(10I5)
   PRINT 25, (KMPL(I), I = 1, NOBJ)
25  FORMAT(' ODNOS OBJEKATA U KOMPLETU S1 : S2 : ... : SN'
*       / 1X, 10(I4, ' :'))
DO 250 I = 1, NOBJ
  M = NVH ( I )
  DO 60 II = 1, M
    XB (II) = XOB (II, I )
    YB (II) = YOB (II, I )

```

```

N = M
  XA (II) = XB (II)
  YA (II) = YB (II)
60  CONTINUE
   PRINT 37, I
37  FORMAT(/// ' H O D O G R A F ', I5, ' . OBJEKTA')

CALL H O K O F

```

```
CALL K O N T A F
```

```
IF ( IERP .EQ. 1 ) S T O P 333
```

```

C
DO 130 II = 1, M1
  FKF (II, I) = FM12 (II)
  NFK ( II, I ) = NT ( II )
130 CONTINUE
C
IF(L1 .EQ. 0) GO TO 135
DO 135 II = 1, L1
  QLK (II, I) = QL (II)
  FLK1(II, I) = FL1(II)
  FLK2(II, I) = FL2(II)
135 CONTINUE
  LK1(I) = L1
C
NN = NVH (I)
DO 220 II = 1, NN
  JJ = NN + 1 - II
  XP(II) = XOB(JJ, I)
  YP(II) = YOB(JJ, I)
220 CONTINUE
C
CALL P O T P O R
C
IF ( IERR .EQ. 1 ) S T O P 444
C
DO 230 II = 1, NH
  PTP(II, I) = H(II)
230 CONTINUE
250 C O N T I N U E
C
NIN = M1
C
NHPOL = NH / 2
NHKVA = NH / 4
C
DO 252 K = 1, NSID
  SSIR = SIR + (K-1) * DSIR
  SDUZ = DUZ + (K-1) * DDUZ
  PRINT 252, SDJZ, SSIP
252 FORMAT(1H1, ' ZA PRAVOKUTNIK DUŽINE', F10.5, ' I ŽIPINE', F10.5)
C
FCMAX = 0.0
NI = 0
J = 0
C
DO 251 I = 1, NCBJ
  IQZ (I) = 1
  IQZN(I) = 1
  ISN (I) = 0
  NKMP(I) = 0
251 CONTINUE
C
DO 260 I = 1, NCBJ
  J1 = IQZN(I)
  J2 = J1 + NHKVA
  IF(J2 .GT. NH) J2 = J2 - NH
  J3 = J1 + NHPOL

```

```

IF(J3 .GT. NH) J3 = J3 - NH
J4 = 3 * NHKVA + J1
IF(J4 .GT. NH) J4 = J4 - NH

```

```

TE(I) = F K F ( J2, I )
NBP(I) = (SSIR - PTP(J1,I) - PTP(J3,I)) / TE(I) + 1
HB(I) = PTP(J2, I) + PTP(J4, I)
RKMP = KMPL(I)
RNBP = NBP(I)

```

```

253 XI(I) = RKMP / RNBP
ISN(I) = ISN(I) + 1
NKMP(I) = IFIX( NBP(I) * ISN(I) / KMPL(I) )
PRINT 255, I, NI, J1, J2, J3, J4, TE(I), NBP(I), HB(I),
* XI(I), ISN(I), NKMP(I)
255 FORMAT( / 1X, 6I4, F10.5, I5, 2F10.5, 2I5 )
260 CONTINUE

```

```

IZ = 1
NI = NKMP(1)
DO 263 I = 2, NOBJ
IF (NKMP(I) .GE. NI) GO TO 263
NI = NKMP(I)
IZ = I
263 CONTINUE
PRINT 264, IZ, NI
264 FORMAT( ' IZ = ', I5, ' NI = ', I5 )

```

```

IF (NI .NE. J) GO TO 270
I = IZ
GO TO 253

```

```

270 C O N T I N U E

```

```

AREST = SDUZ
DO 265 I = 1, NOBJ
AREST = AREST - HB(I) * ISN(I)
265 CONTINUE

```

```

IF (AREST .GE. C.0) GO TO 370

```

```

DO 330 IM = 1, NOBJ
I = NOBJ + 1 - IM
J0 = IQZ(I)
DO 325 II = 1, NHPOL
J1 = J0 + II
IF ( J1 .GT. NH ) J1 = J1 - NH
J2 = J1 + NHKVA
IF(J2 .GT. NH) J2 = J2 - NH
J3 = J1 + NHPOL
IF(J3 .GT. NH) J3 = J3 - NH
J4 = 3 * NHKVA + J1
IF(J4 .GT. NH) J4 = J4 - NH

PP = PTP(J2, I) + PTP(J4, I)

RAZ = HB(I) - PP
IF (RAZ .LE. C.0) GO TO 325
TEN = F K F ( J2, I )
NEPN = (SSIR - PTP(J1,I) - PTP(J3,I)) / TEN + 1
JJ = IFIX( NEPN * ISN(I) / KMPL(I) )

```

```

                IF( JJ .LT. J ) GO TO 325
                NBP(I) = NPPN
                TE (I) = TEN
                RKMPL = KMPL(I)
                RNBP  = NBP (I)
                XI (I) = RKMPL / RNBP
                IGZN (I) = J1
                HB (I) = PP
                NKMP(I) = IFIX ( NBP(I) * ISN(I) / KMPL(I) )
                IF (NKMP(I) .GE. NI) GO TO 300
                NI = NKMP(I)
                IZ = I
                PRINT 264, IZ, NI
300          CONTINUE
C
                AREST = AREST + RAZ * ISN(I)
                PRINT 255, I, J, J1, J2, J3, J4, TE(I), NBP(I), HB(I), XI(I), ISN(I),
                * NKMP(I)
                PRINT 320, AREST, RAZ
320          FORMAT(2X, 2F15.5)
                IF (AREST .LT. 0.0) GO TO 325
                GO TO 370
325          CONTINUE
330          C O N T I N U E
C
C
                G O T O 400
C
C
370          J = NI
                IZZ = 1
                NIZ = NKMP(I)
                DO 373 I = 2, NOBJ
                IF(NKMP(I) .GE. NIZ) GO TO 373
                IZZ = I
                NIZ = NKMP(I)
373          CONTINUE
C
                IF(NIZ .LE. NI) GO TO 375
                J = NIZ
                IZ = I
C
375          PK = 0.0
C
                DO 380 I = 1, NOBJ
                MIB(I) = KMPL(I) * J
                PK      = PK + S(I) * MIB(I)
                ISL(I) = ISN(I)
                IQZ(I) = IQZN(I)
380          CONTINUE
C
                FC = PK / (SSIP * SDUZ)
                IF(FC .LE. FCMAX) GO TO 385
                FCMAX = FC
                DO 383 II = 1, NOBJ
                IQOP(II) = IQZ(II)
383          CONTINUE
C
385          PRINT 386
386          FORMAT(1X)

```

```

DO 390 I = 1, NOBJ
  TETA = ( IQZ(I)-1 ) * DQS
  PRINT 388, I, TETA, ISL(I), NBP(I), HB(I), TE(I),
    MIB(I), FC, J
388   FORMAT(1X, I5, F8.1, Z17, 2F10.5, I10, F10.5, I5)
390   CONTINUE

```

```

I = IZ
GO TO 253

```

```

400 PRINT 500, FCMAX
500 FORMAT(1H1, ' ISKORIŠTENJE PRI OPTIMALNOM RAZMJEŠTAJU JE', F10.5)

```

```

DO 550 I = 1, NOBJ
  DQSZ = ( ICOP(I)-1 ) * DQS
  THETA = DQSZ * 3.141592654 / 180.0

  N = NVH(I)
  DO 510 II = 1, N
    XA(II) = XOB(II, I)
    YA(II) = YOB(II, I)
510   CONTINUE

  CALL H R O T O R ( XA, YA, N, THETA )

  PRINT 520, I, DQSZ
520   FORMAT( ' KOORDINATE', I3, ' . OBJEKTA ROTIRANOG ZA',
    FE.1, ' STUPNJEVA' )
  PRINT 530, (XA(II), YA(II), II = 1, N)
530   FORMAT(3(2X, '( ', F10.5, ' ', F10.5, ' )' ))
550   CONTINUE
800   CONTINUE

STOP
END

```

SUBROUTINE H R O T O R (XHKE, YHKE, N, THETA)

.....
 * NAMJENA: ODREĐIVANJE KOORDINATA VRHOVA ZAROTIRANOG POLIGONA
 * (HODOGRAF KONTAKT FUNKCIJE - HKF)


```

DIMENSION XHKE(90), YHKE(90)

SIN TA = SIN ( THETA )
COS TA = COS ( THETA )

DO 10 I = 1, N
  X = XHKE ( I )
  Y = YHKE ( I )
  XHKE ( I ) = X * COSTA + Y * SINTA
  YHKE ( I ) = -X * SINTA + Y * COSTA
10 CONTINUE

THST = 180. * THETA / 3.141592654

RETURN
END

```

$X, Y = x, y$

SUBROUTINE KONTIS (S, AT, INA, IOD, JH)

NAMJENA: IZPACUNAVANJE ISKORISTIVOSTI U ZAVISNOSTI OD
KUTA ZAROTIRANOSTI OBJEKATA ZA SLUCAJ SREJSTANJA
U TRAKU KONSTANTNE DULJINE.

COMMON /FOTP/ H(360), NH
• /MORAY/ DGS
• /IERR/ IERR

REAL MI(400), THETA(400)
DIMENSION IOF(1), INA(1)

IERR = 0
NTM = 1
NF = NH/2

DO 60 II = 1, JH
ID1 = INA(II)
ID2 = NF
IF (II .LT. JH) ID2 = INA(II+1) - 1

DO 20 I = ID1, ID2
BTR = H(I) + H(I+NF)
NTM = I + 1
IF (NTM .LE. 400) GO TO 5
P R I N T 3

3 FORMAT(' PDE, GRACENA DIMENZIJA VEKTORA MI(400)')
IERR = 1
R E T U R N

5 THETA(NTM) = DGS * I
MI (NTM) = (S * IOD(II)) / (AT * BTR)
20 C O N T I N U E

60 C O N T I N U E

THETA(1) = 0.0
MI (1) = MI(NTM)

WRITE(10, 70) NTM

70 FORMAT(I4)

WRITE(10, 80) (THETA(I), MI(I), I = 1, NTM)
80 FORMAT(4(F6.1, F10.5))

E N D F I L E 10

R E T U R N
E N D

— * —

SUBROUTINE F U N I J A (S)

NAMJENA: IZRAČUNAVANJE FUNKCIJE ISKORISTIVOSTI U ZAVISNOSTI OD
 PUTA ZAROTIRANOSTI OBJEKTA ZA SLUČAJ JEDNOREDNOG
 RAZMJESTAJA KONGRUENTNIH OBJEKATA U POLUBESKONACNU TRAKU

```

COMMON /FOTP/ H(360), NH
• /KORAK/ DQS
• /HEMA/ QM(360), NT(360), FM12(1000), IFM, NQN
• /LELA/ QL(20), FL1(20), FL2(20), LI
• /IERR/ IERP
  
```

```

REAL MI(400), THETA(400)
IERR = 0
NTH = 1
IFM = 1
ILI = 0
NF = NH/2
  
```

```
DO 20 I = 1, NF
```

```
  NTH = NTH + 1
```

```
  IF(NTH .LE. 400) GO TO 5
```

```
  PRINT 3
```

```
3  FORMAT(' PREKORACENA DIMENZIJA VEKTORA MI(400)')
```

```
  IERR = 1
```

```
  RETURN
```

```
5  IF(LI .EQ. 0 .OR. QL(ILI+1) .GT. QM(I)) GO TO 10
```

```
  ILI = ILI + 1
```

```
  FMIN = FL1(ILI)
```

```
  IF(FL2(ILI) .LT. FMIN) FMIN = FL2(ILI)
```

```
  IF(QL(ILI) .EQ. QM(I)) GO TO 18
```

```
  THETA(NTH) = DQS * I
```

```
  MI (NTH) = S / ((H(I) + H(NF+I)) * FMIN)
```

```
  NTH = NTH + 1
```

```
  IF(NTH .LE. 400) GO TO 10
```

```
  PRINT 3
```

```
  IERR = 1
```

```
  RETURN
```

```
10 FMIN = FM12(IFM)
```

```
  IF(NT(I) .EQ. 1) GO TO 18
```

```
  NTDO = NT(I)
```

```
  DO 15 J = 2, NTDO
```

```
    JJ = IFM + J
```

```
    IF(FMIN .LE. FM12(JJ)) GO TO 15
```

```
    FMIN = FM12(JJ)
```

```
15  CONTINUE
```

```
18  IFM = IFM + NT(I)
```

```
  THETA(NTH) = DQS * I
```

```
  MI (NTH) = S / ((H(I) + H(NF+I)) * FMIN)
```

```
20  CONTINUE
```

```
THETA(I) = 0.0
```

```
MI (I) = MI(NTH)
```

```

      PRINT 25, NTH
25  FORMAT('1 KOORDINATE FUNKCIJE ISKORISTIVOSTI OD', IS, ' TACAKA
      ' FORMA ISFISA JE (KUT I ISKORISTIVOST')
      PRINT 26, (THETA(I), NI(I), I = 1, NTH)
26  FORMAT(5(2X,'I',F6.1,' ',F10.5,' '))

      UPIS NA DISK DSI = 10
      WRITE (10, 30) NTH
30  FORMAT(14)

      WRITE(10, 40) (THETA(I), NI(I), I = 1, NTH)
40  FORMAT(4(F6.1, F10.5))

      END FILE 10

      RETURN
      END

```

*
-----*

SUBROUTINE L O K M I N I F, N, IND, NI, MINEL)

.....

• NAMIJENA: ODREĐIVANJE INDEKSA MINIMALNE VRIJEDNOSTI FUNKCIJE
 • RAZDJELENE PO INTERVALIMA A ZADANE U VIDU VEKTORA
 • NJENIH VRIJEDNOSTI I VEKTORA PODJELE NA INTERVALE

.....

```

      DIMENSION F(360), IND(50), MINEL(50)

      DO 10 I = 1, NI
        IB = IND(I)
        IT = N
        IF ( I .LT. NI ) IT = IND(I+1) - I
        FMIN = F( IB )
        MINEL(I) = IB
        IB = IB + 1
        IF ( IB .GT. IT ) GO TO 10

        DO 5 J = IB, IT
          IF ( F(J) .GE. FMIN ) GO TO 5
          FMIN = F( J )
          MINEL(I) = J
        5 CONTINUE
      10 CONTINUE

```

```

      PRINT 15
15  FORMAT(///5X,'INDENS',10X,'FUNKCIJA')

      DO 20 I = 1, NI
        J = MINEL(I)
        PRINT 25, J, F(J)
20  CONTINUE
25  FORMAT(5X,14,10X,F10.5)

      RETURN
      END

```

*
-----*

SUBROUTINE EXTREM

NAMJENA: ODREĐIVANJE EKSTREMALNIH TACA NA NEPREKIDUTE
JEDNOZNAČNO DEFINIRANE FUNKCIJE.

```

COMMON /FEXT/ F(1000), N
      /INEXT/ INEX(50), K
      /IERR/ IERR
IERR = 0
ISIGN = 0
NH1 = 0
K = 0
NH1 = N - 1
PRINT 65
DO 50 II = 1, NH1
  DIF = F(II+1) - F(II)
  IF(DIF) 10, 50, 30
10  IF(ISIGN,LT,0) GO TO 45
  IF(ISIGN,GT,0) GO TO 15
  IF(II,NE,1) GO TO 15
  ISIGN = -1
  GO TO 45
15  ISIGN = -1
  GO TO 40

30  IF(ISIGN,GT,0) GO TO 45
  IF(ISIGN,LT,0) GO TO 35
  IF(II,NE,1) GO TO 35
  ISIGN = 1
  GO TO 45

35  ISIGN = 1

40  NH2 = II + 1
  K = K + 1
  IF(K,GT,50) GO TO 60
  INEX(K) = (NH1 + NH2) / 2
  PRINT 72, II, DIF, NH1, NH2, ISIGN, K, INEX(K)
  NH1 = NH2 - 1
  GO TO 50

45  NH1 = II
50  CONTINUE
  RETURN

60  PRINT 70
  IERR = 1
  RETURN

65  FORMAT(/5X,'VRIJEDNOSTI VARIJABLI PODPROGRAMA EXTREM'
      /5X,'II',2X,'DIFERENCIJA',2X,'NH1',5X,'NH2',5X,'ISIGN',
      5X,'K',4X,'INEX(K)'/)
70  FORMAT(5X,'NADJEHO VISE OD 50 EXTREMA - STO JE OGRANICENJE')
72  FORMAT(5X,12,F10.5,3I8,3X,16,I8)
END

```

— * —

```
SUBROUTINE MULFUN(X, INX, NX, Y, INY, NY, Z, INZ, NZ)
```

```

C .....
C • NAMJENA: MNOZENJE DVIJU FUNKCIJA KOJE SU NA DEFINIRANOM INTERVALU
C • PO DJELOVIMA KONSTANTNE
C .....
C
  INTEGER X(50), Y(50), Z(100)
  DIMENSION INX(50), INY(50), INZ(100)
  COMMON /IERR/ IERR
  IERR = 0

C
  DO 1 I = 1, 100
    Z(I) = 0.0
    INZ(I) = 0
  1 CONTINUE

C
  KRAJ = NX + 1
  IF (INY(NY), LT, INX(NX)) KRAJ = NY + 1

C
  JX = 1
  JY = 1
  JZ = 0
  5 IF (JX, EQ, KRAJ, OR, JY, EQ, KPAJ) GO TO 50
  JZ = JZ + 1
  IF (JZ, GT, 100) GO TO 100
  IF (INX(JX), EQ, INY(JY)) GO TO 10
  IF (INX(JX), GT, INY(JY)) GO TO 30

C
  Z(JZ) = X(JX) * Y(JY-1)
  INZ(JZ) = INX(JX)
  IF (JX, LE, NX) JX = JX + 1
  GO TO 5

C
  10 Z(JZ) = X(JY) * Y(JY)
  INZ(JZ) = INX(JX)
  IF (JX, LE, NX) JX = JX + 1

C
  IF (JY, LE, NY) JY = JY + 1
  GO TO 5

C
  30 Z(JZ) = X(JX-1) * Y(JY)
  INZ(JZ) = INY(JY)
  IF (JY, LE, NY) JY = JY + 1
  GO TO 5

  50 NZ = JZ
  PRINT 51, (INZ(I), Z(I), I = 1, NZ)
  51 FORMAT(/5X, 'PRODUKT FUNKCIJA', 10(15X, 2I10))
  RETURN

C
  •• GRESKA ZBOG PREGORAČENA DIMENZIJA MATRICE Z I INZ ••
  100 IERR = 1
  PRINT 101
  101 FORMAT(/5X, 'FUNKCIJA Z IMA VIŠE OD 100 KONSTANTNIH INTERVALA')
  RETURN
  END

```

———— * ————

SUBROUTINE NPOLIG(PT,NOI,IOB,N)

• NAMJENA: OPREDJIVANJE FUNKCIONALNE ZAVISNOSTI BROJA OBJEKATA
 • NOB(N) (POLIGONA) OD KUTA POD KOJIM OBJEKAT LEZI U
 • ODNOSU NA X-OSI GLEDAJUĆI U SMJERU KAZALJKE NA SATU.

COMMON /POTP/ H(360), NH
 • /HEMA/ CH(360), HT(360), FN12(1000), KI, NI
 COMMON /IERR/ IERR

DIMENSION NOB(50), IOB(50)

IERR = 0
 NIPOL = NH / 2
 NHKVA = NH / 4
 ILO = 0
 N = 0

PRINT IO
 IC FORMAT(/5X,'POTPROGRAM NPOLIG DAJE:')
 • /5X,'INTERVAL INDEKS-KUTA BR.OBJ/TRACI PERIOD')

DO 50 I = 1, NIPOL
 JQ = NH - NHKVA + I
 IF (JQ .GT. NH) JQ = JQ - NH
 JP = NHKVA + I
 IL = (BT - (P(JQ) + P(JP))) / FN12(I) + 1
 IF (IL .EQ. ILO) GO TO 50
 N = N + 1
 IF (N .GT. 50) GO TO 100
 NOB(N) = IL
 IOB(N) = I
 ILO = IL

PRINT 40, N, IOB(N), NOB(N), FN12(I)

40 FORMAT(5X, I5, 2(7X, I5), F15.5)

50 CONTINUE
 RETURN

100 PRINT IOI
 IOI FORMAT(/5X,'ODBAČEN ZAHITJEV ZA SNJEŠTANJE VIŠE OD 50 OBJEKATA')
 RETURN
 END

— * —

Univerzitet u Beogradu
 Prirodno-matematički fakultet
 MATEMATIČKI FAKULTET
 BIBLIOTEKA

Broj _____ Datum _____

SUBROUTINE NTRAKA(AT,NTR,ITR,N)

C
 C * NAMJENA: ODREĐJIVANJE FUNKCIONALNE ZAVISNOSTI BROJA TRAKA NTR(
 C * OD KUTA POD KOJIM OBJEKAT LEZI U ODNOSU NA X-OSU
 C * GLEDAJUĆI U SMJERU KAZALJKE NA SATU.
 C
 C

COMMON /POTP/ H(360), NH
 * /IERR/ IERR

DIMENSION NTR(50), ITR(50)

IERR = 0

NHPOL = NH / 2

ILD = 0

N = 0

P R I N T 55

DO 50 I = 1, NHPOL

SIT = H(I) + H(NHPOL + 1)

IL = AT / SIT

IF (IL, EQ, ILD) GO TO 50

N = N + 1

IF (N, GT, 50) GO TO 100

NTR (N) = IL

ITR (N) = I

ILD = IL

C
 P R I N T 65, N, ITR(N), NTR(N), SIT

50 CONTINUE
 RETURN

C
 C
 55 FORMAT(/5X, 'POTPROGRAM N T R A K A D A J E : '
 * /5X, 'INTERVAL INDEKS-KUTA BROJ-TRAKA SIRINA-TRAKE')
 65 FORMAT(5X, I5, 2(9X, I5), F15, 5)

C
 100 P R I N T 101
 101 FORMAT(/5X, 'ODEBAČEN ZAHTJEV ZA PODJELOM NA VIŠE OD 50 TRAKA')
 IERR = 1
 RETURN
 END

— * —

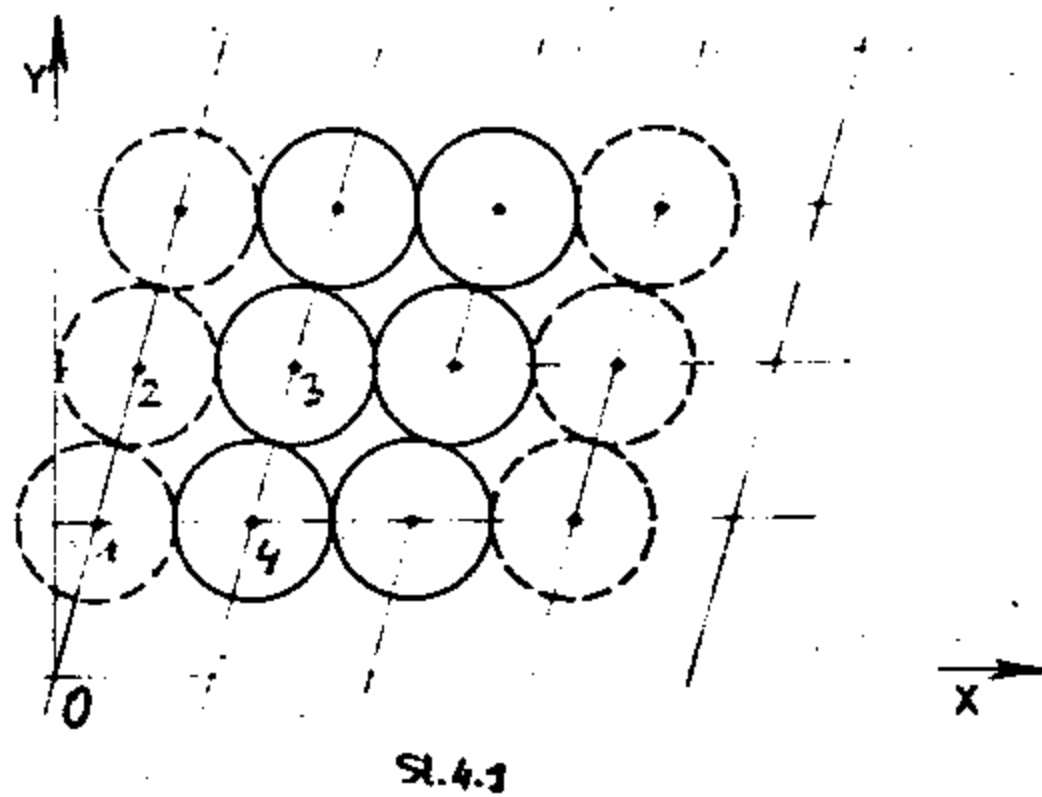
G L A V A I V

REŠETKASTI RASPORED OBJEKATA

1. Formulacija problema i smještanje u ravnini

Neka je zadan skup ograničenih kongruentnih objekata $S \in \{S\}$ proizvoljna oblika koje treba smjestiti rešetkastno u ravnini. Uzimimo dvije familije pravaca P_1 i P_2 .

Pravci $p_i \in P_1$ neka su međusobno paralelni i jednako udaljeni a $p_j \in P_2$ također međusobno paralelni i jednako udaljeni, a sa p_i zatvaraju neki kut. Pravci p_i i p_j čine rešetku W u ravnini, sl.4.1.



Ako u vrhove rešetke smjestimo polove međusobno orijentiranih objekata $S \in \{S\}$ tako da objekti mogu imati samo rubne točke zajedničke, onda takav smještaj objekata zovemo REŠETKASTI SMJEŠTAJ.

U vezi s tim postavlja se problem:

Objekte $S \in \{S\}$ smjestiti u R^2 tako da koeficijent popunjenosti bude maksimalan.

Vidimo da O_1, O_4, O_3, O_2 predstavlja paralelogram R i da je ravnina prekrivena jednakim paralelogramima bez rupa i preklapanja. Pored toga vidimo da se u jedan paralelogram smjesti površina upravo jednog objekta. Prema tome je očito da je funkcija cilja, postići što manju površinu paralelograma.

Tvrđnja 4.1.

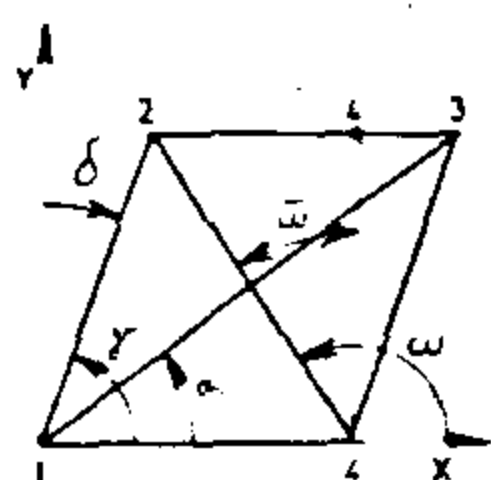
Ako za proizvoljne objekte S_i, S_j $i \neq j$, kod fiksnog kuta zarc tiranosti θ objekata, vrijedi

$$S_i \setminus s_i \cap S_j \setminus s_j = \emptyset, \quad (4.1.1)$$

a površina paralelograma $P(R)$ je minimalna, tada je koeficijen-
popunjenosti maksimalan, i dan je izrazom

$$\mu = \frac{P(S)}{P(R)} \quad (4.1.2)$$

Koji se objekti diraju a koji se ne diraju možemo opisati pomo-
ću teorije grafova, sl.4.2. [39]. Uvodjenjem tzv. matrice di-
ranja imamo i matični zapis situacija. Situaciji na sl.4.1. od-
govara matrica diranja A:

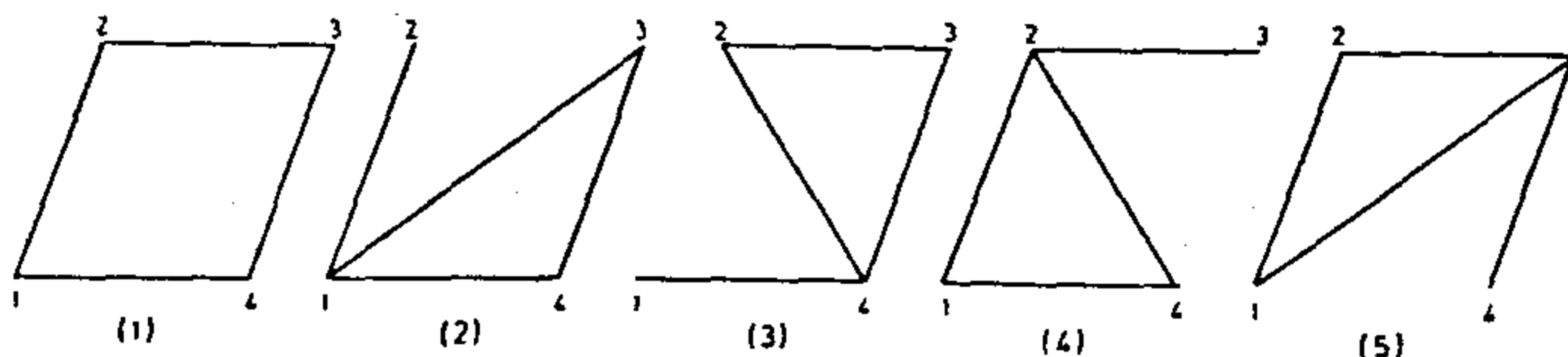


Sl.4.2

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Tvrđnja 4.2.

Ako je θ -fiksna a ispunjeni su uvjeti (4.1.1), onda se $\min P(R)$
realizira kada imamo bilo koju situaciju predstavljenu na jed-
nom od slijedećih pet grafova, sl.4.3, odnosno pripadnih pet ma-
trica diranja A_i ($i = 1,5$).



Sl.4.3

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}; A_2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}; A_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.1.3)$$

$$A_4 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; A_5 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Navedena tvrdnja slijedi iz uvjeta nepresjecanja (4.1.1), načina formiranja rešetke i nasuprotnih stranica paralelograma. Min $P(R)$ će se postići tada kada se dira maksimalni broj objekata.

Tvrdnja 4.3.

Minimum površine paralelograma R moguće je odrediti iz bilo koje od situacija na sl.4.3.

Pretpostavimo da imamo slučaj kome odgovara graf Γ) odnosno matrica A_1 (vidi sl.4.3).

Pošto su $S_i, S_j \in \{S\}$ kongruentni i međusobno orijentirani, za HKF- g_{ij} i g_{ji} vrijedi $g_{ij} = g_{ji} \quad (\forall_{i,j} \quad i \neq j)$ a također za kontakt funkcije vrijedi

$$K_{ij}(\theta) = K_{ji}(\theta); \quad \forall_{i,j}; \quad i \neq j.$$

Na osnovi svojstava HKF i KF za objekte prednjih osobina, a s obzirom na naprijed izrečene tvrdnje, imamo slijedeću tvrdnju:

Tvrdnja 4.4.

Minimum $P(R)$ se postiže kad vrijedi slijedeći sistem nejednazi - uvjeta nepresjecanja:

$$\begin{aligned} \rho_{14}(O_1, O_4) - K(\theta) &= 0 \\ \rho_{12}(O_1, O_2) - K(\theta + \gamma) &= 0 \\ \rho_{13}(O_1, O_3) - K(\theta + \alpha) &\geq 0 \\ \rho_{43}(O_4, O_2) - K(\theta + \omega) &\geq 0 \end{aligned} \quad (4.1.4)$$

gdje su:

$$0 \leq \theta \leq \pi \quad 0 < \gamma, \omega < \pi$$

$$\gamma = \angle O_2, O_1, O_4; \quad \omega = \angle O'_2, O'_4, O'_4; \quad \alpha = \angle O_3, O_1, O_4$$

$$\alpha = \arctg \frac{K(\theta + \gamma) \sin \gamma}{K(\theta) + K(\theta + \gamma) \cos \gamma}$$

Iz prve dvije jednačbe sistema (4.1.4) slijedi da vrhovi paralelograma O_2 i O_4 uvijek leže na HKF - g_{ij} objekata S_i i S_j . Prema tome površina paralelograma čiji se minimum traži, tj. koju predstavlja funkciju cilja dana je izrazom:

$$P[R(\theta, \gamma)] = K(\theta) \times K(\theta + \gamma) \cdot \sin \gamma; \quad \theta, \gamma \in \Gamma \quad (4.1.5)$$

Oblast Γ je definirana sa (4.1.4)

Tvrđnja 4.5.

$$\gamma_1 \leq \gamma \leq \gamma_2; \quad \gamma_1 = \gamma(\theta); \quad \gamma_2 = \gamma_2(\theta)$$

$\gamma_2(\theta)$ je dan izrazom

$$\gamma_2 = \pi - \arctg \frac{K(\theta + \gamma_1) \sin \gamma_1}{K(\theta) - K(\theta + \gamma_1) \cos \gamma_1} \quad (4.1.6)$$

a γ_1 se dobije kao korijen jednačbe

$$K(\theta + \gamma) \sin \gamma - K(\theta + \omega) \cdot \sin \omega = 0 \quad (4.1.7)$$

gdje je:

$$\omega = \pi - \arctg \frac{K(\theta + \gamma) \cdot \sin \gamma}{K(\theta) - K(\theta + \gamma) \cdot \cos \gamma} \quad (4.1.7)$$

Ako konstruiramo HKF s polom u O_1 a zatim s polom u O_4 i nađemo točku njihovog presjeka, vidjet ćemo da je to točka O_2 . Vektor $\vec{O_1 O_2}$ upravo čini kut $\gamma_1(\theta)$. Kut $\gamma_2(\theta)$ je najveći mogući kut γ za zadanu orijentaciju θ objekata a da egzistira rešetka za koju su ispunjeni uvjeti nepresjecanja objekata (4.1.4).

Neka svojstva funkcija $\gamma_1(\theta)$ i $\gamma_2(\theta)$ [40]

1. $\gamma_1(\theta) = \gamma_1(\theta \pm n\pi)$; $\gamma_2(\theta) = \gamma_2(\theta \pm n\pi)$; $n = 1, 2, \dots$
2. Ako su $S \in \{S\}$ konveksni, onda je $\gamma_1(\theta)$ jednoznačna, ^{u ravnini $\gamma_1 \geq 0$} inače je višeznačna i ima toliko grana koliko KF ima točaka infleksije.
3. Općenito se $\gamma_1(\theta)$ i $\gamma_2(\theta)$ ne daju eksplicite izraziti.
4. $\gamma_1(\theta), \gamma_2(\theta) \in C[0, \pi]$.
5. Ako je kontakt funkcija $K(\theta) \in C[0, \pi]$, onda je svaka grana $\gamma_1^i(\theta)$; $i = 1, \dots, v$ po dijelovima glatka ^{u oblasti egzistencije}.

Tvrđnja 4.7.

U ravnini postoje oblasti Z_j ($j = 2, 4, \dots, n-2$) u kojima nije ispunjena četvrta nejednadžba u (4.1.4). Oblasti Z_j su određene sistemom nejednadžbi [40].

$$\begin{aligned} \gamma - \gamma_1^j(\theta) &\geq 0 \\ -\gamma + \gamma_1^{j+1}(\theta) &\geq 0 \quad j = 2, 4, \dots, v-2 \end{aligned}$$

uz uvjet da je $\gamma_1^1(\theta) < \gamma_1^2(\theta) < \dots < \gamma_1^v(\theta)$. U tim oblastima $P[R(\theta, \gamma)]$ ne postiže ekstreme. Naime, imamo situaciju da je za $\gamma = \gamma_1^2(\theta) - \gamma_1^1(\theta)$ četvrta jednadžba u (4.1.4) ispunjena, tj. 0_2 leži na HKF, dok za $\gamma = \gamma_1^3(\theta) - \gamma_1^2(\theta)$ nije ispunjena, jer 0_2 leži unutar oblasti G_{ij} , čija je granica HKF, s polom u 0_4 .

Iz izloženog slijedi da se problem optimalnog smještaja kongruentnih objekata u ravnini svodi na određivanje:

$$P^*[R(\theta, \gamma)] = \min_{\theta, \gamma \in \Gamma} P[R(\theta, \gamma)] = \min_{\theta, \gamma \in \Gamma} [K(\theta) \times K(\theta + \gamma) \sin \gamma] \quad (4.1.9)$$

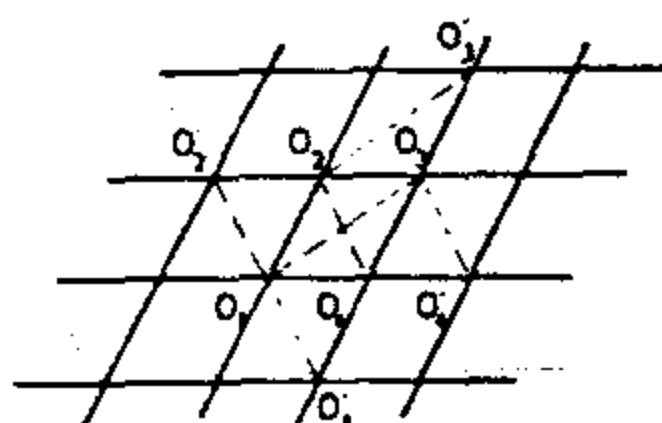
$$\Gamma = D \setminus \bigcup_{j=2}^{n-1} Z_j; \quad D - \text{ je oblast određena sa nejednadžbama}$$

$$\begin{aligned} -\gamma + \gamma_2(\theta) &\geq 0 \\ \gamma_j - \gamma_1(\theta) &\geq 0 \\ \pi - \theta &\geq 0 \\ \theta &\geq 0 \end{aligned} \quad (4.1.10)$$

Iz poznatih svojstava KF možemo zaključiti da je, općenito uzeši, $P^*[R(\theta, \gamma)]$ mnogoekstremalna i da su gradientne metode gotovo neupotrebljive.

Tvrđnja 4.8.

Za različite $\theta, \gamma \in [0, \pi]$ funkcija $P[R(\theta, \gamma)]$ ima bar pet jednakih vrijednosti. Odgovarajuće paralelograme vidimo na sl.4.4.



Sl.4.4

To su paralelogrami:

$$O_1O_2O_3O_4; O_1O_3O_3' O_2; O_2O_2' O_1O_4; O_2O_3O_4' O_4; O_1O_1' O_4O_2.$$

Iz toga slijedi vrlo važna činjenica, a to je da pomoću izračunatih parametara osnovnog paralelograma $O_1O_2O_3O_4$ kome odgovara optimalni smještaj možemo odmah izračunati parametre za bar još četiri optimalna smještaja, što bitno smanjuje vrijeme rješavanja problema.

Kao što vidimo iz (4.4.9) kod računanja površine paralelograma radimo samo sa pripadnom KF. Međutim, kada se radi o konkavnim objektima, onda se ponekad pokazuje da je zgodnije direktno raditi s jednažbama HKF danim sa $F(x'_1, y') = 0$.

Naime, ako ishodište fiksnog koordinatnog sistema smjestimo u pol O_1 , a os OX se podudara sa stranicom rešetke kroz O_1 i O_4 tada je, za proizvoljni θ jednažba HKF s polom u O_1 , dana izrazom

$$F(x \cos \theta + y \sin \theta; -x \sin \theta + y \cos \theta) = 0, \quad (4.1.11)$$

a jednažba HKF s polom u O_4 je dana sa:

$$F[(x-x^*) \cos \theta + y \sin \theta; (-x+x^*) \sin \theta + y \cos \theta] = 0. \quad (4.1.12)$$

Duljina stranice O_1O_4 paralelograma R jednaka je $x^*(\theta) = K(\theta)$, a dobije se tako da u (4.1.11) stavimo $y = 0$, tj. imamo

$$F(x\cos\theta, -x\sin\theta) = 0 \quad (4.1.13)$$

Ordinatu $y_2 = y^*(\theta)$ dobijemo tako da (4.1.11) i (4.1.12) riješimo kao sistem, tj. da nađemo presjek HKF s polom u O_1 i HKF s polom u O_4 . Znači da se problem svodi na određivanje

$$P^*[R(\theta, \gamma)] = \text{Min} x^*(\theta) \cdot y^*(\theta) \quad \theta \in [0, \pi].$$

Tako za slučaj elipse s osima a, b imamo da je

$$x^*(\theta) = \frac{+}{-} \frac{2ab}{[b^2 \cos^2 \theta + a^2 \sin^2 \theta]^{1/2}}$$

$$y^*(\theta) = \frac{+}{-} \frac{\sqrt{3}}{[b^2 \cos^2 \theta + a^2 \sin^2 \theta]^{1/2}}$$

pa je $P^*[R(\theta, \gamma)] = 2 \cdot \sqrt{3}ab$.

Odnosno koeficijent popunjenosti je dan sa:

$$\mu = \frac{\pi}{2\sqrt{3}} = 0,9069\dots(*)$$

i ne zavisi od kuta θ kao i od veličine osi a i b .

Specijalno za $a=b=r$ imamo koeficijent popunjenosti za krugove

$$\mu = \frac{\pi}{2\sqrt{3}} = 0,9069\dots(**)$$

Tvrđnja-4.9.

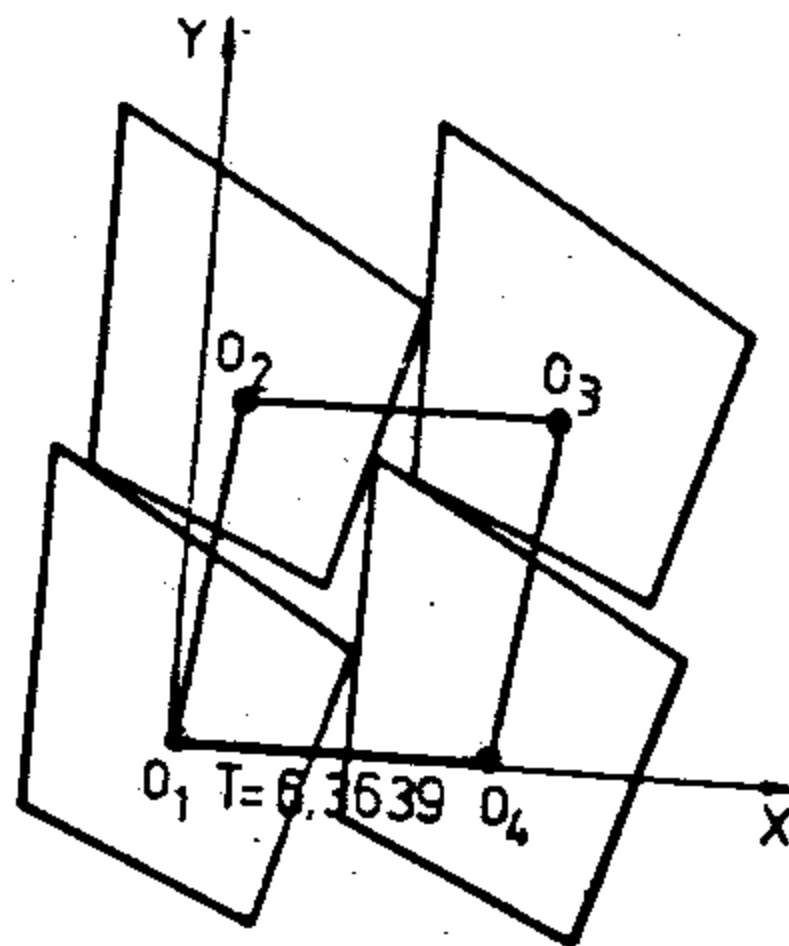
Ako su objekti $\{S\}$ konveksni, problem se svodi na određivanje:

$$P^*[R(\theta, \gamma)] = \text{Min} P[R(\theta, \gamma(\theta))] \quad \theta \in [0, \pi].$$

Za krugove i elipse je $\gamma = 60^\circ$, tj. imamo tzv. sačasto smještanje. U [42] i [51] dobiveni su isti rezultati kao i (*) (**)
i to metodama diskretne (kombinatorne) matematike.

U [9*] je dokazano, također metodama diskretne matematike, da za svaku konveksnu figuru smještenu rešetkasto u R^2 vrijedi $\mu(S) > 2/3$, a znak jednakosti vrijedi za trokut. Naime, trokut je najnepovoljnija figura kod rešetkastog smještanja.

Prema tome rezultati [9] [42] i [51] u smislu izračunavanja koeficienta iskorištenja, specijalni su slučaj naših rezultata, služili su nam za testiranje programa na točnost i efikasnost. Na osnovi ovih razmatranja i nekih drugih činjenica koje će biti objašnjene u slijedećem paragrafu, vršeni su razni pokušaji i eksperimenti dok se nije uspjelo složiti takav algoritam, a po njemu i program za računar koji je dao zadovoljavajuće rezultate. Demonstraciju paketa programa 'RERAR' vidimo na rešetkastom smještaju objekta $S_3: A_1(2,-6); A_2(-4,1); A_3(1,4); A_4(3,-1)$ Parametri smještaja su: $O_1(0,0); O_2(0,77609; 6,63991), O_4(6,36396,0) \theta = 136^\circ$. Koeficijent iskorištenja $\mu = 0,89928$, sl.4.5.

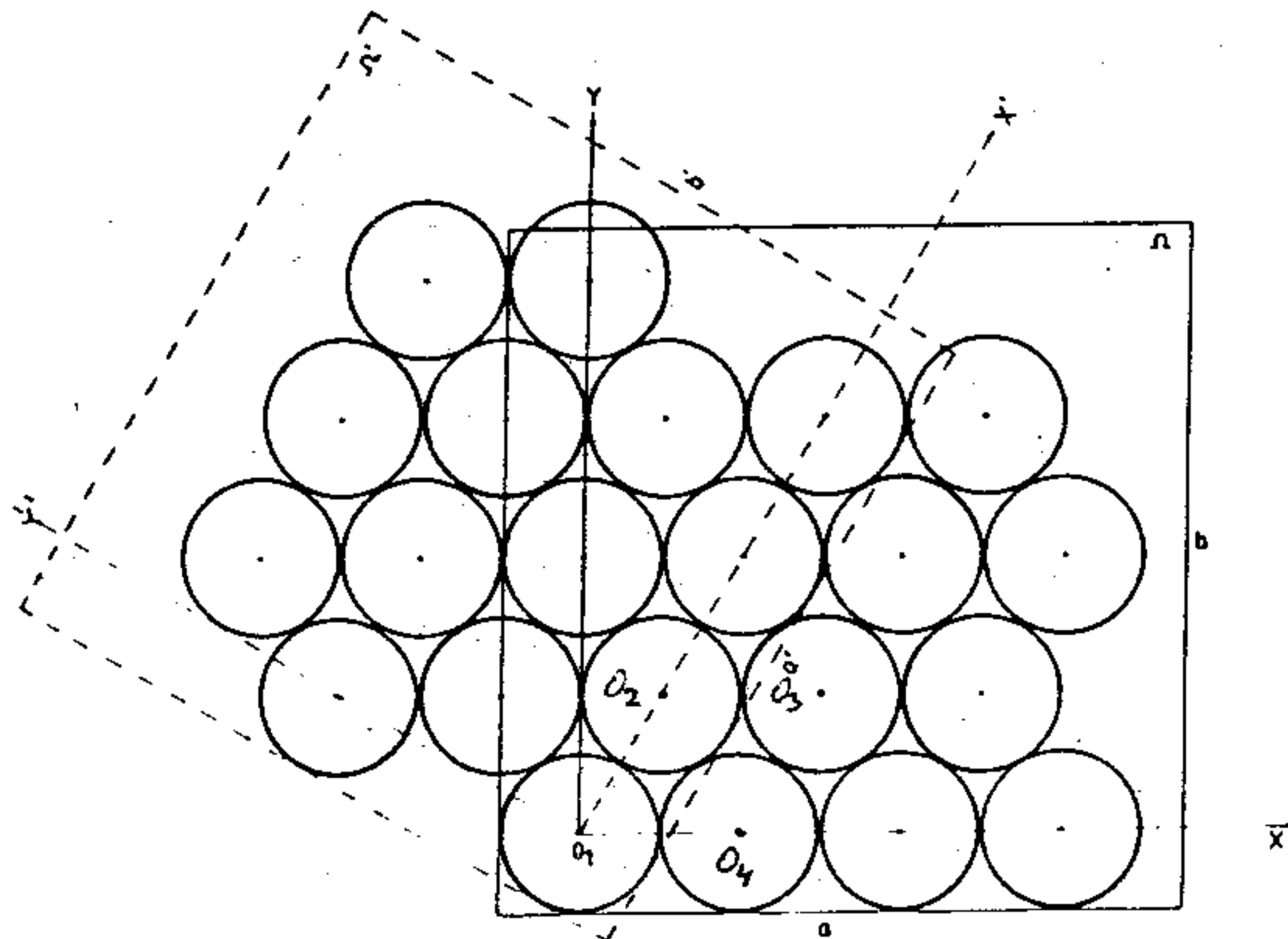


SLIKA 4.5

2. Rešetkasto smještanje u pravokutnik

Neka je zadan pravokutnik $\Omega = (axb)$ u koji treba rešetkasto smjestiti maksimalni broj objekata $S \in \{S\}$.

Objekti su međusobno orijentirani, tj. zakrenuti za isti kut θ . Pravokutnik Ω smjesimo tako da stranica a i b budu paralelne koordinatama OX (OY) fiksnog sistema koordinata, sl.4.6.



SLIKA 4.6

Kako je pol O_1 smješten u ishodište koordinatnog sistema a stranica rešetke W je izabrana tako da je paralelna sa OX , onda su stranice a i b (donji, odnosno lijevi) potporni pravci objekta S_1 .

Neka su $O_1 O_2 O_3 O_4$ vrhovi osnovnog paralelograma R .

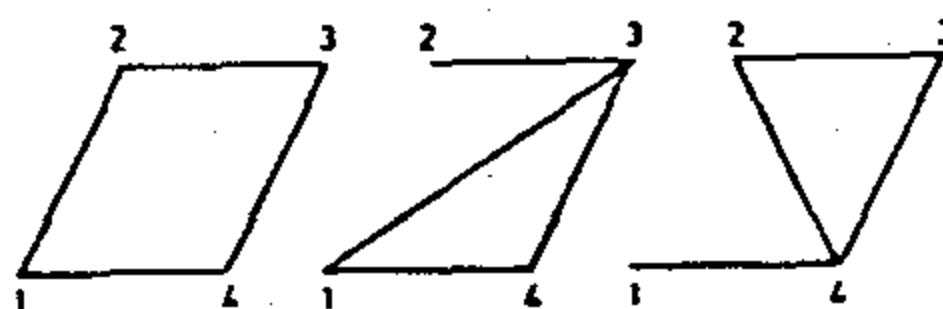
Neka je $\delta = \angle YO_1 O_2$; $\bar{\omega} = \angle O_2 O_4 O_4'$, $\alpha = \angle O_3 O_1 O_4$, $\gamma = \angle O_2 O_1 O_4$.

Kako su S_1 i S_2 u kontaktu, to je $\rho(O_1, O_4) = K(\theta)$.

Koordinate polova objekata $S_i \in \{S\}$ $i = (1, 4)$ dane su izrazima

$$\begin{aligned} x_1 &= 0; & x_2 &= \frac{K(\theta)}{\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}} \operatorname{tg}\delta & x_3 &= x_2 + K(\theta); & x_4 &= K(\theta) \\ y_1 &= 0; & y_2 &= \frac{K(\theta)}{\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}}; & y_3 &= y_2; & y_4 &= 0. \end{aligned} \quad (4.2.1)$$

Neka za objekte smještene u R^2 vrijedi jedan od slijedeća tri grafa sl.4.7;



Sl.4.7

Jednostavnosti radi uzimimo da je

$$\frac{\max p(\theta)}{\min K(\theta)} = C \leq 1, \quad \text{tada je}$$

uvjet nepresjecanja za sve $\{S\} \in \Omega$ dan sistemom:

$$\begin{aligned} K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \cos\delta \cdot k(\theta + \gamma) &\geq 0 \\ K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \sin\alpha \cdot K(\theta + \alpha) &\geq 0 \\ x_4 - K(\theta) &\geq 0 \end{aligned} \quad (4.2.2)$$

No kako je u sva tri slučaja $\rho(S_1, S_2) = 0$, to je $x_4 = K(\theta)$ i (4.2.2) možemo predstaviti sa tri nejednadžbe:

$$\begin{aligned} K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \cos\delta \cdot k(\theta + \gamma) &\geq 0 \\ K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \sin\alpha \cdot k(\theta + \alpha) &\geq 0 \\ K(\theta) - (\operatorname{tg}\gamma + \operatorname{tg}\bar{\omega}) \cos\bar{\omega} \cdot K(\theta + \omega) &\geq 0 \end{aligned} \quad (4.2.3)$$

Bar jedna nejednadžba u (4.2.3) treba da bude ispunjena kao stroga jednakost.

Ako je $N \left[\frac{\max p(\theta)}{\min K(\theta)} \right] + 1 = C_1 > 1$ onda (4.2.3) prelazi u sistem:

$$\begin{aligned} K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \cos\delta \cdot k(\theta + \gamma) &\geq 0 \\ K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \sin\delta \cdot K(\theta + \alpha) &\geq 0 \\ K(\theta) - (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) \cos\bar{\omega}_{1i} \cdot K(\theta + \omega_{1i}) &\geq 0; \quad (i=1, q) \end{aligned} \quad (4.2.4)$$

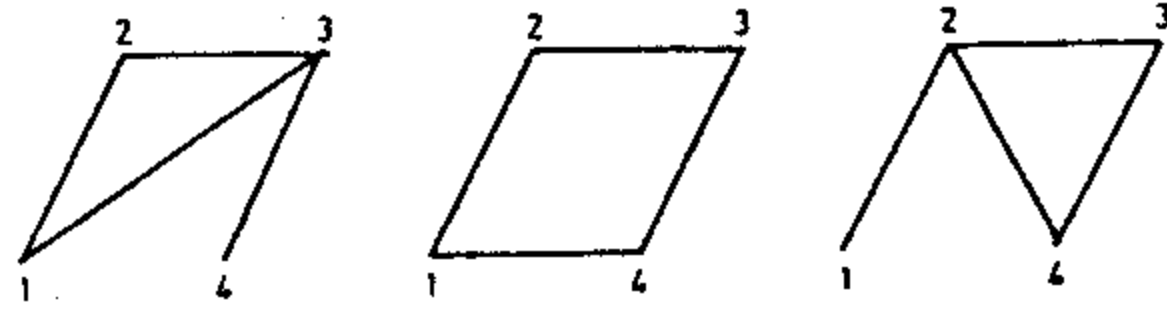
gdje je:

$$\bar{\omega}_{11} = \bar{\omega}, \quad \bar{\omega}_{1i} = \operatorname{arc} \operatorname{tg} \frac{iK(\theta) - x_2}{y_2}; \quad (i=2, q),$$

odnosno:

$$\bar{\omega}_{1i} = \operatorname{arc} \operatorname{tg} [i(\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) - \operatorname{tg}\delta].$$

Ako su objekti smješteni optimalno, onda osim situacija koje odgovaraju grafovima na sl.4.7. imamo još tri situacije kojima odgovaraju grafovi na sl.4.8.



Sl.4.8

Kao što vidimo na sl.4.8. u sva tri slučaja O_2 leže na HKF, tj. $\rho(O_1, O_2) = K(\theta + \frac{\pi}{2} - \delta)$, pa se koordinate (x_2, y_2) mogu izraziti sa

$$x_2 = K(\theta + \frac{\pi}{2} - \delta) \sin \delta \quad (4.2.5)$$

$$y_2 = K(\theta + \frac{\pi}{2} - \delta) \cos \delta$$

Izaberemo koordinatni sistem $X'O_1Y'$ tako da os O_1X' prolazi kroz polove O_1 i O_2 paralelograma R rešetke, sl.4.6. Sada su opet stranice Ω' potporni pravci objekta S_1 smještenog u O_1 . U ovom novom sistemu koordinata $X'O_1Y'$ za situacije predstavljene na sl.4.8, uvjeti međusobnog nepresjecanja objekata dani su također sa (4.2.4).

Funkcija \mathcal{B} u našem problemu predstavlja broj objekata $\{S\} \in \Omega$, s tim da se objekti ne presjecaju, tj. da je odnos objekata predstavljen sa grafovima na sl.4.7. i sl.4.8. i da vrijedi (4.2.5).

Analitički izraz za broj objekata je slijedeći:

$$\mathcal{B}(\theta, \delta, \bar{\omega}) = \mathcal{B}_1(\theta, \delta, \bar{\omega}) \vee_1 \mathcal{B}_2(\theta, \delta, \bar{\omega}) \quad (4.2.6)$$

gdje je:

$$\mathcal{B}_1(\theta, \delta, \bar{\omega}) = \sum_{r=0}^{M_1} \{N_{1r} \left[\frac{a - [p(\theta + \frac{\pi}{2}) + p(\theta + \frac{3\pi}{2})] + K(\theta)}{K(\theta)} - r \frac{\operatorname{tg} \delta}{\operatorname{tg} \delta + \operatorname{tg} \bar{\omega}} \right] + N_{1r} \left[r \cdot \frac{\operatorname{tg} \delta}{\operatorname{tg} \delta + \operatorname{tg} \bar{\omega}} \right] \} \quad (4.2.7)$$

$$\mathcal{B}_2(\theta, \delta, \bar{\omega}) = \sum_{r=0}^{M_2} \left[N_{1r} \frac{[a - [p(\theta + \pi - \delta) + p(\theta - \delta)] \cdot (\operatorname{tg} \delta + \operatorname{tg} \bar{\omega}) \cos \delta + K(\theta)]}{K(\theta)} \right] + \\ + N_{2r} \left[\frac{r}{2} \sin 2\delta (\operatorname{tg} \delta + \operatorname{tg} \bar{\omega}) \right] \quad (4.2.8)$$

Vrijednosti M_1 odnosno M_2 su dane izrazima:

$$M_1(\theta, \delta, \bar{\omega}) = N_1 \left[\frac{b - [p(\theta) + p(\theta + \pi)] \cdot (\operatorname{tg} \delta + \operatorname{tg} \bar{\omega})}{K(\theta)} \right] \quad (4.2.9)$$

$$M_2(\theta, \delta, \bar{\omega}) = N_2 \left[\frac{b - p(\theta + \frac{\pi}{2} - \delta) + p(\theta + \frac{3\pi}{2} - \delta)}{K(\theta + \frac{\pi}{2} - \delta) \cos \delta} \right] \quad (4.2.10)$$

gdje je: V_1 - oznaka za R-disjunkciju.

Znači da se problem sveo na to da se nađe

$$\max \mathcal{B}(\theta, \delta, \bar{\omega})$$

Iz prednjeg slijedi da se izraz 4.1.9 za ravninu može napisati u obliku

$$P^*[R(\theta, \delta)] = \frac{[K(\theta)]^2}{\operatorname{tg} \delta + \operatorname{tg} \bar{\omega}}. \quad (4.2.12)$$

Kod izbora algoritma za rešetkasto smještanje u ravnini kao i pravokutniku Ω koriste se neke činjenice koje su za oba problema zajedničke:

Činjenica 1.

Ako su $K(\theta)$, $p(\theta) \in C_{[0, \pi]}$, njih možemo s proizvoljnom, unaprijed zadanom točnošću, zamijeniti sa $K_\epsilon(\theta)$, $p_\epsilon(\theta) \in C_{[0, \pi]}^{(2)}$.

No, $K_\epsilon(\theta) \neq 0$ na $[0, \pi]$, a $\operatorname{tg} \delta + \operatorname{tg} \bar{\omega}$ je ograničena funkcija, jer su $0 < \theta < \pi$; $0 \leq \delta \leq \frac{\pi}{2}$; $0 \leq \omega \leq \frac{\pi}{2}$

dok δ i $\bar{\omega}$ za rešetku ne mogu istovremeno biti jednaki nula. Zato se stacionarne točke funkcije cilja (4.2.12) podudaraju sa stacionarnim točkama aproksimirajuće kontakt funkcije $K_\epsilon(\theta)$.

Naime $\frac{\partial R}{\partial \theta} = \frac{2K'_\varepsilon(\theta) K_\varepsilon(\theta)}{\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}} = 0$ postiže se jedino za $K'_\varepsilon(\theta) = 0$.

Činjenica 2.

Ako $\delta, \omega \in \Gamma \setminus c$ imamo $\operatorname{tg}\delta + \operatorname{tg}\bar{\omega} < (\operatorname{tg}\delta + \operatorname{tg}\omega) \setminus c$,

onda se ekstremalna vrijednost funkcije (4.2.12) postiže na granici oblasti Γ .

Činjenica 3.

Neka su θ_i ($i=1, s$) točke u kojima kontakt funkcija $K_\varepsilon(\theta)$: $\theta \in [0, \pi]$ ima minimume i neka je:

$$\begin{aligned} \max (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) &\geq C_{1i} \\ \min (\operatorname{tg}\delta + \operatorname{tg}\bar{\omega}) &\geq C_{2i} \end{aligned} \quad \delta, \omega \in [0, \frac{\pi}{2}] .$$

Ukoliko je

$$\frac{K_\varepsilon^2(\theta_i)}{C_{1i}} < \frac{K_\varepsilon^2(\theta_j)}{C_{2i}} ,$$

tada je

$$\operatorname{Min} P_\varepsilon [R(\theta_i, \delta, \bar{\omega})] < \operatorname{Min} P_\varepsilon [R(\theta_j, \delta, \bar{\omega})] \quad i \neq j.$$

Prednje činjenice su iskorištene za izradu algoritma za rešetkasto smještanje objekata u ravnini. Isprobano je nekoliko algoritama pa je izabran onaj koji je pokazao više prednosti. Po istom je sastavljen program „**RERAP**”

Jedan od algoritama za rješavanje problema rešetkastog smještanja kongruentnih objekata $S_i \in \{S\}$ u pravokutnik dimenzije $a \times b$ može se izvesti na osnovi slijedećih razmatranja.

Ako površinu pravokutnika kome veličina stranica teži u beskonačnost, podijelimo s brojem objekata koji se u njega smjeste, moramo dobiti površinu paralelograma rešetke, u koju se smjesta površina upravo jednog objekta. Znači da treba da vrijedi slijedeća tvrdnja koju nisam uspio do kraja dokazati.

Tvrđnja 4.10.

$$\lim_{\substack{a \rightarrow \infty \\ b \rightarrow \infty}} \frac{a \times b}{\max \mathcal{B}(\theta, \delta, \bar{\omega})} = \min P^* [R(\theta, \delta, \bar{\omega})] \quad (4.2.13)$$

Na ovaj način je ujedno uspostavljena izvjesna veza između problema smještanja u ravninu i smještanja u pravokutnik, tako da može biti jedan dio algoritma za oba problema zajednički. Algoritam za smještaj u pravokutnik može se izvesti na osnovi slijedećih razmatranja:

Ako uzmemo da je

$$\ell_1(\theta) = \frac{a \times b}{b - [p(\theta) + p(\theta + \pi)] \{a - [p(\theta + \frac{\pi}{2}) + p(\theta + \frac{3\pi}{2})] + K(\theta)\}} \quad (4.2.14)$$

onda je s jedne strane:

$$\frac{a \times b}{\mathcal{B}(\theta, \delta, \bar{\omega})} \leq \ell_1(\theta) \cdot P[R(\theta, \delta, \bar{\omega})] \quad (4.2.15)$$

a s druge strane, za iste vrijednosti tekućih varijabla $\theta, \delta, \bar{\omega}$ imamo da je:

$$\frac{a \times b}{\mathcal{B}(\theta, \delta, \bar{\omega})} \geq P[R(\theta, \delta, \bar{\omega})] \quad (4.2.16)$$

Znači postoji nekakav $\ell_2(\theta, \delta, \bar{\omega})$ da je:

$$\frac{a \times b}{\mathcal{B}(\theta, \delta, \bar{\omega})} = \ell_2(\theta, \delta, \bar{\omega}) P[R(\theta, \delta, \bar{\omega})] \quad (4.2.17)$$

Znači da je $\ell_1(\theta) \geq \ell_2(\theta, \delta, \bar{\omega}) \geq 1$ (4.2.18)

No, ako je naša tvrđnja istinita onda postaje takvi $\Delta\theta, \Delta\delta, \Delta\bar{\omega}$ da za odgovarajuće $\theta, \delta, \bar{\omega}$ vrijedi

$$\begin{aligned} \ell_2(\theta, \delta, \bar{\omega}) \cdot P[R(\theta, \delta, \bar{\omega})] &= P[R(\theta + \Delta\theta, \delta + \Delta\delta, \bar{\omega} + \Delta\bar{\omega})] \\ &\leq \ell_1(\theta) P[R(\theta, \delta, \bar{\omega})] = P[R(\theta + \Delta\theta^*, \delta, \bar{\omega})] \end{aligned} \quad (4.2.19)$$

Iz (4.2.19) imamo:

$$\Delta P' [R(\theta, \delta, \bar{\omega})]_{\theta} = P [R(\theta, \delta, \bar{\omega})] [\ell_1(\theta) - 1] = P [R(\theta + \Delta\theta^*, \delta, \bar{\omega}) - P [R(\theta, \delta, \omega)]] ,$$

odnosno

$$\Delta P' [R(\theta, \delta, \bar{\omega})]_{\theta} = \frac{[K(\theta)]^2}{\text{tg}\delta + \text{tg}\bar{\omega}} [\ell_1(\theta) - 1] = \frac{[K(\theta + \Delta\theta^*)]^2 - [K(\theta)]^2}{\text{tg}\delta + \text{tg}\bar{\omega}}$$

Točnost rezultata koji se dobiju zavise od točnosti s kojima računamo kontakt i potpornu funkciju.

Treba napomenuti da $\Delta\theta^*$ uvijek ne možemo naći, na primjer u točkama gdje se postiže $\max P [R(\theta, \delta, \omega)]$ uz uvjet da je $\ell_1(\theta) > 1$.

Na sl.4.9. dan je optimalan rešetkasti raspored objekta S_3 u pravokutniku 60×60 .

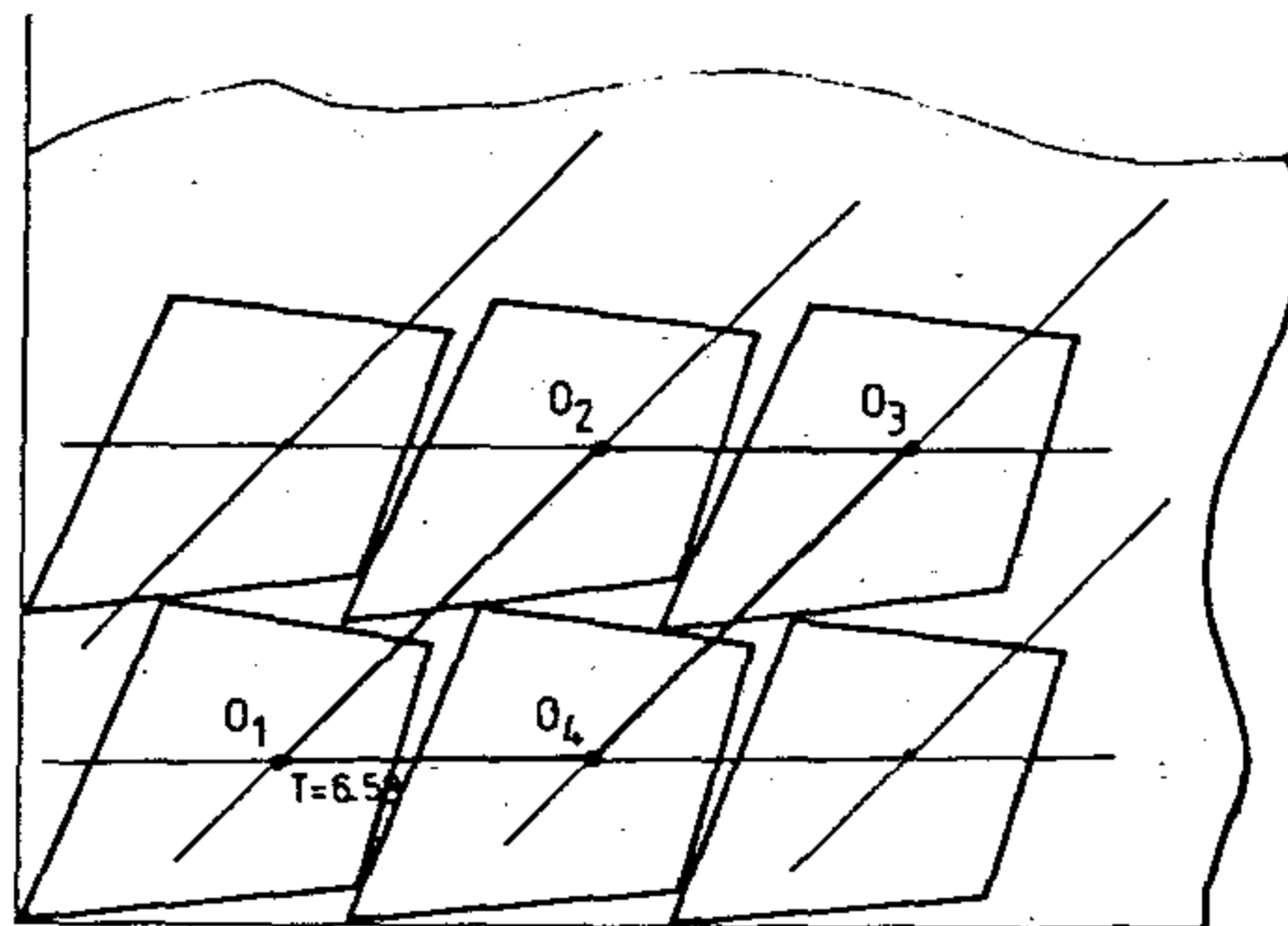
U pravokutnik se smjesti ukupno 64 objekta s maksimalnim koeficijentom iskorištenja 0,67956.

Imamo 8 redova, u svakom se smjesti po 8 objekata.

$\theta = 36^\circ$, $T(\theta) = 6,58021$; $O_1(0,0)$, $O_2(6,72428, 6,46611)$.

U zavisnosti od θ koeficijent iskorištenja se kreće od 0,52778 do navedenog maksimuma.

Kod izrade algoritma maksimalno su korišteni rezultati iz n-rednog periodičnog rasporeda.



SLIKA 4.9

Posebni slučajevi I:

Ako raspoložemo s više dimenzija pravokutnika Ω_i i želimo izabrati onaj u koji se rešetkasto objekti najbolje smještaju, tada se problem svodi na određivanje

$$\max \mu(X) : X(\theta, \delta, \bar{\omega}) \in \Gamma$$

gdje je:

$$\mu(X) = \bigvee_{\alpha}^m \mu_i(X)$$

$$\mu_i(X) = \frac{P(S) \beta_i}{a_i \times b_i}$$

gdje je β_i - broj objekta $\{S\} \subset \Omega_i$, a $P(S)$ površina objekta S . U tabeli 4.1. vidimo rezultate dobivene za pet različitih dimenzija kada se smješta objekt

$$S_3 : (-2,6) \quad (-4,1) \quad (1,4) \quad (3,-1).$$

TABELA 4.1.

Red. br.	Dimenzije ploča	Br. REDOVA	Ukupno objek.	BROJ OBJEKATA U POJEDINOM REDU																		Iskorištenost
				1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	
1	60 x 60	8	64	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	0,675556	
2	75 x 75	10	110	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	0,74311	
3	70 x 80	9	108	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	0,73286	
4	80 x 80	11	125	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	0,74219	
5	90 x 40	18	74	4	3	4	3	4	4	3	4	4	3	4	4	4	4	4	4	5	0,78111	

 $P_1 = P_2 = P_3 = P_4 = 10,5\%$

Rezultati su dobiveni pomoću izvjesnih modifikacija programa RECAP-I.

Posebni slučajevi II:

Eksperimentalno je pokazano da zapravo ne postoji dimenzija pl
če Ω_i koja bi bila najbolja za sve objekte, pa je interesantn
odabrati onu dimenziju koja je u prosjeku najbolja za zadani o
jekat. Za navedenih pet dimenzija iz tabele 4.1. napravljeni s
pored objekata S_3 za još dva objekta S_1 :

$A_1(-4,-4), A_2(-3,4), A_3(3,3), A_4(4,-1);$ i S_2 :

$A_1(0,-5), A_2(-5,2), A_3(-1,3), A_4(6,-1),$

Pomoću modificiranog programa „RERAP II” dobiveni su rezultati
dani u tabeli 4.2.

Iz tabele 4.2. vidimo da pravokutnik 90 x 40 ima bitno bolje
iskorištenje (oko 8%) za sva tri objekta, od pravokutnika iste
površine 60 x 60. Ovim primjerom je ujedno pokazano da nije
uvijek točno da se kod jednakih površina bolje iskorištenje po-
stiže kod kvadrata.

TABELA 4.2.

Red. br.	Dimenzije ploča	S ₁		S ₂			S ₃			Prosječna iskorišćenost		
		θ	n	μ(θ)	θ	n	μ(θ)	θ	n		μ(θ)	
1.	60 x 60	66	49	0,57167	123	49	0,57847	39	64	0,67556	0,60856	
2.	75 x 75	12	84	0,62720	135	83	0,62711	123	110	0,74311	0,66580	
3.	70 x 80	6	86	0,64500	70	81	0,61473	125	108	0,73126	0,66363	
4.	80 x 80	3	95	0,62344	138	96	0,63750	123	125	0,74219	0,66771	
5.	90 x 40	9	56	0,65333	138	53	0,62569	81	74	0,78111	0,68670	
		$P_1 = P_8$		$\Delta\mu = 7,8\%$				$\Delta\mu = 4,7\%$				$\bar{\mu} = 7,8\%$

Napomena:

Rad programa je testiran na poznatom rezultatu za rešetkasti raspored elipsa [51] gdje je $\mu = 0,9069\dots$, a rezultat je dobiven metodama kombinatorke (diskretne) geometrije.

PROGRAMI UZ GLAVU IV

.....
 * NAIJENA: ODREDJIVANJE PARAMETARA RESET K L RAZMJEŠTAJA OBJEKATA
 * U RAVNINI.
 *

```
COMMON /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
*       /KORAK/ DGS
*       /RLINE/ XR(360, 10), YR(360, 10)
*       /VIFKF/ FKF(1000,5), NFK(360,5)
*       /BGEND/ POP(360, 2)
*       /NPINT/ NIN
*       /PERIOD/ TE(360)
*       /CILJ/ FC(360)
*       /IERR/ IERR
*       /HEMA/ QH(360), NT(360), FM12(10,5), KI, NI
```

```
DIMENSION XZ(50), YZ(50)
```

```
IERR = 0
```

```
CALL U L A Z
```

```
IF(IERR .EQ. 1) S T O P 111
```

```
NVH(2) = NVH(1)
```

```
NOBJ = 2
```

```
NZ = NVH(2)
```

```
DO 5 J = 1, NZ
```

```
  XOB(J,2) = XOB(J,1)
```

```
  YOB(J,2) = YOB(J,1)
```

```
  XZ(J) = XOB(J,1)
```

```
  YZ(J) = YOB(J,1)
```

```
5 CONTINUE
```

```
CALL P O T I O N
```

```
S = S P ( XZ, YZ, NZ )
```

```
PRINT 33
```

```
FCMAX = 0.0
```

```
DO 80 I = 1, NIN
```

```
  TE(I) = FKF(I, 1)
```

```
  QH(I) = DGS * (I - 1)
```

```
CALL X I Y I ( I, 1, TE(I) )
```

```
33  FORMAT(///' KUT OBJ.', 8X, 'X', 8X, 'Y', 8X,  

  *       'POVRŠ. PARAL. PERIOD' ISKRIS_ENJE')
```

```
60  PP = YR(I, 1) * TE(I)  

  FC(I) = S / PP
```

```
65  PRINT 65, QH(I), XR(I, 1), YR(I, 1), PP, TE(I), FC(I)  

  *       FORMAT(1X, F8.1, 5F12.5)
```



```

CALL U L A Z
  IF ( IERR , EQ , 1 ) S T O P 111

NVH(2) = NVH(1)
NOBJ   = 2
NZ     = NVH(2)

DO 10 I = 1, NZ
  XOB(I,2) = XOB(I,1)
  YOB(I,2) = YOB(I,1)
  XZ (I)   = XOB(I,1)
  YZ (I)   = YOB(I,1)
10 CONTINUE

CALL P O T K O N

S = S P ( XZ , YZ , NZ )

NIN = NH

DO 30 I = 1, NIN
  TE(I) = F K F ( I , 1 )
  QM(I) = DQS * ( I - 1 )
  CALL X I Y I ( I , 1 , TE(I) )
30 C O N T I N U E

DO 110 IS = 1, NSIR
  SSIR = SIR + DSIR * ( IS - 1 )
  PRINT 32, SSIR
32  FORMAT(1H1, ' 7A SIRINU ROLNE ', F10.5)

  PRINT 33
33  FORMAT(///' NUT OBJ. BR,RED.', 8X, 'X', 8X, 'Y', 8X,
  ' P.OBJEKATA PERIOD ISKORISTENJE BR. OBJEK, '
  ' BROJ OBJEKATA U REDOVINA')

FCHAX = 0.0
PP     = SSIR * DUZ

DO 80 I = 1, NIN

  I1 = NIN / 4 + 1
  IF(I1 ,GT, NIN) I1 = I1 - NIN
  I2 = 3 * NIN / 4 + 1
  IF(I2 ,GT, NIN) I2 = I2 - NIN
  I3 = NIN / 2 + 1
  IF(I3 ,GT, NIN) I3 = I3 - NIN

  BRRED = (SSIR - POP(I,1) - POP(I3,I)) / YR(I,1)

  UHA = 0.0

DO 35 J = 1, BRRED

```

```

PRIV(J) = ( J - 1 ) * XR(I,1)
IF( PRIV(J) .GE. 0.0 .OR. PRIV(J) .GE. UMA) GO TO 35
UMA = PRIV(J)
35 CONTINUE
C
IF (UMA .EQ. 0.0) GO TO 40
DO 40 J = 1, NRED
PRIV(J) = PRIV(J) - UMA
40 CONTINUE
C
NOBP = 0.0
C
DO 50 J = 1, NRED
PRID = DUZ - PRIV(J)
NOST = PRIV(J) / TE(I)
NNNN = (PRID - POP(I1,1) - POP(I2,1)) / TE(I)
NOBR(J) = NNNN + NOST + 1
NOBP = NOBP + NOBR(J)
50 CONTINUE
C
POBJ = NOBP * S
FC(I) = POBJ / PP
C
PRINT 65, QH(I), NRED, XR(I,1), YR(I,1), POBJ, TE(I), FC(I),
NOBF, (NOBF(I), I = 1, NRED)
65 FORMAT(IX, F8.1, 17, 5F12.5, 3X, 16, 3X, 10( 5I4 /))
C
IF(YR(I,1) .LE. 0.0) GO TO 80
C
IF(FCMAX .GE. FC(I)) GO TO 80
FCMAX = FC(I)
IZ = 1
80 CONTINUE
C
CALL OUTPUT( IZ )
C


---


WRITE(10, 85) NH
WRITE(10, 90) (QH(I), FC(I), I = 1, NH)
85 FORMAT(15)
90 FORMAT (4(F8.1, F12.5))
C
110 CONTINUE
C
ENDFILE(10)
C
STOP
END

```

— * —

G L A V A V

PERIODIČNO SMJEŠTANJE U n-REDOVA

1. Opis i definicija problema

U trećoj glavi smo definirali jednorodno periodično smještanje i naveli programe za rješavanje čitave jedne klase problema koji se javljaju u praksi. Pokazaće se da je jednorodno periodično smještanje samo poseban slučaj n-rednog, a s druge strane da se n-redno dađe izvesti iz rešetkastog.

Naime, recimo da imamo konačan skup prebrojivih skupova proizvoljnih objekata $\{S_i\}$ ($i = 1, 2, \dots, n$) čiji su parametri $(x_{ij}, y_{ij}, \theta_{ij})$ ($i = 1, m, j = 1, \dots$).

Skup $\{S_i\}$ čine kongruentni objekti. Uzmemo iz svakog $\{S_i\}$ samo po jedan element i smjestimo u R^2 s parametrima $(x_{ij}, y_{ij}, \theta_{ij})$ i promatramo ih kao jedan složeni objekt:

$$S_i^k = \bigcup_{i=1}^m S_i \quad (5.1.1)$$

čiji su parametri $(x_i^k, y_i^k, \theta_i^k)$. Veze medju parametrima su dane sa

$$\begin{pmatrix} x_{ij} - x_i^k \\ y_{ij} - y_i^k \end{pmatrix} = (a_{ij}, b_{ij}) \begin{pmatrix} \cos \theta_i^k & \sin \theta_i^k \\ -\sin \theta_i^k & \cos \theta_i^k \end{pmatrix} \quad (5.1.2)$$

$$\theta_{ij} = \theta_{ij}^* + \theta_i^*$$

gdje su $(a_{ij}, b_{ij}, \theta_{ij}^*)$ - parametri objekta S_i u pomičnom koordinatnom sistemu $(x_i^k, y_i^k, \theta_i^k)$.

Pretpostavimo da su na taj način generirani objekti $\{S_i^k\}$ uređjeni tako da za V_i vrijedi $i - 1 < i < i + 1$,

tada za $S_i^k \in \{S^k\}$ kažemo da su periodično rasporedjeni ako vrijedi

$$x_{i+1}^k = x_i^k + T_x^k; \quad y_{i+1}^k = y_i^k + T_y^k; \quad \theta_{i+1}^k = \theta_i^k + T_\theta^k.$$

Parametri se, općenito uzevši, mogu mijenjati nezavisno.

Poseban slučaj:

Ako za parametre smještaja objekata $\{S^k\}$ vrijede veze:

$$x_i^k = iT_x^k; \quad y_i^k = iT_y^k; \quad \theta_i^k = i2\pi \quad (i = 0^+1^+2^+\dots)$$

dobijemo rešetkasti raspored [42] [51]. Kod toga su svi objekti međusobno orijentirani.

U praksi se uvijek na parametre objekata postavljaju dodatni uvjeti dani sa $\phi(x,y,\theta) = 0$. Kada zadamo parametar θ_i^k , onda parametri trebaju zadovoljavati $\phi(x,y) = 0$, što u R^2 predstavlja neku liniju na koju se smještaju polovi objekata. Sa $\phi(x,y+r)=0$ dobijemo liniju l_r , nastalu translacijom linije paralelno osi OX za veličinu r. Konstruiramo dvije linije

$l_{\min} = \phi(x, y+r_{\min})=0$ i $l_{\max} = \phi(x, y+r_{\max}) = 0$ takve da one formiraju pojas najmanje širine u kome su smješteni objekti, koji sa $l_{\min}(l_{\max})$ imaju zajedničke samo granične točke. Linije $l_{\min}(l_{\max})$ su u stvari potporne linije.

Konstruiramo potporne linije $l_{\min}^i(l_{\max}^i)$ svakom objektu S_i koji učestvuje u formiranju $\{S^k\}$ a to su:

$$l_{\min}^i = \phi(x, y+r_{\min}^i) = 0; \quad l_{\max}^i = \phi(x, y+r_{\max}^i) = 0.$$

Definicija 1.

Periodični raspored objekata S_i ($i=1,m$) zovemo n-redni periodični raspored, ako parametri $x_i = vT_x^k$ ($v=0^+1^+2^+\dots$) i y_i zadovoljavaju jednadžbu $\phi(vx_i^k, y+r_i) = 0$ ($i = 1,m$) i vrijedi skup nejednadžbi

$$\begin{aligned} r_{\max}^{i_{v+1}} &> r_{\max}^{i_v} \\ r_{\min}^{i_{v+1}} &> r_{\min}^{i_v} \end{aligned} \quad v = 1, 2, \dots, n < m \quad (5.1.3)$$

Poseban slučaj:

Kod jednorednog periodičnog smještaja imamo $r_{\max} = r_{\max}^{i_v}$
 $r_{\min} = r_{\min}^{i_v}$, tj. pojas u kome se smještaju objekti formiraju
 potporne linije l_{\max} i l_{\min} , jednog te istog objekta tj. (5.1.3)
 nije ispunjeno ni za jedan v ($v=1, n$).

U praksi se najčešće susrećemo sa slučajevima da $l = \phi(x, y) = 0$
 predstavljaju pravce, tj. imamo linearno n-redno periodično
 smještanje.

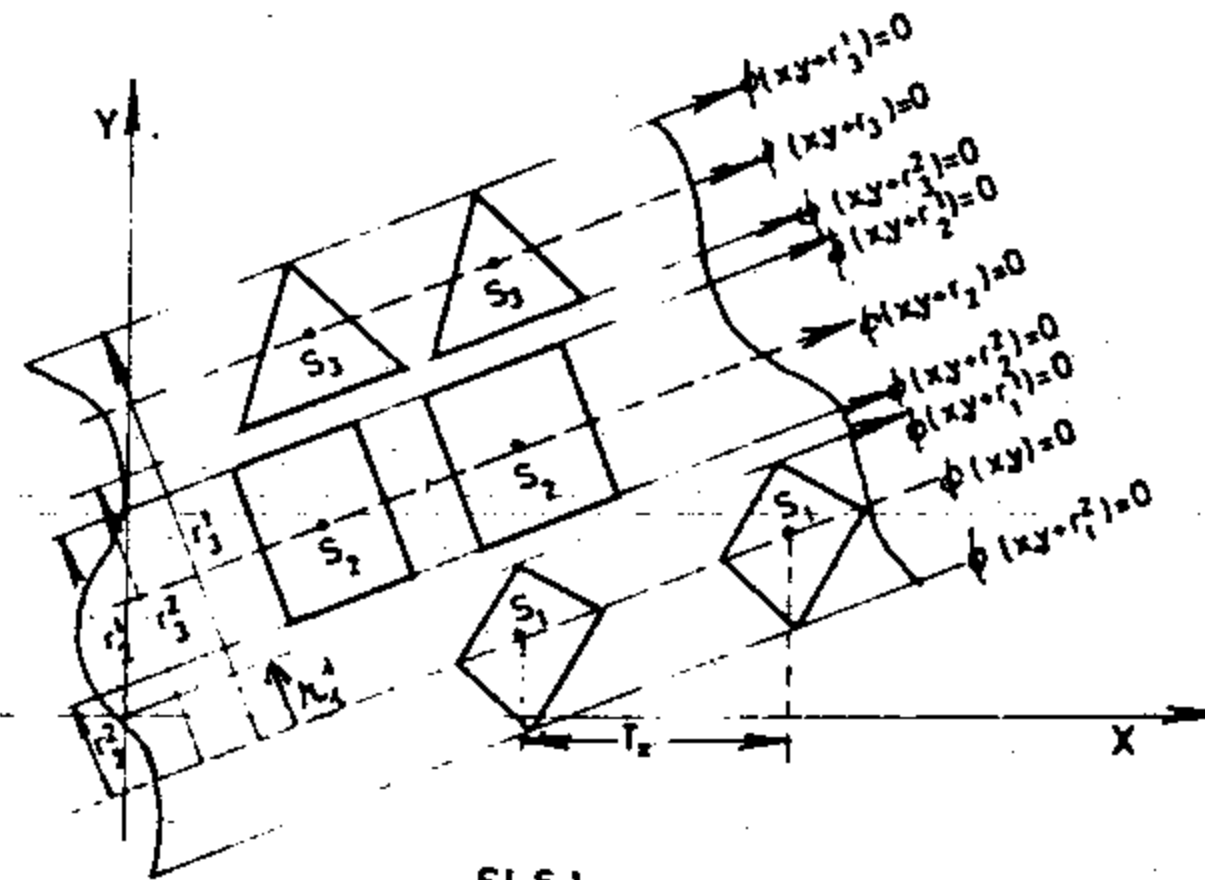
Definicija 2.

Ako parametri $x_i = vT_x$ ($v=0^+1^+2^+\dots$) y_i , i $\theta_i = \theta^* = \text{konstanta}$,
 zadovoljavaju jednakost

$$AvT_x^k + B(y+r_i) + c = 0 \quad (i = 1, m), \quad (5.1.4)$$

a ispunjena je (5.1.3), onda se takav smještaj zove linearni
 n-redni periodični smještaj.

Kratkoće radi, u buduće ćemo takav smještaj zvati n-smještaj.
 Na sl.5.1. vidimo primjer smještaja za $n = 3$.



Sl.5.1

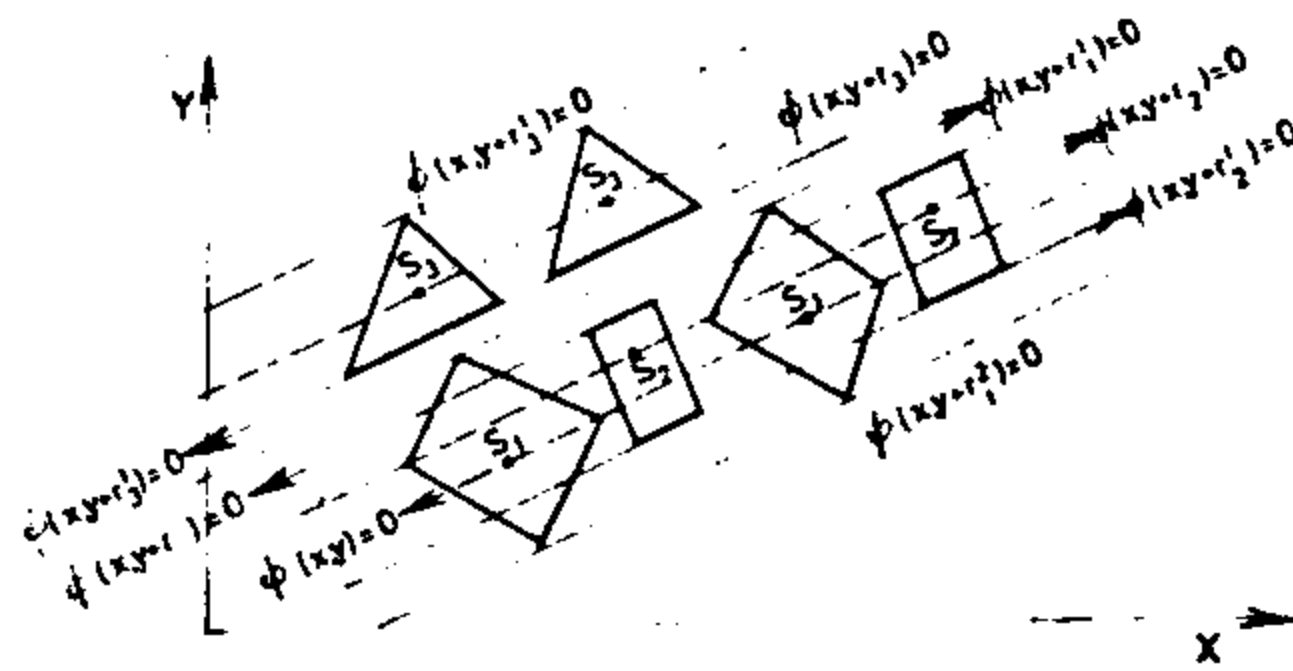
Polovi $\{S_1\}$, $\{S_2\}$, $\{S_3\}$ se nalaze na pravcima:

$$\phi(x, y) = 0; \quad \phi(x, y+r_2) = 0; \quad \phi(x, y+r_3) = 0$$

a imamo ispunjene uvjete (5.1.3) jer je:

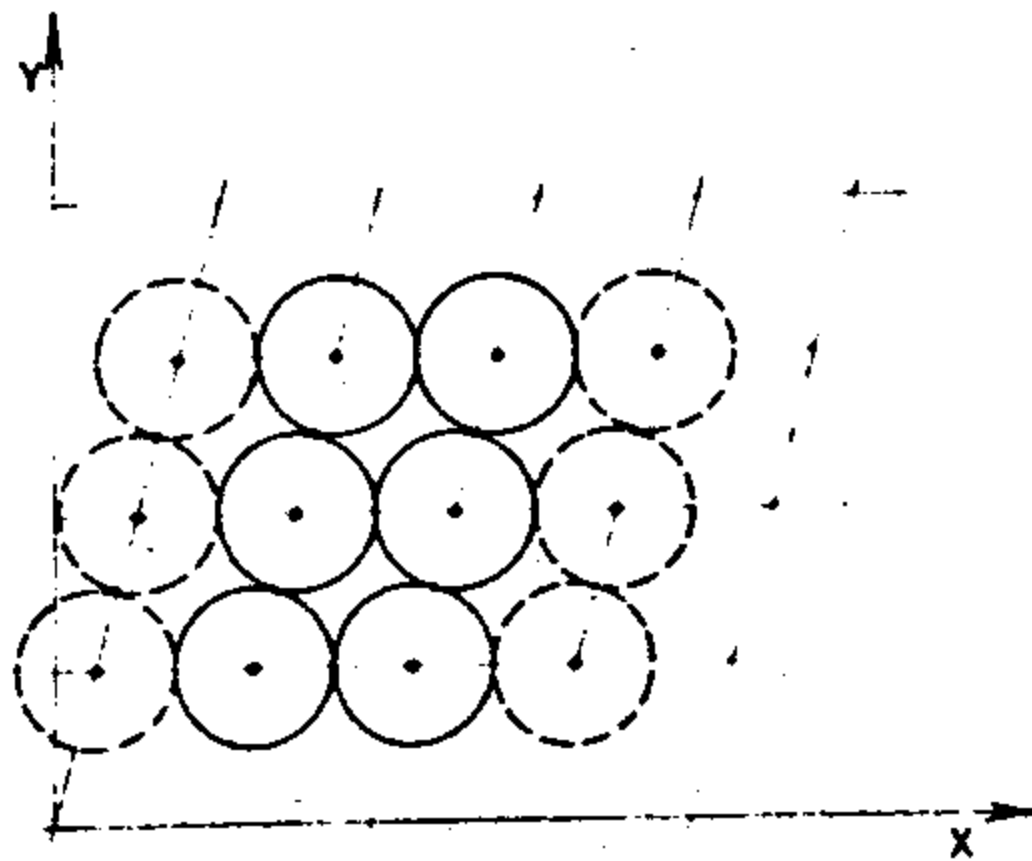
$$r_1^1 < r_2^1 < r_3^1; \quad r_1^2 < r_2^2 < r_3^2.$$

Na slici 5.2. vidimo slučaj rasporeda tri objekta ali dvoredno jer za troredni raspored nisu ispunjeni uvjeti (5.1.3).



Sl.5.2

Pretpostavimo da je zadan skup kongruentnih objekata $S \in \{S\}$ i da su oni smješteni rešetkasto sl.5.3.



Sl.5.3

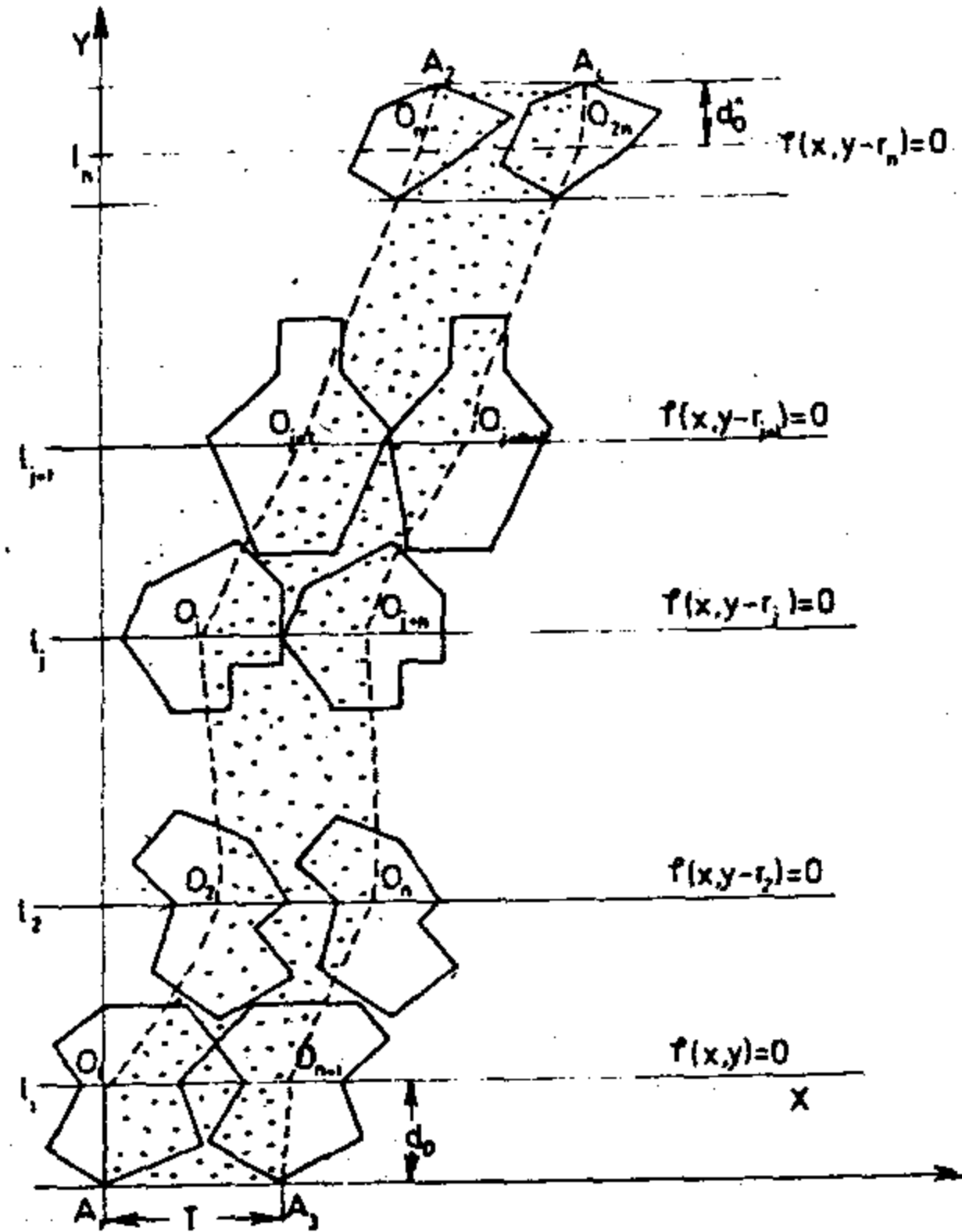
Izaberemo pojas L određenog smjera koji u sebi sadrži tri pravca jednako udaljena na kojima su smješteni polovi objekata. Granice L su potporni pravci n -rednog smještaja.

Dobiveni raspored je $n=3$ redni periodični raspored objekata $S \in \{S\}$ kada je $\phi(x,y) = ax + by + c = 0$.

Izborom raznih smjerova može se iz rešetkastog dobiti određen broj n -rednih periodičnih rasporeda.

2. Formalizacija problema n-rednog smještaja

Uzmimo beskonačnu traku L u koju smjestimo m -različitih objekata u n -redova i to označimo sa $L(n,m)$ [40]. Neka je $m < n$, što implicira da se u svakom redu nalazi samo jedna vrst objekata, sl.5.4.



SI.54

Neka je $\phi(x,y) = y = 0$, tj. pravac se podudara sa osi OX fiksnog sistema. Objekte S_j ($j = 1, n$) smjestimo tako da se njima pripadne osi OX^j koordinatnog sistema $X'O_jY'$ podudaraju s paralelnim pravcima $\phi_j = \phi(x,y-r_j) = y-r_j = 0$ ($j = 1, n$). Neka su granice pojasa L pravci $\phi(x,y-r_0) = 0$ i $\phi(x,y-r_n^*) = 0$.

Znači da su oni i potporni pravci. Poligon $R = (A_1, O_1, O_2, \dots, O_n, A_2, A_4, O_{2n}, O_{n+1}, A_3)$, zovemo osnovni paralelogram rasporeda $L(n,m)$. Označimo paralelograme $A_1 O_1 O_{n+1} A_3 = \bar{R}$, $O_j O_{j+1} O_{j+n+1} O_{j+n} = R_j$ i $O_n A_2 A_4 A_{2n} = \bar{\bar{R}}$.

Iz definicije n -smještaja $L(n,m)$ slijedi da su baze svih potparalelograma međusobno jednake tj.

$$O_1 O_{n+1} = O_2 O_{n+2} = \dots = O_j O_{n+j} = \dots = O_n O_{2n} = T \quad (5.2.1)$$

Udaljenost između dva pravca na kojima su smješteni polovi dana je sa:

$$d_j = |r_{j+1} - r_j| \quad (j = 1, n-1).$$

Ako imamo takav smještaj $L(n, m)$ da je udaljenost $d_j = 0$, onda za takav potparalelogram R_j kažemo da je degeneriran. Znači da u najopćenitijem slučaju za potparalelograme R_j, R_{j+t}, R^2 smještaja $L(n, m)$ vrijedi

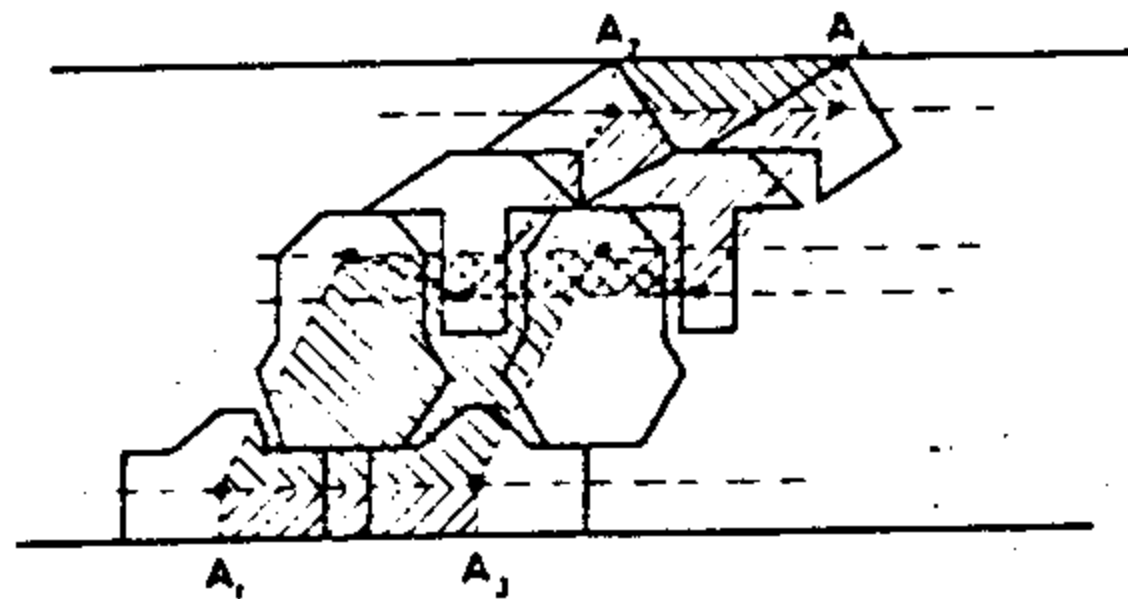
$$R_j \cap R_{j+t} \neq \emptyset \quad (5.2.3)$$

$$j = 1, 2, \dots, n-3; \quad t = 2, 3, \dots, n-3; \quad j+t < n$$

Znači da se R_j i R_{j+t} , mada nisu susjedni, mogu presjecati. Naime, ako pogledamo ordinate polova $O_{j+1}(x_{j+1}, y_{j+1})$ i $O_{j+t}(x_{j+t}, y_{j+t})$ u sistemu XOY onda je sasvim moguće da bude

$$y_{j+t} < y_{j+1}. \quad (5.2.4)$$

Na sl.5.5. vidimo slučaj $L(4, 4)$ gdje se P_1 i P_3 presjecaju.



Sl.5.5

Znači da je osnovni paralelogram dan sa:

$$R = \bar{R} \cup \bar{R} \cup (R_j \cap R_{j+t}) \quad (5.2.6)$$

a da je površina dana sa:

$$P(R) = P(\bar{R}) + P(\bar{R}) + \sum_{j=1}^{n-1} P(R_j) - \sum_{t=2}^{n-2} P(R_{j+t}) \quad (5.2.6)$$

Mi ćemo proučavati slučajeve gdje je $R_j \cap R_{j+t} = \emptyset$ ($t \geq 2$) pa (5.2.5) prelazi u

$$R = \bar{R} \cup \bar{R} \cup \bigcup_{n=1}^{n-1} R_j, \quad (5.2.7)$$

odnosno (5.2.6) prelazi u

$$P(R) = P(\bar{R}) + \sum_{j=1}^{n-1} P(R_j) + P(\bar{R}) \quad (5.2.8)$$

Neka za bilo koji par objekata S_j^a i S_t^b smještenih u čvorovima (O_a, O_b) potparalelograma $R_j \subset R^2$, vrijedi:

$$S_i^a \setminus S_i^a \cap S_t^b \setminus S_t^b = \emptyset \quad (5.2.9)$$

$i, t = 1, 2, a \neq b, a, b = 1, \dots, 2n$

tada je očito da će kod minimalne površine osnovnog paralelograma koeficijent iskorištenja biti najveći. Sa ovako definiranim osnovnim paralelogramom ravninu možemo prekriti bez rupa i preklapanja, a kako je površina objekata invarijantna na položaj, znači da koeficijent iskorištenja možemo definirati izrazom

$$\mu = \frac{\sum P(S_j)}{P(\bar{R}) + \sum_{j=1}^{n-1} P(R_j) + P(\bar{R})} \quad (5.2.10)$$

znači da trebamo naći takav raspored $L(n, m)$ da bude

$$P(R) = \min (H \cdot T) \quad (5.2.11)$$

gdje H - širina pojasa L , a T - period smještaja.

Znači da je

$$P(R) = \min \left[T \left(d_0 + \sum_{n=1}^{n-1} |r_{j+1} - r_n| + d_0^* \right) \right] \quad (5.2.12)$$

d_0 - udaljenost od linije l_1 - na kojoj su polovi objekta S_1 do donjeg potpornog pravca p_1 ,

d_0^* - udaljenost od linije l_n - na kojoj su smješteni polovi S_n do gornjeg potpornog pravca,

$d_j = |r_{j+1} - r_j|$ - udaljenost između l_{j+1} i l_j .

Budući da izraz $H = d_0 + \sum_{j=1}^{n-1} d_j + d_0^*$ može biti i negativna, a s obzirom da je površina uvijek pozitivna, to je površina dana sa:

$$P(R) = \min \left[d_0 + \sum_{j=1}^n d_j + d_0^* \mid x T \right]. \quad (5.2.13)$$

Ovakvi problemi u industriji se nazivaju n-redno štampanje. Predmet našeg izučavanja u ovoj glavi bit će slijedeći problemi koje sada samo navodimo:

Problem I:

Odrediti onu širinu beskonačne trake L i parametre smještaja objekata $S_j \in \{S\}$ ($j = 1, m$), kada su objekti smješteni n-redno, za koju je koeficijent iskorištenosti maksimalan.

Problem II:

Rolnu zadane širine H izrezati na trake L_i - u kojima su n-redno smješteni objekti S_j ($j = 1, m$), tako da koeficijent iskorištenosti bude maksimalan.

II-1. Iz skupa raspoloživih širina rolni $\{H_i\}$ odabrati onu za koju je koeficijentat iskorištenja maksimalan.

II-2. Iz skupa raspoloživih širina rolni $\{H_i\}$ odabrati onu za koju je prosječni koeficijenat iskorištenja minimalan.

Problem III:

U rolnu zadane širine H smjestiti jednu vrst objekta $S \in \{S\}$. Odrediti onaj broj redova n - kod koga je iskorištenje najveće.

Problem IV:

U pravokutnu oblast $\Omega = (a \times b)$ treba n - redno i periodično smjestiti objekte S_j ($j = 1, m$) tako da koeficijent iskorištenja bude maksimalan.

pošto se pojasevi mogu smještati paralelno stranici a , odnosno b , to je broj pojaseva dan sa:

$$K_a = N \left[\frac{b}{H} \right], \quad (5.2.14)$$

odnosno

$$K_b = N \left[\frac{a}{H} \right]. \quad (5.2.15)$$

Adekvatno tome imamo i pripadne koeficijente iskorćštenja:

$$\mu_a = \frac{K_a \sum_{j=1}^n v_{ja} P(S_j)}{a \times b} \quad (5.2.16)$$

$$\mu_b = \frac{K_b \sum_{j=1}^n v_{jb} P(S_j)}{a \times b} \quad (5.2.17)$$

gdje v_{ja} , v_{jb} - broj objekata j -tog reda - kod smještaja pojaseva paralelno sa stranicom a , odnosno b ,

$P(S_j)$ - površina objekta S_j .

Prema tome, treba naći

$$\mu_{\Omega} = \max \{ \mu_a, \mu_b \}, \quad (5.2.18)$$

tj. nalazimo

$$\mu_{\Omega}^* = \max \mu_{\Omega} = \frac{N\left[\frac{b}{H}\right] \left[\sum_{j=1}^n v_{ja} P(S_j) \right] \vee N\left[\frac{a}{H}\right] \left\{ \sum_{j=1}^n v_{jb} P(S_j) \right\}}{a \times b} \quad (5.2.19)$$

Problem IV-1.

Iz skupa raspoloživih pravokutnika Ω_i izabrati onaj koji daje najbolje iskorištenje kod n -rednog smještaja zadanih objekata.

Problem IV-2.

Iz skupa Ω_i izabrati onaj koji ima najbolje prosječno iskorištenje kod n -rednog smještaja objekata.

Problem V.

U zadani Ω kod n -rednog periodičnog rasporeda smjestiti najveći mogući broj kompleta.

Problem V-1.

Iz skupa zadanih dimenzija Ω_i odabrati onu dimenziju za koju je za zadani komplet iskorištenje maksimalno.

Problem V-2.

Iz skupa zadanih dimenzija Ω_i odabрати onu za koju je prosječni koeficijent iskorištenja za razne komplete maksimalan.

Svodjenje problema na matematičko programiranje i moguće situacije

Nas interesira medju kakvim rasporedima treba tražiti onaj kod kojeg je osnovni paralelogram R najmanji, tj. možemo li odbaciti one paralelograme koji nisu takvi.

A) Vidimo da kod konstantne širine pojasa dane sa

$$H = d_0 + \sum_{j=1}^{n-1} d_j + d_0^* = \text{const}, \text{ površina } R \text{ zavisi od perioda } T.$$

Sastavljeni objekt S_i^k dan sa (5.1.1) možemo premiještati paralelno l_j samo dotle dok ne dodje do dodira objekata na jednoj od linija l_j , tj. dok ne bude

$$s_j^j \cap s_j^{j+n} \neq \emptyset \quad (5.2.20)$$

$$s_j^j \setminus s_j^j \cap s_j^{j+n} \setminus s_j^{j+2} = \emptyset$$

Onaj red kod koga se prvog realizira (5.2.20) definira period T. Budući da kod bilo kog reda može doći najprije do ispunjenja (5.2.20), period T_j n-rednog smještaja dan je sa:

$$T = \max \{T_1, T_2, \dots, T_n\} \quad (5.2.21)$$

B) S druge strane za $T = \text{const.}$ minimum osnovnog paralelograma R se postiže kada je udaljenost izmedju prvog i posljednjeg reda minimalna. Budući da promatramo samo one slučajeve gdje nema medjusobnog preklopa R_j , znači da trebaju biti visine svih podparalelograma R_j - minimalne, a za to je potrebno da se objekti u raznim redovima diraju.

Da objekti jednog reda diraju objekte drugog reda možemo formalno zapisati

$$s_j^{l_j} \cap s_{j+t}^{l_{j+t}} \neq \emptyset$$

$$s_j^{l_j} \setminus s_j^{l_j} \cap s_{j+t}^{l_{j+t}} \setminus s_j^{s+t} = \emptyset \quad \begin{matrix} j+t \leq n \\ j=1, n-2; t=1, n-1. \end{matrix} \quad (5.2.22)$$

$S_j^{l_j} (S_{j+t}^{l_{j+t}})$ - objekti smješteni na pravcima $l_j (l_{j+t})$.

Uvjetima (5.2.20) i (5.2.21) stavljena su ograničenja na moguće smještaje objekata, tj. na vrijednosti parametara smještaja.

Znači da n-redni raspored objekata treba tražiti među rasporedima za koje vrijede (5.2.20) i (5.2.22).

Prema tome, formalno n-redni raspored u Ω se svodi na nalaženje optimuma (5.2.13) i (5.2.8) uz ispunjenje uvjeta (5.2.20) i (5.2.22).

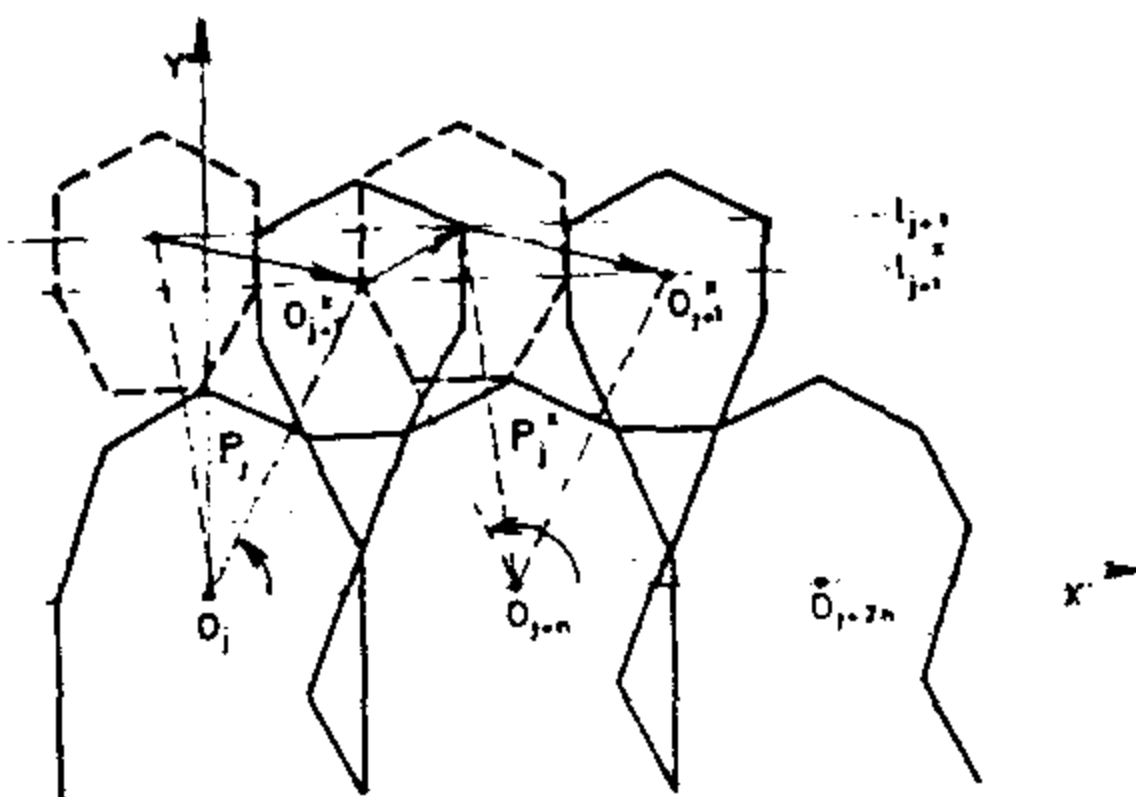
Iz prednjega je jasno da se radi o jednoj specifičnoj klasi problema matematičkog programiranja.

Dalje će biti pokazano da je prednje formalne uvjete moguće izraziti pomoću kontakt i potporne funkcije koje smo definirali u glavi II.

Uočavanje situacija koje zadovoljavaju (5.2.20) i (5.2.22)
(u daljnjem tekstu "KONTAKT SITUACIJE")

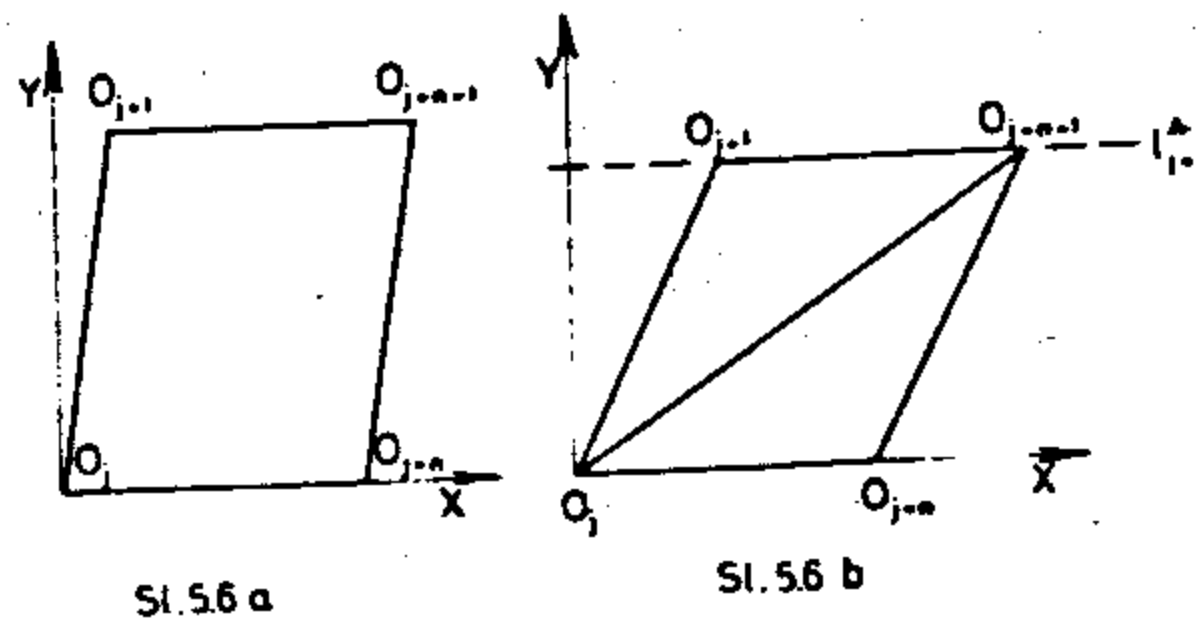
Neka je zadano $L(n,m)$ objekata S_t ($t=1,n$) u kome je već S_τ ($\tau=1,2\dots j$) objekata smješteno s polovima na pripadnim pravcima l_τ , a smještamo novi red objekata S_{j+1} s polovima na l_{j+1} .

Neka se objekti u $(j+1)$ redu premještaju u odnosu na fiksne objekte u smjeru osi OX , ali tako da su uvijek u kontaktu, tj. da vrijedi (5.2.22), sl.5.6.



Sl. 5.6

Vidimo da se kod takvog kontakt premještanja polovi objekata S_{j+1} nalaze na raznim pravcima l_{j+1} (l_{j+1}^*) pa se mijenja i udaljenost $d_j = d_{j+1} - d_j$. Kada se polovi objekata nalaze na l_{j+1} , onda toj kontakt situaciji odgovara graf 5.6a. Dakle, od šest mogućih kontakta medju objektima već rasporedjenim i onima koje smještamo u redu $(j+1)$ imamo četiri kontakta. Medjutim, kada se objekti S_{j+1} nalaze na pravcu l_{j+1}^* , onda imamo pet kontakata, tj. pored onih na sl.5.6a. još jedan novi i to izmedju objekata S_j smještenog u polu O_{j+n} i objekta S_{j+1} smještenog u polu O_{j+1} . U terminima teorije grafova za situaciju na sl.5.6a možemo reći da postoje rebra $(j,j+1)$; $(j,j+n)$; $(j+1,j+n+1)$; $(j+n,j+n+1)$ [39].



Opaska 1:

Ako imamo smještaj $L(n,m)$ a $n = m$, $n \geq 2$ i u jednom redu imamo iste objekte a uvjeti (5.2.22) nisu ispunjeni, onda:

- kad postoji rebro $(j,j+1)$, postoji i rebro $(j+n,j+n+1)$
- kad postoji rebro $(j,j+n+1)$, postoji i rebro $(j+n,j+n+1)$
- kad postoji rebro $(j+n,j+1)$, postoji i rebro $(j+n,j+n+1)$
- kad postoji rebro $(j,j+n)$, postoji i rebro $(j+n,j+n+1)$
- kad postoji rebro $(j+1,j+1+n)$, postoji i rebro $(j+1+n,j+1+n+1)$

(5.2.23)

$$j = (1, n-1), \quad k = i + 1, \quad i = 1, 2, \dots$$

Naime, ako u nekom potparalelogramu R_j n -rednog smještaja imamo neki od slučajeva a) do e), onda uzevši u obzir da paralelogram R pokriva pojas bez rupa i preklapanja, vrijedi naprijed navedeno (a) - (e).

Kod n -rednog rasporeda $L(n,1)$, tj. kada se rasporedjuje samo jedna vrst objekata, svi su potparalelogrami R_j jednaki, pa se na-

prijed navedeni slučajevi (a) - (e) svode na četiri slučaja:

- (a) kada postoji rebro $(j, j+n)$, postoji i rebro $(j+i, j+n+i)$
 (b) kada postoji rebro $(j, j+1)$, postoji i rebro $(j+in, j+1+in)$
 (c) kada postoji rebro $(j, j+n+1)$, postoji i rebro $(j+i, j+n+1+i)$
 (d) kada postoji rebro $(j+n, j+1)$, postoji i rebro $(j+n(i+1), j+in+1)$

(5.2.24)

$$j = (1, n-1), \quad i = (1, 2, \dots).$$

Za dosadašnja razmatranja dovoljan je bio neorijentirani graf, dok će za daljnja razmatranja biti potrebni orijentirani grafovi [39]. Vrhove paralelograma $(j, j+1, j+n+1, j+n)$ u kojima su smješteni polovi $(O_j, O_{j+1}^*, O_{j+n+1}^*, O_{j+n})$ tretiramo kao vrhove grafa, a spojnice vrhova R kao orijentirane stranice (duge) grafa. Označimo sa $(\alpha, \beta) \subset g$ uređen par rebara, gdje je $\alpha \neq \beta$; $\alpha, \beta = j, j+1, j+n+1, j+n$.

Sa $\vec{a}_{\alpha\beta}$ - označimo vektor koji ide iz vrha α u vrh β , a sa $|a_{\alpha\beta}|$ - njemu pripadnu duljinu. Sa $\phi_{\alpha\beta}$ označimo kut $\vec{a}_{\alpha\beta}$ sa osi $O_j X$. Neka se $a_{\alpha\beta}$ i $\phi_{\alpha\beta}$ podudaraju s odgovarajućim stranicama grafa paralelograma R_j^* .

Vežu medju vrhovima grafa nazovimo mrežom i označimo sa D .

a) Neka se mreža D nalazi u I i II kvadrantu sistema $X'O_j Y'$ ($j=1, n-1$) i neka joj se vrhovi podudaraju s vrhovima $R_j \subset R$. Da bi vrh O_{j+1} imao najmanju apscisu po modulu, treba da bude nedegenerirana jedna od duga $(j, j+1)$, $(j, j+n+1)$, $(j+n, j+1)$.

b) Za mrežu D uvijek postoji takav broj $K = 1, 2, \dots$ da vrijedi

$$0 < |a_{\alpha\beta}| \cos \phi_{\alpha\beta} < kT; \quad T - \text{period.} \quad (5.2.25)$$

c) Smjer vektora $\vec{a}_{\alpha\beta}$ podudara se sa dugom (α, β) . Ako imamo (β, α) onda duga mijenja smjer, tj. imamo:

$$\phi_{\alpha\beta} = \begin{cases} \phi_{\alpha\beta} + \Pi & \text{za } \phi_{\alpha\beta} < \Pi \\ \phi_{\alpha\beta} - \Pi & \text{za } \phi_{\alpha\beta} \geq \Pi \end{cases} \quad (5.2.26)$$

gdje:

$$\alpha \neq \beta; \quad \alpha, \beta = (j, j+1, j+n+1, j+n).$$

Definicija.

Ako su u mreži diranja poznati svi parovi brojeva:

$$\{ |a_{\alpha\beta}|, \phi_{\alpha\beta} \} \quad (\alpha \neq \beta; \alpha, \beta = (j, j+n, j+n+1, j+1),$$

onda takvu mrežu zovemo "POTPUNA MREŽA DIRANJA".

Ako u vrhove R_j smještamo dva različita objekta, onda općenito uzevši možemo imati 24 različite mreže diranja [40] tabela 5.1.

Kod toga, kad nalazimo kontakt situacije, imamo razne slučajeve [40].

Slučaj 1.

Za $m = 1, n = 2$, tada u paralelogramu R_1 možemo imati samo slučajeve mreža D_1 do D_4 ; D_{11} do D_{12} .

Slučaj 2.

$n = m = 2$, onda u R_1 možemo imati samo mreže D_1 do D_{18} .

Slučaj 3.

Za $n > 2, m = n, t > 2$, a uvjet (5.2.22) nije ispunjen, onda u $R_j \subset R, (j=1, n-1)$ možemo imati samo mreže D_1 do D_{21} .

Slučaj 4.

Za $n > 2, m = n, t > 2$, a (5.2.22) vrijedi, onda u $R_j \subset R (j = 1, n-1)$ možemo imati D_1 do D_{24} .

Ako imamo potpune mreže diranja, onda imamo slijedeće slučajeve a) i b):

Slučaj a:

Potpune mreže diranja D_1 do D_{18} ; D_{22} i D_{23} definiraju vrijednost perioda T, n - rednog rasporeda.

Slučaj b:

Ako su D_1 do D_{18} potpune mreže diranja, onda pomoću parametara $|a_{\alpha\beta}|$ i $\phi_{\alpha\beta}$ možemo odrediti parametre smještaja objekata. Tako za slučaj mreže D_4 , ako je smjestimo u $X'O_jY'$ tako da se vrh "j" podudara sa O_j , a vrh $(j+n)$ leži na osi O_jX , imamo slijedeće

parametre smještaja u R_j :

$$x_j=0; \quad x_{j+1}=|\vec{a}_{\alpha\beta}| \cos\phi_{\alpha\beta}; \quad x_{j+n}=|a_{j,j+n}|;$$

$$x_{j+n+1} = x_{j+1} + |\vec{a}_{j,j+n}|$$

$$y_j=0; \quad y_{j+1}=|\vec{a}_{\alpha\beta}| \sin\phi_{\alpha\beta}; \quad y_{j+1}=0; \quad y_{j+n+1} = y_{j+1}.$$

Kao što će dalje biti vidljivo sve naprijed navedeno bit će moguće matematički definirati pomoću kontakt funkcije.

TABELA 5.1

Red. br.	Oblik mreže	Br. reb.	Red. br.	Oblik mreže	Br. reb.
1		6	13		3
2		5	14		3
3		5	15		2
4		4	16		2
5		4	17		2
6		4	18		2
7		4	19		2
8		4	20		1
9		3	21		1
10		3	22		1
11		3	23		1
12		3	24		0

Formalni opis tehnoloških ograničenja

Očigledno je da općenito uzevši kod n -rednog smještanja možemo dobiti veće iskorištenje nego kod jednorednog, međutim, jednoredno štampanje se dosta upotrebljava, jednostavno zato što raspoloživa tehnika ne omogućava da se štampa više redova od jednom.

Kada se problemi rješavaju objektivnim metodama, onda treba po mogućnosti probleme definirati jedinstvenim aparatom, što mi činimo koristeći kontakt, potporne i R-funkcije.

Izgleda se najveća količina limova, zbog masovnosti, štampa pomoću preša čija se snaga kreće od nekoliko desetina pa do nekoliko stotina tona. Taj proces nazivaju hladno izvlačenje. Neke preše imaju fiksne glave a neke promjenljive.

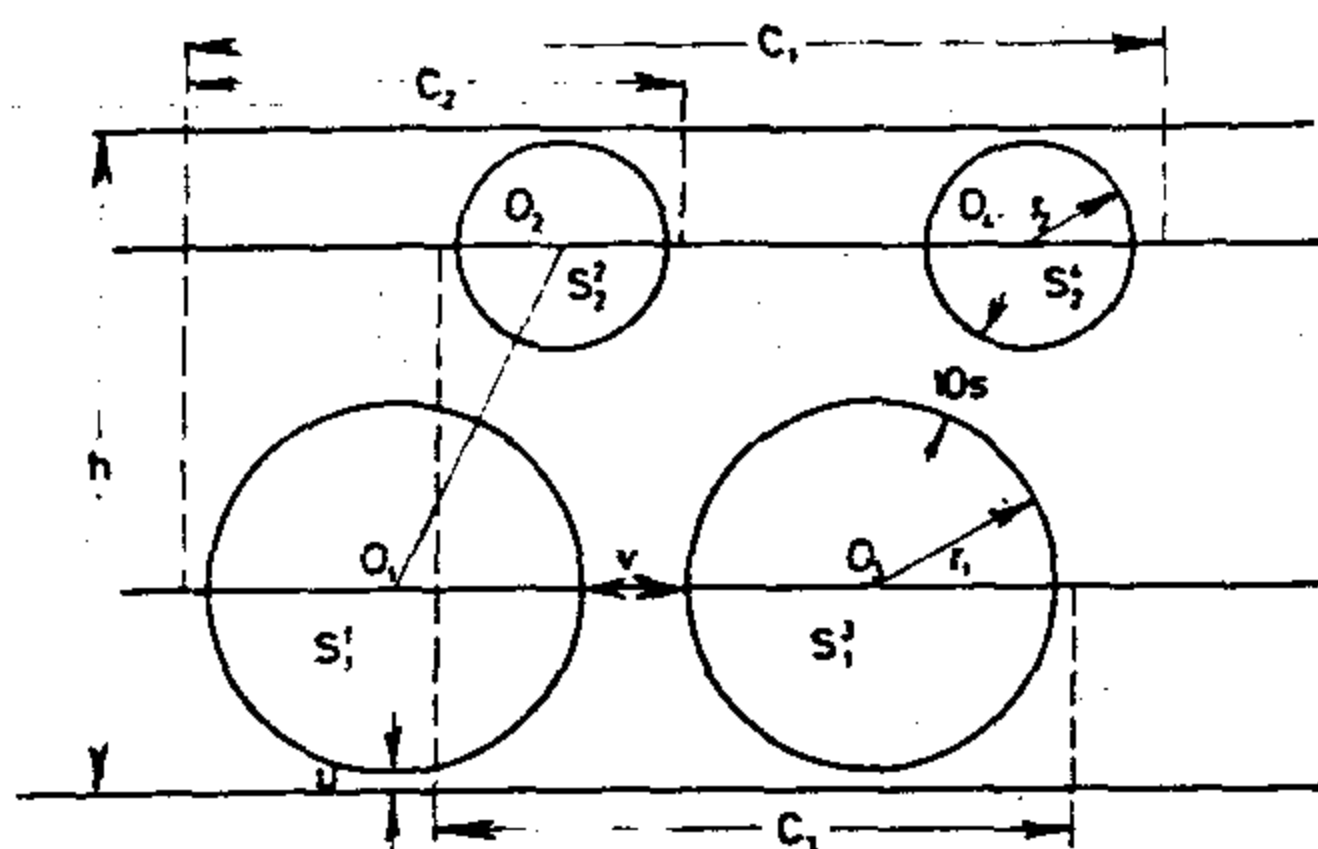
Brzine traka su različite i kreću se do nekoliko desetina metara u minuti s različitom duljinom koraka. Sve preše imaju ograničenja na maksimalnu širinu trake u kojoj su n -redno raspoređeni objekti, kao i na najveću udaljenost objekata koji se istovremeno štampaju.

Ni proizvođači, kao ni potrošači ne žele imati posla sa previše različitih dimenzija ploča i to zbog transporta, uskladištenja, itd. Kod velikih ploča, zbog nesavršenstva dimenzioniranja, ne može se govoriti u matematičkom smislu o pravokutnim pločama ili trakama konstantne širine. Rubovi materijala su iz raznih razloga često oštećeni o čemu se mora voditi računa kada se vrši proračun. Iz tehnoloških razloga često se ne može objekte smjestiti do samog ruba. Treba voditi računa da u zavisnosti od debljine limova, udaljenost između objekata koji se istovremeno štampaju ne smije biti manja od neke unaprijed zadane veličine (obično 7-12 debljina materijala).

Kako preše koriste velike količine energije može se desiti da troškovi utroška energije i ostalog premaše vrijednost smanjenog otpada, mada su to rijetki slučajevi.

Formalni prikaz kako se mogu u obzir uzeti neka ograničenja demonstrira slijedeći primjer.

Recimo da su objekti krugovi radiusa r_1 i r_2 koje treba smjestiti dvoredno. Neka je debljina lima "s", a rubu ploče ne smiju s krugovi primaknuti bliže od u , zatim udaljenost medju objektim ne smije biti manja od v , a udaljenost medju objektima smještanim u $O_1O_2O_3O_4$ koji se istovremeno štampaju ne smiju biti manji od $10s$, sl.5.7.



Sl.5.7

Neka su C_1, C_2, C_3 - duljine radnih polja štampanja, a h dozvoljena širina radnog polja štampanja.

Neka je C - dopustiva duljina radnog polja. Tada moraju biti zadovoljeni slijedeći uvjeti:

$$h = r_1 + r_2 + (r_1 + r_2 + 10s) \sin \alpha + 2u \leq h \quad (5.2.27)$$

$$C_1 \vee C_2 \vee C_3 \leq C \quad (5.2.28)$$

gdje je

$$C_1 = 3r_1 + r_2 + 2u + v + (r_1 + r_2 + 10s) \cos \alpha$$

$$C_2 = r_1 + r_2 + 2u + (r_1 + r_2 + 10s) \cos \alpha$$

$$C_3 = r_1 + r_2 + 2u - (r_1 + r_2 + 10s) \cos \alpha.$$

Kasnije će se vidjeti da je ove uvjete moguće izraziti pomoću kontakt funkcija.

3. Uvjeti nepresjecanja simetrija, suženja granica kutnih parametara

Na osnovu opisanih mreža diranja D_1 do D_{24} i svojstava KF možemo ukazati na više primjedbi o kojima treba voditi računa kod rješavanja problema.

Primjedba 1.

Ako su objekti $S_j, S_{j+1} \in \{S\}$ smješteni n-redno u podparalelogramu $R_j^t \in R$ ($j = 1, n-1$) $j+t < n$; $t = 1, 2, \dots$) a odgovara im jedna od mreža diranja D_1 do D_{24} , onda se udaljenost medju pravcima ℓ_j i ℓ_{j+t} na kojima su polovi može odrediti pomoću KF.

Naime, to slijedi iz činjenice da su u D_1 do D_{21} nedegenerirane duge $(j, j+t)$, $(j, j+n+t)$ ili $(j+n, j+t)$. Označimo kuteve u R_j^t sa:

$$\begin{aligned} \gamma_j^t &= \angle O_{j+t} O_j O_{j+n} \\ \alpha_j^t &= \angle O_{j+n+t} O_j O_{j+n} \\ \omega_j^t &= \angle O_{j+t} O_{j+n} O_{j+2n} \end{aligned} \quad (5.3.1)$$

onda udaljenost medju redovima j i $j+t$ možemo izraziti pomoću KF:

$$\begin{aligned} K_{j, j+t} (\theta + \gamma_j^t) \sin \gamma_j^t & \text{ za nedegeneriranu dugu } (j, j+t) \\ d_j = r_{j+t} - r_j = K_{j, j+t} (\theta + \alpha_j^t) \sin \alpha_j^t & \quad " \quad (j, j+n+t) \\ K_{j, j+1} (\theta + \omega_j^t) \sin \omega_j^t & \quad " \quad (j+n; j+t) \end{aligned} \quad (5.3.2)$$

$$j=1, n-1; \quad j+t \leq n; \quad t=1, 2, \dots$$

Mada se u praksi takvi slučajevi ne pojavljuju, moguće je da imamo:

$$d_j = d_j(\theta, \gamma, \omega) = d_j(\theta, \gamma_1 \gamma_2 \dots \gamma_{n-1}, \omega_1, \omega_2 \dots \omega_{n-1}).$$

Uočimo da su medjusobno nezavisni slijedeći parovi kutnih parametara:

$$(\theta, \gamma_j^t); \quad (\theta, \alpha_j^t) \quad \text{i} \quad (\theta, \omega_j^t), \quad \text{jer je:}$$

$$\omega_j^t = \begin{cases} \Pi\text{-arc tg } \frac{A(\theta, \gamma_j^t) \sin \gamma_j^t}{|A(\theta, \gamma_j^t) \cos \gamma_j^t - T|} & \text{za } A(\theta, \gamma_j^t) \cos \gamma_j^t < T \\ \text{arc tg } \frac{A(\theta, \gamma_j^t) \sin \gamma_j^t}{A(\theta, \gamma_j^t) \cos \gamma_j^t - T} & \text{za } A(\theta, \gamma_j^t) \cos \gamma_j^t \geq T \end{cases} \quad (5.3.3)$$

$$\gamma_j^t = \begin{cases} \Pi\text{-arc tg } \frac{B(\theta, \alpha_j^t) \sin \alpha_j^t}{B(\theta, \alpha_j^t) \cos \alpha_j^t - T} & \text{za } B(\theta, \alpha_j^t) \cos \alpha_j^t < T \\ \text{arc tg } \frac{B(\theta, \alpha_j^t) \sin \alpha_j^t}{B(\theta, \alpha_j^t) \cos \alpha_j^t - T} & \text{za } B(\theta, \alpha_j^t) \cos \alpha_j^t \geq T \end{cases} \quad (5.3.4)$$

$$\gamma_j^t = \begin{cases} \Pi\text{-arc tg } \frac{C(\theta, \omega_j^t) \sin \omega_j^t}{C(\theta, \omega_j^t) \cos \omega_j^t - T} & \text{za } C(\theta, \omega_j^t) \cos \omega_j^t < T \\ \text{arc tg } \frac{C(\theta, \omega_j^t) \sin \omega_j^t}{C(\theta, \omega_j^t) \cos \omega_j^t - T} & \text{za } C(\theta, \omega_j^t) \cos \omega_j^t \geq T \end{cases} \quad (5.3.5)$$

$$j = 1, n-1; \quad j+t < n; \quad t = 1, 2, \dots$$

$$\text{gdje su: } A(\theta, \gamma_j^t) = K_{j, j+t}(\theta + \gamma_j^t); \quad B(\theta, \alpha_j^t) = K_{j, j+t}(\theta + \alpha_j^t) \quad (5.3.6)$$

$$C(\theta, \omega_j^t) = K_{j, j+t}(\theta + \omega_j^t); \quad T\text{-period } n\text{-rednog smještaja.}$$

Primjedba 2.

Ako je $t = 1$, onda sve udaljenosti d_j medju redovima zavise samo od dva parametra, tj.

$$d_j = d_j^*(\theta, \gamma_j^*) = d_j[\theta, \gamma_j, \gamma_j(\omega_j)] \quad (5.3.7)$$

a takvi slučajeви su u praksi najinteresantniji.

Primjedba 3.

Ako imamo diranje objekata koji se nalaze u istom redu, tj. vrijedi (5.2.20), onda je period T dan izrazom

$$T = \max\{K_{11}(\theta), K_{22}(\theta), \dots, K_{nn}(\theta)\} \quad \theta \in [0, 2\pi],$$

odnosno pomoću R-disjunkcije

$$T = \bigvee_{j=1}^m K_{jj}(\theta) \quad (5.3.8)$$

Primjedba 4.

Kada u R_j uvjet (5.2.20) nije ispunjen a imamo mrežu diranja D_{13} ili D_{14} , onda je period T , dan izrazima:

$$T = \begin{cases} A(\theta, \gamma_j^t) \cos \gamma_j^t + C(\theta, \omega_j^t) \cos \omega_j^t & \text{za } D_{13} \\ B(\theta, \alpha_j^t) \cos \alpha_j^t - A(\theta, \gamma_j^t) \cos \gamma_j^t & \text{za } D_{14} \end{cases} \quad (5.3.9)$$

$$j = 1, n-1; \quad j+t < n; \quad t = 1, 2, \dots$$

Primjedba 5.

Ako (5.2.20) nije ispunjen a u R_j imamo D_{20} ili D_{21} , a u svim potparalelogramima $R_{j+1}, R_{j+t}, R_{j+t-1}$ imamo mrežu diranja D_{12} , onda je veličina perioda T dana sa:

$$T = \begin{cases} \sum_{j=j}^{j+t-1} K_{s,s+1}(\theta + \gamma_s) \cos \gamma_s + K_{j,j+1}(\theta + \omega_j^t) \cos \omega_j^t & \text{za } D_{20} \text{ u } R_j^t \\ K_{j,j+t}(\theta + \alpha_j^t) \cos \alpha_j^t + \sum_{s=j}^{j+t-1} K_{s,s+1}(\theta + \gamma_s) \cos \gamma_s & \text{za } D_{21} \text{ u } R_j \end{cases} \quad (5.3.10)$$

$$j = 1, n-1; \quad j+t < n; \quad t = 1, 2$$

Primjedba 6.

Kod dvorednog rasporeda dva različita objekta, tj. $L(2,2)$, mreže D_{12} do D_{24} ne dolaze u obzir. Naime, za ovaj slučaj bilo da nije određen period (D_{19} do D_{21} i D_{24}) bilo da nije određena udaljenost medju redovima (D_{22} , D_{24}) u potparalelogramu R_1

Primjedba 7.

Kod smještaja $L(n,m)$ period T se određuje pomoću (5.3.8).

Primjedba 8.

Zbog primjedbe 1, formalno uvjet da se objekti u R_j^t međusobno ne presjecaju možemo napisati:

$$\rho_{j,j+t}(s_j^j, s_{j+t}^{j+t}) \geq 0$$

$$\rho_{j,j+n}(s_j^j, s_j^{j+n}) \geq 0$$

$$\rho_{j,j+n+t}(s_j^j, s_{j+t}^{j+n+t}) \geq 0$$

$$\rho_{j+t,j+n+t}(s_{j+t}^{j+t}, s_{j+t}^{j+n+t}) \geq 0$$

$$\rho_{j+n,j+t}(s_j^{j+n}, s_{j+t}^{j+t}) \geq 0$$

$$\rho_{j+n,j+n+t}(s_j^{j+n}, s_{j+t}^{j+n+t}) \geq 0$$

$$j = 1, n-1; \quad j+t < n; \quad t = 1, 2, \dots$$

(5.3.11)

Prednji uvjet nepresjecanja da se napisati pomoću kontakt funkcije.

$$\rho_{j,j+t}(O_j, O_{j+t}) - K_{j,j+t}(\theta + \gamma_j^t) \geq 0$$

$$\rho_{j,j+n}(O_j, O_{j+n}) - T \geq 0$$

$$\rho_{j,j+t+n}(O_j, O_{j+n+t}) - K_{j,j+t}(\theta + \gamma_j^t) \geq 0$$

(5.3.12)

$$\rho_{j+t, j+n+t} (O_{j+t}, O_{j+n+t}) - T \geq 0$$

$$\rho_{j+n, j+t} (O_{j+n}, O_{j+t}) - K_{j, j+t} (\theta + \omega_j^t) \geq 0$$

gdje je T-period koji se u zavisnosti od tipa mreže D_i određuje jednim od izraza (5.3.8) - (5.3.10).

U zavisnosti od toga koja je duga R_j^t negenerirana, imamo različite izraze za koordinate vrhova R_j^t .

a) ako je nedegenerirana duga $(j, j+t)$ onda su koordinate dane sa:

$$x'_j = 0; x'_{j+t} = A(\theta, \gamma_j^t) \cos \gamma_j^t; x'_{j+n+t} = x_{j+t} + T; x'_{j+n} = T \quad (5.3.13)$$

$$y'_j = 0; y'_{j+t} = A(\theta, \gamma_j^t) \sin \gamma_j^t; y'_{j+n+t} = y'_{j+t}; y'_{j+n} = 0$$

b) ako je nedegenerirana duga $(j+n; j+t)$, imamo:

$$x'_j = 0; x'_{j+t} = x_{j+n+t} - T; x'_{j+n+t} = B(\theta, \alpha_j^t) \cos \alpha_j^t; x'_{j+n} = T \quad (5.3.14)$$

$$y'_j = 0; y'_{j+t} = y_{j+n+t}; y'_{j+n+t} = B(\theta, \alpha_j^t) \sin \alpha_j^t; y'_{j+n} = 0$$

c) ako je nedegenerirana duga $(j+n; j+t)$, imamo:

$$x'_j = 0; x'_{j+t} = T - C(\theta, \omega_j^t) \cos \omega_j^t; x'_{j+n+t} = x_{j+t} + T; x'_{j+n} = T \quad (5.3.15)$$

$$y'_j = 0; y'_{j+t} = C(\theta, \omega_j^t) \sin \omega_j^t; y'_{j+n+t} = y'_{j+t}; y'_{j+n} = 0$$

Što se tiče granica promjene kutnih parametara, imamo dva slučaja:

A) ako imamo presjecanje, tj. $R_j \cap R_{j+t} \neq \emptyset$,

onda imamo općenito:

$$0 \leq \theta, \gamma_j^t, \omega_j^t, \alpha_j^t \leq 2\pi$$

B) ako imamo $R_j \cap R_{j+t} = \emptyset$, onda je

$$\begin{aligned} 0 &< \theta < 2\pi \\ 0 &< \gamma_j^t, \omega_j^t, \alpha_j^t < \pi \end{aligned} \quad (5.3.16)$$

Prema tome ako je $R_j \cap R_{j+t} = \emptyset$, tada uvjete međusobnog nepresjecanja (5.3.12) objekata smještenih u vrhovima R_j^t za sve u početku nabrojene probleme I, II, III, IV i V možemo napisati pomoću izraza:

$$\begin{aligned} x'_{j+t}(\theta, \gamma_j^t, \omega_j^t) - K_{j,j+t}(\theta + \gamma_j^t) \cos \gamma_j^t &> 0 \\ y'_{j+t}(\theta, \gamma_j^t, \omega_j^t) - K_{j,j+1}(\theta + \gamma_j^t) \sin \alpha_j^t &> 0 \end{aligned} \quad (5.3.17)$$

$$x'_{j+n} - T > 0$$

$$y'_{j+t}(\theta, \gamma_j^t, \omega_j^t) - K_{j,j+t}(\theta + \omega_j^t) \sin \omega_j^t > 0$$

$$j = 1, n-2; j+t < n; t = 1, 2, \dots$$

gdje su koordinate vrhova R_j , u zavisnosti koja je duga nedege-nerirana, dane izrazima (5.3.13) - (5.3.15).

Pojava simetrije

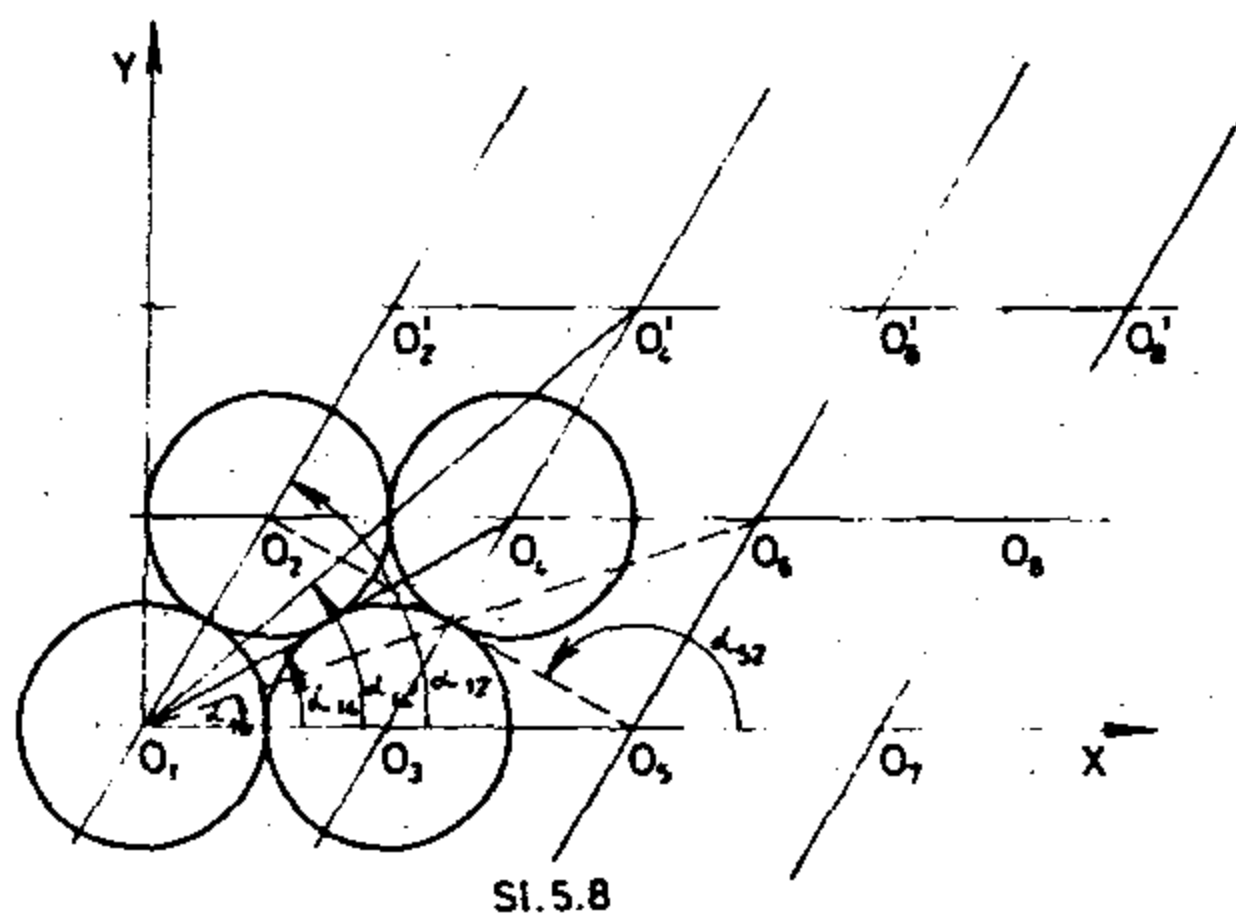
Vidimo da u stvari imamo dvije klase problema smještaja $L(n,m)$ i to, klasu gdje je $R_j \cap R_{j+t} \neq \emptyset$, i drugu klasu koju ćemo mi dalje promatrati, gdje je $R_j \cap R_{j+t} = \emptyset$, tj. gdje nema diranja među objektima u redu j , s objektima iz reda $(j+t)$ za $t \geq 2$.

Pogledajmo n -redni smještaj kada se smješta samo jedna vrsta objekata [40], tj. kada imamo $L(n,1)$. Vidjet ćemo da postoje neke vrlo važne veze ovog rasporeda sa rešetkastim rasporedom što se koristi za bitno smanjenje obima računanja.

Naime, recimo da imamo smještene rešetkaste $S_1 \in \{S\}$, onda se iz takvog rešetkastog rasporeda mogu generirati n -redni rasporedi o čemu je bilo riječi u Glavi III.

Kada smo iz rešetkastog rasporeda izdvojili neki raspored $L(n,1)$, onda tom rasporedu možemo pridružiti odgovarajuću kontakt mrežu kao i kontakt matricu.

Recimo da smo iz rešetkastog rasporeda izdvojili dvoredni raspored $L(2,1)$. Neka su poznati parametri rešetkastog rasporeda θ_1, γ , kut rebra $(1,2)$, koordinate pola $\theta_2(x_2, y_2)$, $T(\theta) = x_3$. Recimo da rešetkastom rasporedu odgovara mreža D_1 za osnovni paralelogram $O_1O_2O_3O_4$, sl.5.8.



Uvedimo oznake:

$$\begin{aligned}
 \alpha_{14} &= \angle O_4 O_1 O_3 & \alpha_{52} &= \angle O_2 O_5 O_7 \\
 \alpha_{14'} &= \angle O_4 O_1 O_3 & \alpha_{32'} &= \angle O_2' O_3 O_5 \\
 \alpha_{16} &= \angle O_6 O_1 O_3 & \alpha_{32} &= \angle O_2 O_3 O_5 \\
 \alpha_{12} &= \angle O_2 O_1 O_3
 \end{aligned} \tag{5.3.17}$$

Sve naprijed navedene parametre možemo odrediti pomoću parametara osnovnog paralelograma i to:

$$\begin{aligned}
 \alpha_{14} &= \arctg \frac{y_2}{x_2 + x_3} & \alpha_{32'} &= \pi - \arctg \frac{y_2}{x_3 - 2x_2} \\
 \alpha_{14'} &= \arctg \frac{y_2 + T}{2x_2 + T} & \alpha_{32} &= \omega \\
 \alpha_{16} &= \arctg \frac{y_2}{x_2 + 2x_3} & \alpha_{12} &= \gamma \\
 \alpha_{52} &= \pi - \arctg \frac{y_2}{2x_3 - x_2}
 \end{aligned}$$

Pokazuje se da u zavisnosti od tipa mreže D_1 , koja se promatra, na osnovu nadjenih parametara za taj raspored možemo naći od tri do osam novih raspoređa za različite $\theta \in [0, 2\pi]$ koji su jednako dobri a radimo pri tom samo sa parametrima već nadjenog raspoređa. Ova činjenica bitno utječe na smanjenje rada računara.

U tabeli 5.2. vidimo parametre za sedam dodatnih n -raspoređa izvedenih iz parametara polaznog raspoređa, tj. za slučaj mreže D_1 .

Ako je rednost $n > 2$, biti će i više mogućih pojaseva. Dakle, broj mogućih pojaseva zavisi od vrste mreže.

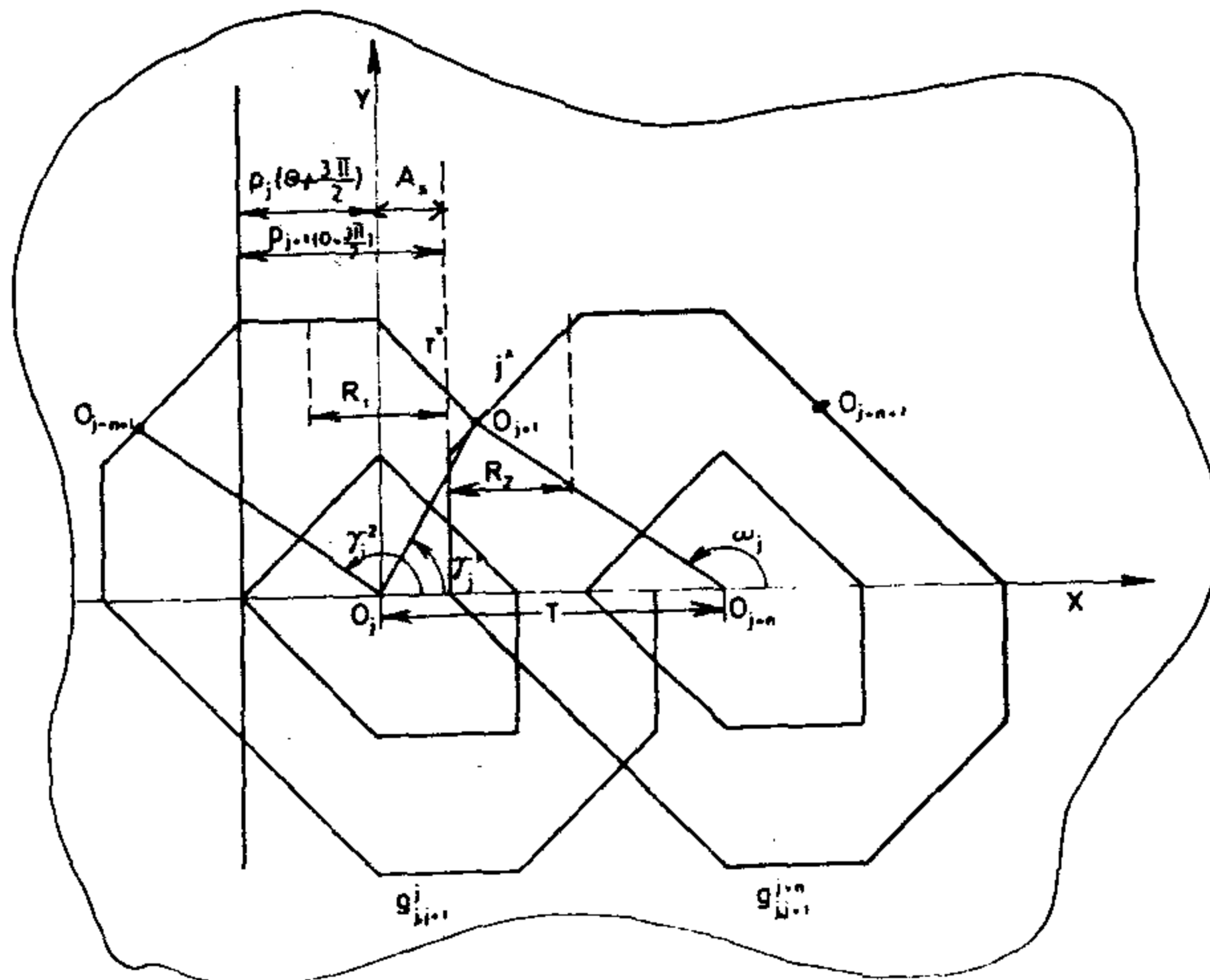
Za $n = 3$ pokazuje se da pored osam smjerova iz dvorednog raspoređa imamo još dodatnih osam smjerova; za $n = 4$ imamo osam dodatnih osim onih za $n = 3$, tj. 24; za $n = 5$ imamo 40 i za $n = 6$ imamo 48 mogućih pojaseva. Dalji slučajevi nisu ni interesantni jer se u praksi ne pojavljuju.

JASEVI	KUT ZAOKRETA		PERIOD SMJEŠTAJA	KOORDINATE POLA 0 2	ŠIRINA POJASA H
	FIGURE	DUGE (1, 2)			
T ₁₃	θ	γ	$K(\theta)$	$x_2^{13} = K(\theta + \gamma) \cos \gamma$ $y_2^{13} = K(\theta + \gamma) \sin \gamma$	$p_1(\theta) + y_2^{13} + p_1(\theta + \pi)$
T ₁₄	$\theta + \alpha_{14}$	$\gamma - \alpha_{14}$	$\frac{y_2^{13}}{\sin \alpha_{14}}$	$x_2^{14} = K(\theta + \gamma) \cos(\gamma - \alpha_{14})$ $y_2^{14} = K(\theta + \gamma) \sin(\gamma - \alpha_{14})$	$p_1(\theta + \alpha_{14}) + y_2^{14} + p_1(\theta + \alpha_{14} + \pi)$
T ₁₂	$\theta + \gamma$	$\omega - \gamma$	$K(\theta + \gamma)$	$x_2^{12} = K(\theta + \omega) \cos(\omega - \gamma)$ $y_2^{12} = K(\theta + \omega) \sin(\omega - \gamma)$	$p_1(\theta + \gamma) + y_2^{12} + p_1(\theta + \gamma + \pi)$
T ₃₂	$\theta + \omega$	$\pi - \omega$	$\frac{y_2^{13}}{\sin(\pi - \omega)}$	$x_2^{32} = K(\theta) \cos(\pi - \omega)$ $y_2^{32} = K(\theta) \sin(\pi - \omega)$	$p_1(\theta + \omega) + y_2^{32} + p_1(\theta + \omega + \pi)$
T ₅₂	$\theta + \alpha_{52}$	$\pi - \alpha_{52}$	$\frac{y_2^{13}}{\sin(\pi - \alpha_{52})}$	$x_2^{52} = K(\theta) \cos(\pi - \alpha_{52})$ $y_2^{52} = K(\theta) \sin(\pi - \alpha_{52})$	$p_1(\theta + \alpha_{52}) + y_2^{52} + p_1(\theta + \alpha_{52} + \pi)$
T _{32'}	$\theta + \alpha_{32'}$	$\omega - \alpha_{32'}$	$\frac{2y_2^{13}}{\sin \alpha_{32'}}$	$x_2^{32'} = K(\theta + \omega) \cos(\omega - \alpha_{32'})$ $y_2^{32'} = K(\theta + \omega) \sin(\omega - \alpha_{32'})$	$p_1(\theta + \alpha_{32'}) + y_2^{32'} + p_1(\theta + \alpha_{32'} + \pi)$
T _{14'}	$\theta + \alpha_{14'}$	$\gamma - \alpha_{14'}$	$\frac{2y_2^{13}}{\sin \alpha_{14'}}$	$x_2^{14'} = K(\theta + \gamma - \alpha_{14'}) \cos(\gamma - \alpha_{14'})$ $y_2^{14'} = K(\theta + \gamma - \alpha_{14'}) \sin(\gamma - \alpha_{14'})$	$p_1(\theta + \alpha_{14'}) + y_2^{14'} + p_1(\theta + \alpha_{14'} + \pi)$
T ₁₆	$\theta + \alpha_{16}$	$\alpha_{14} - \alpha_{16}$	$\frac{y_2^{13}}{\sin \alpha_{16}}$	$x_2^{16} = K(\theta + \alpha_{14}) \cos(\alpha_{14} - \alpha_{16})$ $y_2^{16} = K(\theta + \alpha_{14}) \sin(\alpha_{14} - \alpha_{16})$	$p_1(\theta + \alpha_{16}) + y_2^{16} + p_1(\theta + \alpha_{16} + \pi)$

Gra nice promjena parametara

Pokazuje se da kutevi $\gamma = \gamma(\gamma_1, \gamma_2, \dots, \gamma_{n-1})$; $\omega = \omega(\omega_1, \omega_2, \dots, \omega_{n-1})$ praktično imaju znatno uže granice od onih danih u (5.3.16). Naime, to slijedi iz konstrukcije $L(n, m)$. Recimo da su R_{j-1} već konstruirani i da se konstruira R_j i to pomoću hodografa kontakt funkcije. To znači da već imamo red j na kome su smješteni objekti S_j s periodom T , sl.5.9. Polovi HKF su smješteni na istom pravcu kao i objekti S_j u točkama O_{j+in} . Neka se $g_{j,j+1}^{j+(i-1)n}$ i $g_{j,j+1}^{j+2n}$ presjecaju i formiraju neku oblast za koju pretpostavimo da je jednostruko suvisla (općenito nije). Uočimo točke O_{j-n+1} , O_{j+1} , O_{j+n+1} presjeka hodografa sa maksimalnim ordinatama. Kada objekt S_j i S_{j+1} nisu konveksni, onda općenito imamo više presjeka pa otuda i različite vrijednosti ordinata y_{j+1}^i . Ako su objekti konveksni, onda je HKF konveksan i točna presjeka je jedinstvena, pa je i y_{j+1} jedinstvena.

Ordinate točaka O_{j-n+1} , O_{j+1} , O_{j+n+1} su jednake, jer su polovi objekata smješteni na istom pravcu, a u jednom redu su samo objekti jedne vrste i k tome smješteni periodično. Udaljenost medju tim točkama je konstantna i jednaka periodu T .



Sl. 5.9

Moguća mjesta polova objekata S_{j+1} su na toj izlomljenoj liniji koja je dio HRF objekata S_j i S_{j+1} . Prema tome za γ_j vrijedi

$$\bar{\gamma}_j^1 \leq \gamma \leq \bar{\gamma}_j^2 \quad (5.3.19)$$

gdje je: $\bar{\gamma}_j^1 = \min \gamma_j^1$; $\bar{\gamma}_j^2 = \max \gamma_j^2(\theta)$.

U stvari γ_j je kut vektora $\overrightarrow{O_j O_{j+1}}$ s početkom u točki O_j pomičnog sistema XO_jY i vrhova u O_{j+1} , tj. u točki gdje se sijeku hodografi $g_{j,j+1}^j$ i $g_{j,j+1}^{j+n}$; $\bar{\gamma}_j^2$ - je u stvari najveći mogući γ_j i on je jednak kutu $\omega = \angle O_{j+1}, O_{j+n}, O_{j+2}$, $\theta \in [0, 2\pi]$.

Zbog toga što je $\bar{\gamma}_j^1$ određuje kut vektora $\overrightarrow{O_j O_{j+1}}$ sa osi $O_j X$, možemo ga odrediti kao korijen jednadžbe [40].

$$K_{j,j+1}(\theta + \gamma_j) \sin \gamma_j - K_{j,j+1}(\theta + \omega_j) \sin \omega_j^* = 0 \quad (5.3.20)$$

gdje je:

$$\omega_j^* = \pi - \arctg \frac{K_{j,j+1}(\theta + \gamma_j) \sin \gamma_j}{T(\theta) - K_{j,j+1}(\theta + \gamma_j) \cos \gamma_j} \quad (5.3.21)$$

$$(j = 1, \dots, n-1)$$

Zbog uzajamne veze γ_j i ω_j , sl.5.9, slijedi da ω_j zadovoljava sistem nejednadžbi

$$\bar{\gamma}_j^2 \leq \omega_j < \omega_j^* \quad (5.3.22)$$

Prema tome očito da su granice γ_j , ω_j manje od intervala $[0, \pi]$. Što se tiče svojstava funkcija $\gamma_j^1(\theta)$ odnosno $\gamma_j^2(\theta)$, ona su navedena kod rešetkastog rasporeda, a ostala će biti navedena uz svaki pojedini problem.

4. Formuliranje tehnoloških ograničenja pomoću KF PF i R-funkcija

Uzmemo li izraz (5.2.13) širinu beskonačne trake (Problem 1) možemo definirati pomoću KF i PF u obliku:

$$h = p_1(\theta) + \sum_{j=1}^{n-1} d_j(\theta, \gamma_j, \omega_j) + p_n(\theta + \pi) \quad (5.4.1)$$

gdje: $d_j(\theta, \gamma_j, \omega_j)$ udaljenost između j i $j+1$ reda objekata S_j i S_{j+1} , $p_1(\theta)$; $p_n(\theta)$ - potporne funkcije.

Na širinu se često postavlja i uvjet kratnosti, tj. da vrijedi

$$\bar{h} = \varepsilon \left[N \left[\frac{h}{\delta} \right] + 1 \right]$$

gdje: δ - koeficijent kratnosti, obično je 5 ili 10. Ovo uvođenje kratnosti, pored tehnološke strane, ima za posljedicu da proračuni bitno kraće traju.

Pored toga postoje ograničenja na širinu trake u vezi sa širinom radnog stola štampanja H . Taj se uvjet može zapisati:

$$\delta \left[N \left[\frac{p_1(\theta) + \sum_{j=1}^{n-1} d_j(\theta, \gamma_j, \omega_j) + p_n(\theta + \pi)}{\delta} \right] + 1 \right] \leq H. \quad (5.4.2)$$

Ako udaljenost među objektima koji se istovremeno štampaju nije manja od (Kx_s) , $K \in [5, 10]$, onda za R_j tehnološka ograničenja oblika (5.2.28) možemo izraziti pomoću R-funkcija i (5.4.2) jednačbom

$$\bigwedge_{j=1}^{n-1} (c_1^j \vee_1 c_2^j \vee_1 c_3^j) \wedge c_4 = 1 \quad (5.4.3)$$

gdje:

$$c_1^j = \left\{ \text{Sign} \left[p(O_j, O_{j+1}) - K_{j,j+1}(\theta + \gamma_j) - K \cdot s \right] \cdot \text{Sign} \left[B - p\left(\theta - \frac{\pi}{2}\right) - p_{j+1}\left(\theta + \frac{\pi}{2}\right) - K_{j,j+1}(\theta + \gamma_j) \cos \gamma_j \right] \right\}$$

$$C_2^j = \{ \text{Sign}[\rho(O_j, O_{j+n}) - K_{j,j+1}(\theta + \alpha_j) - K \cdot s] \cdot \text{Sign}[B - p_j(\theta - \frac{\pi}{2}) - p_{j+1}(\theta + \frac{\pi}{2}) - K_{j,j+1}(\theta + \alpha_j) \cos \alpha_j] \},$$

$$C_3^j = \{ \text{Sign}[\rho(O_{j+1}, O_{j+n}) - K_{j,j+1}(\theta + \omega_j) - K \cdot s] \text{Sign}[B - p_j(\theta + \frac{\pi}{2}) - p_{j+1}(\theta - \frac{\pi}{2}) - K_{j,j+1}(\theta + \omega_j) \cos \alpha_j] \}$$

$$C_4 = \text{Sign}[H - \bar{h}]$$

$$\text{Sign}(X_{j+1}) = \begin{cases} 0 & \text{za } x_{j+1} \leq 0 \\ 1 & \text{za } x_{j+1} > 0 \end{cases} \quad (5.4.4)$$

$$\alpha = \text{arc tg} \frac{K_{j,j+1}(\theta + \gamma_j) \sin \gamma_j}{K_{j,j+1}(\theta + \gamma_j) \cos \gamma_j + T(\theta)} \quad (5.4.5)$$

$$\gamma_j \in (0, \frac{\pi}{2}); \quad \omega_j \in (0, \pi); \quad \alpha_j \in [0, \frac{\pi}{2}); \quad \theta \in [0, 2\pi];$$

V_1 - operacija R-disjunkcije, $x_{j,j+1}$ - apscisa pola O_{j+1} .

$B \times H$ - dopustive granice radnog polja štampanja,

$\text{Max}T$ - granična vrijednost perioda n -rednog smještanja.

Vrlo često kod smještanja objekata funkcija cilja je mnogoekstremalna pa se dešava da neka od njih zadovoljavaju i tehnološka ograničenja dok druga ne zadovoljavaju. Zato se često problemi rješavaju kao da tehnoloških ograničenja nema, a onda kad se rješenja nadju, vrši se provjera da li zadovoljavaju ograničenja.

Što se tiče ograničenja na udaljenost medju objektima njih se uključi u same KF i PF.

Vidimo da navedena ograničenja za traku vrijede i kod rješenja problem IV i IV-I s tim da se sam (5.4.2) modificira u:

$$\delta \left[N \left[\frac{p_1(\theta) + d_j(\theta, \delta_j, \omega_j) + p_n(\theta + \pi) + s}{\delta} \right] + 1 \right] \leq H, \quad (5.4.6)$$

jer se pravokutnik Ω reže u ^{s razmaknutim figurama} pojaseve, pa se na to troši jedan dio. ($t = 2u + v$) Sl. 5.7.

5. Formulacija problema smještanja pomoću KF, PF i R-funkcija specifičnosti i realizacija pomoću kompjuterskih programa

Općenito rješenje $L(n,m)$ problema uključuje uvjete na objekte u jednom redu (5.2.20), uvjete diranja medju redovima (5.2.22) kao i tehnološka ograničenja (5.2.23) koje smo koristeći KF, PF i R-funkcije definirali u odjeljku 4 ove glave. Uvjeti da se objekti smještaju u oblasti sadržani su već u samoj funkciji cilja, vidi (5.2.13), (5.2.19). Prema tome kod beskonačne trake kao i kvadrata treba pripadne funkcije cilja (5.2.13), (5.2.19) definirati kao funkcije parametara smještaja objekata. Ako se radi o beskonačnoj traci, onda izraz (5.2.13) treba izraziti pomoću kontakt i potporne funkcije.

Iz svojstva potporne funkcije slijedi da je udaljenost d_0 , od donjeg potpornog pravca p_1 dana sa $d_0 = p_1(\theta)$, a udaljenost od gornjeg potpornog pravca do pola objekta S_n sa $d_0^* = p_n(\theta + \pi)$. Parametri smještaja su izraženi pomoću (5.3.2) - (5.3.5), a u zavisnosti od tipa mreže D_i , sa (5.3.8) - (5.3.10) definirali smo period smještaja, pa funkcija cilja za beskonačnu traku glasi:

$$P[R(\theta, \gamma, \omega)] = [p_1(\theta) + \sum_{j=1}^{n-1} |d_j(\theta, \gamma, \omega)| + p_n(\theta + \pi)] \cdot T(\theta) \quad (5.5.1)$$

gdje

$$\gamma = \gamma(\gamma_1, \gamma_2, \dots, \gamma_{n-1}); \quad \omega = \omega(\omega_1, \omega_2, \dots, \omega_n).$$

Prema tome rješavanje problema smještaja u beskonačnu traku se svodi na odredjivanje

$$P^*[R(\theta, \gamma, \omega)] = \min_{\theta, \gamma, \omega \in \Gamma} P[R(\theta, \gamma, \omega)] \quad (5.5.2)$$

gdje je oblast Γ - definirana sa (5.3.16) i (5.3.17).

Zbog nekih zajedničkih svojstava smještaja u beskonačnu traku i pravokutnik, ovdje odmah dajemo i analitički izraz za funkciju cilja (5.2.18) za smještaj u pravokutnik.

Pošto se u pravokutniku pojasevi mogu smještati paralelno stranici a kao i b , treba izračunati oba koeficijenta iskorištenja μ_a, μ_b .

Njih izrazimo pomoću KF; PF i R-funkcija, tj. kao funkciju parametara θ, γ, ω . Uzevši u obzir kako smo definirali period T sa (5.3.9)-(5.3.10) i izraze za koordinate vrhova potparalelograma R_j u zavisnosti od toga koja je duga u R_j nedegenerirana (5.3.13) - (5.3.15) kao i uvjeta da objekt S_j u redu j -om dira lijevu stranu pojasa, to izraz za funkciju cilja glasi:

$$\begin{aligned} u_a = & \frac{1}{a \times b} \bigvee_{j=1}^n \left\{ \left[\sum_{k=1}^j P(S_k) \left[N_1 \left[\frac{a - p_j(\theta - \frac{\pi}{2}) - \sum_{i=1}^k [T(\theta) - x_i] - p_k(\theta + \frac{\pi}{2})}{T(\theta)} + 1 \right] + \right. \right. \right. \\ & + N_2 \left[\frac{p_j(\theta - \frac{\pi}{2}) + \sum_{i=1}^k [T(\theta) - x_i] - p_k(\theta - \frac{\pi}{2})}{T(\theta)} \right] \left. \right. \left. \right\} + \\ & + \sum_{k=j}^{n-1} P(S_k) \left[N_3 \left[\frac{a - \sum_{i=j}^k x_{i+1} - p_j(\theta - \frac{\pi}{2}) - p_{k+1}(\theta + \frac{\pi}{2})}{T(\theta)} + 1 \right] + \right. \\ & \left. + N_4 \left[\frac{p_j(\theta - \frac{\pi}{2}) + \sum_{i=j}^k x_{j+1} - p_k(\theta + \frac{\pi}{2})}{T(\theta)} \right] \right] \times N_5 \left[\frac{b}{p_1(\theta) + \sum_{j=2}^{n-1} y_j + p_n(\theta + \pi)} \right] \end{aligned} \quad (5.5)$$

gdje $\gamma = \gamma(\gamma_1, \gamma_2, \dots, \gamma_{n-1})$; $\omega = \omega(\omega_1, \omega_2, \dots, \omega_{n-1})$.

Pošto se vrlo često pojavljuju dvoredni smještaji, to je izraz (5.5.3) za dvoredni slučaj dan izrazom:

$$\begin{aligned} u_a = & \frac{1}{a \times b} \left\{ P(S_1) N_1 \left[\frac{a - p_1(\theta - \frac{\pi}{2}) - p_1(\theta + \frac{\pi}{2}) + T(\theta)}{T(\theta)} \right] + \right. \\ & \left. + P(S_2) N_2 \left[\frac{a - p_1(\theta - \frac{\pi}{2}) - K_{12}(\theta + \gamma) \cos \gamma - p_2(\theta + \frac{\pi}{2}) + T(\theta)}{T(\theta)} \right] \right\} V_1 \\ & V_1 \left\{ P(S_1) N_3 \left[\frac{a - p_2(\theta - \frac{\pi}{2}) - K_{12}(\theta + \gamma) \cos \gamma + T(\theta) - p_1(\theta + \frac{\pi}{2})}{T(\theta)} \right] + \right. \\ & \left. + P(S_2) N_4 \left[\frac{a - p_2(\theta - \frac{\pi}{2}) - p_2(\theta + \frac{\pi}{2}) + T(\theta)}{T(\theta)} \right] \right\} \times \\ & \times N_5 \left[\frac{b}{m N_b \left[\frac{p_1(\theta) + K_{12}(\theta + \gamma) \sin \gamma + p_2(\theta + \gamma) + m}{m} \right]} \right] \end{aligned}$$

Ako je $P(S_v)=1$ za \forall_v , onda izraz u brojniku predstavlja broj objekata koji se dobiju iz $a \times b$.

Opaska:

Broj objekata u jednom redu može se razlikovati od broja objekata u drugom redu samo za jedan.

Znači da se problem n -rednog smještanja u Ω svodi na nalaženje

$$\mu^* = \max_{\theta, \gamma, \omega \in \Gamma} (\theta, \gamma, \omega) = \max_{\theta, \gamma, \omega \in \Gamma} [\mu_a(\theta, \gamma, \omega) \bigvee_1 \mu_b(\theta, \gamma, \omega)] \quad (5.5.4)$$

Γ - definiran sa (5.3.16) i (5.3.17)

Iz analitičkih izraza (5.5.2) i (5.5.4), te uzevši u obzir svojstva kontakt i potporne funkcije, nameću se slijedeći zaključci:

Problemi I - IV su problemi matematičkog programiranja ali toliko specifični da zahtijevaju specijalne metode rješavanja i to zbog slijedećih osobina:

1. Funkcije (5.5.2) i (5.5.4) imaju oblast definicije zadane sa sistemom nejednadžbi (5.3.17) koji je nelinearan, složen i glomazan. Naime, nelinearnost (5.3.17) slijedi iz nelinearnosti KF i izraza za parametre kuteva ω_j, γ_j , dane sa (5.3.3) - (5.3.5).

2. Funkcija (5.5.2) je nelinearna i po dijelovima glatka, što slijedi iz svojstva KF i PF, kao i samog izraza (5.5.2).

3. Funkcija cilja (5.5.4) je po dijelovima konstantna, jer vrijednost (5.5.3) promatramo za cijeli broj objekata smještenih u Ω .

4. Broj nejednadžbi koje definiraju Γ nije manji od $v=6n-5$, gdje je n -rednost smještanja. Naime, za formiranje jednog potparalelograma R_j treba realizirati četiri nejednadžbe kojima su zadani uvjeti međusobnog nepresjecanja objekata, a zatim tri nejednadžbe koje određuju granice promjena kutnih parametara (5.3.16). Pošto imamo $(n-1)$ potparalelogram, te da su objekti međusobno orijentirani, dobijemo $v=6(n-1)+1$. Pored toga možemo imati i razna tehnološka ograničenja.

5. Ako je $n=m$, onda se traženje optimuma funkcije cilja (5.5.2) za beskonačnu traku odvija u prostoru dimenzije $R^{(2n-1)}$ a za Ω u prostoru $R^{(2n+1)}$.

Mi se upravo zbog toga ograničavamo na slučaj gdje (5.2.22) nije ispunjen, tj. promatramo slučajeve samo diranja medju susjednim redovima.

Treba voditi računa da postoje oblasti Z_j gdje $P[R_j(\theta, \gamma_j, \omega_j)]$ ne postiže ekstreme (vidi rešetkasti smještaj formule (4.1.10), tako da ekstreme (5.5.2) treba tražiti na nešto užoj oblasti D , tj.

$$P^*(R) = \min_{\theta, \gamma, \omega \in D} P[R(\theta, \gamma, \omega)] \quad (5.5.5)$$

S obzirom da je

$$K_{j,j+1}(\theta + \bar{\gamma}_j^1) \sin \bar{\gamma}_j^1 = K_{j,j+1}(\theta + \bar{\gamma}_j^2) \sin \bar{\gamma}_j^2 \quad (5.5.6)$$

te zbog osobitosti funkcija $\bar{\gamma}_j^1(\theta)$ i $\bar{\gamma}_j^3(\theta)$, oblast D u kojoj je funkcija (5.5.2) definirana može se zapisati u slijedećem obliku:

$$- \bar{\gamma}_j^2 + \gamma_j^2(\theta) > 0$$

$$- \gamma_j^0 - \bar{\gamma}_j^1(\theta) \geq 0$$

$$2\pi - \theta \geq 0$$

$$\theta \geq 0 \quad j = (1, n-1). \quad (5.5.7)$$

Ako je Γ_j - oblast promjene γ_j i ω_j , onda se pokazuje da je općenito ova oblast nekonveksna, pa čak i višestruko suvisla. Ekstremi funkcije (5.5.2) i (5.5.4) mogu se postizati kako unutar Γ_j - tako i na rubu.

Pošto smo se mi ograničili na mogućnost medjurednog diranja samo medju susjednim redovima, to ima za posljedicu da je udalje-

nost medju redovima dana sa $d_j = d_j(\theta, \gamma_j)$. To uzrokuje da se potparalelogrami R_j i R_{j+1} mogu nezavisno konstruirati, tj. problem se svodi na $(n-1)$ nezavisne konstrukcije R_j . Iz navedenog slijedi da je funkcija cilja (5.5.1) za smještaj u beskonačni pojas oblika:

$$P[R(\theta, \gamma)] = [p(\theta) + \sum_{j=1}^{n-1} d_j(\theta, \gamma_j) + p_n(\theta + \pi)] xT(\theta). \quad (5.5.5)_a$$

Znači da se rješavanje navedenog problema svodi na nalaženje

$$P^*[R(\theta, \gamma)] = \min_{\theta, \gamma \in \Gamma} P[R(\theta, \gamma)] = \min_{\theta, \gamma \in \Gamma} [p_1(\theta) + \sum_{j=1}^{n-1} \min |d_j(\theta, \gamma_j)| + p_n(\theta + \pi)] xT(\theta) \quad (5.5.6)_a$$

gdje je: Γ - oblast definirana sa (5.3.16) i (5.3.17).

Na osnovi iznjetog isprobani su razni algoritmi pa je izabran jedan koji se na temelju opsežnih eksperimenata pokazao najbolji.

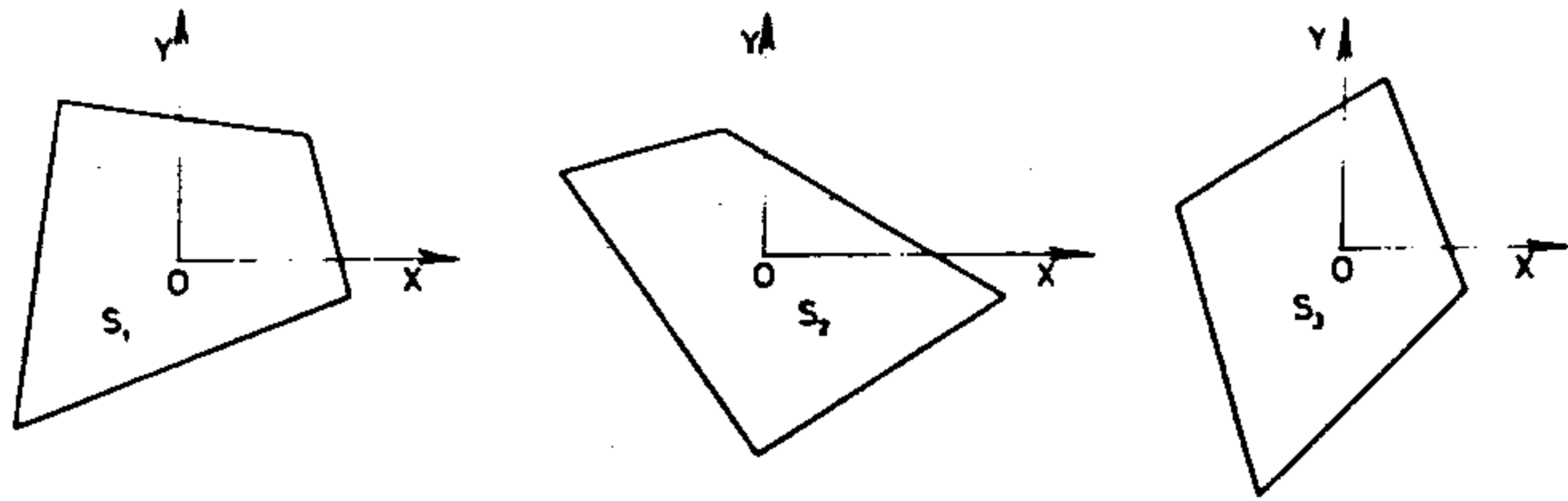
Na osnovi tog algoritma sastavljen je paket programa za računar "POJAS". Pošto su poznati rezultati da je za jednorodni periodični raspored elipsa $\mu = 0,785$ a za dvoredni $\mu = \frac{\pi}{2 + \sqrt{\pi}} = 0,841\dots$

naš program je dao iste rezultate.

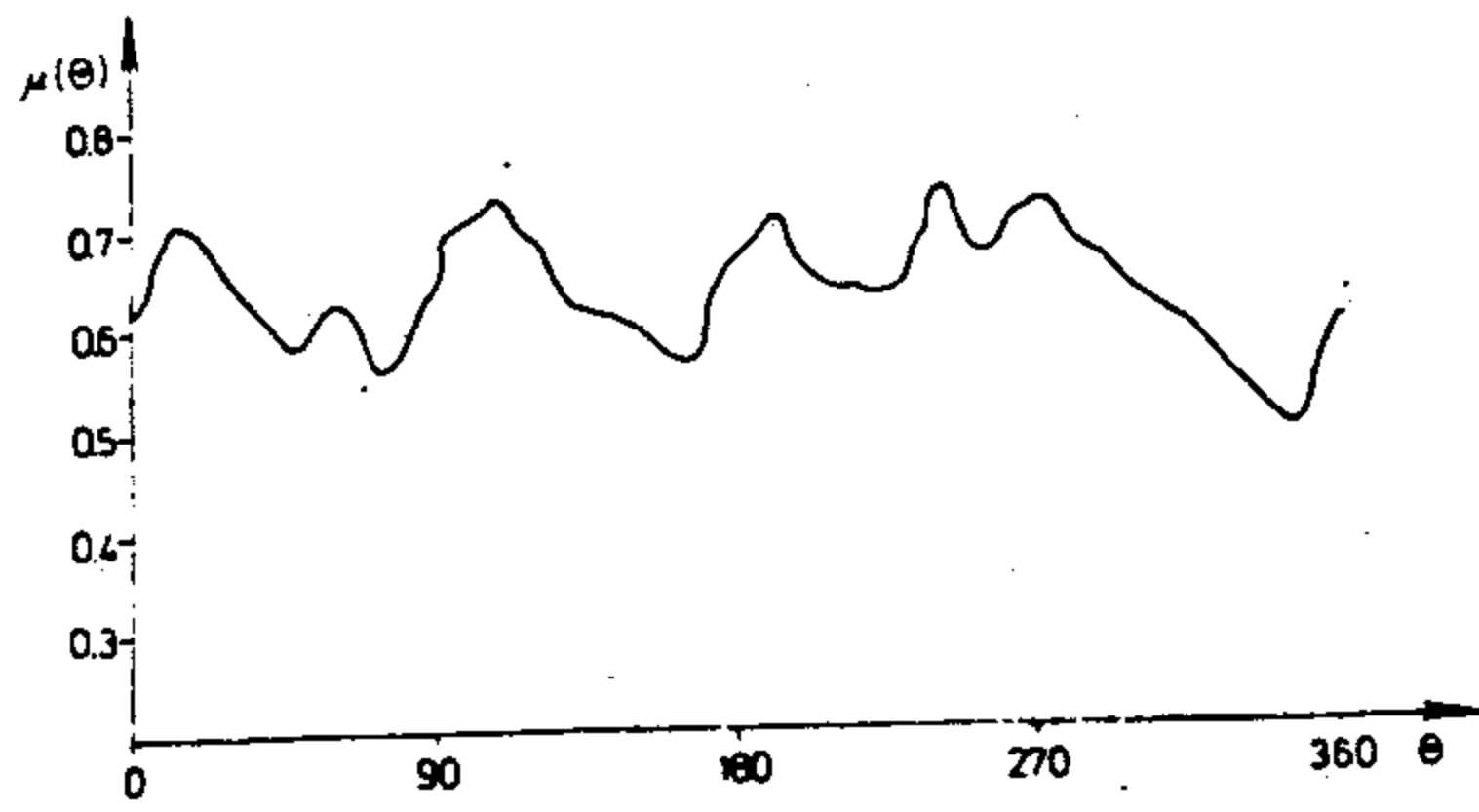
Na sl.5.10. vidimo tri objekta koje treba smjestiti troredno u pojas beskonačne duljine da otpad bude minimalan. Kako se mijenja koeficijent iskorištenja u zavisnosti od θ , vidimo na sl. 5.11. Optimalni raspored je prikazan na sl.5.12.

Parametri smještaju su:

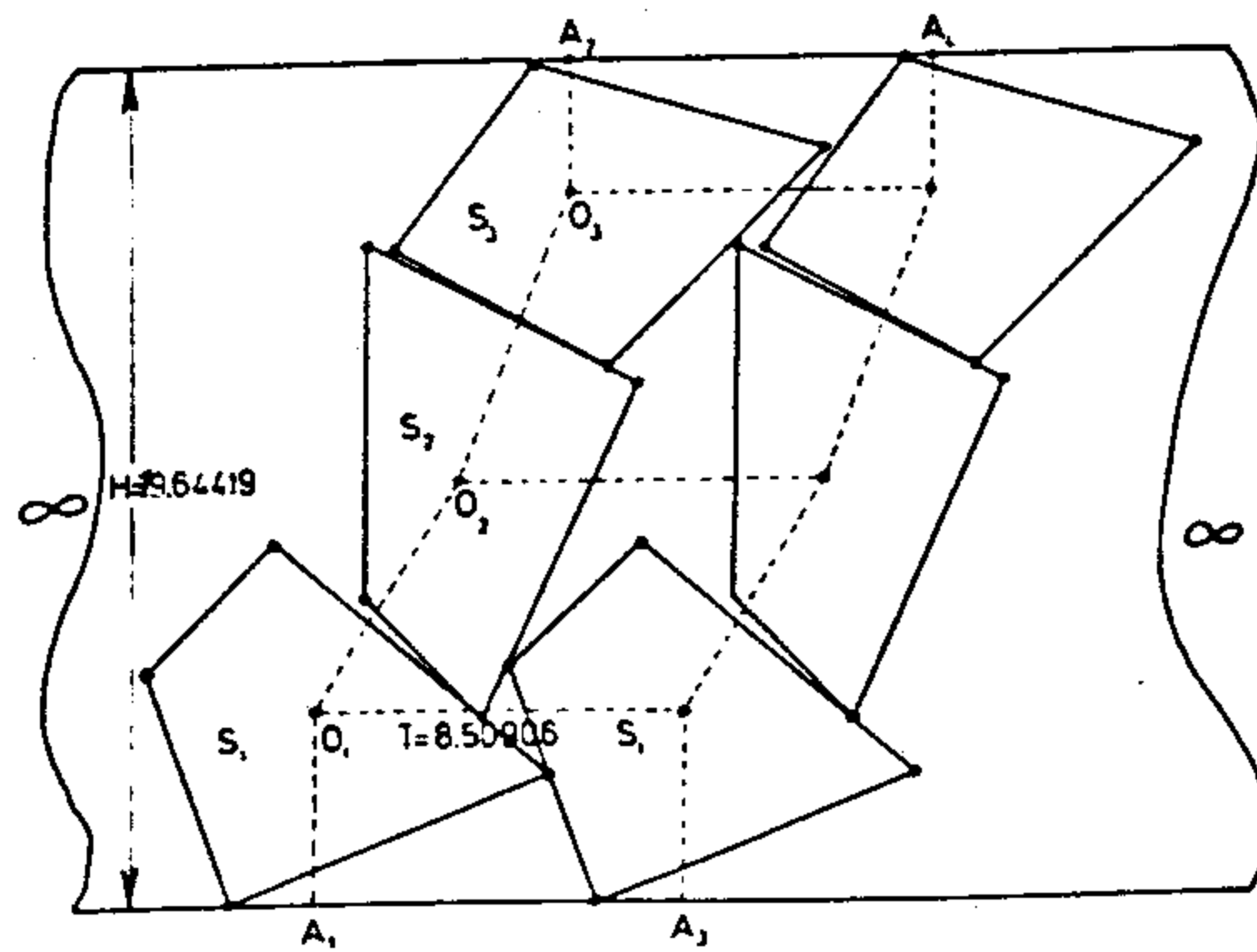
Optimalni $\theta^* = 242$; $T(\theta^*) = 8,50906$; $\mu(\theta^*) = 0,73286$;
 $H(\theta^*) = 19,64419$; $O_1(0,0)$; $O_2(3,28755; 5,08232)$; $O_3(2,63$;
 $6,79327)$. Minimalni koeficijent iskorištenja se realizira kod
 $\theta = 345^\circ$ a $\mu(345^\circ) = 0,49838$. Dakle $\Delta\mu(\theta) = 23,4\%$.



SI.5.10



SI.5.11



SI.5.12

Smještaj u kvadrat zadanih dimenzija

U odnosu na beskonačnu traku ovdje se pojavljuju dvije specifičnosti.

I specifičnost

Budući da je duljina trake ograničena, tj. jednaka a (b), to je broj objekata u redu " j " ograničen a to onda povlači ograničenje na položaj O_{j+1}^* , gdje se smješta pol objekata S_{j+1} . Naime, sa slike 5.9. vidimo da je:

$$\begin{aligned} E_2 &= E_{j+1} = a - C_{j+1} \cdot T \\ A_s &= p_{j+1}(\theta + \frac{3\pi}{2}) - p_j(\theta + \frac{3\pi}{2}); \quad \theta \in [0, 2\pi] \end{aligned} \quad (5.5.7)$$

gdje: a - duljina stranice pravokutnika Ω paralelno kojoj se smještaju pojasevi.

$$C_{j+1} = N \left[\frac{a - p_{j+1}(\theta + \frac{3\pi}{2}) - p_j(\theta + \frac{3\pi}{2})}{T} \right] \quad (5.5.8)$$

mogući broj objekata u $j+1$ redu; T -period koga se određuje pomoću (5.3.8), $p_{j+1}(\theta + \frac{3\pi}{2})$; $p_j(\theta + \frac{3\pi}{2})$ potporne funkcije objekata S_j i S_{j+1} . Pol svakako leži na izlomljenoj liniji O_{j+kn+1} , $O_{j+(k+1)n+1}$ ($k=1, 2, \dots$), a to znači da vrijedi:

$$A_s \leq x_{j+1}^* \leq E_{j+1} \quad (5.5.9)$$

Ako postoji ostatak E_j^* , koji leži u granicama

$$0 \leq E_j^* \leq B \quad (5.5.10)$$

gdje $B = \min_j E_j$, onda lijeva granica u (5.5.9) može biti smanjena. Ako ta mogućnost postoji, onda (5.5.9) možemo zapisati:

$$A_s - E_j^* \leq x_{j+1} \leq E_{j+1}.$$

U slučaju kada u Ω vršimo dvoredno smještanje dva objekta, onda je veličina ostatka dana sa

$$E_1^* = a - N \left[\frac{a - p_1(\theta + \frac{3\pi}{2}) - p_1(\theta + \frac{\pi}{2})}{T} \right] \times T .$$

II specifičnost

Treba odrediti broj pojaseva širine H_1 koje se smještaju u Ω gdje je H_1 dan izrazom

$$H_1 = p_1(\theta) + \sum_{j=1}^{n-1} y_j + p_n(\theta + \pi) \quad (5.5.12)$$

gdje y_j - udaljenost između redova j i $j-1$;

$p_1(\theta)$ - potporna funkcija od S_1 , $p_n(\theta + \pi)$ - potporna funkcija od S_n .

Označimo li sa v_a - broj pojaseva koji se mogu smjestiti u Ω kod paralelnosti sa stranicom a , onda je

$$v_a = N \left[\frac{b}{H_1} \right], \quad (5.5.13)$$

s tim da duljina stranice b mora zadovoljiti uvjet:

$$b = v_a H_1 \geq 0 . \quad (5.5.14)$$

Ako se radi o smještaju u pojaseve paralelne stranici a , koeficijent popunjenosti dan je sa:

$$\mu = \frac{v_a \sum \xi_j P(S_j)}{a \times b} \quad (\mu = \frac{v_a \sum \xi_j S_j^T}{\rho \times a \times b} .) \quad (5.5.15)$$

gdje: ξ_j - broj objekata S_j , $P(S_j)$ - površina objekata S_j , $a \times b$ površina Ω . U zagradi je dana druga formula za računanje koeficijenta iskorištenja,

gdje (S_j^T - težina objekta S_j , ρ - specifična težina materijala).

Vidimo da funkcija cilja, tj. koeficijent iskorištenja (5.5.15), zavisi od parametara svih potparalelograma tj.

$$\mu = \mu(\theta, \gamma_1, \gamma_2, \dots, \gamma_{n-1}) \quad (5.5.16)$$

pa se rješavanje svodi na računanje

$$\mu = \max_{\theta, \gamma \in \Gamma} \mu(\theta, \gamma) \quad (5.5.17)$$

Γ - oblast definirana sa (5.3.16), (5.3.17) i (5.5.11).

Ovdje su poteškoće kod nalaženja optimalnog rješenja u tome što funkciju cilja (5.5.16) ne možemo prikazati kao sumu nezavisnih funkcija od kojih bi svaka funkcija (sumand) zavisila od dva parametra.

S druge strane ovakve funkcije su jako nelinearne, ograničenja isto tako, a oblast definicije obično višestruko suvisla. Ekstremi se mogu postići unutar oblasti kao i na rubu. Sve to, kao i nepoznavanje boljeg načina (na čemu se radi) problem je riješen ipak dekompozicijom, pa se zato može govoriti o racionalnom a ne optimalnom rješenju. Međutim, provjerom mnogih brojnih lista iz prakse gdje se ovakvi problemi javljaju, konstatirano je da dobivena racionalna rješenja prevazilaze ona iz prakse gotovo u pravilu 5-10%, a ponekad i preko 20%. Rezultati koji slijede dobiveni su pomoću paketa programa A x B.

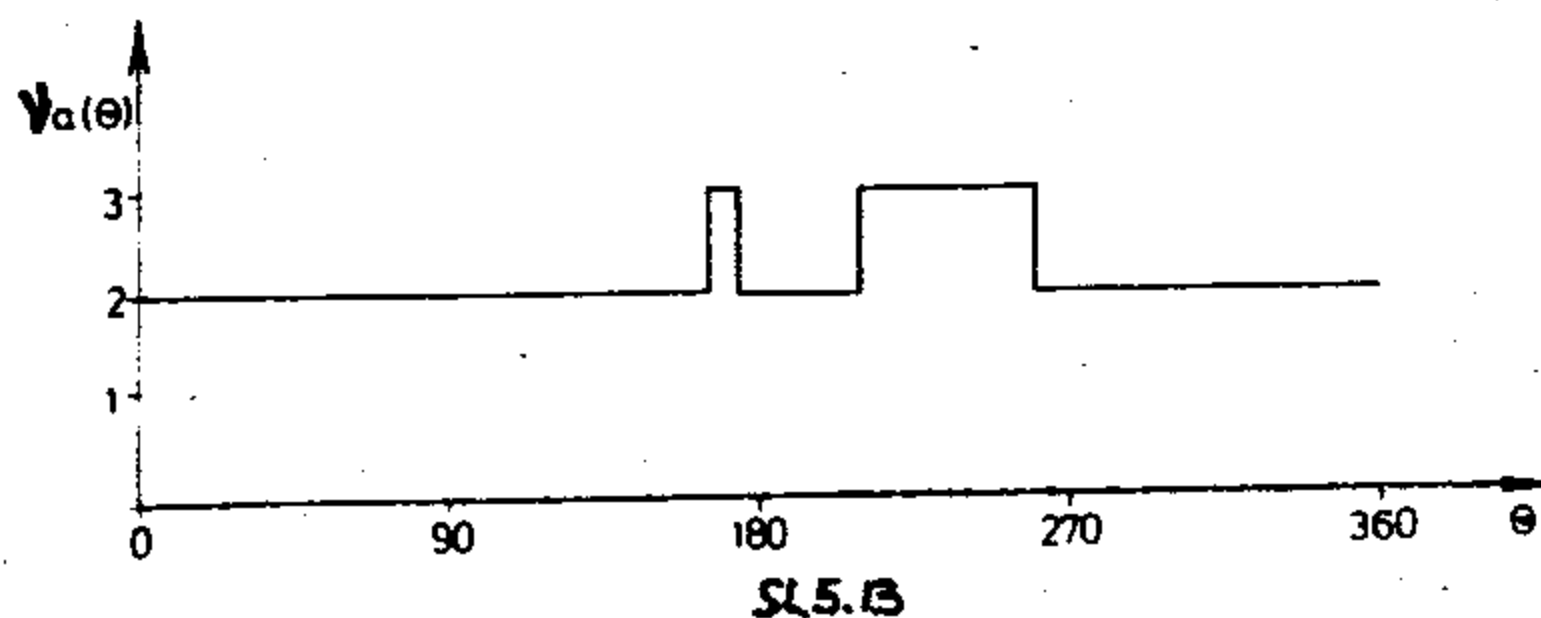
Primjer:

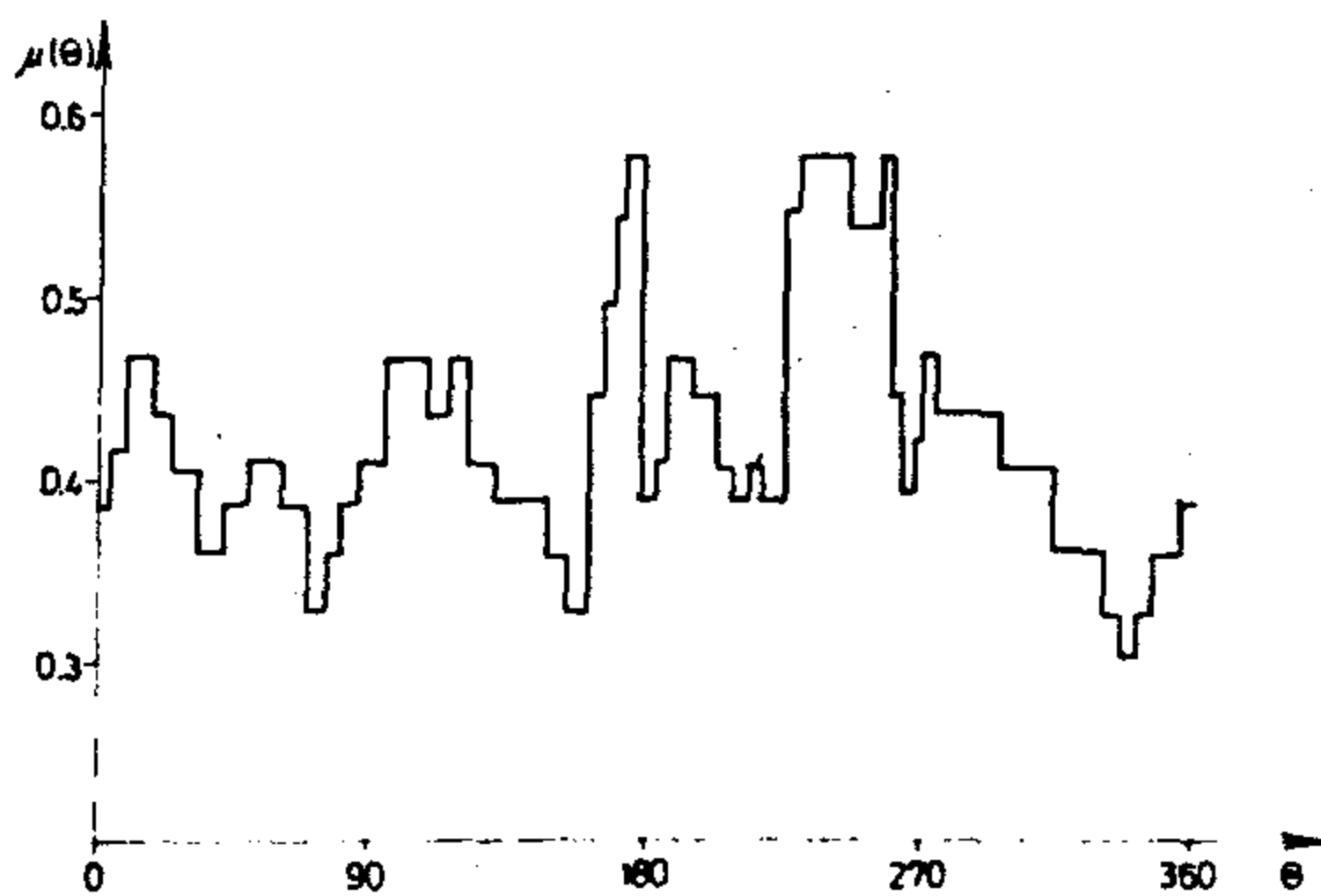
Objekte na sl.5.10. treba smjestiti troredno i periodički u pravokutnik dimenzije 50 x 60.

Kako se mijenja broj pojaseva u zavisnosti od θ , vidimo na sl.5.13.

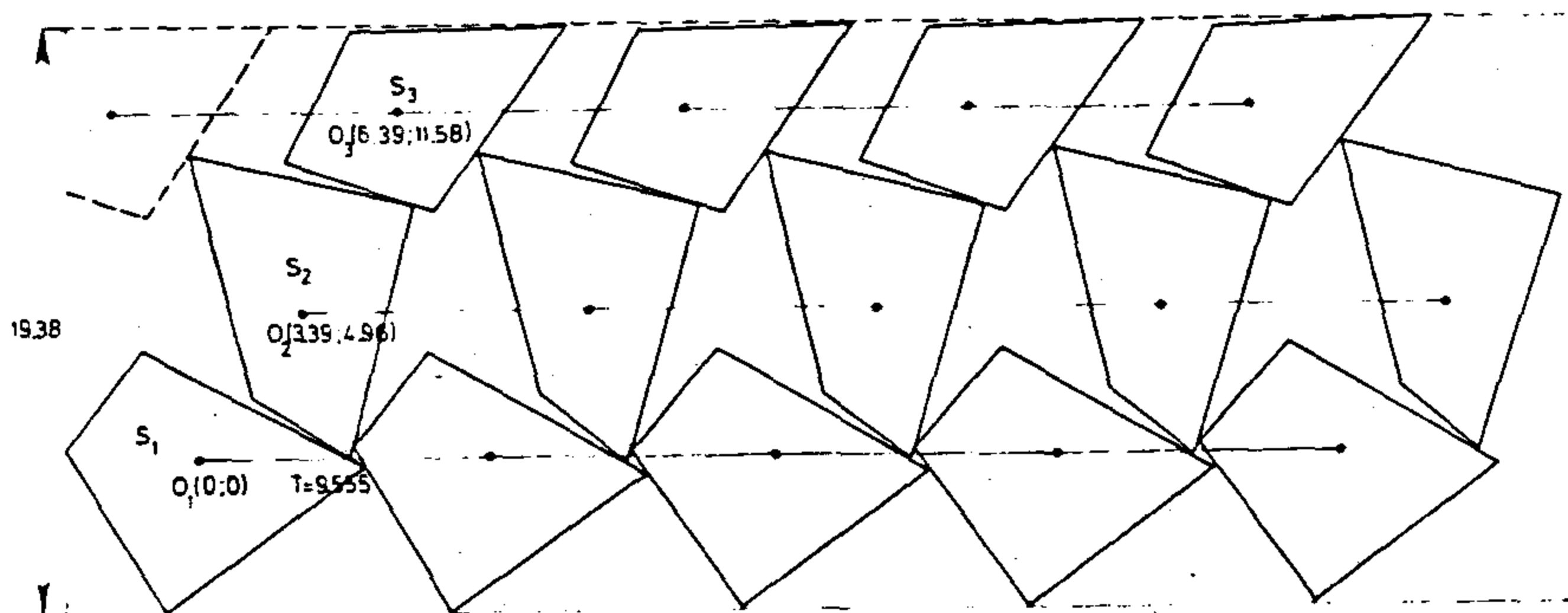
Promjenu koeficijenta iskorištenja s promjenom θ vidimo na sl. 5.14.

Optimalni raspored objekata prikazan je na sl.5.15.





Sl. 5.14



Sl. 5.15

Parametri smještaja: $\theta^* = 228^\circ$; $T(\theta^*) = 9,55539$;
 $\mu = (\theta^*) = 0,5745$; Broj pojaseva = 3; širina pojasa = 19,38;
 Broj elemenata u jednom pojasu $S_1 = 5$; $S_2 = 5$; $S_3 = 4$;
 Koordinate polova $O_1(0,0)$; $O_2(3,39, 4,96)$; $O_3(6,39; 11,58)$.

Smještaj u rolnu zadane širine

U praksi se često pojavljuje problem n-rednog smještanja u rolnu čija je širina zadana. Znači da se zadana rolna reže u pojaseve u svakom od kojih se zadani objekti smještaju n-redno. Širina pojasa određuje se na isti način kao u problemu smještanja u beskonačni pojas. Funkcija cilja je dana izrazom

$$\mu = \frac{m \sum_{j=1}^n P(S_j)}{H \times T} \quad (5.5.18)$$

gdje je: m - broj pojaseva smještenih u rolnu; $P(S_j)$ - površina objekta S_j ; n - broj redova u jednom pojasu, $T(\theta)$ - period izračunat pomoću (5.3.8); $H(\theta)$ - zadana širina rolne.

Prema tome treba naći takve parametre smještaja da bude

$$\mu^*(\theta^*) = \max_{\theta, \gamma \in \Gamma} \mu(\theta, \gamma) \quad (5.5.19)$$

gdje je: Γ - oblast definirana sa (5.3.16), (5.3.17).

Na osnovi razradjenog algoritma, u kome se pretpostavlja da imamo medjuredno diranje samo medju susjednim redovima, tj. da vrijedi (5.3.7), sačinjen je paket programa NR-ROLNA pomoću koga su dobiveni podaci koji slijede.

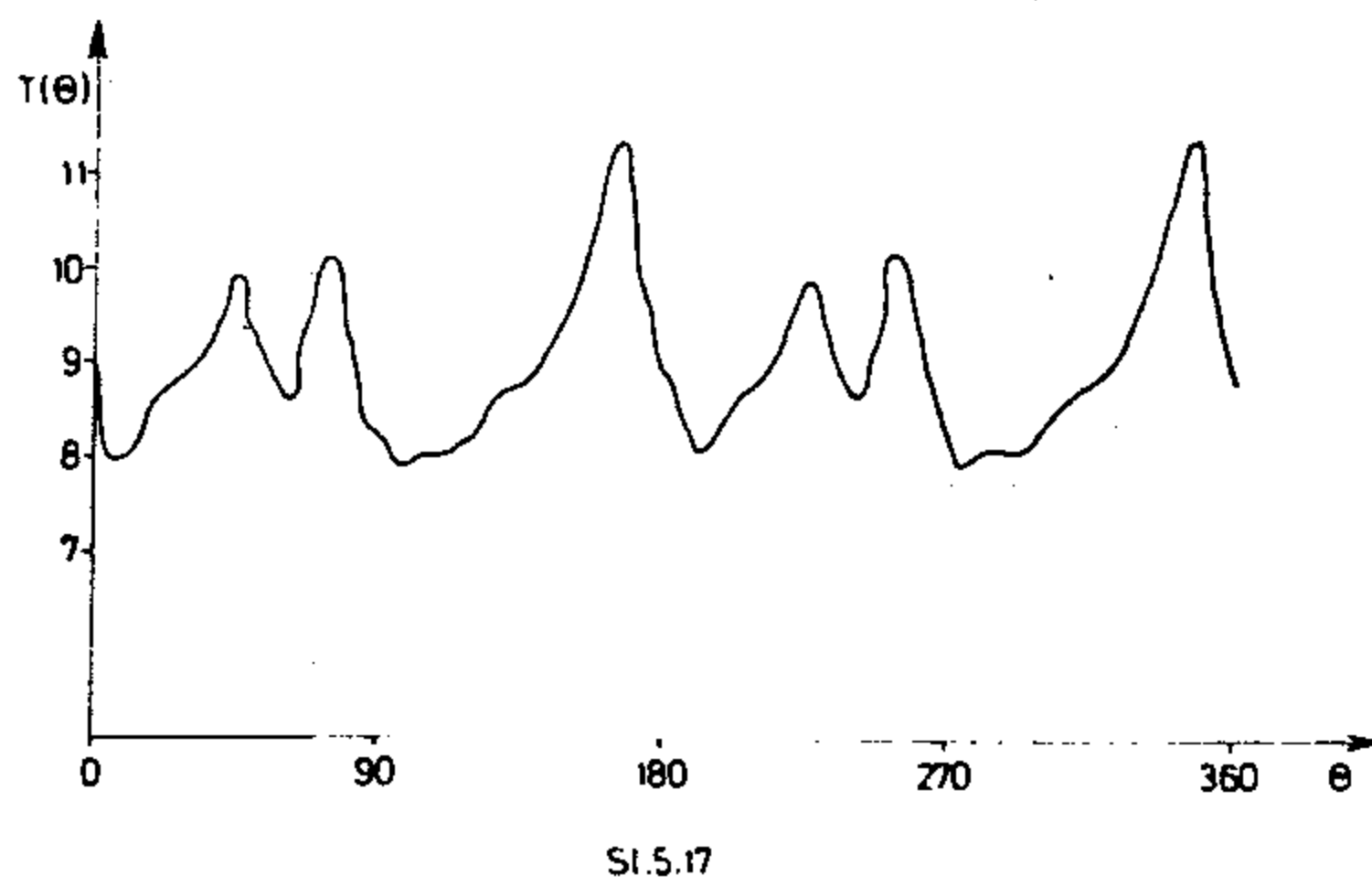
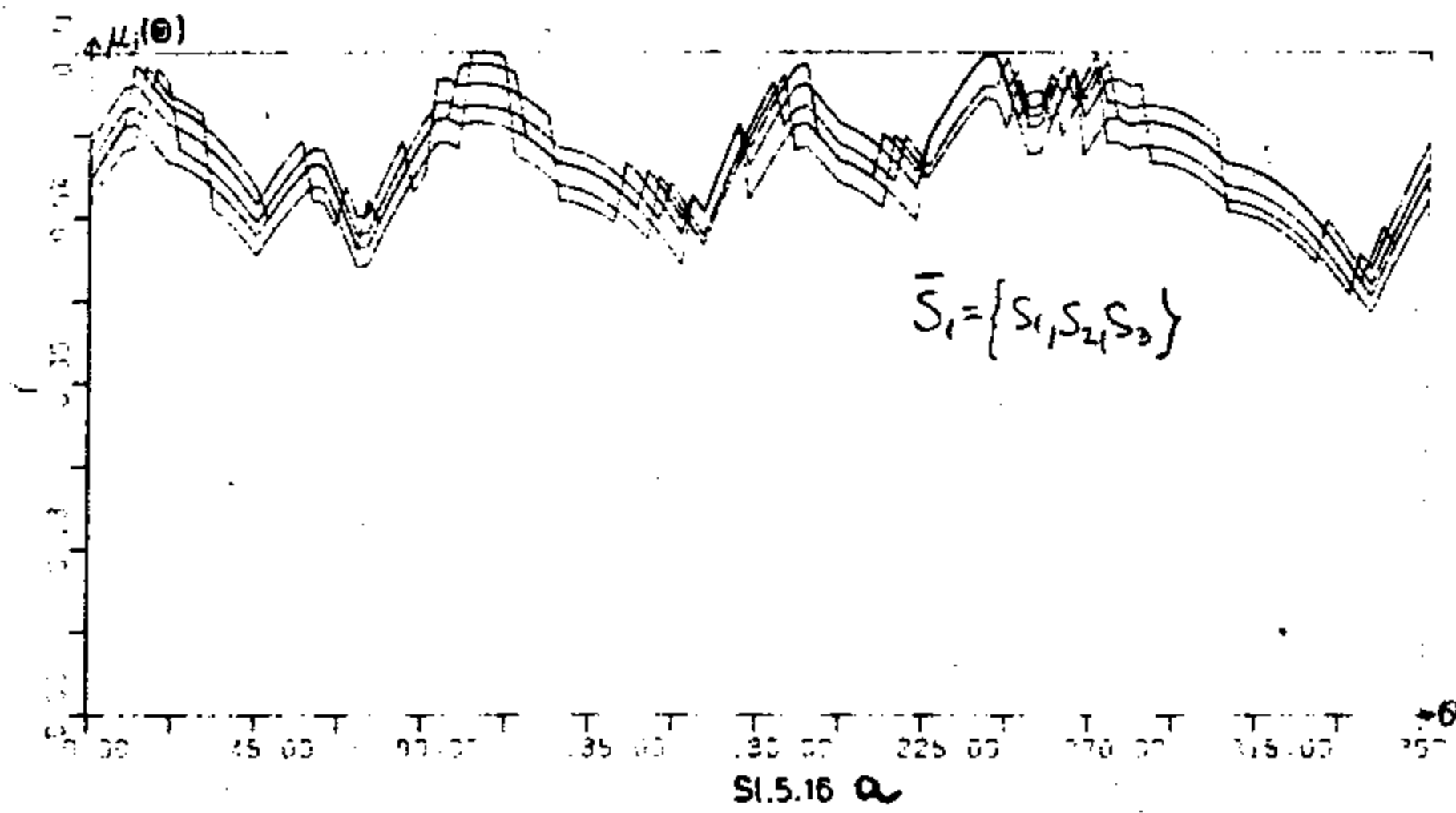
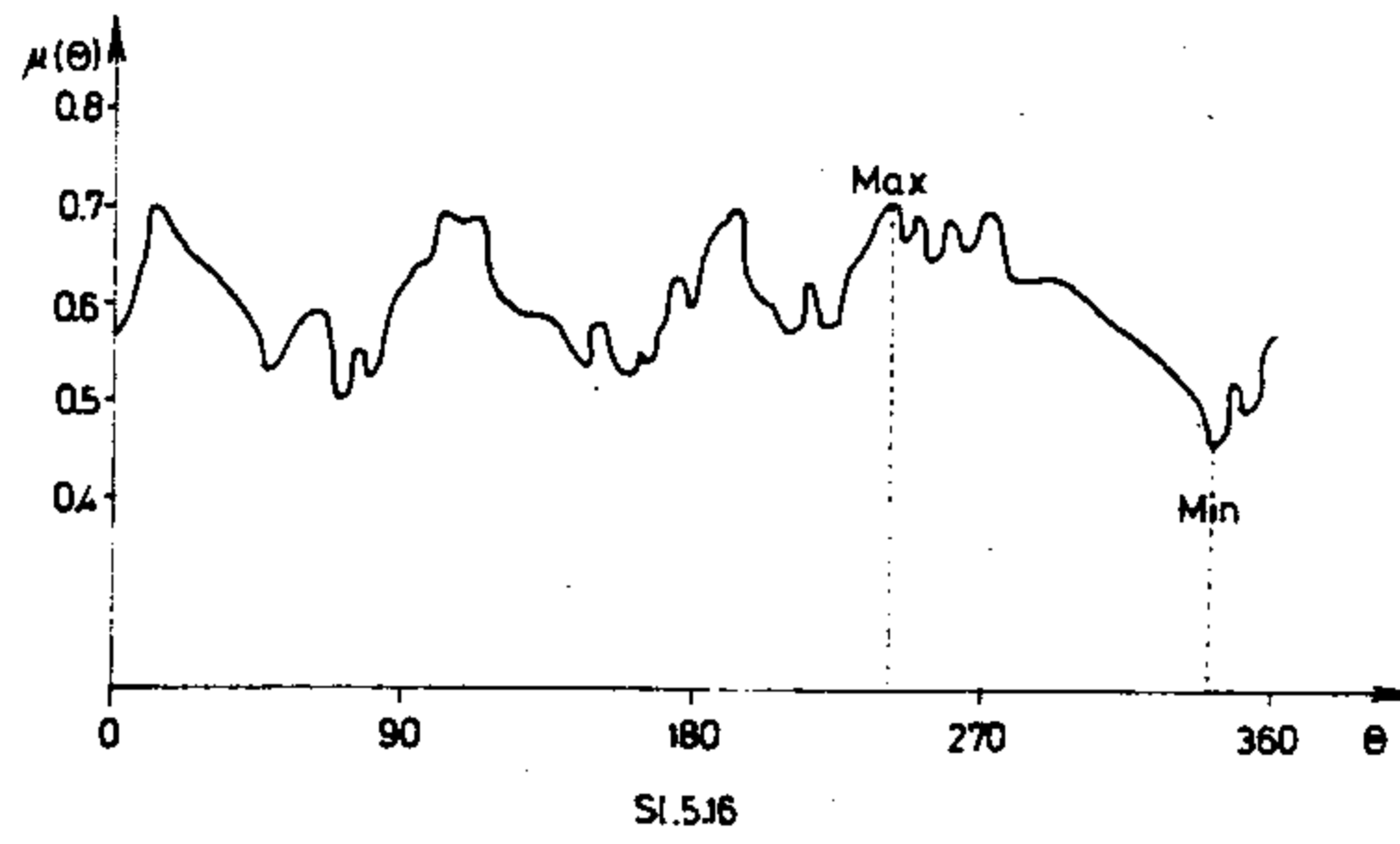
Zadana su tri objekta sa sl.5.10. koje treba smjestiti troredno u rolnu širine $H = 240$.

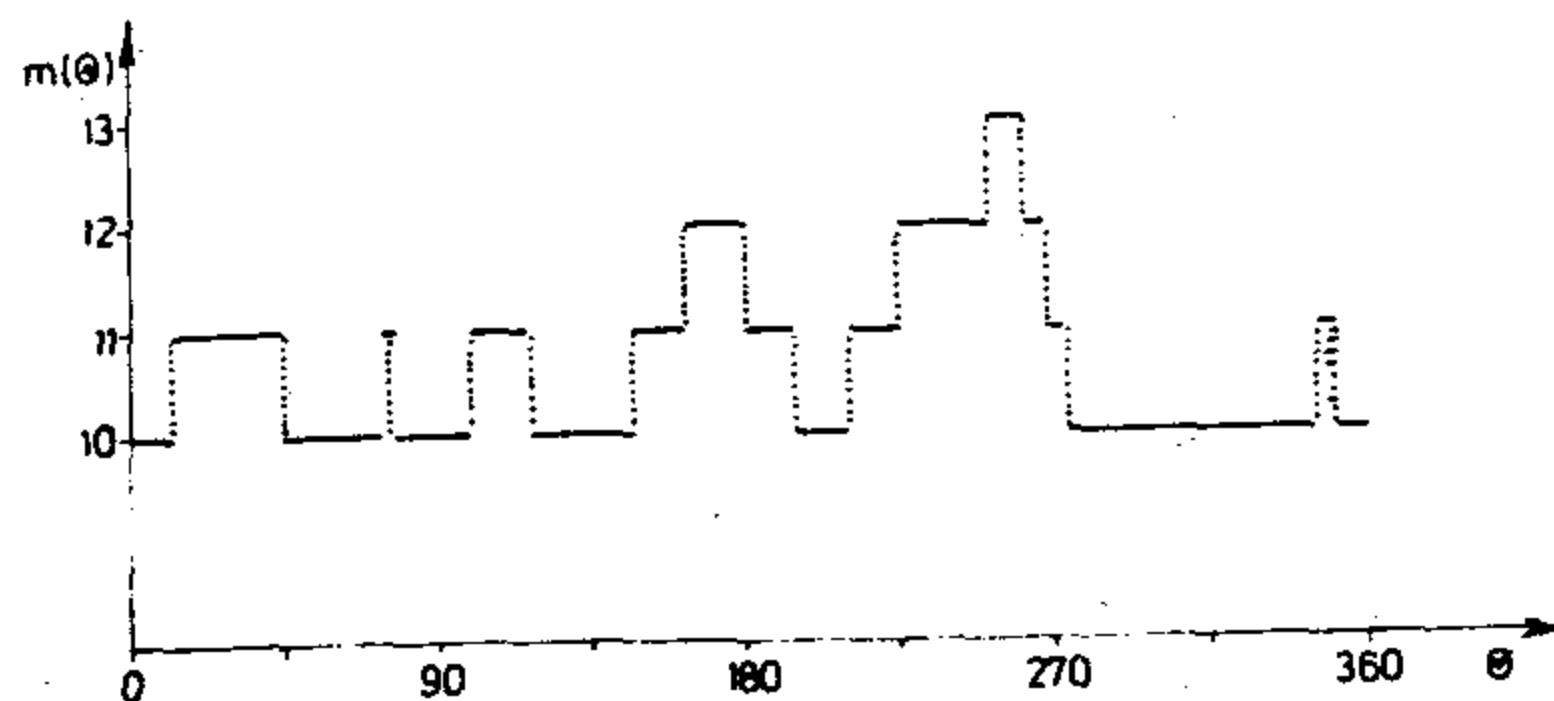
Na sl.5.16. vidimo kako se mijenja koeficijent iskorištenja u zavisnosti od θ . Na sl.5.17. predstavljena je funkcija $T(\theta)$, a na sl.5.18. $m(\theta)$. Na sl.5.19. vidimo optimalni smještaj.

Parametri smještaja su slijedeći:

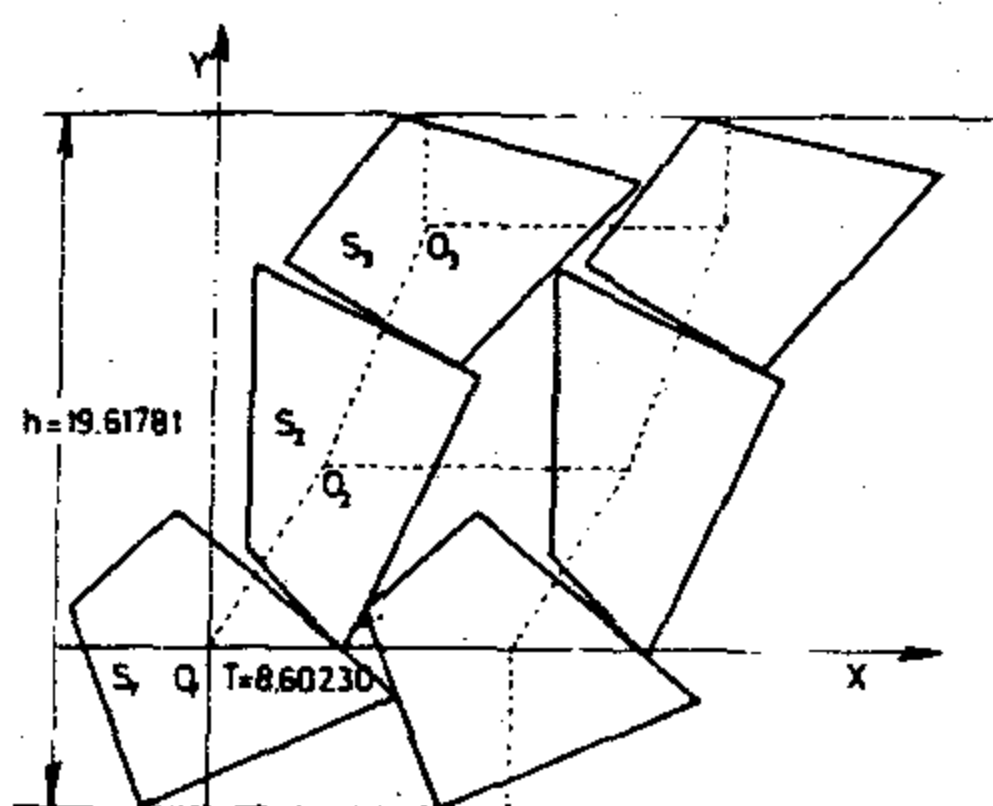
$\theta^* = 240^\circ$; $\mu(240^\circ) = 0,71202$; $T(\theta) = 8,60230$; Širina pojasa $h(240^\circ) = 19,61781$; Broj pojaseva = 12; Ukupna širina $\bar{h}(240^\circ) = 235,41377$; $O_1(0,0)$; $O_2(3,14952; 5,13843)$; $O_3(5,69926; 11,92166)$.

$\min \mu(\theta) = \mu(342^\circ) = 0,46582$ tj. $\Delta\mu = 24,6\%$.





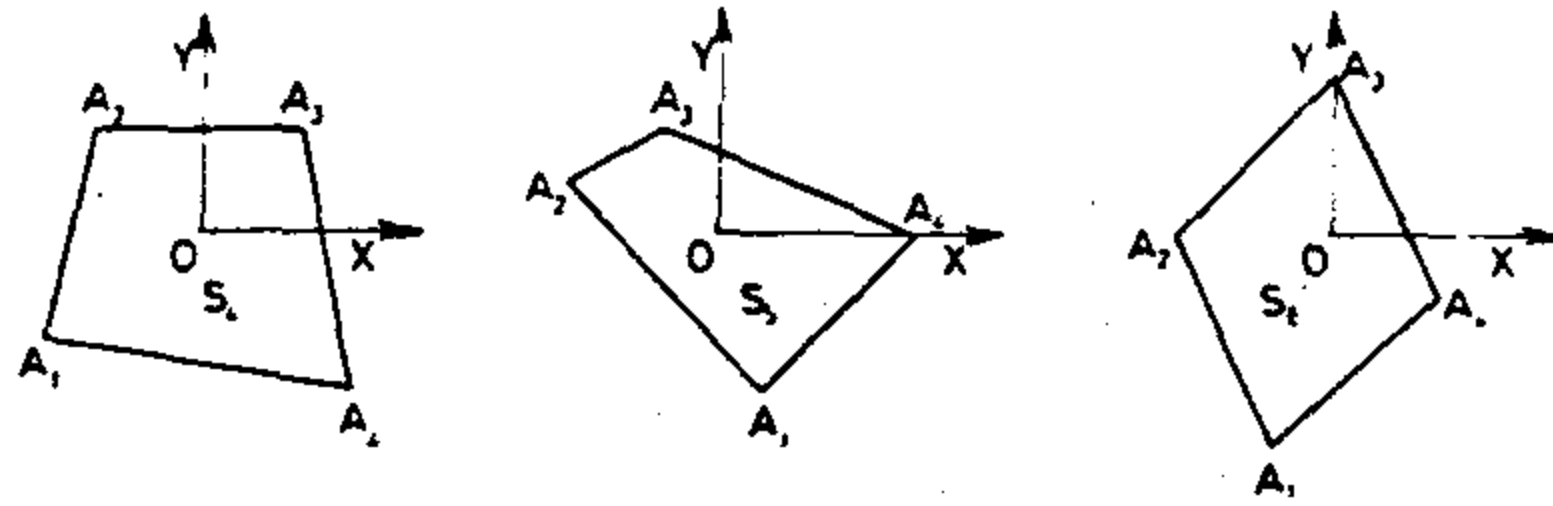
Sl. 5.18



Sl. 5.19

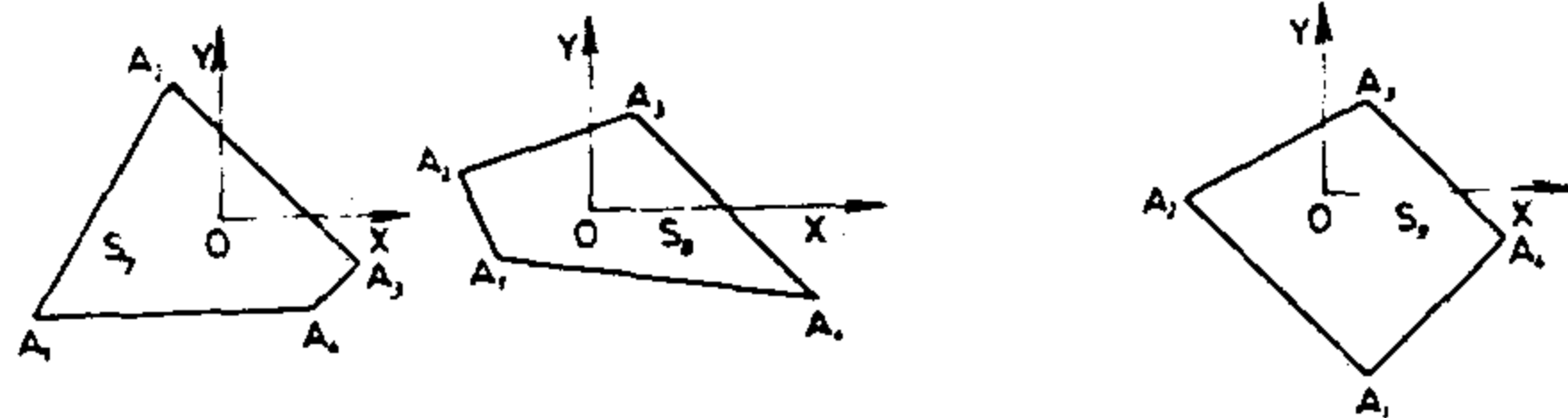
Poznato je da se rolne proizvode u raznim širinama. Zato je interesantno odabrati onu rolnu iz skupa zadanih širina za koju je otpad za zadani skup objekata koji se kroje minimalan. U tabeli 5.3. vidimo parametre za pet različitih širina, dobivene pomoću modificiranog paketa programa ROLNA I.

Vidimo da je za zadane objekte maksimalni koeficijent iskorištenja praktično jednak. Međutim, to općenito nije slučaj. Naime, ako bi uzeli da se širine mijenjaju ne za $K \times 30$ ($K \in \mathbb{N}_+$) već za neku drugu veličinu ΔH , a pogotovo za slučaj kada je $\Delta H_i \neq \Delta H_j$, onda se koeficijenti iskorištenja mogu znatno razlikovati. Zavisnost koeficijenta iskorištenja od θ za navedenih pet širina vidimo na sl. 5.16a.



Sl. 5.20

Nije rijedak slučaj da netko kroji više raznih objekata. Međutim, postoje tehnološka ograničenja na maksimalni broj redova. Pretpostavimo da je $n = 3$, a da se kroje tri grupe objekata i to objekti sa sl.5.10, sl.5.20. i sl.5.21.



Sl.5.21.

Sada je interesantno prosječno iskorištenje za grupu objekata. U tabeli 5.4. vidimo razne parametre smještanja za tri navedene grupe objekata a za zadanih pet širina rolna.

Vidimo da je za sve tri grupe objekata \bar{S}_1 , \bar{S}_2 , \bar{S}_3 maksimalni koeficijent iskorištenja stabilan u odnosu na zadane širine rolna i da je najmanje iskorištenje = 71,2% kod grupe \bar{S}_1 , a najveće = 79% kod grupe \bar{S}_2 .

Vidimo da je najbolje iskorištenje kod \bar{S}_1 za $H = 150$, za \bar{S}_2 za $H = 180$ kao i kod \bar{S}_3 . Ako je kriterij izbora prosječni koeficijent iskorištenja, onda vidimo da su sve širine gotovo jednako dobre.

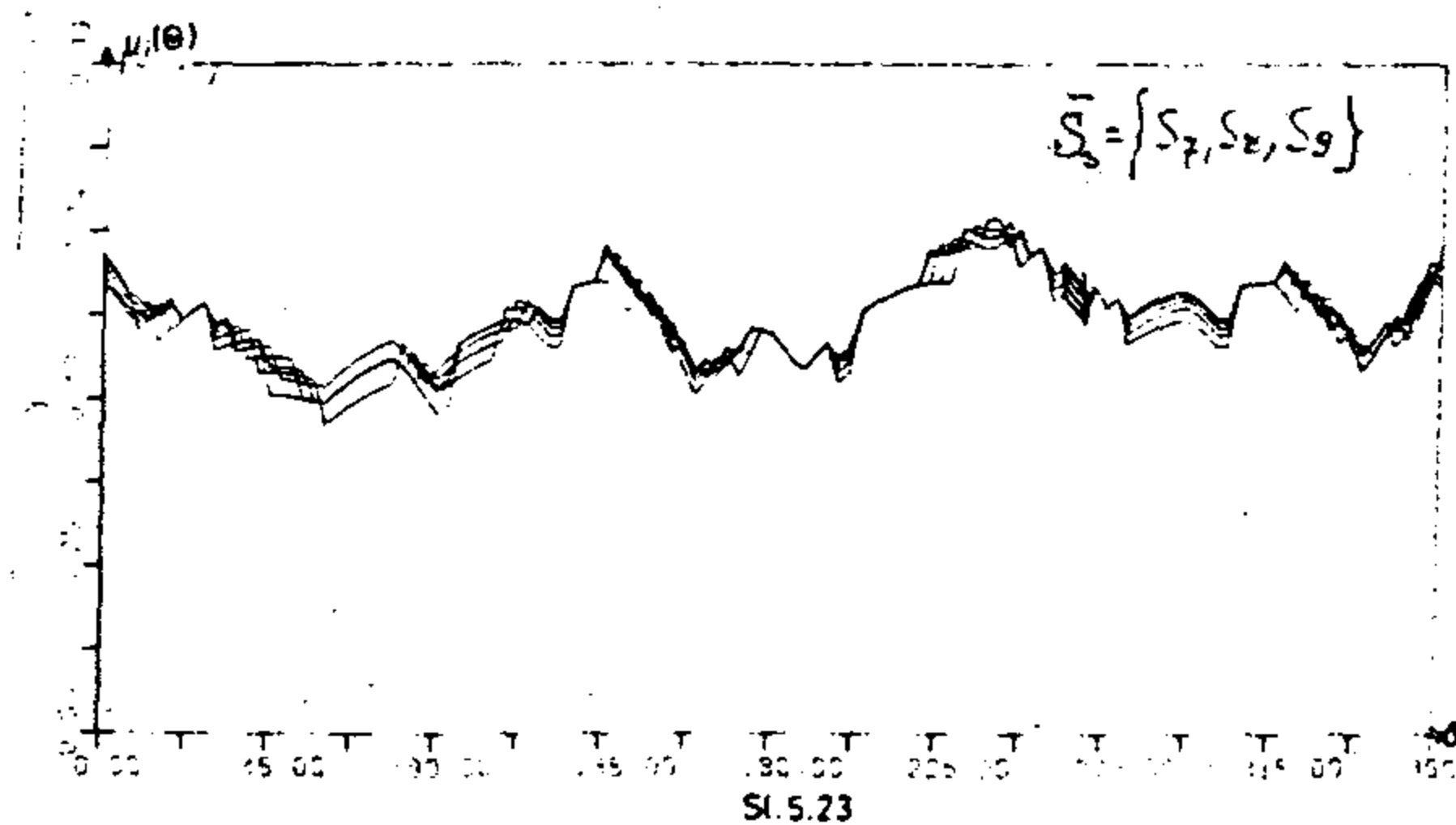
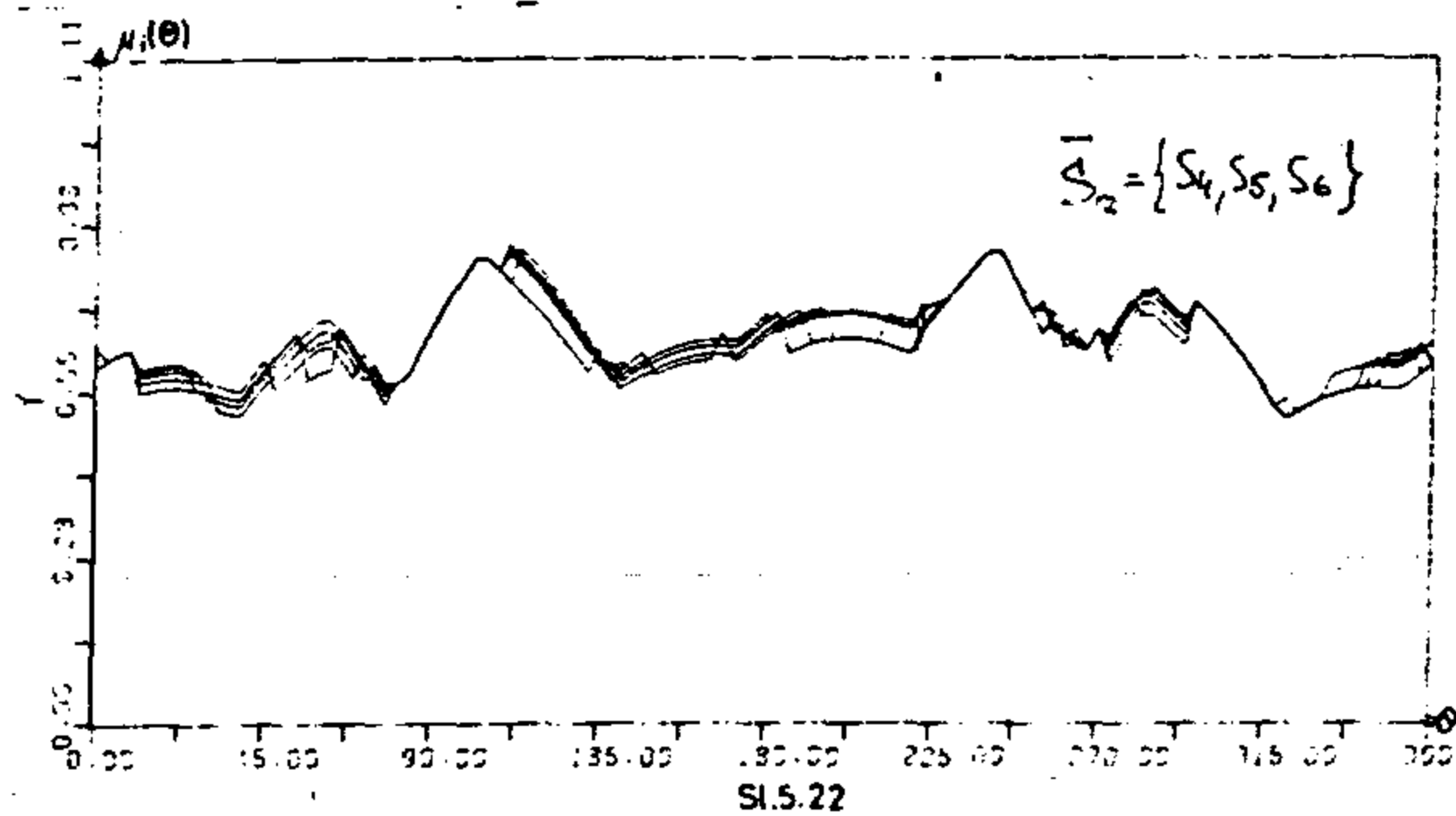
TABELA 5.3.

Širine rolna	S ₁			S ₂			S ₃			Neizo- tropni koeffi- c. ($\sigma=180$)	
	KORDINATE OB S ₁			KORDINATE OB S ₂			KORDINATE OB S ₃				
	X ₂	Y ₂	X ₃	Y ₂	Y ₃	X ₃	Y ₂	Y ₃	X ₃		Y ₃
120	3,15	5,14	5,70	11,92	240°	8,60	6	19,62	117,7	0,71202	0,57628
150	5,40	8,12	11,14	13,72	105°	8,00	7	21,00	147,0	0,71405	0,64541
180	3,15	5,14	5,70	11,92	240°	8,60	9	19,62	176,6	0,71202	0,61469
210	4,18	5,85	7,50	13,88	270°	8,20	10	20,88	209,0	0,71138	0,65860
240	3,15	5,14	5,70	11,92	240°	8,60	12	19,62	235,41	0,71202	0,63390

TABELA 5.4.

Širina rolne	S ₁ = S ₁ , S ₂ , S ₃		S ₂ = S ₄ , S ₅ , S ₆		S ₃ = S ₇ , S ₈ , S ₉		S ₁ , S ₂ , S ₃ μ (0)
	Θ ₁	μ(Θ ₁)	Θ ₂	μ(Θ ₂)	Θ ₃	μ(Θ ₃)	
120	240	0,71202	243	0,78623	243	0,75513	0,75113
150	105	0,71405	114	0,79052	243	0,73835	0,74764
180	240	0,71202	111	0,79756	240	0,76115	0,75695
210	270	0,71138	243	0,78623	234	0,75161	0,74974
240	240	0,71202	111	0,78223	231	0,74499	0,74691
μ(S ₁)		≈ 718		≈ 798		≈ 758	

Na sl.5.16a. vidimo promijene koeficijenta iskorištenja za grupu objekata \bar{S}_1 , za navedenih pet širina, a na sl.5.22. i sl.5.23. za grupe objekata \bar{S}_2 odnosno \bar{S}_3 .



Sa slika 5.16, 5.22. i 5.23. vidimo da su koeficijenti iskorištenja za određene intervale kuta θ isti za svih pet širina, a na više intervala imamo za dvije ili tri širine iste koeficijente iskorištenja.

Ovakvi komparativni grafovi omogućavaju i neke druge, za praksu, važne zaključke, pogotovo kada su zadana razna tehnološka i druga ograničenja.

Smještaj u rolnu jednog objekta

Kada se u rolnu zadane širine H smješta samo jedna vrsta objekta, onda koeficijent iskorištenja nije samo funkcija parametara potparalelograma već i broja redova tj. $\mu = \mu(\theta, \gamma, n)$. Prema tome rješavanje problema za $n > 2$ sastoji se u nalaženju

$$\mu^* = \max \mu(\theta, \gamma, n).$$

Broj redova se računa pomoću izraza

$$n = N \left[\frac{H - p_1(\theta^*) - p_1(\theta^* + \pi)}{Y_2^*} \right]$$

gdje je: Y_2^* - udaljenost medju susjednim redovima.

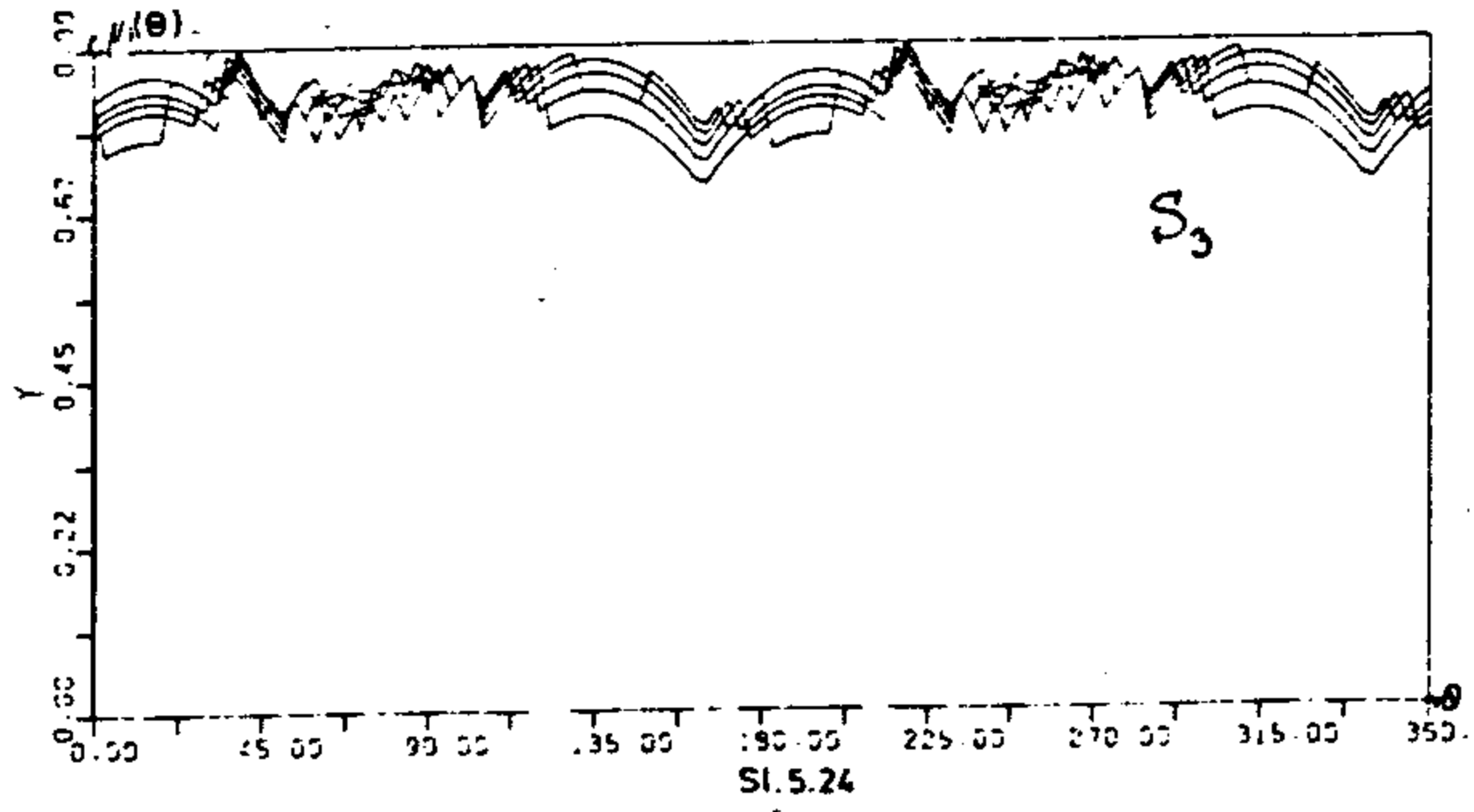
Koeficijent iskorištenja računa se pomoću (5.5.18) s tim da je sada $m = 1$.

Pomoću paketa programa ROLNA I dobiveni su podaci za smještaj objekta S_3 sa sl.5.10. u pet rolna različitih širina TABELA 5.5.

TABELA 5.5.

Širina rolne	Min.		Max.		$\Delta\mu$
	θ	$\mu(\theta)$	θ	$\mu(\theta)$	
60	165° i 345°	0,70301	36° i 216°	0,86623	≈ 16%
72	165° i 345°	0,73230	36° i 216°	0,88228	≈ 15%
84	165° i 345°	0,75322	39° i 219°	0,87543	≈ 12%
96	165° i 345°	0,76892	39° i 219°	0,88383	≈ 12%
109	171° i 351°	0,77419	39° i 219°	0,89040	≈ 12%

Na sl.5.24. vidimo kako se mijenjaju navedeni koeficijenti iskorištenja. Vidimo da postoje određeni intervali θ gdje je čak svih pet koeficijenata iskorištenja jednako, dok ima više intervala gdje se podudaraju dva koeficijenta iskorištenja.



Izbor optimalnog pravokutnika

Specijalno u metalnoj, drvnoj i tekstilnoj industriji postoji veliki broj pravokutnih materijala sa širokim spektrom standardnih dimenzija. Očito da smještaji raznih objekata u razne dimenzije daju različite koeficijente iskorištenja. Međutim, da bi se moglo izabrati najpovoljniju dimenziju za zadane objekte, trebalo bi unaprijed izvršiti proračun za sve moguće raspoložive dimenzije pa onda s obzirom na otpad i druge uvjete izabrati optimalnu dimenziju.

Dakle, ako je zadana familija pravokutnika Ω_i s dimenzijama $a_i \times b_i$, $i = (1, n)$, onda po analogiji sa (5.5.7), (5.5.8), (5.5.11), (5.5.13) i (5.5.15) za svaki Ω_i treba izračunati:

$$C_{j+1}^{(i)} = N \left[\frac{a_i - p_{j+1}(\theta + \frac{3\pi}{2}) - p_{j+1}(\theta + \frac{\pi}{2})}{T} \right] \quad (5.5.21)$$

$$E_{j+1}^{(i)} = a_i - TC_{j+1}^{(i)} \quad (5.5.22)$$

$$A_s^{(i)} = p_{j+1}(\theta + \frac{3\pi}{2}) - p_j(\theta + \frac{3\pi}{2}) \quad (5.5.23)$$

$$A_j^i \leq x_{j+1}^{(i)} \leq E_{j+1}^i \quad (5.5.24)$$

$$H_1^{(i)} = p_1(\theta) + \sum_{j=1}^{n-1} y_j^{(i)} + p_n(\theta + \pi) \quad (5.5.25)$$

$$v_i = N \left[\frac{b_i}{H_1^i} \right] \quad (5.5.26)$$

$$\mu_i = \frac{\sum_{j=1}^n \xi_j P(S_j)}{\rho \times a_i \times b_i} \quad (5.5.27)$$

Budući da se radi o n različitih dimenzija ploča, na osnovi R-disjunkcije, imamo:

$$\mu(\theta, \gamma) = \prod_{i=1}^n \mu_i(\theta^i, \gamma^i) \quad (5.5.28)$$

gdje: $\theta = (\theta^1, \theta^2, \dots, \theta^n)$; $\gamma = \gamma(\gamma^1, \gamma^2, \dots, \gamma^n)$.

Znači da se problem svodi na nalaženje

$$\mu^* = \max_{\theta^i, \gamma^i \in \Gamma^i} \left[\bigvee_{i=1}^n \mu_i(\theta^i, \gamma^i) \right] \quad (5.5.29)$$

Γ^i - oblasti definirane sa (5.3.16), (5.3.17) i (5.5.24) za svaki Ω_i .

Odredjenim modifikacijama algoritma za kvadrat sastavljen je algoritam a po njemu program A x B-I pomoću koga su dobiveni rezultati za tri objekta sa sl.5.10, a za pet dimenzija pravokutnika, tabela 5.5.

Najbolje iskorištenje se postiže za dimenziju 50 x 90. Vidimo da su maksimalni koeficijenti iskorištenja isti za dimenzije 50 x 60, 50 x 80, i 50 x 100.

Kako se mijenja koeficijent iskorištenja za dimenziju 50 x 90 kod smještanja S_1, S_2, S_3 vidimo na sl.5.25, a za dimenzije 50 x 60 i 50 x 70 na sl.5.26.

Kada se kroji više grupa raznih objekata, onda je interesantno odabrati onu dimenziju kvadrata koja je najbolja u prosjeku. U Tabeli 5.6. vidimo podatke za tri grupe objekata dane na sl. 5.10, sl.5.20. i sl.5.21. Pokazuje se da je u prosjeku najbolja dimenzija 50 x 90.

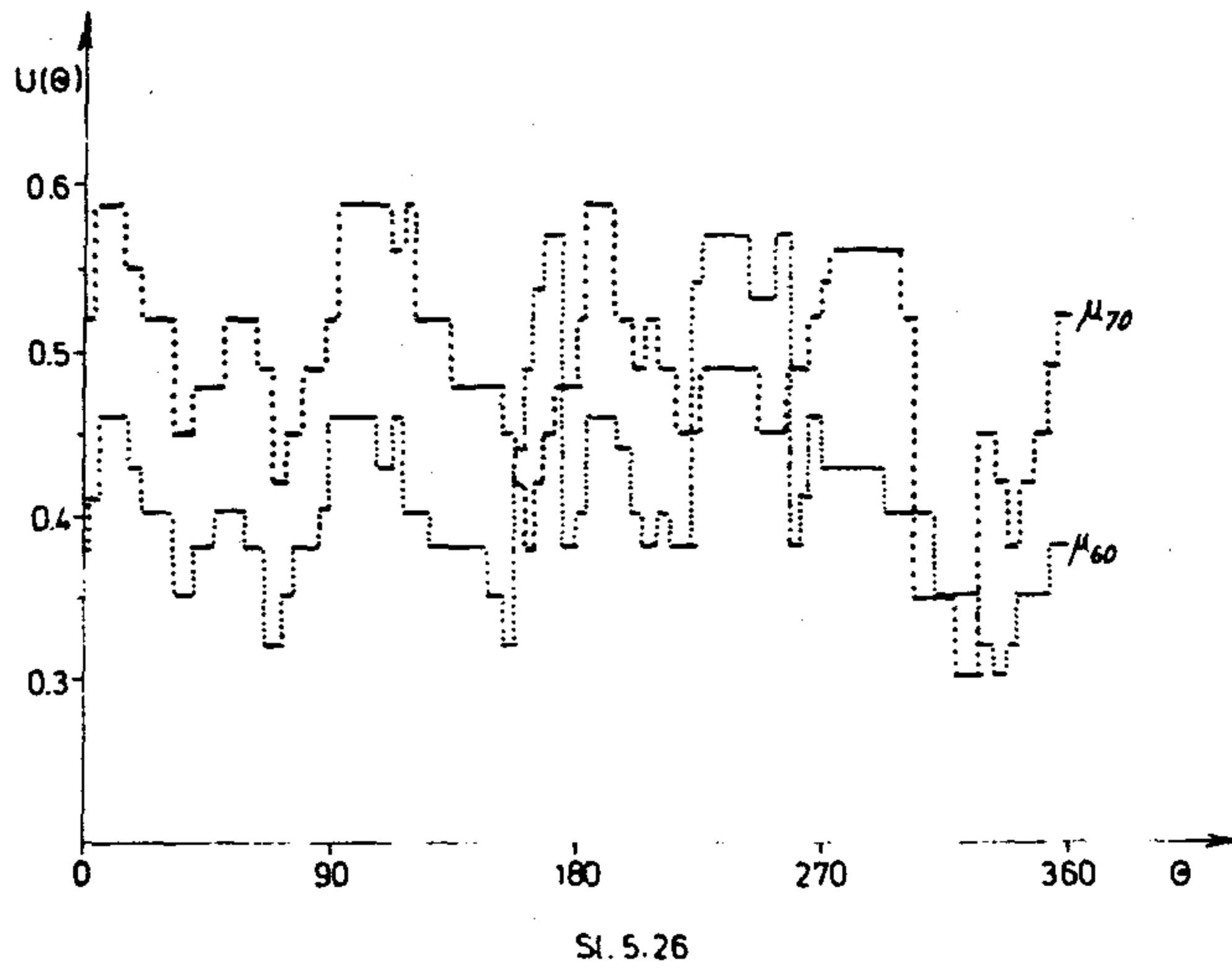
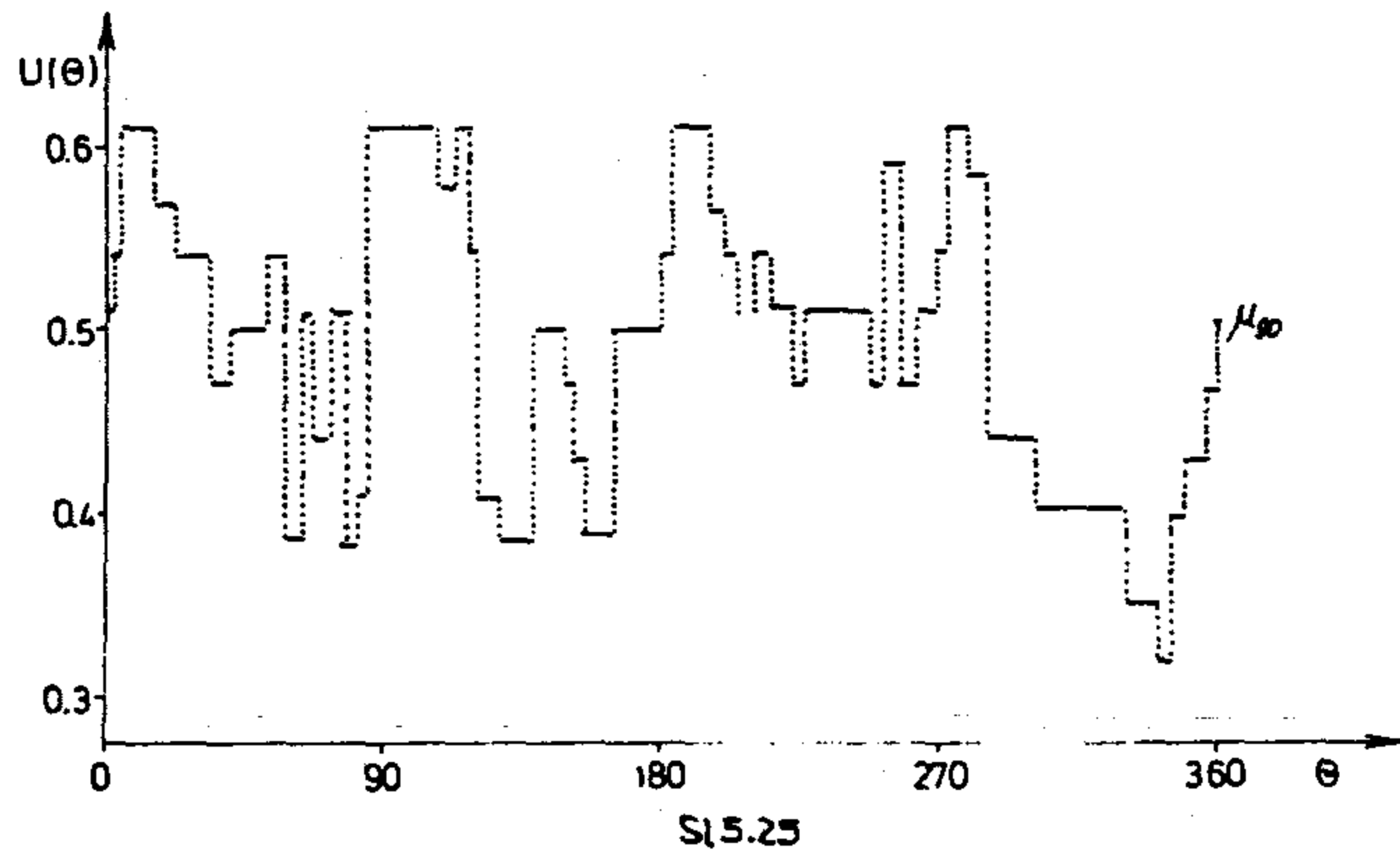
Naime, vidimo da je za $\{S_1, S_2, S_3\}$ najbolje iskorištenje kod dimenzije 50 x 90, za $\{S_4, S_5, S_6\}$ kod dimenzije 50 x 70, a za $\{S_7, S_8, S_9\}$ kod dimenzije 50 x 100.

TABELA 5.5.

Dimenzije	θ		T(θ)	Šir. po- jasa	Broj po- jasa	Broj objekata u pojasu			Ukupan broj objekata			$\mu(\theta)$	Neizotro- pni koe- ficienti $\theta = 0$
	OD	DO				S ₁	S ₂	S ₃	S ₁	S ₂	S ₃		
50 x 60	228°	246°	9,58	19,38	3	5	5	4	15	15	12	0,57450	0,38000
50 x 70	6°	15°	8,28	22,17	3	6	5	6	18	15	18	0,59357	0,48857
50 x 80	228°	246°	9,55	19,38	4	5	5	4	20	20	16	0,57450	0,42750
50 x 90	6°	15°	8,28	22,17	4	6	5	5	24	20	20	0,61556	0,50667
50 x 100	228°	246°	9,55	19,38	5	5	5	4	25	25	20	0,57450	0,45600

TABELA 5.6.

Dimenzije pravokut- nika	S ₁ , S ₂ , S ₃		$\mu(\theta)$	S ₄ , S ₅ , S ₆		$\mu(\theta)$	S ₇ , S ₈ , S ₉		$\mu(\theta)$	Prosječni $\mu(\theta)$
	θ	θ		θ	θ					
							θ	θ		
50 x 60	228°	228°	0,57450	102°	102°	0,70800	225°	225°	0,67800	0,65350
50 x 70	6°	6°	0,59357	110°	110°	0,70922	246°	246°	0,67286	0,65855
50 x 80	228°	228°	0,57450	102°	102°	0,66375	232°	232°	0,67800	0,63875
50 x 90	6°	6°	0,61556	102°	102°	0,70800	250°	250°	0,67800	0,66385
50 x 100	228°	228°	0,57450	108°	108°	0,69510	248°	248°	0,68880	0,65280



Krojenje neizotropnih materijala

Neizotropni materijali se javljaju jer se kao takvi izrađuju. To je slučaj kod izrade specijalnih limova u elektro industriji, panel ploča u drvnoj industriji, prugastih i još nekih materijala u tekstilnoj industriji itd. No, neizotropnost se može javiti i kod svih drugih materijala kao posljedica grešaka kod izrade ili oštećenja prilikom transporta.

Problem se rješava tako da se uoči kut neizotropnosti pa se poštujući taj kut vrši smještanje. Do kakvih drastičnih smanjenja koeficijenata iskorištenja može doći, vidi se iz posljednjih stupaca table 5.3. za rolnu i tebele 5.5. za kvadrat.

Smještaj kompleta n-redno i periodički u pravokutnik

U glavi III §5. formuliran je problem jednorednog periodičnog smještaja kompleta. Ovdje se vrši poopćenje na n-redno smještaje objekata, čime se općenito uzevši postiže ušteda. Naime, općenito je najnižu rubna točka objekta u $j+1$ - rednu niža od najviše rubne točke objekta u j -tom redu. Sada je broj s -tih objekata u jednom pojasu dan izrazom

$$n_{sb} = N \left[\frac{b - p_s(\theta_s) - p_s(\theta_s + \pi) + K_s(\theta_s)}{K_s(\theta_s)} \right] + \\ + \sum_{i=2}^n N \left[\frac{b - p_s(\theta_s) - p_s(\theta_s + \pi) - x_{2i}^s + K_s(\theta_s)}{K_s(\theta_s)} \right]$$

gdje je: x_{2i}^s - apscisa s -tog objekta u i -tom redu,
 $p(\theta)$ - potporna a $K(\theta)$ - kontakt funkcija.

Ukupan broj objekata je dan sa

$$m_{sb} = n_{sb} \cdot z'_i.$$

Funkcija cilja je dana, za slučaj smještaja paralelno stranici b , izrazom

$$K_b(\theta', x'_2, z') = \min \left\{ \frac{m_{1b}}{r_1}, \frac{m_{2b}}{r_2}, \dots, \frac{m_{sb}}{r_s} \right\}$$

gdje je: $\theta' = (\theta'_1, \theta'_2, \dots, \theta'_s)$, $z' = (z'_1, z'_2, \dots, z'_s)$

$$x'_2 = (x_{21}^1, \dots, x_{2n}^1; x_{21}^2, \dots, x_{2n}^2, \dots, x_{21}^s, \dots, x_{2n}^s)$$

Za slučaj smještaja paralelno stranici a imamo:

$$K_a(\theta', x'_2, z') = \min \left\{ \frac{m_{1a}}{r_1}, \frac{m_{2a}}{r_2}, \dots, \frac{m_{sa}}{r_s} \right\}.$$

Pomoću R-disjunkcije i konjukcije možemo zapisati izraz za funkciju cilja

$$K(\theta', \theta'', x'_2, x''_2, z'_1, z''_1) = \left[\bigwedge_{i=1}^s \frac{m_{ib}}{r_i} \right] \vee_1 \left[\bigwedge_{i=1}^s \frac{m_{ia}}{r_i} \right].$$

Rješavanje problema se svodi na određivanje

$$\max K(\theta', \theta'', x_2', x_2'', z_1', z_1''),$$

tj. nalaženje maksimuma po dijelovima konstantne funkcije u G_s -dimenzionalnom prostoru. Zato se problem rješava korištenjem principa optimalnosti Belmana, kao i kod jednorednog smještaja.

Sada se biraju za S_1 parametri θ_1' i x_{2i} ($i = 2, 3, \dots, s$) i to $\theta_1' = \theta_1^{1(1)}$, $x_{2i}^1 = x_{2i}^{1(1)}$ ($i = 2, 3, \dots, n$)

$$n_{1b} = N[x_1' + 1] = N_1 \left[\frac{r_1}{n_{1b}} + 1 \right]$$

Širina ostatka računa se pomoću izraza

$$a'_{1b} = a - n_{1b} \cdot H_1[\theta_1^{1(1)}, x_{22}^{1(1)}, \dots, x_{2n}^{1(1)}]$$

gdje je: $H_1[\theta_1^{1(1)}, x_{22}^{1(1)}, \dots, x_{2n}^{1(1)}]$ - širina pojasa u kome su n -redno smješteni objekti S_1 .

Nakon što se smjeste S_1, S_2, \dots, S_s imamo ostatak

$$a_{sb}^1 = a - \sum_{i=1}^s n_{ib} \cdot H_i[\theta_i^{1(1)}, x_{22}^{i(1)}, \dots, x_{2n}^{i(1)}].$$

PROGRAMI UZ GLAVU V

NAMJENA : ODREĐIVANJE HODOGRAFA, KONTAKT I POTPORNIH FUNKCIJA
OBJEKATA KOJI SE RAZMJESTAJU

I N T E G E R T

```

COMMON /APOL/ XA(50), YA(50), N
      /BEPO/ XB(50), YB(50), M
      /VHKF/ XC(90), YC(90), T
      /XEP/ XP(90), YP(90), NN
      /QVRH/ QV(50), QA(50)
      /ALFA/ ALFA(90)
      /POTP/ H(360), NH
      /HEMA/ HM(360), HT(360), FM12(1000), KI, MI
      /LELA/ QL(20), FL1(20), FL2(20), LI
      /KORAK/ DQS
      /IERR/ IERR
      /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
      /POJA/ SIR, DUZ
      /VIHOD/ XHK(90, 10), YHK(90, 10), NHK(10)
      /VIGAM/ XGAM(90, 10), YGAM(90, 10), NGAM(10)
      /VIFKF/ FKF(1000, 5), NFK(360, 5)
      /KLELA/ GLK(20, 10), FLK1(20, 10), FLK2(20, 10), LK1(10)
      /VIFNG/ FSO(1000, 10), NSO(360, 10), RF(360)
      /SLELA/ GLS(20, 10), FLS1(20, 10), FLS2(20, 10), LSI(10)
      /BGEND/ POP(360, 2)
      /NPINT/ NIN

```

```

DO 100 I = 1, NOBJ
  N = NVH ( I )

```

```

DO 10 II = 1, N
  XB (II) = XOB (II, I)
  YB (II) = YOB (II, I)

```

```

10 CONTINUE

```

```

IF (I .EQ. 1) GO TO 52

```

```

35 II = I - 1
37 PRINT 37, II, I
  FORMAT(///' H O D O G R A F', I5, ', ', I', I5, ', ', OBJEKTA')

```

```

CALL H O K O F

```

```

  NGAM (II) = T
  DO 52 II = 1, T
    XGAM (II, II) = XC (II)
    YGAM (II, II) = YC (II)

```

```

52 CONTINUE

```

```

  N = N
  DO 60 II = 1, N
    XA (II) = XB (II)
    YA (II) = YB (II)

```

```

60 CONTINUE

```

```

P R I N T 37, 1, 1
C
CALL H O K O F
DO 65 II = 1, T
  XHK (II, 1) = XC (II)
  YHK (II, 1) = YC (II)
65 CONTINUE
  NHK(1) = T
100 C O N T I N U E
  DO 150 I = 1, NOBJ
    P R I N T 110, I
    FORMAT(// ' F K O N G U E N T, O B J E K T A', 15, ', ', R E D A R A Z N, '/)
    T = NHK (I)
    DO 120 II = 1, T
      XC (II) = XHK (II, 1)
      YC (II) = YHK (II, 1)
120 CONTINUE
    CALL K O N T A F
    DO 130 II = 1, MI
      FKF (II, 1) = FM12 (II)
      NFK (II, 1) = NT (II)
130 CONTINUE
C
    IF (LI .EQ. 0) GO TO 150
    DO 135 II = 1, LI
      QLK (II, 1) = QL (II)
      FLK1 (II, 1) = FL1 (II)
      FLK2 (II, 1) = FL2 (II)
135 CONTINUE
    LK1(1) = LI
C
150 C O N T I N U E
    NIN = MI
C
    NMI = NOBJ - 1
    DO 200 I = 1, NMI
      II = I + 1
      P R I N T 160, I, II
      FORMAT(// ' K F O B J E K T A', 15, ', ', I, ', 15, ', ', R E D A R A Z N, '/)
      T = NGAM (I)
      DO 170 II = 1, T
        XC (II) = YGAM (II, 1)
        YC (II) = YGAM (II, 1)
170 CONTINUE
      CALL K O N T A F
      DO 180 II = 1, MI
        FSO (II, 1) = FM12 (II)
        NSO (II, 1) = NT (II)
        QF (II) = QO (II)
180 CONTINUE
C
      IF (LI .EQ. 0) GO TO 200
      DO 185 II = 1, LI
        GLS (II, 1) = GL (II)
        FLS1 (II, 1) = FL1 (II)
        FLS2 (II, 1) = FL2 (II)
185 CONTINUE
      LSI(1) = LI
C
200 C O N T I N U E
C

```

```

D O 250 I = 1, 2
      NN = NVH (I)
      DO 220 II = 1, NN
          JJ = NN + 1 - II
          IX = 1
          IF ( I ,EQ, 2 ) IX = NOBJ
          XP (II) = XOB (JJ, IX)
          YP (II) = YOB (JJ, IX)
220  CONTINUE
      CALL P O T P O R
      DO 230 II = 1, NH
          POP (II, I) = H (II)
230  CONTINUE
250  C O N T I N U E

```

```

R E T U R N
E N D

```

— * —

C
C
C
C
SUBROUTINE V R I S K (S, AT, INA, ITA, JH)
NAMJENA: IZRACUNAVANJE FUNKCIJE ISKORISTIVOSTI U ZAVISNOSTI OD
KUTA ZAROTIRANOSTI OBJEKTA ZA SLUCAJ VISEREDNOG
RAZHJESTAJA KONGRUENTNIH OBJEKATA

C
C
C
C
COMMON /POTP/ H(360), NH
* /KORAK/ DQS
* /NEMA/ NM(360), NT(360), FM12(1000), IFM, NQN
* /LELA/ QL(20), FL1(20), FL2(20), LI
* /IERR/ IERR

C
C
C
C
REAL NI(400), THETA(400)
DIMENSION ITA(1), INA(1)

C
C
C
C
IERR = 0
NTM = 1
IFM = 1
ILI = 0
NF = NH/2

C
C
C
C
DO 60 II = 1, JH
 ID1 = INA(II)
 ID2 = NF
 IF(II .LT. JH) ID2 = INA(II+1) - 1

C
C
C
C
DO 20 I = ID1, ID2
 NTM = NTM + 1
 IF(NTM .LE. 400) GO TO 5
 PRINT 3
3 FORMAT(' PREKORACENA DIMENZIJA VEKTORA NI(400)')
 IERR = 1
 RETURN

```

5      IF(L1, EQ, 0 .OR. OL(IL1+1), GT, CM(I)) GO TO 10
      IL1 = IL1 + 1
      FMAX = FL1(IL1)
      IF(FL2(IL1), GT, FMAX) FMAX = FL2(IL1)
      IF(OL(IL1), EQ, CM(I)) GO TO 18
      THETA(NTM) = DQS * I
      NI (NTM) = S/AT * ITA(II)/ FMAX
      NTM = NTM + 1
      IF(NTM .LE. 400) GO TO 10
      PRINT 3
      IERR = 1
      RETURN

C
10     FMAX = FM12(IFM)
      IF(NT(I), EQ, 1) GO TO 18
      NTDO = NT(I)

      DO 15 J = 2, NTDO
        JJ = IFM + J
        IF(FMAX .GE. FM12(JJ)) GO TO 15
        FMAX = FM12(JJ)

15     CONTINUE
18     IFM = IFM + NT(I)
      THETA(NTM) = DQS * I
      NI (NTM) = S/AT * ITA(II)/FMAX
20     C O N T I N U E

C
60    C O N T I N U E

C
      THETA(I) = 0.0
      NI (I) = NI(NTM)

C
      PRINT 65, NTM
65    FORMAT('1 KOORDINATE FUNKCIJE ISKORISTIVOSTI OD', I5, ' TACAKA
      *      ' FORMA ISPISA JE I KUT ; ISKORISTIVOST )')
      PRINT 66, (THETA(I), NI(I), I = 1, NTM)
66    FORMAT(5(2X, '(F6.1, ', F10.5, ' )'))

C
      UPIS NA DISK PSI = 10
      WRITE(10, 70) NTM
70    FORMAT(14)

C
      WRITE(10, 80) (THETA(I), NI(I), I = 1, NTM)
80    FORMAT(4(F6.1, F10.5))

C
      E N D F I L E 10

C
      R E T U R N
      E N D

```

— * —

SUBROUTINE X I Y I (I, J, PER)

.....
 * NAMJENA: ODREĐIVANJE KOORDINATA MOGUĆEG POLOZAJA OBJEKATA
 * DVIJE S'ISJEDNE LINIJE RASPOREDA (LINIJE J I J+1)
 *

DIMENSION XHKF(90), YHKF(90), XHP1(90),
 * AX(3), AY(3), BX(3), BY(3),
 * XSEC(20), YSEC(20)
 COMMON /RLINE/ XR(360, 10), YR(360, 10)
 * /VIFHQ/ FSO(1000, 10), NSO(360, 10), QF(360)
 * /VIGAM/ XGAM(90, 10), YGAM(90, 10), NGAM(10)

QZ = QF (I)
 NG = NGAM (J)
 DO 10 II = 1, NG
 XHKF (II) = XGAM (II, J)
 YHKF (II) = YGAM (II, J)
 10 CONTINUE

THETA = QZ * 3.141592654 / 180.0

CALL H R O T O R (XHKF, YHKF, NG, THETA)

DO 20 II = 1, NG
 XHP1 (II) = XHKF (II) + PER
 20 CONTINUE

IY = 1
 YMIN = YHKF(1)

DO 25 II = 2, NG
 IF (YMIN .LE. YHKF(II)) GO TO 25
 IY = II
 YMIN = YHKF(II)
 25 CONTINUE

DO 30 II = 1, NG
 I1 = IY - 1 + II
 IF(I1 .GT. NG) I1 = I1 - NG
 I2 = IY + II
 IF(I2 .GT. NG) I2 = I2 - NG

IF(YHKF(I1) .LE. 0.0 .AND. YHKF(I2) .GE. 0.0) GO TO 35
 30 CONTINUE

35 IREG = I1

DO 40 II = 1, NG
 I1 = IY + 1 - II
 IF(I1 .LE. 0) I1 = I1 + NG
 I2 = IY - II
 IF(I2 .LE. 0) I2 = I2 + NG
 IF(YHKF(I1) .LE. 0.0 .AND. YHKF(I2) .GE. 0.0) GO TO 45
 40 CONTINUE

```

45 IEND = II
   NST = IEND - IBEG
   IF (NST .LE. 0) NST = NST + NG
C
   NSEC = 0
   DO 70 II = 1, NST
     IA = IBEG + II - 1
     IF ( IA .GT. NG ) IA = IA - NG
     IC = IBEG + II
     IF ( IC .GT. NG ) IC = IC - NG
     AX(1) = XHKF(IA)
     AY(1) = YHKF(IA)
     BX(1) = XHKF(IC)
     BY(1) = YHKF(IC)
C
     DO 70 JJ = 1, NST
       IF ( II .EQ. JJ ) GO TO 70
       IB = IBEG + JJ - 1
       IF ( IB .GT. NG ) IB = IB - NG
       ID = IBEG + JJ
       IF ( ID .GT. NG ) ID = ID - NG
       AX(2) = XHKF(IB)
       AY(2) = YHKF(IB)
       BX(2) = XHKF(ID)
       BY(2) = YHKF(ID)
C
C
       CALL P R E S E R ( AX, AY, BX, BY, IND )
C
       IF ( IND .EQ. 0 ) GO TO 70
       NSEC = NSEC + 1
       XSEC(NSEC) = AX(3)
       YSEC(NSEC) = AY(3)
C
C
       IF ( IND .EQ. 1 ) GO TO 70
       NSEC = NSEC + 1
       XSEC(NSEC) = PX(3)
       YSEC(NSEC) = PY(3)
C
70 C O N T I N U E
C
       IF ( NSEC .GT. 0 ) GO TO 80
       YR (I, J) = 0,0
       XR (I, J) = X P O I N T (XHKF, YHKF, NG )
       R E T U R N
C
80 DO 95 II = 1, NSEC
   J2 = II + 1
   DO 90 JJ = 1, NSEC
     IF (YSEC(II) .LE. YSEC(JJ)) GO TO 90
     TEMP = YSEC(II)
     YSEC(II) = YSEC(JJ)
     YSEC(JJ) = TEMP
     TEMP = XSEC(II)
     XSEC(II) = XSEC(JJ)
     XSEC(JJ) = TEMP
90 CONTINUE
95 C O N T I N U E
C

```

```

DO 100 II = 1, NSEC
  IF (YSEC(II) .LT. 0.0) GO TO 100
  YR (I, J) = YSEC (II)
  XR (I, J) = XSEC (II)
  RETURN
100 CONTINUE

```

```

YR (I, J) = 0.0
XR (I, J) = X P O I N T (XHKF, YHKF, HG)
RETURN
END

```

— * —

SUBROUTINE U L A Z

* * * * *
 * NAMJENA : UCITAVANJE I STAMPANJE ULAZNIH PODATAKA ZA N-REDNO
 * RAZMJEŠTANJE OBJEKATA.
 * * * * *

* * * * *

```

COMMON /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
  /POJA/ SIR, DUZ
  /KORAK/ DQS
  /IERR/ IERR

```

```
IERR = 0
```

```

5 READ 5, SIR, DUZ, DQS, NOBJ
  FORMAT ( 3(F15.5, SX), 15 )
7 IF (SIR .NE. 0.0) PRINT 7, SIR
  FORMAT ( 1H1, 'SIRINA POJASA =', F15.5 )
9 IF (DUZ .NE. 0.0) PRINT 9, DUZ
  FORMAT ( 1X, 'DUZINA POJASA =', F15.5 )
10 PRINT 10, DQS, NOBJ
  FORMAT ( 1X, 'KUT ROTACIJE =', F12.2,
    /1X, 'BROJ OBJEKATA =', 19 )

```

```
IF (NOBJ .GT. 0) GO TO 13
```

```

IERR = 1
PRINT 11
11 FORMAT ( ' BROJ OBJEKATA MORA BITI VECI OD NULE' )
RETURN

```

C
C
C
C

```
13 CONTINUE
```

```

DO 50 I = 1, NOBJ
  READ 15, N
  15 FORMAT ( 15 )
  PRINT 20, I, N
  20 FORMAT ( //1X, 'KOORDINATE OBJEKTA BROJ', I6, ' OD',
    16, ' VRHOVA'// )
  READ 25, ( XOB( J, I ), YOB( J, I ), J = 1, N )
  25 FORMAT( 8F10.5 )
  NVH ( I ) = N
  PRINT 30, ( XOB( J, I ), YOB( J, I ), J = 1, N )
  30 FORMAT ( 3(2X, '(', F10.5, ' ', F10.5, ' ') ) )

```

C

```

50 CONTINUE
RETURN
END

```

— * —

SUBROUTINE P R E S E K (AX, AY, BX, BY, IND)

```

C .....
C * NAMJENA: ODREĐJIVANJE ODNOSA MEDJU DUZIMA A1-B1 I A2-B2
C * -DUZINE SE MOGU PRESJECATI U KOM SLUCAJU SE ODREĐUJU
C *   KOORDINATE PRESJECNE TACKE;
C * -DUZINE SE MOGU FREKLAPATI U KOM SLUCAJU SE ODREĐUJU
C *   KOORDINATE RUBNIH TACAKA DUZINE FREKLAPANJA;
C * -DUZINE MOGU BITI U ODNOSU DA NEMAJU ZAJEDNIČKU TACKU
C .....
C
C   D I M E N S I O N   AX(3), AY(3), BX(3), BY(3)
C
C   IND = 0
C   DX1 = BX(1) - AX(1)
C   DY1 = BY(1) - AY(1)
C   DX2 = BX(2) - AX(2)
C   DY2 = BY(2) - AY(2)
C
C   IF ( DX1 .EQ. 0. )   GO TO 50
C     ALFA = DY1 / DX1
C     B1   = AY(1) - ALFA * AX(1)
C     IF ( DX2 .EQ. 0. )   GO TO 40
C       BETA = DY2 / DX2
C       B2   = AY(2) - BETA * AX(2)
C       IF ( ABS(ALFA) .EQ. ABS(BETA) )   GO TO 30
C         AX(3) = ( B2 - B1 ) / ( ALFA - BETA )
C         GO TO 100
C
C   30   IF ( B1 .NE. B2 )   P E T U R N
C         CALL C O N D A ( AX, AY, BX, BY, IND )
C         R E T U R N
C
C   ... SLUCAJ KADA JE DRUGA DUZ OKONITA ...
C   40   AX(3) = AX(2)
C         GO TO 100
C
C   50   IF ( DX2 .EQ. 0. )   GO TO 90
C
C     BETA = DY2 / DX2
C     B2   = AY(2) - BETA * AX(2)
C     AX(3) = AX(1)
C     AY(3) = BETA * AX(3) + B2
C
C     GO TO 102
C
C   90   IF ( AX(1) .NE. AX(2) )   P E T U R N
C         CALL C O N D A ( AX, AY, BX, BY, IND )
C         R E T U R N
C
C
C
C   100  AY(3) = ALFA * AX(3) + B1
C   102  IF ( (DX1.GT.0..AND.(AX(3).LT.AX(1).OR.AX(3).GT.BX(1))) .OR.
C *      (DX1.LT.0..AND.(AX(3).LT.BX(1).OR.AX(3).GT.AX(1))) .OR.
C *      (DX2.GT.0..AND.(AX(3).LT.AX(2).OR.AX(3).GT.BX(2))) .OR.
C *      (DX2.LT.0..AND.(AX(3).LT.BX(2).OR.AX(3).GT.AX(2))) )
C *      R E T U R N
C   103  IND = 1
C         R E T U R N
C         E N D

```

PROGRAM R O L N A 1

.....
 * NAMIJENA: U ROLNU ZADANE SIRINE OPTIMALNO SMJESTITI ISTI OBJEKAT,
 * I ODREDITI FUNKCIJU REDNOSTI SMJESTANJA I ISKORISTIVOST.
 *

```
COMMON /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
*   /KORAK/ DGS
*   /RLINE/ XR(360, 10), YR(360, 10)
*   /POJA/ SIR, DUZ
*   /VIFKF/ FRF(1000,5), NFK(360,5)
*   /BGEND/ POP(360, 2)
*   /NPINT/ NIN
*   /PERIOD/ TE(360)
*   /CILJ/ FC(360)
*   /IERR/ IERR
*   /MEMA/ QM(360), NT(360), FM12(1000), KI, MI
```

```
DIMENSION XZ(50), YZ(50)
DIMENSION NPO(50), IPO(50)
DIMENSION SIP(360)
```

```
READ 2, NSIR, DSIR
2 FORMAT(I2, F10.5)
IERR = 0
```

```
CALL U L A Z
IF(IERR .EQ. 1) S T O P 111
NVH(2) = NVH(1)
```

```
NOBJ = 2
NZ = NVH(2)
DO 5 J = 1, NZ
  XOB(J,2) = XOB(J,1)
  YOB(J,2) = YOB(J,1)
  XZ(J) = XOB(J,1)
  YZ(J) = YOB(J,1)
```

```
5 CONTINUE
CALL P O T R O N
S = S P ( XZ, YZ, NZ )
```

```
DO 30 I = 1, NIN
  TE(I) = FRF(I, 1)
  QM(I) = DGS * I
```

```
CALL X I Y I ( I, I, TE(I) )
```

```
30 CONTINUE
```

```
DO 110 IS = 1, NSIR
  SSIR = SIR + DSIR * ( IS - 1 )
  PRINT 32, SSIR
32 FORMAT(I11, ' ZA SIRINU ROLNE ', F10.5)
```

```

33 PRINT 33
   FORMAT(///' KUT OBJ. BR,RED. SIRINA POJ. ISKORIS_ENJE'
          ' PERIOD', BX, 'X', BX, 'Y')

```

```

C
   FCHAX = 0.0
   IL     = 0
   N      = 0

```

```

C
C
C
   DO 80 I = 1, NIN
   H1 = POP(I, 1)
   I2 = NIN/2 + I
   IF ( I2 .GT. NIN ) I2 = I2 - NIN
   H2 = POP(I2, 1)

```

```

C
   NRED = (SSIR - H1 - H2) / YR(I, 1) + I
   IF (NRED .EQ. IL) GO TO 60

```

```

   N = N + 1
   IF (N .LE. 50) GO TO 50

```

```

35 PRINT 35
   FORMAT(' PREPORACENA DIMENZIJA VEKTORA NPO(50)')
   IERR = 1
   S T O P

```

```

C
40 NPO(N) = NRED
   IPO(N) = I
   IL     = NRED

```

```

C
60 PP = SSIR * TE(I)
   SIP(I) = ( NPO(I) - 1 ) * YR(I, 1) + H1 + H2
   FC(I) = S * NPO(N) / PP

```

```

C
65 PRINT 65, QM(I), NRED, SIP(I), FC(I), TE(I), XE(I,1), YR(I,1)
   FORMAT(IX, F8.1, 16, F12.5, F15.5, 3F10.5)

```

```

C
   IF(YR(I, 1) .LE. 0.0) GO TO 80

```

```

C
   IF(FCHAX .GE. FC(I)) GO TO 60
   FCHAX = FC(I)
   IZ = I

```

```

C
80 C O N T I N U E

```

```

C
   CALL O U T P U T ( IZ )

```

```

   KPI = KI + 1
   QMULA = 0.0
   WRITE(10, 85) KPI
   WRITE(10, 90) QMULA, FC(KI), (QM(I), FC(I), I = 1, 3)
   WRITE(10, 90) (QI(I), FC(I), I = 4, KI)
85 FORMAT(15)
90 FORMAT (4(F8.1, F12.5))

```

```

C
110 C O N T I N U E

```

```

C
   ENDFILE(10)

```

```

C
   S T O P
   E N D

```

PROGRAM P O J A S

• • • • •
 • NAMJENA: ODREĐJIVANJE ŠIRINE POLUPESKONAČNOG POJASA I PARAMETARA
 • RAZMJEŠTAJA N OBJEKATA SMJEŠTENIH U N REDOVA TAKO DA
 • KOEFICIJENTAT POPUNJENOSTI POJASA BUDE MAKSIMALAN.
 • • • • •

```
COMMON /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
• /KOPAK/ DQS
• /RLINE/ XR(360, 10), YR(360, 10)
• /FOJA/ SIR, DUZ
• /VIFKF/ FKF(1000,5), NFK(360,5)
• /BGEND/ POP(360, 2)
• /NPINT/ NIN
• /PERIOD/ TE(360)
• /CILJ/ FC(360)
• /IERR/ IERR
• /RENA/ QH(360), NT(360), FM12(1000), KI, MI
```

```
DIMENSION XZ(50), YZ(50)
```

```
IERR = 0
```

```
CALL U L A Z
IF(IERR .EQ. 1) S T O P 111
CALL P O T F O R M
```

```
S = 0
DO 6 I = 1, NOBJ
  NZ = NVH(I)
  DO 5 J = 1, NZ
    XZ(J) = XOB(J, I)
    YZ(J) = YOB(J, I)
```

```
5 CONTINUE
```

```
S = S + SP(XZ, YZ, NZ)
```

```
8 CONTINUE
```

```
P R I N T 9, NOBJ, S
9 FORMAT(' POVSINA', I4, ' OBJEKATA S =', F12.5)
```

```
PRINT 10
10 FORMAT(///' KUT OBJEKTA', I0X, 'ZAUZETA POVRŠINA',
  I0X, 'ŠIRINA POJASA', 5X, 'ISKORISTIVOST', 5X, 'PERIOD')
PZ = 2.0 ** 32 = 1.0
DO 80 I = 1, NIN
  PER = FKF(I, 1)
  DO 20 II = 2, NOBJ
    PERI = FKF(I, II)
    IF(PERI .LE. PER) GO TO 20
  PER = PERI
20 CONTINUE
TE(I) = PER
QZ = DQS * I
QH(I) = QZ
```

```

C
  NMI = NOPJ - 1
  DO 28 J = 1, NMI
    CALL X I Y I (I, J, PER)
  PRINT 26, J, XR(I,J), J, YR(I,J)
26  FORMAT('X',I2,' ' =',F10,5,' Y',I2,' ' =',F10,5)
28  CONTINUE

C
    H1 = POP ( I, 1 )
    I2 = NIN / 2 + 1
    IF ( I2 .GT. NIN ) I2 = I2 - NIN
    H2 = POP ( I2, 2 )
    NR = NODJ - 1
    RJK = 0.0
    DO 30 JJ = 1, NR
      RJK = RJK + YR(I, JJ)
30  CONTINUE
    SIR = H1 + H2 + RJK
    PNEW = SIR * PER
    FC(I) = S / PNEW
    IF (PNEW .GT. PZ) GO TO 60
    PZ = PNEW
    IZ = I
60  PRINT 70, PZ, PNEW, SIR, FC(I), PER
70  FORMAT(2X,F8,1,I2X,F12,5,I4X,F12,5,5X,F12,5,6X,F10,5)
80  CONTINUE

C
  CALL OUTPUT ( IZ )

C
C
C  OVAJ ISPIS RALI SE SAHO U SLUCAJU KADA SE ZELI
C  NA DATOTEMU DSI = 10 SACUVATI PODACI O ISKO-
C  RISTIVOSTI U FUNKCIJI OD KUTA, INACE SE VRSI
C  BLOKIRANJE PAREDEI KOJENTAR DEKLARACIJOM.
C
  KPI = KI + 1
  QMNULA = 0.0
  WRITE(10, 85) KPI
  WRITE(10, 90) QMNULA, FC(KI), (QM(I), FC(I), I = 1, 3)
  WRITE(10, 90) (QM(I), FC(I), I = 4, KI)

85  FORMAT(15)
90  FORMAT (4(F8,1, F12,5))
  ENDFILE(10)

C
  STOP
  END

```

— * —

PROGRAM R O L N A

• • • • •
 • NAMJENA: ROLNU ZADANE SIRINE (SIR) IZREZATI NA POJASEVE U KOJIMA
 • SU N-REDNO RAZMJESTENI OBJEKTI S(J), J=1,NOBJ
 • • • • •

```
COMMON /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
  • /KORAK/ DQS
  • /RLINE/ XR(360, 10), YR(360, 10)
  • /POJA/ SIR, DUZ
  • /VIFKF/ FKF(1000,5), NFK(360,5)
  • /BGEND/ POP(360, 2)
  • /NPINT/ NIN
  • /PERIOD/ TE(360)
  • /CILJ/ FC(360)
  • /IERR/ IERR
  • /MEMA/ QH(360), NT(360), FR12(1000), K1, M1
```

```
DIMENSION XZ(50), YZ(50)
DIMENSION NPO(50), IPO(50)
DIMENSION SIP(360)
```

```
READ 2, NSIR, DSIR
2 FORMAT(I2, F10.5)
IERR = 0
```

```
CALL U L A Z
  IF(IERR .EQ. 1) STOP 111
CALL P O T K O N
```

```
S = 0
DO 8 I = 1, NOBJ
  NZ = NVH(I)
  DO 5 J = 1, NZ
    XZ(J) = XOE(J, I)
    YZ(J) = YOB(J, I)
5 CONTINUE
```

```
S = S + SP(XZ, YZ, NZ)
8 CONTINUE
```

```
DO 30 I = 1, NIN
  PER = FKF(I, 1)
  DO 20 II = 2, NOBJ
    PERI = FKF(I, II)
    IF( PERI .LT. PER ) GO TO 20
  PER = PERI
20 CONTINUE
  TE(I) = PER
  QH(I) = DQS * I
  PRINT 25, QH(I)
25 FORMAT(/' KUT ROTACIJE = ', F8.1)
```

```
NN1 = NOBJ - 1
DO 28 J = 1, NN1
  J1 = J + 1
```

```

      CALL X I Y I (I, J, PER)
      PRINT 26, J1, XR(I,J), J1, YR(I,J)
26  FORMAT(' X', I2, ' =', F10.5, ' Y', I2, ' =', F10.5)
28  CONTINUE

```

```

C
      NOB = NOBJ
      SIP(I) = SIRINA(I, NOB)
30  CONTINUE

```

```

C
C
C
      DO 110 IS = 1, NSIR
      SSIR = SIR + DSIR * (IS - 1)
      PRINT 32, SSIR
32  FORMAT(III1, ' ZA SIRINU ROLNE ', F10.5)

```

```

C
      PRINT 33
33  FORMAT(///' KUT OBJ. DR. POJ. SIRINA POJ. UK. SIRINA'
          ' POTR. POVREINA ISKORIS_ENJE PERIOD')

```

```

      FCMAX = 0.0
      IL = 0
      N = 0

```

```

C
C
C
      DO 80 I = 1, NIP
      NBP = SSIR / SIP(I)

```

```

      IF (NBP .EQ. IL) GO TO 60
      N = N + 1
      IF (N .LE. 50) GO TO 40
      PRINT 35
35  FORMAT(' PREDRACENA DIMENZIJA VEKTORA HPO(50)')
      IERR = 1
      STOP

```

```

C
40  HPO(N) = NBP
      HPO(N) = 1
      IL = NBP

```

```

C
60  PP = SSIR * TE(I)
      FC(I) = S * HPO(N) / PP
      SUK = SIP(I) * HPO(N)

```

```

C
      PRINT 65, QM(I), NBP, SIP(I), SUK, PP, FC(I), TE(I)
65  FORMAT(IX, F8.1, 17, 2F12.5, F15.5, 2F12.5)
      IF (FCMAX .GE. FC(I)) GO TO 80
      FCMAX = FC(I)
      IZ = I

```

```

80  CONTINUE

```

```

C
      CALL OUTPUT (IZ)

```

```

C
      KPI = KI + 1
      QMULA = 0.0
      WRITE(10, 85) KPI
      WRITE(10, 90) QMULA, FC(KI), (QM(I), FC(I)), I = 1, 3)
      WRITE(10, 90) (QI(I), FC(I)), I = 4, KI)
85  FORMAT(I5)
90  FORMAT(4(F8.1, 112.5))

```


SUBROUTINE C O N T A (AX, AY, BX, BY, IND)

```

C
C
C * * * * *
C * NANJETA: ODREĐJIVANJE RUBNIH TACAKA DUZINE PREKLAPANJA DVIJU
C * DUZI ILI PODIRNE TACKE AYO SE DUZI CIJI SE PRAVCI PRE
C * KLAPAJU, PODIRUJU U JEDNOJ TACCI.
C * * * * *
C
C DIMENSION AX(3), AY(3), BX(3), BY(3), MESTO(4), X(4), IPER(24)
C DATA IPER /1234, 4321, 1243, 3421, 1324, 4231, 1342, 2431,
C * 1432, 2341, 1423, 3241, 2134, 4312, 2143, 3412,
C * 2314, 4132, 2413, 3142, 3124, 4213, 3214, 4123 /
C
C DATA MESTO / 1, 2, 3, 4 /
C IND = 0
C IF ( AX(2) .EQ. BX(2) ) GO TO 20
C X(1) = AX(1)
C X(2) = BX(1)
C X(3) = AX(2)
C X(4) = BX(2)
C GO TO 50
C
C 20 X(1) = AY(1)
C X(2) = LY(1)
C X(3) = AY(2)
C X(4) = BY(2)
C AX(3) = AX(1)
C BX(3) = AY(1)
C
C 50 DO 60 I = 1, 3
C K = I + 1
C DO 55 J = K, 4
C IF ( X(I) .LE. X(J) ) GO TO 55
C TEMP = X(I)
C
C X(I) = X(J)
C X(J) = TEMP
C ITMP = MESTO(I)
C MESTO(I) = MESTO(J)
C MESTO(J) = ITMP
C
C 55 C O N T I N U E
C 60 C O N T I N U E
C
C N = 1000 * MESTO(1) + 100 * MESTO(2) +
C * 10 * MESTO(3) + MESTO(4)
C
C IF ( AX(2) .EQ. BX(2) ) GO TO 150
C
C DO 70 I = 1, 24
C IF ( N .EQ. IPER(I) ) GO TO 75
C 70 C O N T I N U E
C
C 75 GO TO ( 101, 101, 102, 102, 103, 103, 104, 104, 104, 104,
C * 105, 105, 106, 106, 107, 107, 108, 108, 109, 109,
C * 110, 110, 110, 110 ), I
C
C 101 IF ( BX(1) .EQ. AX(2) ) GO TO 301
C RETURN
C 102 IF ( BX(1) .EQ. BX(2) ) GO TO 301
C RETURN
C 103 IF ( BY(1) .EQ. AX(2) ) GO TO 301
C AX(3) = AX(2)
C BX(3) = BX(1)
C GO TO 235

```

```

104 AX(3) = AX(2)
    BX(3) = BX(2)
    GO TO 240

105 IF ( BX(2) ,EQ, BX(1) ) GO TO 301
    AX(3) = BX(2)
    BX(3) = BX(1)
    GO TO 255

106 IF ( AX(1) ,EQ, AX(2) ) GO TO 306
    R E T U R N
107 IF ( AX(1) ,EQ, BX(2) ) GO TO 306
    R E T U R N
108 IF ( AX(1) ,EQ, AX(2) ) GO TO 306
    AX(3) = AX(2)
    BX(3) = AX(1)
    GO TO 285

109 IF ( BX(2) ,EQ, AX(1) ) GO TO 306
    AX(3) = BX(2)
    BX(3) = AX(1)
    GO TO 295

110 AX(3) = AX(1)
    BX(3) = BX(1)
    GO TO 300

150 DO 160 I = 1, 24
    IF ( N ,EQ, IFEP(1) ) GO TO 165

160 C O N T I N U E

165 GO TO ( 210, 210, 220, 220, 230, 230, 240, 240, 240, 240,
    *      250, 250, 260, 260, 270, 270, 280, 280, 290, 290,
    *      300, 300, 300, 300 ), I

210 IF ( BY(1) ,EQ, AY(2) ) GO TO 302
    R E T U R N
220 IF ( BY(1) ,EQ, BY(2) ) GO TO 302
    R E T U R N
230 IF ( BY(1) ,EQ, AY(2) ) GO TO 302
235  AY(3) = AY(2)
    BY(3) = BY(1)
    GO TO 410

240 AY(3) = AY(2)
    BY(3) = BY(2)
    GO TO 410

250 IF ( BY(2) ,EQ, BY(1) ) GO TO 302
255  AY(3) = BY(2)
    BY(3) = BY(1)
    GO TO 410

260 IF ( AY(1) ,EQ, AY(2) ) GO TO 307
    R E T U R N
270 IF ( AY(1) ,EQ, BY(2) ) GO TO 307
    R E T U R N

```

```

280 IF ( AY(1) .EQ. AY(2) ) GO TO 307
285   AY(3) = AY(2)
      BY(3) = AY(1)
      GO TO 410
C
290 IF ( BY(2) .EQ. AY(1) ) GO TO 307
295   AY(3) = BY(2)
      BY(3) = AY(1)
      GO TO 410
C
300 AY(3) = AY(1)
      BY(3) = BY(1)
      GO TO 410
C
C
301 AX(3) = BX(1)
302 AY(3) = BY(1)
      GO TO 400
C
306 AX(3) = AX(1)
307 AY(3) = AY(1)
C
C
400 IND = 1
      R E T U R N
410 IND = 2
      R E T U R N
      E N D

```

— * —

FUNCTION X P O I N T (XHKF, YHKF, NG)

```

C
C .....
C * NAMJENA: ODREĐIVANJE PODREĐIATE APSCISE AKO JE ORDINATA = C.O.
C *          KADA JE VRIJEDNOST FUNKCIJE VISEZNAČNA JER JE POLIGON
C *          (HODOGRAF) KONKAVAN ODABIRA SE NAJMANJA VRIJEDNOST
C .....
C
C DIMENSION XHKF(90), YHKF(90), XPT(20)
C
C PRINT 5
5 FORMAT(' ..... ULAZ U SUBR- X P O I N T .....')
C
RX = 0
NGPI = NG + 1
XHKF(NGPI) = XHKF(1)
YHKF(NGPI) = YHKF(1)
DO 40 I = 1, NG
  XA = XHKF(I)
  YA = YHKF(I)
  XB = XHKF(I + 1)
  YB = YHKF(I + 1)
  IF (ABS(YA) + ABS(YB) .EQ. ABS(YA + YB)) GO TO 40
  DX = XB - XA
  LY = YB - YA
  IF (DX .EQ. 0.0) GO TO 15
  A = DY / DX
  IF (LY .EQ. 0.0) GO TO 16
  B = YA - A * XA

```

```

      KX = KX + 1
      XPT(KX) = -E / A
      IF (XPT(KX) .GT. 0.01) GO TO 40
      KX = KX - 1
      GO TO 40
15  IF (XA .LT. 0.0) GO TO 40
      KX = KX + 1
      XPT(KX) = XA
      GO TO 40
16  KX = KX + 1
      IF (XA .LT. XB) GO TO 28
      IF (XB .LT. 0.0) GO TO 30
      XPT(KX) = XB
      GO TO 40
28  IF (XA .LT. 0.0) GO TO 35
      XPT(KX) = XA
      GO TO 40
30  IF (XA .LT. 0.0) GO TO 37
      XPT(KX) = XA
      GO TO 40
35  IF (XB .LT. 0.0) GO TO 37
      XPT(KX) = XB
      GO TO 40
37  KX = KX - 1
40  C O N T I N U E

```

```

      XPOINT = XPT(I)
      IF (KX .EQ. 1) R E T U R N
45  FORMAT (' FUNKCIJE XPOINT NE DAJE VRIJEDNOST APSCISE')
      IF (KX .NE. 0) GO TO 50
      P R I N T 45
      R E T U R N

50  DO 60 I = 2, KX
      IF (XPOINT .LE. XPT(I)) GO TO 60
      XPOINT = XPT(I)
60  C O N T I N U E

      R E T U R N
      E N D

```

SUBROUTINE OUTPUT (IZ)

```

* * * * *
* NAMJENA : ODREĐIVANJE KOORDINATA OBJEKATA KOJI SE RAZMJESTAJU
*           ZAROTIRANIH ZA KUT OPTIMALNOG RAZMJESTAJA GZ
*           OBJEKTI SU PAZMJESTANI U RELATIVNIM KOORDINATNIM
*           SISTEMIMA, A SADA ĆE BITI DATE NJIHOVE KOORDINATE
*           U ZAJEDNIČKOM KOORDINATNOM SISTEMU
* * * * *

```

```

COMMON /INPUT/ XOBJ(50,10),YOBJ(50,10),NVII(10),NOBJ
*       /KORAK/ DQS
*       /IERR/ IERP
*       /RLINE/ XR(360,10),YR(360,10)
*       /NPINT/ IIP
*       /PERIOD/ TE(360)
*       /CILJ/ FC(360)

```


PROGRAM AXB

* * * * *
 * NAMJENA: U PRAVOKUTNIK ZADANE DUZINE I SIRINE N-REDNO SMJESTITI *
 * OBJEKTE S(J), J=1, NOBJ, TAKO DA SE PRAVOKUTNIK DIJELI *
 * NA POJASEVE UZDUZ STRANICE DUZINE.
 * * * * *

```

COMMON /INPUT/ XOB(50, 10), YOB(50, 10), NVH(10), NOBJ
  * /KORAK/ DGS
  * /RLINE/ XR(360, 10), YR(360, 10)
  * /FOJA/ SIR, LUZ
  * /VIFKF/ FKF(1000, 5), NFK(360, 5)
  * /EGEND/ POP(360, 2)
  * /NPINT/ NIN
  * /PERIOD/ TE(360)
  * /CILJ/ FC(360)
  * /IERR/ IERR
  * /MENA/ QM(360), NT(360), FM12(1000), KI, HI
  * /EXEP/ XP(90), YP(90), NN
  * /POTP/ H(360), NH
  * /POFOS/ PTP(360, 5)
  
```

```

DIMENSION XZ(50), YZ(50)
DIMENSION S(10), NOBR(10), POVR(10), PRIV(10)
DIMENSION SIF(360)
  
```

```

READ 2, NSIR, DSIR
2 FORMAT(I2, F10.5)
  
```

```
IERR = 0
```

```

CALL U L A Z
  IF(IERR .EQ. 1) STOP 111
CALL P O T K O N
  
```

```

IF(NOBJ .LE. 2) GO TO 8
  NMI = NOBJ - 1
  DO 6 I = 2, NMI
    NH = NVH(I)
    DO 3 II = 1, NH
      JJ = NH + 1 - II
      XP(II) = XOB(JJ, 1)
      YP(II) = YOB(JJ, 1)
    3 CONTINUE
  
```

```
3 CONTINUE
```

```
CALL P O T P O R
```

```
DO 4 II = 1, NH
  PTP(II, 1) = H(II)
```

```
4 CONTINUE
```

```
6 C O N T I N U E
```

```
DO 7 I = 1, 2
```

```
IX = 1
```

```
IF (I .EQ. 2) IX = NOBJ
```

```
DO 7 II = 1, NIN
```

```
PTP(II, IX) = POP(II, I)
```

```
7 CONTINUE
```

```

8 DO 15 I = 1, NOBJ
  NZ = NVH(I)
  DO 9 J = 1, NZ
    XZ(J) = XOB(J, I)
    YZ(J) = YOB(J, I)
9 CONTINUE

```

```

C
S(I) = SF(XZ, YZ, NZ)
C

```

```

15 CONTINUE
C

```

```

DO 30 I = 1, NIN
  PER = FKF(I, I)
  DO 20 J1 = 2, NOBJ
    PER1 = FKF(I, J1)
    IF (PER1 .LE. PER) GO TO 20
    PER = PER1
20 CONTINUE
  TE(I) = PER
  QH(I) = DRS * I
  PRINT 25, QH(I)
25 FORMAT(/' KUT ROTACIJE = ', F8.1)
C

```

```

C
NII = NOBJ - 1
DO 28 J = 1, NII
  J1 = J + 1
  CALL X I Y I (I, J, PER)
28 CONTINUE
C

```

```

NOE = NOBJ
SIF(I) = S I R I M A (I, NOE)
30 CONTINUE
C

```

```

DO 110 IS = 1, NSIR
  SSIR = SIR + USIR * (IS - 1)
  PRINT 32, SSIR
32 FORMAT(III, ' ZA SIRINU ROLNE ', F10.5)
C

```

```

PRINT 33
33 FORMAT(///' KUT OBJ. DR. POJ. SIRIJA DOJ. UR. SIRIJ
  ' PRIV. UMANJENJE ISKORISTENJE PERIOD.
  ' BROJ OBJEKATA U REDOVINA FOJASA')
C

```

```

FCHAX = 0.0
C

```

```

DO 80 I = 1, NIN
  WEP = SSIR / SIF(I)
  I1 = NIN / 4 + 1
  IF (I1 .GT. NIN) I1 = I1 - NIN
  I2 = 3 * NIN / 4 + 1
  IF (I2 .GT. NIN) I2 = I2 - NIN
C

```

```

UHA = 0.0
PRIV(I) = 0.0

```

```

DO 35 J = 2, NOBJ
  PRIV(J) = PTP(I2, J-1) - PTP(I2, J) + PRIV(J-1) + XR(I, J-1)
  IF (PRIV(J) .GE. 0.0 .OR. PRIV(J) .GE. UMA) GO TO 35
  UMA = PRIV(J)
35 CONTINUE

IF (UMA .EQ. 0.0) GO TO 40
DO 40 J = 1, NOBJ
  PRIV(J) = PRIV(J) - UMA
40 CONTINUE

DO 50 J = 1, NOBJ
  PRID = DUZ - PRIV(J)
  NOST = PRIV(J) / TE(I)
  NNNN = (PRID - PTP(I1, J) - PTP(I2, J)) / TE(I)
  NOBR(J) = NNNN + NOST + 1
  POVR(J) = S(J) * NOBR(J)
50 CONTINUE

UKP = 0.0
DO 60 J = 1, NOBJ
  UKP = UKP + POVR(J)
60 CONTINUE

FC(I) = (UKP * NEP) / (DUZ * SSIR)
SUK = SIP(I) * MBP

PRINT 65, QH(I), NEP, SIP(I), SUK, PRIV(1), FC(I), TE(I),
      (NOBR(I1), I1=1, NOBJ)
65 FORMAT(IX, F8.1, I7, 2F12.5, F15.5, 2F12.5, 6I5)

DO 70 J = 1, NM1
  IF (YR(I, J) .LE. 0.0) GO TO 80
70 CONTINUE

IF (FCMAX .GE. FC(I)) GO TO 80
FCMAX = FC(I)
IZ = I
80 CONTINUE

CALL OUTPUT (IZ)

KPI = KI + 1
QMNULA = 0.0
WRITE(10, 85) KPI
WRITE(10, 90) QMNULA, FC(K1), (QH(I), FC(I), I = 1, 3)
WRITE(10, 90) (QH(I), FC(I), I = 4, K1)
85 FORMAT(I5)
90 FORMAT(4(F8.1, F12.5))

110 CONTINUE

ENDFILE(10)

STOP
END

```

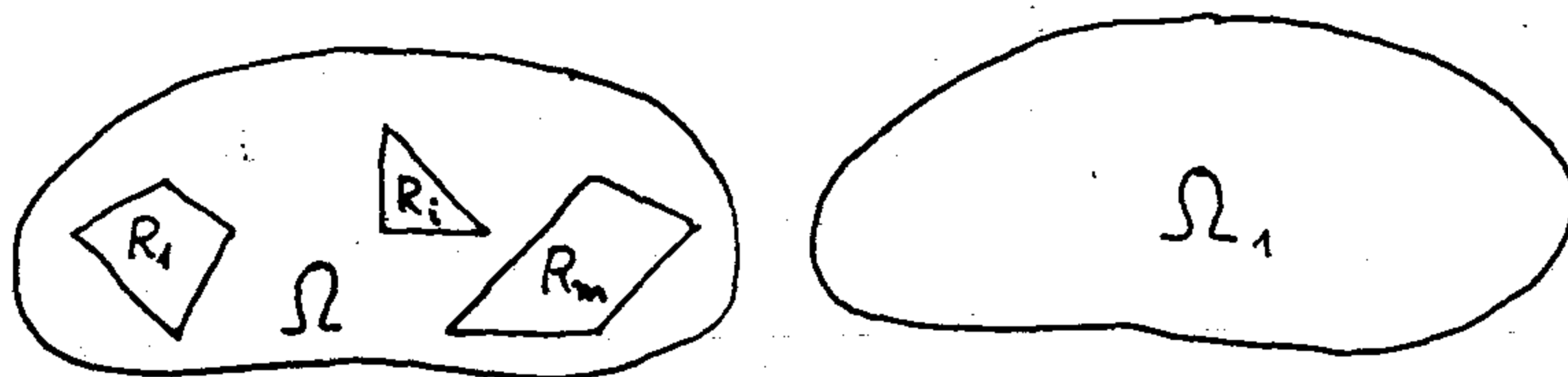

G L A V A V I

NEREGULARNO SMJEŠTANJE ORIJENTIRANIH OBJEKATA U RAVNINI

1. Formulacija problema, uvjeti međusobnog nepresjecanja objekata i uvjeti smještanja objekata u oblast

Neka su zadani međusobno orijentirani ograničeni objekti $\{S_i\}_n$ koje treba smjestiti u zadanu višestruko suvislu oblast Ω proizvoljna oblika. Neka u oblasti Ω_1 postoje podoblasti R_i $i = (1, m)$ u koje se objekti ne smiju smještati, tj.

$$\Omega = \Omega_1 \setminus \left(\bigcup_{i=1}^m R_i \right) \quad (6.1.1)$$



Sl. 6.1

Da se objekti S_i i S_j ne presjecaju, možemo izraziti pomoću kontakt funkcije

$$(x_i - x_j)^2 + (y_i - y_j)^2 - K_{ij}^2(\theta) \geq 0 \quad (6.1.2)$$

gdje je:

$$\theta = \arctg \frac{y_i - y_j}{x_i - x_j}.$$

U zavisnosti od vrste objekata koji se smještaju (razdjeljivi, nerazdjeljivi, konveksni, nekonveksni itd.) kao i od izbora po-

la, kontakt funkcija može biti jednoznačna, ali općenito je višeznačna, tj.

$$K_{ij}(\theta) = K_{ij}^r(\theta) \quad r = 1, 2, \dots, m(\theta) \quad (6.1.3)$$

Naime, ako izaberemo pol HKF $O_{ij} \in G_{ij}$, onda je $m(\theta)$ paran broj, inače je neparan. Ako je $m(\theta)$ paran, onda se uvjeti nepresjecanja S_i i S_j mogu prikazati pomoću sistema nejednažbi

$$\begin{aligned} & (x_i - x_j)^2 + (y_i - y_j)^2 - [K_{ij}^{2n+2}(\theta)]^2 \geq 0 \\ & - (x_i - x_j)^2 - (y_i - y_j)^2 + [K_{ij}^{2n+1}(\theta)]^2 \geq 0 \end{aligned} \quad (6.1.4)$$

$$n = 0, 1, 2, \dots, \frac{m(\theta) - 2}{2}$$

Pomoću operacije R_2 - disjunkcije \bigvee_1 i konjunkcije \bigwedge_1 sistem (6.1.4) možemo zapisati pomoću jedinstvenog izraza

$$\bigvee_{n=0}^{\frac{m-2}{2}} \left\{ [K_{ij}^{2n+2}(\theta)]^2 - (x_i - x_j)^2 - (y_i - y_j)^2 \bigwedge_1 (x_i - x_j)^2 + (y_i - y_j)^2 - [K_{ij}^{2n+1}(\theta)]^2 \right\} \geq 0 \quad (6.1.5)$$

gdje je:

$$K_{ij}^{2n+1}(\theta) \leq K_{ij}^{2n+2}(\theta); \quad n = 0, 1, 2, \dots, \frac{m(\theta) - 2}{2}$$

Za $m(\theta)$ neparno i $m(\theta) > 3$ imamo

$$\begin{aligned} & \bigvee_{n=0}^{\frac{m-1}{2}} \left\{ [K_{ij}^{2n}(\theta)]^2 - (x_i - x_j)^2 - (y_i - y_j)^2 \bigwedge_1 (x_i - x_j)^2 + (y_i - y_j)^2 - \right. \\ & \quad \left. - [K_{ij}^{2n+1}(\theta)]^2 \right\} \geq 0 \end{aligned} \quad (6.1.6)$$

Ako označimo sa $K_{tj}(\theta)$ kontakt funkciju oblasti R_t i objekta S_j , a (c_t, d_t) koordinate pola R_t , a $\xi(\theta)$ označava broj grana

$K_{tj}(\theta)$, onda za $\xi(\theta)$ parno imamo uvjete nepresjecanja R_t i S_j .

$$\sqrt[n=0]{\frac{\xi-2}{2}} \{ [K_{tj}^{2n+1}(\theta)]^2 - (x_j - c_t)^2 - (y_j - d_t)^2 \wedge_1 (x_j - c_t)^2 + (y_j - d_t)^2 - [K_{tj}^{2n+2}(\theta)]^2 \} \geq 0 \quad (6.1.7)$$

a za $\xi(\theta)$ neparan

$$\sqrt[n=0]{\frac{\xi-1}{2}} \{ [K_{tj}^{2n}(\theta)]^2 - (x_j - c_t)^2 - (y_j - d_t)^2 \wedge_1 (x_j - c_t)^2 + (y_j - d_t)^2 - [K_{tj}^{2n+1}(\theta)]^2 \} \geq 0 \quad (6.1.8)$$

Ako je točka $(x_r, y_r) \in g_{ij}$ izolirana točka, onda je $K^{2n}(\theta) = K^{2n+1}(\theta)$.

Kanonska jednadžba HKF objekata S_i i S_j u odnosu na pol O_i je oblika

$$z_{ij}(x', y') = 0, \quad (6.1.9)$$

odnosno u odnosu na fiksni sistem XOY

$$z_{ij}(x' - x_i, y' - y_i) = 0.$$

Sada uvjete medjusobnog nepresjecanja S_i i S_j možemo predstaviti sa:

$$z_{ij}(x_j - x_i, y_j - y_i) \geq 0. \quad (6.1.10)$$

Uvjete medjusobnog nepresjecanja oblasti zabrane za smještaj R_t i objekata S_i čiji je HKF dan kanonskom jednadžbom

$$z_{ti}(x', y') = 0 \quad (6.1.11)$$

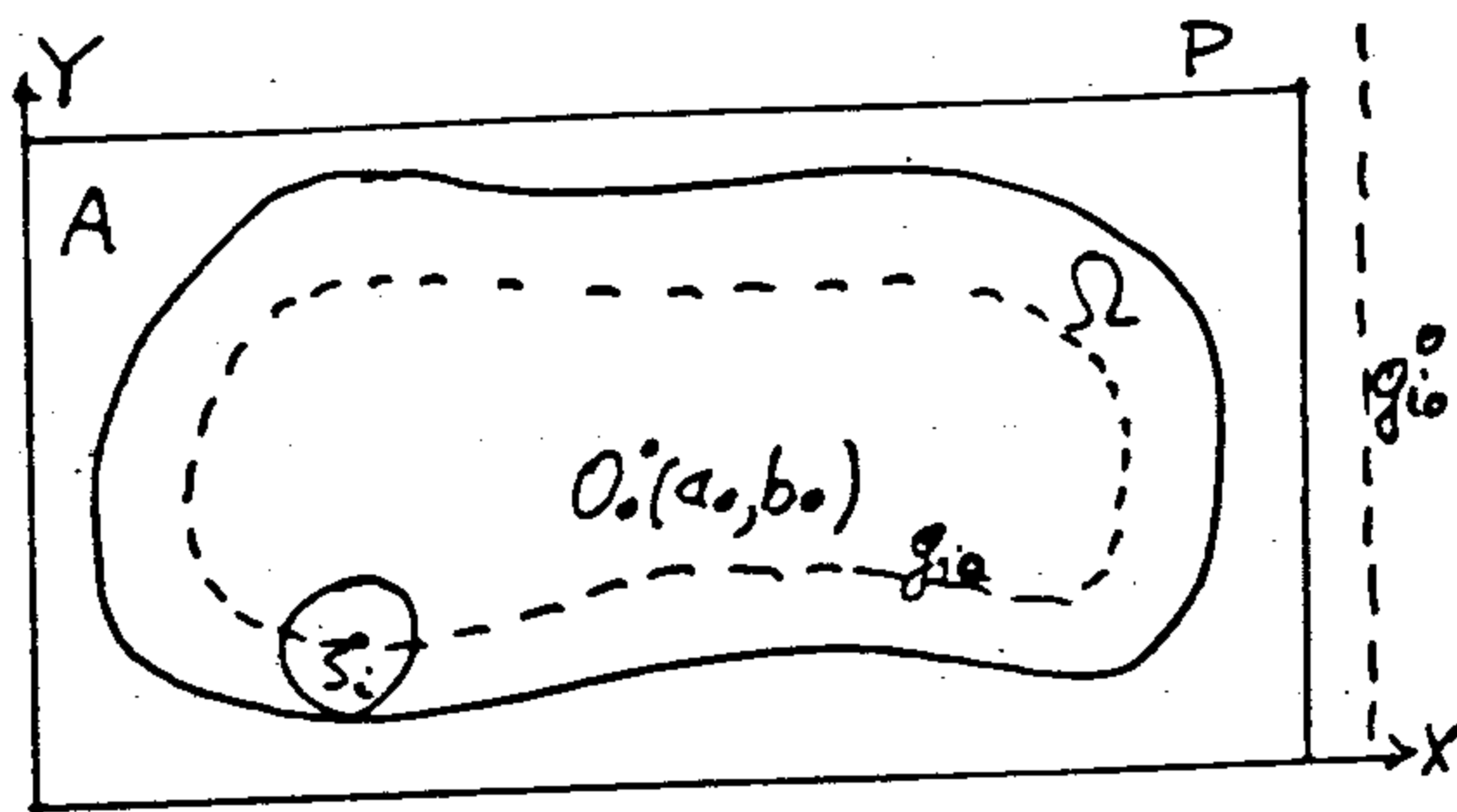
možemo izraziti sa:

$$z_{ti}(x_i - c_t, y_i - d_t) \geq 0.$$

Prema tome objekti koji se smještaju u neku oblast Ω ne smiju se presjecati ni medjusobno ni sa oblastima zabrane.

Uvjeti da se objekt smješta u Ω

Neka je zadana suvisla oblast Ω , sl.6.2, u koju treba smjestiti objekt S_i . U tu svrhu promatramo višestruko suvislu oblast $A = P \setminus \Omega$ gdje je $\Omega \subset P$.



sl. 6.2

Jednostavnosti radi pravokutnik P smjestimo kao na slici 6.2. Pol objekta A izaberemo tako da bude $O_0(a_0, b_0) \in \Omega$. HKF objekta S_i i A se sastoji od dva dijela $g_{oi} = g_{oi} \cup g_{oi}^o$ - gdje je g_{io} - unutarnji, tj. onaj koji nas interesira. U zavisnosti od toga gdje smo izabrali pol $O_0(a_0, b_0)$, kao i od oblika oblasti Ω kontakt funkcija može biti jednoznačna ili višeznačna na nekim intervalima. Kod toga broj $n(\theta)$ vrijednosti $K(\theta)$ može biti paran ili neparan. Uvjet da se S_i smješta u Ω za $n(\theta)$ paran je dan izrazom

$$\sum_{n=0}^{\frac{n(\theta)-2}{2}} \left\{ [K_i^{2n+1}(\theta)]^2 - (x_i - a_0)^2 - (y_i - b_0)^2 \wedge 1(x_i - a_0)^2 + (y_i - b_0)^2 - [K_i^{2n+2}(\theta)]^2 \right\} \geq 0, \quad (6.1.12)$$

a za $n(\theta)$ - neparno

$$\frac{\eta-1}{2} \sum_{n=0}^{\infty} \left\{ [K^{2n}(\theta)]^2 - (x_i - a_0)^2 - (y_i - b_0)^2 \sqrt{(x_i - a_0)^2 + (y_i - b_0)^2} - [K^{2n+1}(\theta)]^2 \right\} \geq 0. \quad (6.1.13)$$

Ako je kanonska jednačba unutarnjeg dijela HKF - g_{i0} dana sa $Z_{oi}(x', y') = 0$, onda ona u odnosu na XOY glasi: $Z_{oi}(x - a_0, y - b_0) = 0$ pa se uvjet smještaja S_i u Ω može izraziti sa:

$$Z_{oi}(x_i - a_0, y_i - b_0) \geq 0. \quad (6.1.14)$$

U praksi se pojavljuju razna ograničenja kao npr.:

a) udaljenost između objekata S_i i S_j mora zadovoljavati uvjet

$$\rho_{ij}(S_i, S_j) \geq l_{ij},$$

b) udaljenost između objekta S_i i oblasti zabrane R_t mora zadovoljavati uvjet

$$\rho_{it}(S_i, R_t) \geq l_{it},$$

c) udaljenost od ruba oblasti Ω do objekta S_i mora zadovoljavati uvjete

$$\rho_i(S_i, \Omega) \geq l_i.$$

prednjim uvjetima osiguravaju se razni protupožarni, manipulativni, saobraćajni zahtjevi itd. Jedan od primjera za navedena ograničenja je slučaj kod proširenja rafinerijskih kompleksa. Naime već postojeća postrojenja na terenu moraju se u projektu uzeti u obzir kao oblasti zabrane, između postrojenja mora biti osiguran prostor za mrežu puteva ili pruga, električni agregati ne smiju biti blizu objekata gdje se proizvode neke zapaljive smjese itd.

2. Formalizacija funkcije cilja

Treba naći analitički izraz za funkciju cilja kada oblast u koju se smještaju objekti ima pomične granice [13].

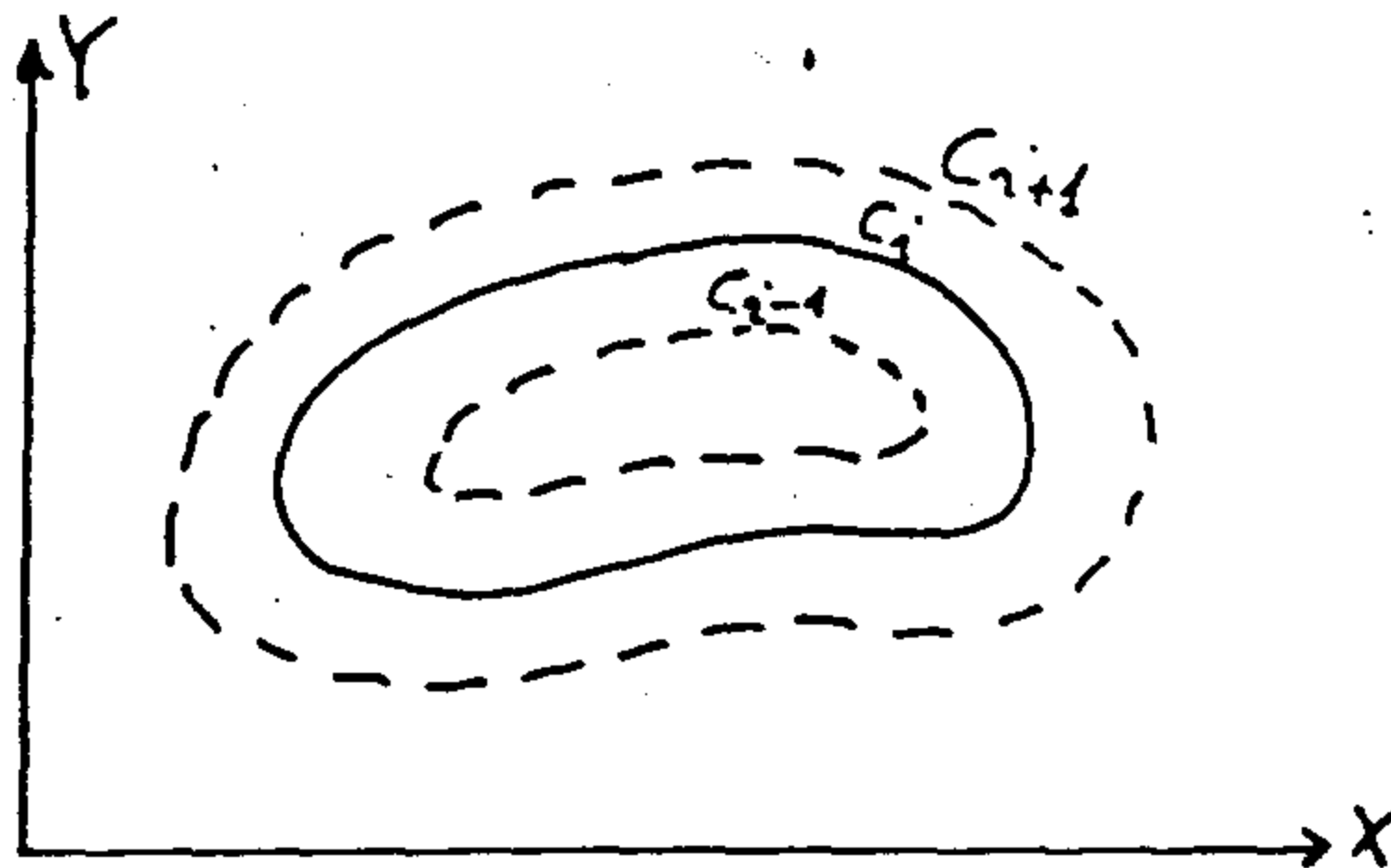
U tu svrhu navedimo neke slučajeve:

I) Skup oblasti je zadan nejednadžbom

$$Z(x, y, C) \geq 0 \quad (6.2.1)$$

gdje parametar C određuje veličinu oblasti.

Ako vrijedi $C_{i-1} < C_i < C_{i+1}$, onda se radi o oblastima koje su uložene jedne u druge, sl.6.3.



sl.6.3

II) Neka se kontura oblasti Ω sastoji od više krivulja

$$\phi_j(x, y) \equiv y - h_j(x) = 0 \quad h_j = 1, m \quad (6.2.2)$$

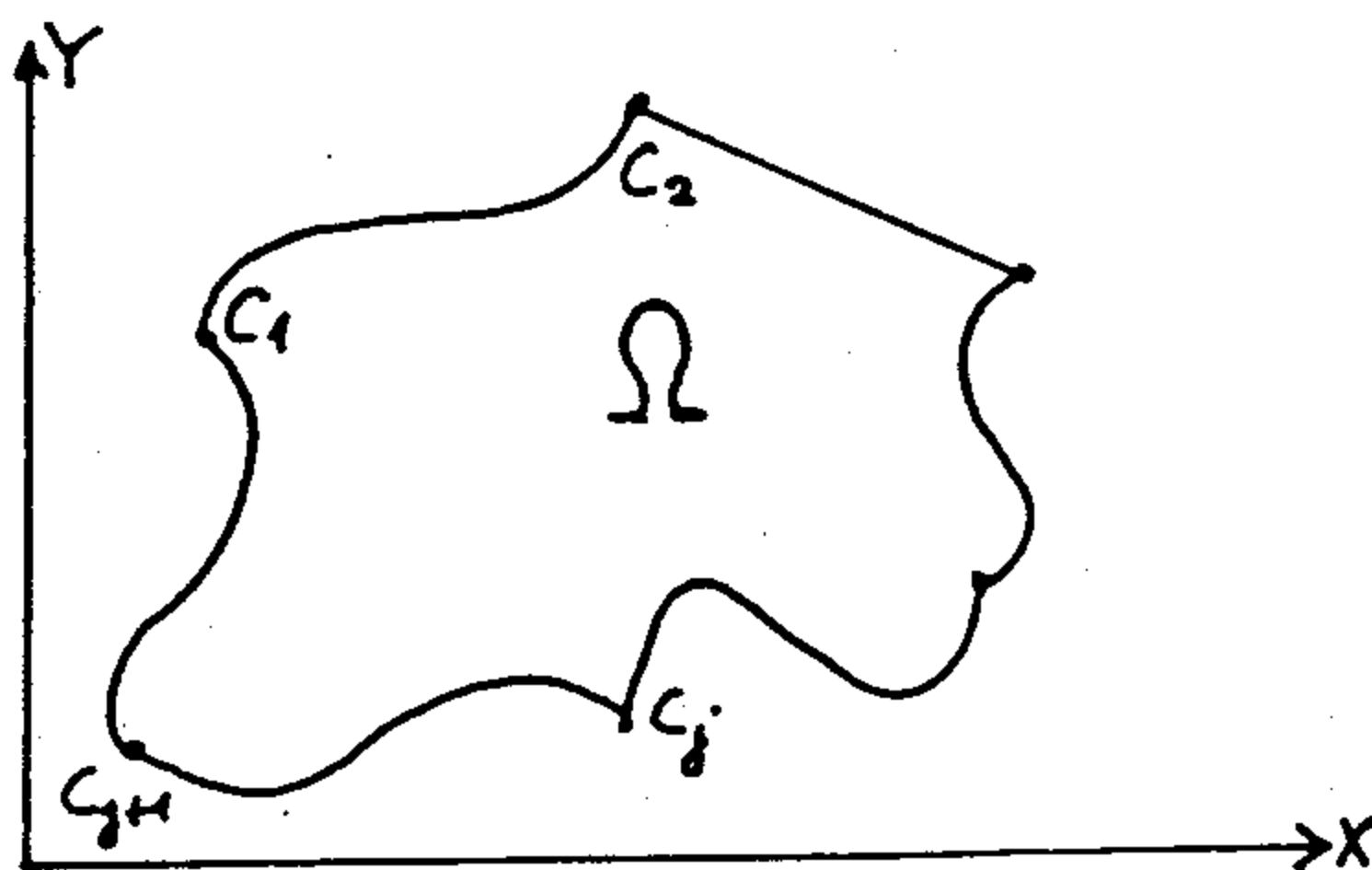
gdje $h_j(x) \in C$. Konturu čine lukovi $C_j \dots C_{j+1}$ navedenih krivulja.

Neka su $(x_{ij}, y_{ij}, \alpha_{ij})$ pripadni parametri smještaja krivulja (6.2.2) u fiksnom sistemu XOY, sl.6.4.

Na parametre $(x_{ij}, y_{ij}, \alpha_{ij})$ možemo stavljati razna ograničenja, recimo ograničenje da se linije mogu premještati paralelno same sebi:

$$\begin{aligned} \alpha_{ij} &= \text{const} \\ y_{ij} &= -\frac{1}{\text{tg} \alpha_{ij}} (x_{ij} - C_j + K_j) \end{aligned}$$

$$\text{ili } x_{ij} = \text{tg} \alpha_{ij} (K_j - y_{ij}) + g_j \quad j(1, m) \quad (6.2.3)$$



Sl. 6.4

Pored toga parametri različitih krivulja od kojih se sastoji kontura Ω mogu biti međusobno povezani linearno ili nelinearno

$$\begin{aligned} x_{ij} &= \psi_{1j}(x_{1K}) \\ y_{ij} &= \psi_{2j}(y_{1K}) \quad l = 1, m; \quad K = 1, m; \quad K \neq j. \end{aligned} \quad (6.2.4)$$

Prema tome kada dodje do promjene parametara x_{ij} ili y_{ij} (ili oba), mijenjaju se i parametri zadani sa (6.2.3) i (6.2.4), dakle, mijenjaju se dimenzije oblasti Ω .

III) Pretpostavimo da objekte S_i treba smjestiti u Ω , tj. da

bude $\bigcup_{i=1}^n S_i \subset \Omega$, onda uvjeti (6.2.4) nisu zadani već parametre smještaja krivulja (x_{ij}, y_{ij}) odredjujemo iz uvjeta da koeficijent popunjenosti bude maksimalan, tj. da površina Ω bude minimalna. Upravo ovaj slučaj ima veliku primjenu pa zato formulirajmo pripadnu funkciju cilja.

Opći položaj krivulja zadanih sa (6.2.2) ima oblik

$$\begin{aligned} \phi_j \left[(X-x_{1j}) \cos \alpha_{1j} + (Y-y_{1j}) \sin \alpha_{1j}; \right. \\ \left. \lambda (X-x_{1j}) \sin \alpha_{1j} + (Y-y_{1j}) \cos \alpha_{1j} \right] = 0 \quad j=1, m \end{aligned} \quad (6.2.5)$$

Ako u (6.2.5) uvedemo ograničenja dana sa (6.2.3) imamo

$$\begin{aligned} \phi_j(X, Y, y_{1j}) &= z_j \left\{ [X - (K_j - y_{1j}) \operatorname{tg} \alpha_{1j} - z_j] \cos \alpha_{1j} + (Y - y_{1j}) \sin \alpha_{1j}; \right. \\ &\left. - [X - (K_j - y_{1j}) \operatorname{tg} \alpha_{1j} - z_j] \sin \alpha_{1j} + (Y - y_{1j}) \cos \alpha_{1j} \right\} = 0; \quad j=1, m \quad (6.2.6) \end{aligned}$$

Pretpostavimo da se (6.2.6) može prikazati eksplicite

$$X = z_j^1(Y, y_{1j}) \quad j = 1, m \quad (6.2.7)$$

ili parametarski

$$\begin{aligned} X &= F_{1j}^{(1)}(t, y_{1j}) \\ Y &= F_{2j}^{(1)}(t, y_{1j}). \end{aligned}$$

Tada se izračunavanje površine Ω svodi na računanje krivuljnog integrala

$$P_{\Omega} = \int_K X dY = \sum_{j=1}^m \int_{C_j}^{C_{j+1}} z_j^{(1)}(Y, y_{1j}) dY. \quad (6.2.8)$$

Koordinate točke $C_j(X_j, Y_j)$: $j = 1, m$ određuju se rješavajući sistem jednažbi

$$\begin{aligned} \phi_j(X, Y, y_{1j}) &= 0 \\ \phi_{j+1}(X, Y, y_{1j}) &= 0 \quad j = 1, m \end{aligned} \quad (6.2.9)$$

Znači da je $X_j = X_j(y_{1j}, y_{1, j+1})$

$$Y_j = Y_j(y_{1j}, y_{1, j+1}). \quad (6.2.10)$$

Slijedi da je površina P_{Ω} - funkcija od y_{1j} ; $j = 1, m$, tj.

$$P_{\Omega}(y_{11}, y_{12}, \dots, y_{1m}) = \sum_{j=1}^m D_j(y_{1j}, y_{1, j+1}). \quad (6.2.11)$$

Pošto tražimo da P_{Ω} - bude minimalna, onda će sigurno svaki luk C_j, C_{j+1} dirati bar jedan od objekata koji se smještaju. Različitom smještaju objekata u Ω odgovaraće i različita diranja objekata i lukova C_j, C_{j+1} . Znači da će različitim položajima $\{S_i\}$ u Ω odgovarati različiti parametri y_{1j} ($j = 1, m$).

Tvrdnja 6.1.

Ako su (x_i, y_i, θ_i) parametri objekta $\{S_i\}_n$, onda vrijedi

$$y_{1j} = y_{1j}(x_1, x_2, x_3, \dots, x_n, y_1, y_2, y_3, \dots, y_n, \theta_1, \theta_2, \theta_3, \dots, \theta_n).$$

Dokaz:

Recimo da objekt S_k dira luk C_j, C_{j+1} . Odredjivanje koordinata diranja (x_{kj}, y_{kj}) svodi se na rješavanje problema uslovnog ekstremuma, tj. na odredjivanje

$$\max (X^2 + Y^2) \quad (*)$$

uz uvjet da je

$$\phi_j(X, Y, y_{1j}) = 0; \quad h_k(X, Y, x_k, y_k, \theta_k) = 0$$

$$h_k(X, Y, x_k, y_k, \theta_k) - Z = 0$$

$h_k(X, Y, x_k, y_k, \theta_k)$ - jednačba općeg položaja objekta S_k , Z - pomoćna varijabla pomoću koje osiguravamo da nijedna točka luka C_j, C_{j+1} ne leži u S_k .

Pretpostavimo da su: $\phi_j(X, Y), H_k(X, Y) \in C^{(1)}$, onda koristeći formulu Lagranža za rješavanje (*) dobijemo:

$$x_{kj} = x_{kj}(x_k, y_k, \theta_k, y_{1j}); \quad y_{kj} = y_{kj}(x_k, y_k, \theta_k, y_{1j}) \quad (**)$$

Supstituiramo (**) mjesto X, Y u (6.2.6) ili (6.2.7) i odredimo

$$y_{1j}^{(1)} = y_{1j}^{(1)}(x_k, y_k, \theta_k).$$

Uzmemo li u obzir da svaki objekt $S_k; k = 1, n$ može dirati (C_j, C_{j+1}) , onda se parametar y_{1j} odredjuje iz

$$y_{1j} = \max \{ |y_{1j}^{(1)}|, |y_{1j}^{(2)}|, \dots, |y_{1j}^{(m)}| \}.$$

Pomoću R_2 - disjunkcije imamo:

$$y_{1j}(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n; \theta_1, \theta_2, \dots, \theta_n) = \bigvee_{i=1}^n y_{1j}^{(1)}(x_i, y_i, \theta_i).$$

Iz tvrdnje 6.1. i (6.2.11) slijedi da je:

$$P_{\Omega}(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n; \theta_1, \theta_2, \dots, \theta_n) =$$

$$= \sum_{j=1}^m D_j(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n; \theta_1, \theta_2, \dots, \theta_n) \quad (6.2.12)$$

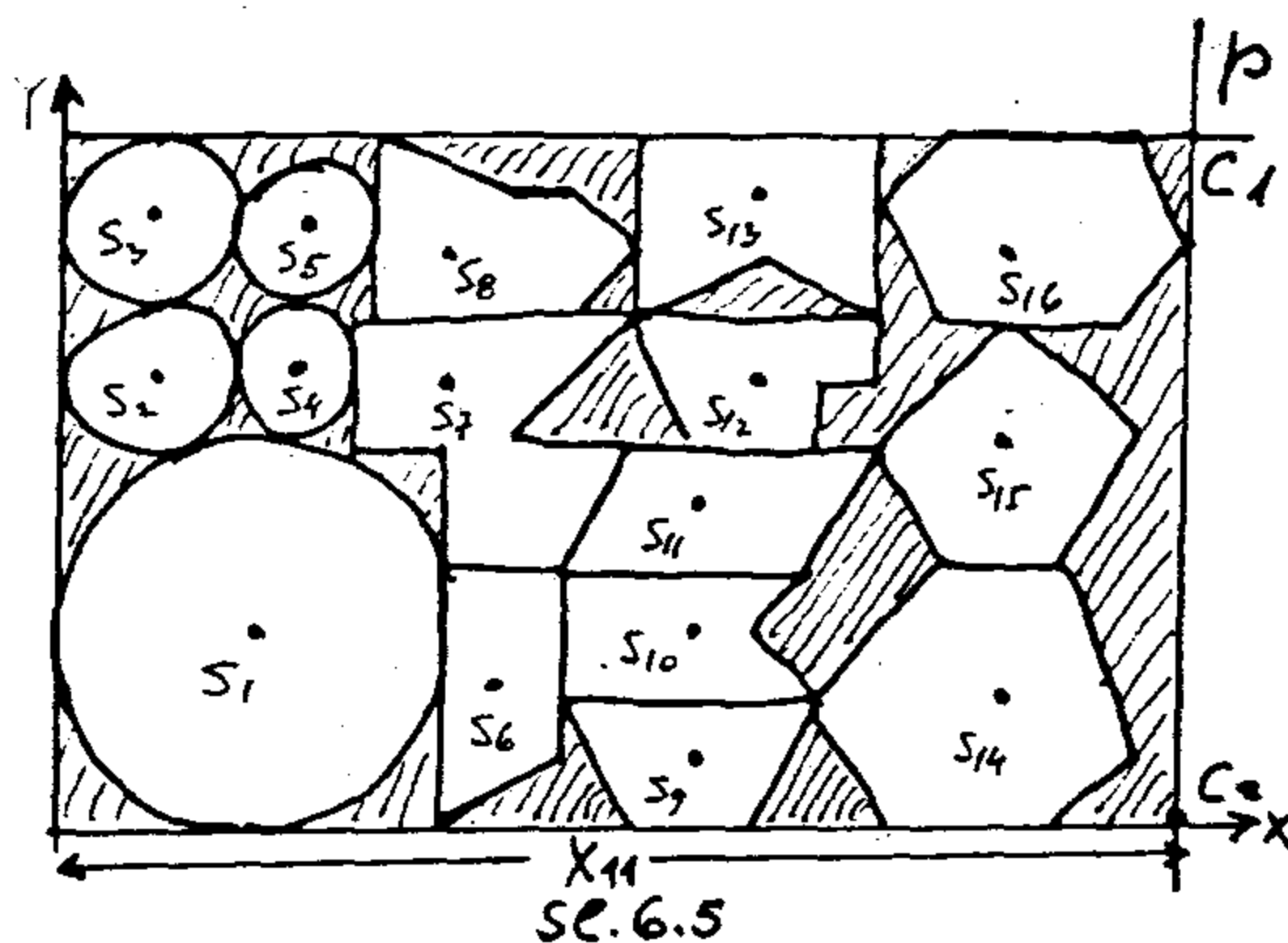
Znači da funkcija cilja u problemima kontakt rasporeda objekata u oblasti s pomičnim granicama zavisi samo od parametara objekata koji se smještaju u Ω .

Tvrdnja 6.2.

Ako je Ω u koju se smješta S_k konveksna oblast, onda se koordinate diranja (X_{kj}, Y_{kj}) objekta S_k i granice Ω podudaraju s koordinatama diranja konveksne ljuske objekta S_k sa Ω .

Specijalni slučajevi koji se u praksi krojenja materijala često susreću:

- A) U polubeskonačnu traku širine H treba smjestiti zadani broj krugova i poligona tako da utrošena duljina trake bude minimalna. Sada je Ω pravokutnik - suvisli. Traku smjestimo kao na sl.6.5.



Neka su parametri krugova (x_i, y_i) $i=(1, n_1)$ a parametri poligonih oblasti $(x_{n_1+1}, y_{n_1+1}, \theta_{n_1+1})$ $i = (1, n_2)$. Funkciju cilja koja predstavlja P_Ω u kojoj su smješteni svi objekti dobijemo na slijedeći način. Uzmemo pravac p koji pomičemo paralelno s osi OY, sve dotle dok ne dirne prvi objekt. Tada je $OC_2 = X_{11}$ zadana duljina iskorištene trake. Analitički izraz je:

$$X_{11} = \bigvee_{i=1}^n X_{11}^{(i)}(x_i, y_i, \theta_i) \quad (6.2.13)$$

$X_{11}^{(i)}(x_i, y_i, \theta_i)$ - apscisa najudaljenije točke objekta S_i od OY.
Za krugove radijusa R_i ($i = 1, n_1$) imamo:

$$X_{11}^{(i)} = x_i + R_i \quad i = 1, n_1 \quad (6.2.14)$$

Neka su (a_i^j, b_i^j) ($i = 1, 2, \dots, k_j$) koordinate vrhova j -tog poligona u vlastitom koordinatnom sistemu X', O', Y' . Koordinate tih vrhova u fiksnom sistemu su dane sa:

$$\bar{X}_{11}^{(i)}(x_j, \theta_j) = a_i^j \cos \theta_j - b_i^j \sin \theta_j + x_j.$$

No, kako je Ω u našem primjeru konveksna oblast, na osnovu tvrdnje 6.2. imamo

$$X_{11}^j(x_j, \theta_j) = \bigvee_i^t \bar{X}_{11}^{(i)}(x_j, \theta_j) \quad (j = n_1+1, \dots, n_1+n_2) \quad (6.2.15)$$

Prema tome funkcija cilja je dana izrazom

$$P_\Omega(x_1, x_2, \dots, x_n, \theta_{n_1+1}, \dots, \theta_{n_2}) = H \cdot \left[\bigvee_{i=1}^n X_{11}^{(i)}(x_i, \theta_i) \right]. \quad (6.2.16)$$

B) Slučaj kada se granice Ω mogu pomicati paralelno sa OX i OY. Neka se u Ω koji je pravokutnik smještaju isti objekti kao i u prethodnom slučaju. Znači da se Ω formira paralelnim pomakom pravaca p_1 sa OY i p_2 sa OX, pa su parametri od p_1 odnosno p_2 dani sa izrazima:

$$x_{11} = \bigvee_{i=1}^n x_{11}^i(x_i, \theta_i); \quad y_{11} = \bigvee_{i=1}^n y_{11}^i(y_i, \theta_i) \quad (6.2.17)$$

Vidimo da je P_{Ω} - potpuno određena parametrima (x_{11}, y_{11}) pa možemo funkciju cilja napisati u obliku:

$$P_{\Omega} = x_{11} \cdot y_{11} = \left[\prod_{i=1}^n x_{11}^i(x_i, \theta_i) \right] \times \left[\prod_{i=1}^n y_{11}^i(y_i, \theta_i) \right] \quad (6.2.18)$$

Slučajevima A i B obuhvaćeni su mnogi primjeri iz prakse krojenja u tekstilnoj i metalnoj industriji.

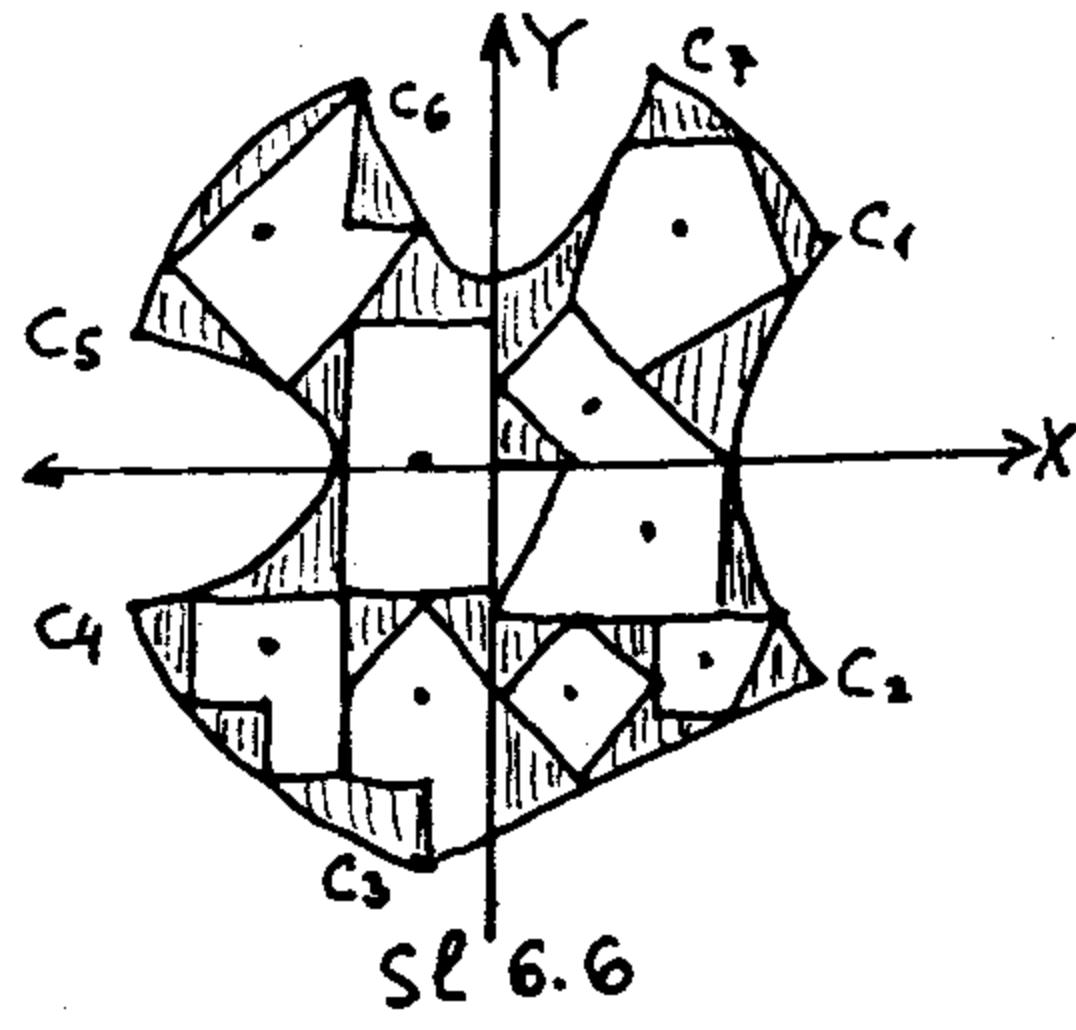
C) Slučaj kada granicu Ω predstavljaju dijelovi pravaca i krivulja II reda, sl.6.4, u koju se smještaju zadani poligoni. Kao što vidimo kontura Ω se sastoji od: C_1-C_2 - dio hiperbole, C_2-C_3 - dio pravca, C_3-C_4 , C_5-C_6 , C_7-C_1 - dio kružnice, C_4-C_5 - dio elipse, C_6-C_7 - dio parabole.

Objekti su peterokuti i četverokuti. Da se dođe do izraza za funkciju cilja u kojoj bi figurirali samo parametri smještaja poligona, trebalo bi:

- koordinate vrhova poligona izraziti pomoću parametara smještaja,
- sukcesivno riješiti odgovarajuće sisteme da se nađu presječne točke C_i izražene pomoću parametara,
- izračunati P_{Ω} , za što je potrebno sve jednačbe krivulja i pravaca napisati u sistemu XOY,
- izraziti da vrhovi objekata diraju konture.

Kada se sve to sredi i napiše onda se dobiju tako glomazni izrazi da za naš primjer na sl.6.6. on ne može stati na jednu stranicu.

Za tako općeniti problem za sada nije uspješno sastavljen odgovarajući algoritam i program za računar, ali se na tome radi.



3. Neke karakteristike neregularnog smještaja objekata u oblasti s pomičnim granicama i svojstva funkcija cilja

Neka je zadana oblast Ω sa (6.1.1) i neka su joj granice pomične. U Ω treba smjestiti međusobno orijentirane objekte $\{S_i\}_n$ s parametrima smještaja (x_i, y_i) . Oblasti zabrane R_i imaju fiksne granice. Znači da se traži:

$$\min_{D \subseteq R^n} P_{\Omega}(x_1, x_2, \dots, x_n; Y_1, Y_2, \dots, Y_n), \quad (6.3.1)$$

Oblast D - definiraju: uvjeti međusobnog nepresjecanja objekata (6.1.5) ili (6.1.6), uvjeti međusobnog nepresjecanja objekata i oblasti zabrane R_i (6.1.7) ili (6.1.8) te uvjeti smještaja objekata $\{S_i\}_n$ u Ω (6.1.12) ili (6.1.13).

Kontakt funkcija u (6.1.12) ili (6.1.13) se mijenja s promjenom oblika oblasti Ω .

Ako navedena ograničenja pojednostavimo, tako da pojednostavimo kontakt funkcije s tim da uvedemo da je:

$$\begin{aligned} r_{ij} &= \min_{\theta \in [0, 2\pi]} K_{ij}(\theta) \\ r_{ti} &= \min_{\theta \in [0, 2\pi]} K_{ti}(\theta) \\ r_i &= \min_{\theta \in [0, 2\pi]} K_i(\theta); \quad i = 1, n, \end{aligned} \quad (6.3.2)$$

tj. da radimo samo sa krugovima.

Pretpostavimo da je pripadni pojednostavljeni sistem jednažbi konzistentan. Taj sistem predstavlja kvadratne forme koje pomoću ortogonalnog operatora A , kome pripada ortogonalna matrica $|A|$, možemo svesti na glavne smjerove, tj. imamo $Z_1^2 + Z_2^2 = \frac{1}{2} r_{12}^2$ što u R^{2n} predstavlja cilindričnu kvadratnu formu s bazom koja s fiksnim koordinatnim sistemom XOY zatvara kut $\pi/4$. Općenito dobijemo $Z_i^2 + Z_j^2 = \frac{1}{2} r_{ij}^2$ $i, j = (1, n); i \neq j$. Znači da su osi

formi međusobno ortogonalne i da prolaze kroz ishodište fiksnog sistema XOY u R^{2n} . Znači da sistem pojednostavljenih jednažbi:

a) koji osigurava nepresjecanje objekata, generira oblast

$$D_1 = \bigcap_{i=1}^n \bigcap_{j=1}^n D_{ij}, \quad D_{ij} = R^{2n} \setminus \bar{C}_{ij}$$

oblast D_1 je zatvorena, neograničena, povezana, nekonveksna, i višestruko suvisla;

b) koji osigurava nepresjecanje objekata i oblasti zabrane

$$D_2 = \bigcap_{i=1}^n \bigcap_{t=1}^m B_{ti}; \quad B_{ti} = R^{2n} \setminus \tilde{T}_{ti}$$

oblast D_2 - je zatvorena, ^{NEOSGRANIČENA} nekonveksna i višestruko suvisla;

c) koji osigurava a) i b) istovremeno

$$D^* = D_1 \cap D_2$$

D^* - je općenito neograničena zatvorena, nekonveksna i nepovezana;

d) koji osigurava da se objekti mogu smjestiti u Ω

$$D^{**} = \bigcap_{i=1}^n A_i$$

D^{**} - je zatvorena, konveksna i ograničena oblast.

Prema tome pojednostavljeni sistem nejednažbi

$$(x_i - x_j)^2 + (y_i - y_j)^2 - r_{ij}^2 \geq 0 \quad i, j=1, n \quad i \neq j$$

$$(x_i - c_t)^2 + (y_i - d_t)^2 - r_{it}^2 \geq 0 \quad i=1, n; \quad t=1, m$$

$$-(x_i - a_0)^2 + (y_i - b_0)^2 + r_i^2 \geq 0 \quad i = 1, n$$

definira oblast $D_0 = D^* \cap D^{**}$ koja je: zatvorena, ograničena,

općenito višestruko suvisla pa čak i nepovezana. Prema tome stvarna oblast $D \subset R^{2n}$ dopustivih rješenja funkcije cilja $P_{\Omega}(X, Y)$ je puno kompliciranija od D_0 .

Iz naprijed iznijetog (pojednostavljenog problema) imamo slijedeće karakteristike neregularnog smještaja:

- 1) oblast rješenja D nije konveksna,
- 2) prostor parametara u kome se traži optimum kod orijentiranih objekata je dimenzije $n_1 = 2n$, a kod neorijentiranih $n_2 = 3n$,
- 3) broj nejednadžbi koje određuju oblast D je $M > \frac{n-1}{2} n + n \cdot m + n$ (n - broj objekata, m - broj zabranjenih područja R_i),
- 4) funkcije koje predstavljaju navedene uvjete nepresjecanja i smještanja su po dijelovima glatke, tj. nisu svuda derivabilne,
- 5) te su funkcije općenito transcendentne, složene i glomazne što se dodatno komplicira zadavanjem uvjeta na minimalne udaljenosti,
- 6) kako se Ω u toku računanja mijenja, mijenjaju se i kontakt funkcije $K_{ij}(\theta)$ pa se na pripadnom sistemu moraju vršiti korekcije
- 7) ako izraze za nelinearnu funkciju cilja napišemo u razvijenom obliku (a ne pomoću \bigwedge_1), izrazi su krajnje glomazni,
- 8) ekstremi funkcije cilja mogu se postizati kako na rubu tako i unutar oblasti definicije D . Čak se mogu postići i u kontinuumu točaka,
- 9) za funkcije oblika

$$F(X) = \bigvee_{i=1}^n F_i(x_i)$$

$$\phi(X) = \bigwedge_{i=1}^n \phi_i(x_i)$$

za $n < \infty$, postoji konačan broj oblasti D_q , u svakoj od kojih možemo izvršiti razlaganje [9]

$$F(X) = \sum_{k=1}^t \psi_k^q [x_{(k-1)r+1}, x_{(k-1)r+2}, \dots, x_{kr}]$$

$$\phi(x) = \sum_{k=1}^t \phi_k^q [x_{(k-1)r+1}, x_{(k-1)r+2}, \dots, x_{kr}]$$

gdje je: $t = \frac{n}{r}$; $r \leq N \left[\frac{1}{2} n^{1/6} \right]$,

10) vrijednosti funkcija oblika

$$F_\alpha(x) = \bigvee_{i=1}^m F_i(x)$$

$$\phi_\alpha(x) = \bigwedge_{i=1}^m \phi_i(x)$$

distribuirane su po zakonu koji asimptotski konvergira k normalnom zakonu za $n \rightarrow \infty$

gdje su: $F_i(x)$; $\phi_i(x)$ $i = 1, \dots, m$ ograničene na zatvorenoj oblasti D ,

11) vrijednosti funkcije cilja $P_\Omega(X, Y, \theta)$

na ograničenoj zatvorenoj oblasti D distribuirane su po zakonu koji asimptotski teži normalnom (log normalnom), ako je izbor vrijednosti nezavisan, i $n \rightarrow \infty$,

12) već kod smještaja konveksnih objekata $\{S_i\}_n$ u polubeskonačni pojas zadane širine, broj lokalnih ekstrema nije manji od $n!$

Naime, ako uzmemo da je $H > \max\{h_1, h_2, \dots, h_n\}$ (h_i - dijаметar od S_i), onda je jasno da objekte možemo u Ω smjestiti na više od $n!$ načina. (Već kod $n = 15$ imamo $15! = 1.307.674.368.000$ mogućih načina smještaja.)

13) optimalne vrijednosti funkcije cilja P_Ω - nalaze se na "repovima" funkcije gustoće vjerojatnosti, pa je vjerojatnost da ih dobijemo minimalna. Znamo da je vjerojatnost da dobijemo vrijednost koja leži lijevo (desno) od $[m-3\sigma, m+3\sigma]$ kod normalnog zakona distribucije $< 0,001$.

Naprijed navedena svojstva oblasti definicije i funkcije cilja dovoljno jasno govore da su gradijentne metode kao i metode Monte-Karlo praktički neupotrebljive. Pokazuje se da određeni postupak izbora početnih vrijednosti, a zatim traženje ekstrema u okolinama koje se smanjuju, omogućava rješenje nekih problema neregularnog smještaja objekata na potpuno zadovoljavajući način, bar što se tiče praktične točke gledišta.

4. Sukcesivni pojedinačni smještaj i njegova realizacija

Neka je zadana funkcija $F(x_1, x_2, \dots, x_n)$ čiji se minimum traži. Poznata je metoda Guass-Seidla (G-S) tzv. postupne ciklične ili pokooridinatne minimizacije [48]. Naime, nalazi se x_1^0 takav da vrijedi

$$F(x_1^0, x_2, \dots, x_n) = \min_{x_1 \in G_1} F(x_1, x_2, \dots, x_n) \quad (6.4.1)$$

G - oblast zadana nekim sistemom nejednadžbi

$$f_i(x_1, x_2, \dots, x_n) \geq 0; \quad (i=1, m) \quad (6.4.2)$$

Zato fiksiramo x_1^0, x_3, \dots, x_n i tražimo takav x_2^0 da vrijedi

$$F(x_1^0, x_2^0, x_3, \dots, x_n) = \min_{x_2 \in G_2} F(x_1^0, x_2, \dots, x_n) \text{ itd.} \quad (6.4.3)$$

Prvi ciklus je gotov kad nadjemo sve $x_1^0, x_2^0, \dots, x_n^0$. U novom ciklusu počnemo sa x_1^1 da vrijedi

$$F(x_1^1, x_2^0, \dots, x_n^0) = \min_{x_1 \in G_1} F(x_1, x_2^0, \dots, x_n^0) \text{ itd.} \quad (6.4.4)$$

Poopćiti metodu G-S možemo tako da promatramo

$$F(X_1, X_2, \dots, X_n) \quad (6.4.5)$$

gdje su $X_i = (x_1, x_2, \dots, x_n)$ - vektori dimenzije s_i .

Neka je (6.4.5) definirana na G, gdje je

$$G = G_1 \times G_2 \times \dots \times G_n. \quad (6.4.6)$$

Treba naći

$$\text{Min } F(X_1, X_2, \dots, X_n) \quad X_i \in G_i. \quad (6.4.7)$$

Ono što smo radili s varijablama sada radimo s vektorima, tj. u prvoj etapi se nadju vektori

$$x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$$

zatim u drugoj

$$x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)} \quad \text{itd.}$$

Općenito, nalaženje vektora $x_{r+1}^{(s)}$ svodi se na rješenje problema minimizacije funkcije

$$F(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{i-1}, \bar{x}_i, \bar{x}_{i+1}, \dots, \bar{x}_n) \rightarrow \min_{x_i \in G_i} \quad \text{za } i = r+1$$

$$\text{te } x_k = \begin{cases} x_k^s & \text{za } k \leq r \\ x_k^{(s-1)} & \text{za } k \geq r+2. \end{cases} \quad (6.4.8)$$

Ako se x_i uzimaju iz G_i , onda vrijednost funkcije $F_s(x_1^{(s)}, x_2^{(s)}, \dots, x_n^{(s)})$ ne raste s rastom s i teži minimumu funkcije $F(x_1, x_2, \dots, x_n)$

Prednost metode je jednostavnost realizacije. Ako se radi o separabilnim funkcijama dobiju se dobri rezultati. Nedostaci su kod neseparabilnih funkcija očiti. Tako npr. kada točka u toku ciklusa padne u dno oblasti zadane funkcijom

$$F(x, y) = \frac{1}{2} |x_1 + x_2| + |x_1 - x_2|,$$

ne može se iz dna izvući.

Ako uzmemo da smještamo orijentirane objekte, onda parametre (x_i, y_i) možemo shvatiti kao komponente vektora $x_i \in \mathbb{R}^2$. Ako se smješta $\{S_i\}_n$ objekata, onda generiramo slučajni niz brojeva $A_t = \{K_1^t, K_2^t, \dots, K_n^t\}$ i prema tom redoslijedu smještamo $\{S_i\}_n$ u Ω . Neka je $A = \{1, 2, 3, \dots, n\}$.

Funkcija cilja je $P_\Omega(x_1, y_1; x_2, y_2; \dots; x_n, y_n)$ gdje (x_i, y_i) koordinate pola objekta S_i .

U Ω prvo smjestimo objekt S_1 što se svodi na to da nađemo takve koordinate $(x_{k_1}^t, y_{k_1}^t) = z_{k_1}^t$ da bude

$$P_\Omega(z_{k_1}^t) = \min_{z_{k_1}^t \in G} P_\Omega(z_{k_1}^t)$$

G - je oblast u kojoj se nalazi pol objekta S_1 uz uvjet da S_1 - potpuno leži u Ω .

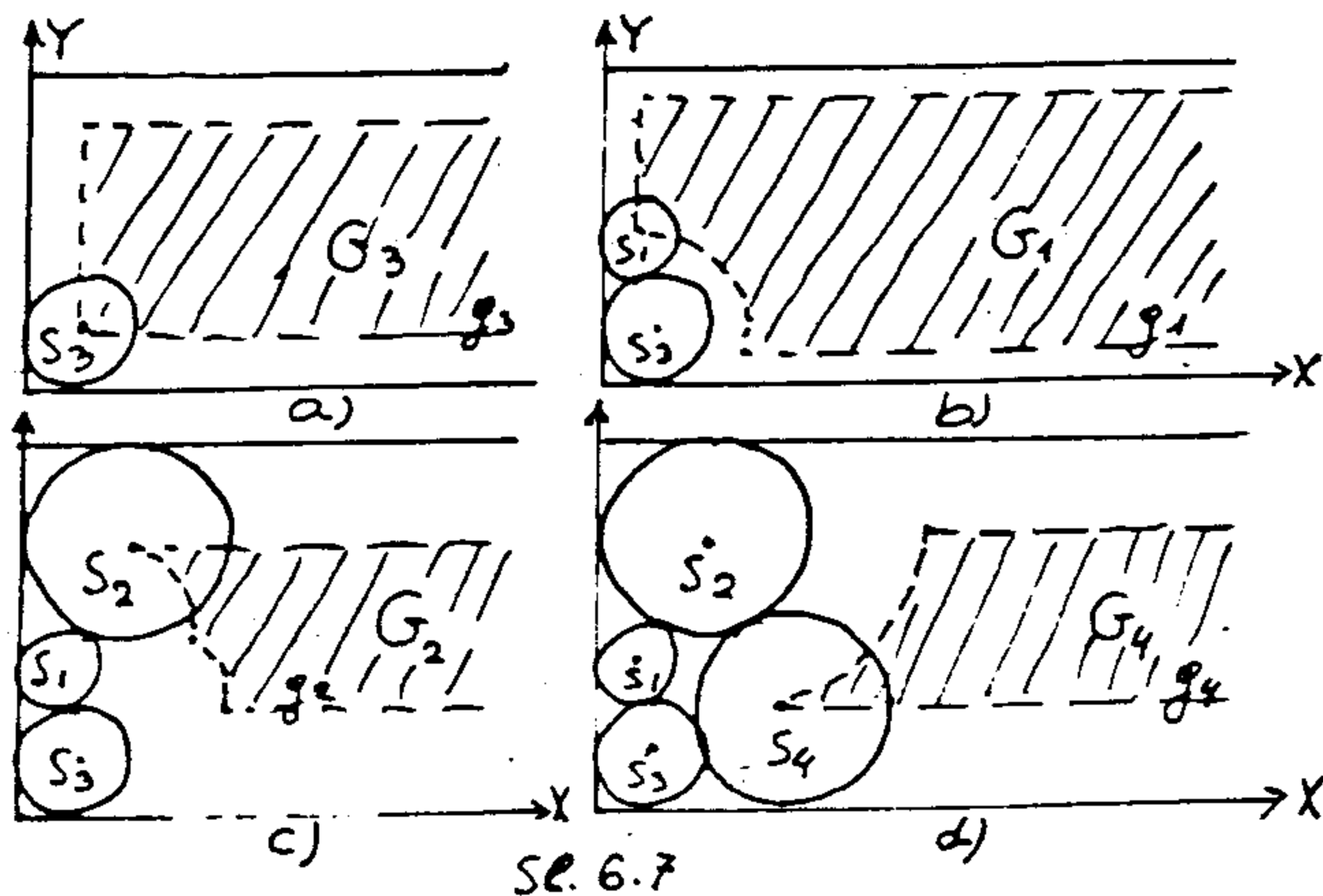
Recimo da su već smješteni objekti $S_{k_1}^t, S_{k_2}^t, \dots, S_{k_{r-1}}^t$ a treba smjestiti $S_{k_r}^t$, tada imamo

$$P_{\Omega}^r(z_{k_1}^*, z_{k_2}^*, \dots, z_{k_r}^*) = \min_{z_{k_r}^* \in G} P_{\Omega}^{r-1}(z_{k_1}^*, z_{k_2}^*, \dots, z_{k_{r-1}}^*, z_{k_r}^*). \quad (6.4.9)$$

Prednji postupak sukcesivnog pojedinačnog smještanja (SPS) može se realizirati pomoću HKF. Velike prednosti kod korištenja HKF demonstrirajmo na slijedećem jednostavnom primjeru.

Primjer:

Zadan je polubeskonačni pojas Ω širine H u koji treba smjestiti 4 kruga radijusa R_i ($i = 1, 2, 3, 4$), sl. 6.7.



Funkcija cilja je oblika

$$P = \max_{i \in [1-4]} (x_i + R_i) \quad (6.4.10)$$

Uvjeti da su objekti smješteni u Ω dani su sa:

$$\begin{aligned} x_i - R_i &\geq 0 \\ y_i - R_i &\geq 0 \\ H - R_i - y_i &\geq 0 \quad i = 1, 2, 3, 4 \end{aligned} \quad (6.4.11)$$

Neka je generiran niz $A_s = \{3, 1, 2, 4\}$ u kom redoslijedu treba smještati krugove.

Uvjeti da se krugovi ne presjecaju glase:

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (R_i + R_j)^2; \quad i, j = 1, 2, 3, 4; \quad i \neq j. \quad (6.4.12)$$

Funkcija cilja je oblika:

$$P_{\Omega}^1(x_3^*) = \min_{x_3 \in G_3} [\max (x_3 + R_3)] = \min (x_3 + R_3) \quad (6.4.13)$$

HKE - g_3 - vidimo na sl. 6.7a Ordinatu y_3 - možemo birati proizvoljno, tj. $R_3 \leq y_3 \leq H - R_3$.

Ako se uzme y_3 - minimalan imamo $y_3 = R_3$.

Za smještaj S_3 nismo trebali formirati HKE - g_3 , mogli smo riješiti sistem nejednadžbi.

$$\begin{aligned} x_3 &\geq R_3 \\ R_3 &\leq y_3 \leq H - R_3 \end{aligned} \quad (6.4.14)$$

Treba uočiti da kod ekstremalne vrijednosti (6.4.13) prva nejednadžba u (6.4.14) prelazi u jednadžbu, tj. minimum se postiže na granici oblasti dopustivih rješenja. Da nađemo gdje treba smjestiti pol drugog po redoslijedu objekta S_1 , treba naći minimalnu vrijednost koordinate x_1 za koju vrijedi

$$\begin{aligned} x_1 &\geq R_1; \quad R_1 \leq y_1 \leq H - R_1 \\ (x_1 - R_3)^2 + (y_1 - R_3)^2 &\geq (R_1 + R_3)^2 \end{aligned}$$

Funkcija cilja je:

$$P_{\Omega}^2(x_3^*, x_1^*) = \min_{x_1 \in G_1} \max(x_3^* + R_3, x_1 + R_1).$$

Fiksirano S_3 i S_1 i smještamo S_2 , i konačno S_4 .

Funkcija cilja je oblika:

$$P_{\Omega}^4(x_4^*, x_1^*, x_2^*, x_3^*) = \min_{x_4 \in G_1} \left[\max_{i \in [1, 2, 3]} (x_i^* + R_i, x_4 + R_4) \right]$$

Vidimo da su granice dopustivih rješenja G_i u stvari HKF i da se ekstremi postižu na rubu oblasti G_i .

Treba obratiti pažnju na neke činjenice:

1) Realizacija metode SPS pomoću HKF omogućava nalaženje točaka koje leže na granici oblasti dopustivih rješenja. Naime, dio nejednadžbi koji opisuje oblast dopustivih rješenja prelazi u jednadžbe, jer se moguća mjesta smještaja polova nalaze na HKF.

2) Kod drugih metoda moramo istovremeno provjeravati sve uvjete koji definiraju G_i , a njih je $\frac{n^2}{2} + \frac{n}{2}$ dok kod SPS istovremeno provjeravamo u prosjeku $\frac{n}{2}$ uvjeta, tj. samo one nejednadžbe u koje ulaze parametri objekta koga smještamo.

3) Nejednadžbe koje opisuju G ciklički zavise od malog broja promjenjivih, pa se zbog ove specifičnosti kod SPS provjera vrši sukcesivno, što smanjuje vrijeme nalaženja rješenja.

4) Rezultati se lako prate.

5) Svaki raspored nije potrebno izvesti do kraja, jer se može uočiti statistički već prije kraja da je neki raspored lošiji od nekog ranijeg.

6) Kako su za praksu najinteresantniji smještaji poligonih $\{S_i\}_n$ objekata u Ω koja je također poligon, onda su i HKF poligoni, pa se ekstremi nalaze u jednom od vrhova. Kod smještaja u beskonačnu traku širine H , to je onaj vrh HKF kome pripada minimalan x_i .

7) Metoda SPS inducira diskretnost. Naime, metoda SPS problem nalaženja minimuma funkcije od više varijabla prevodi u problem nalaženja minimuma funkcionala na skupu permutacija.

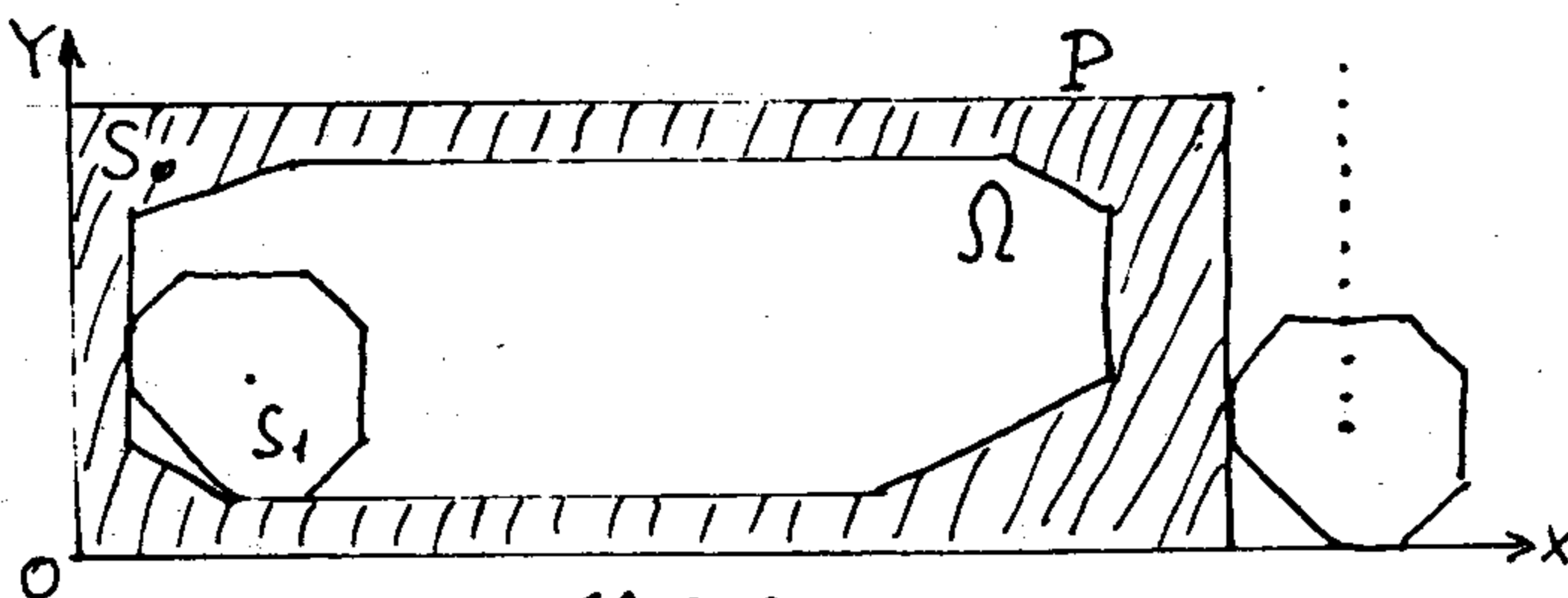
Nedostaci SPS su nedostaci G-S metode. Naime, općenito se točke dobivene pomoću SPS ne podudaraju s točkama u kojima funkcija cilja postiže lokalne ekstreme, zato te točke zovemo racionalne, a ne optimalne točke.

Prema tome s obzirom na strogu nelinearnost funkcije cilja, te složenost oblasti definicije G , za ovakve probleme meni do danas nisu poznate bolje ni determinističke ni probabilističke metode. Zato je, pored svih nedostataka, primjena SPS opravdana za traženje lokalnih ekstrema kod ovakvih funkcija cilja.

Realizacija smještaja u Ω vrši se tako da se generira oblast

$$S_0 = P \setminus \Omega, \text{ sl. 6.8.}$$

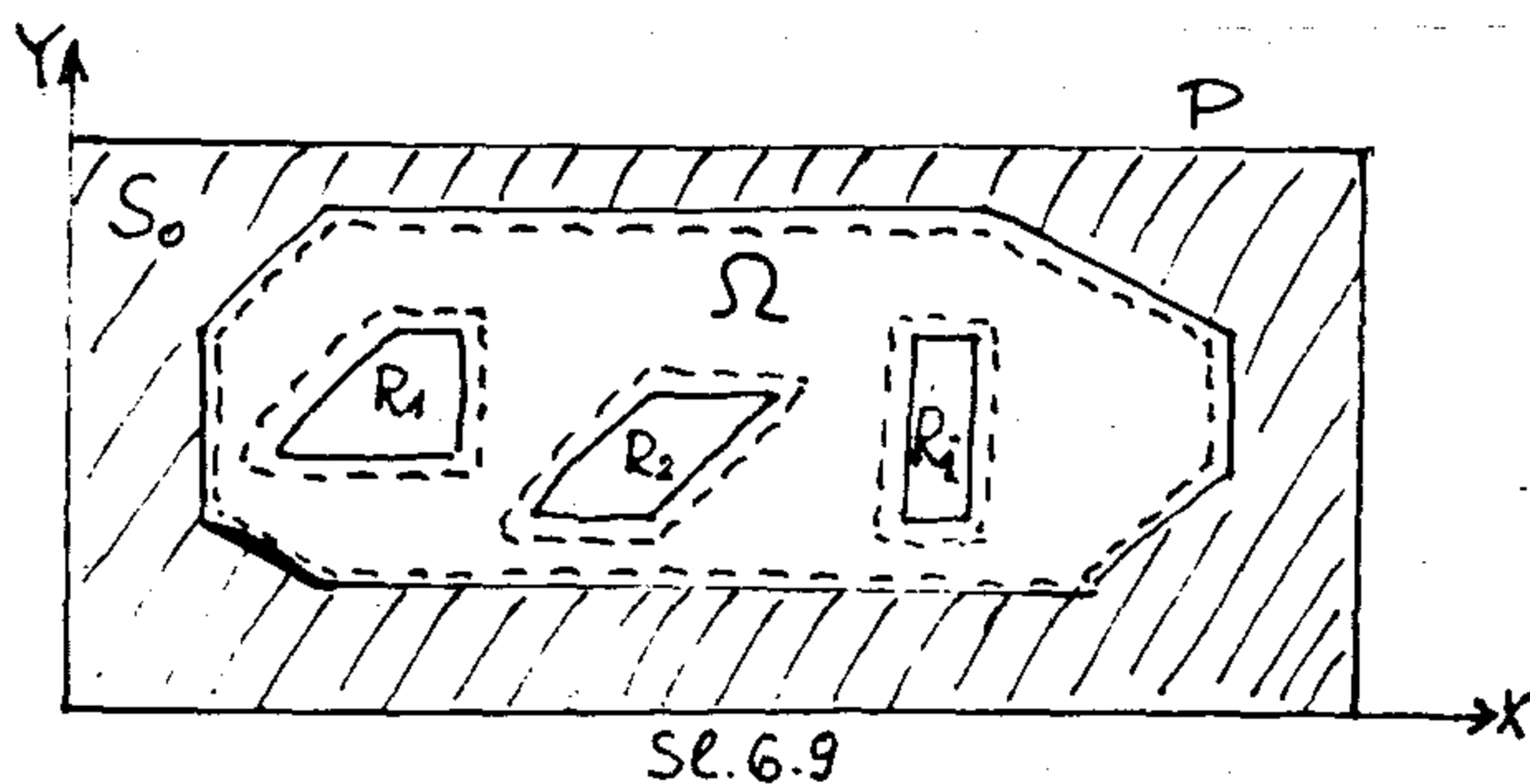
Vanjski dio HKF nije interesantan. Kada smo S_1 smjestili, onda prije nego prijedjemo na smještanje S_2 napravimo uniju $S_0 \cup S_1 = \bar{S}_1$ i nadjemo hodograf za \bar{S}_1 i S_2 , zatim opet napravimo uniju $\bar{S}_1 \cup S_2 = \bar{S}_2$ i nadjemo HKF između \bar{S}_2 i S_3 itd.



Ako su zadane oblasti $\overbrace{R_i}^{\text{ZABRANE}}$, sl. 6.9, onda napravimo $\bar{S}_0 = S_0 \bigcup_{i=1}^n R_i$

i nadjemo hodograf između \bar{S}_0 i S_1 i smjestimo S_1 itd.

Ako su zadane minimalne udaljenosti medju objektima S_i i S_j sa l_{ij} , medju objektima S_j i oblastima zabrane R_i sa l_{ij} , medju objektima S_i i granicom Ω sa l_i - onda se prije nego pristupimo smještaju nekog objekta S_k oko svih već smještenih objekata $S_1, S_2 \dots S_{k-1}$, zatim oko R_i te oko granice Ω formiraju odgovarajuće ekvidistante, na osnovu zadanih matrica udaljenosti. Izvršimo uniju \hat{S}_0 i svih $\tilde{S}_1, \tilde{S}_2 \dots \tilde{S}_{k-1}$ i \tilde{R}_i i tek onda smještamo zadani objekt S_k .



5. Separabilne funkcije i njihova svojstva

U prošlom paragrafu smo razvili metodu SPS, realiziranu pomoću HKF za nalaženje lokalnih ekstrema (racionalnih točaka). U ovom paragrafu dajemo postupak koji osigurava vjerojatnost dobivanja boljih vrijednosti funkcije cilja nego u slučaju kada se objekti smještaju metodama Monte-Karlo.

Neka se traži globalni ekstrem funkcije $F(x)$ zadane na kompaktu $D \subset \mathbb{R}^n$ [48] [49]. Neka SPS omogućava lagano nalaženje lokalnih minimuma, ali pošto je istih puno, teško ih je sve ispitati. Neka je dimenzija \mathbb{R}^n dovoljno velika. Neka funkcija $F(x)$ zadovoljava ograničenja:

$$I) \quad F(x) = \sum_{i=1}^n F_i(x) \quad (6.5.1)$$

gdje svaka $F_i(x)$ zavisi ne više od $(r+1)$ varijable tj.

$$F_i(x) = \begin{cases} g_i(x_i, x_{i+1}, \dots, x_{i+r}) & \text{za } i \leq n-r \\ g_i(x_i, x_{i+1}, x_n, x_p, \dots, x_{i+r-n}) & \text{za } i > n-r \end{cases} \quad (6.5.2)$$

$$II) \quad F_i(x) \in C \begin{matrix} (1) \\ (D) \end{matrix} \quad i = 1, n$$

Definicija:

Funkcija sa prednjim svojstvima zove se r -separabilna funkcija sa stupnjem separabilnosti $r \in \mathbb{N}$. Budući da su $F_i(x)$ neprekidne a definirane na kompaktu, one su jednoliko neprekidne i ograničene, pa vrijedi

$$|F_i(x)| < \frac{A}{n} \quad (6.5.3)$$

A - konstanta jedinstvena za sve i .

Recimo da je SPS zadovoljavajuća metoda za nalaženje lokalnog minimuma u čijoj zoni "privlačnosti" se nalazi proizvoljna točka kompakta D .

Neka je broj lokalnih minimuma funkcije $F(x)$ tako velik da njihovo dobivanje, kod slučajnog izbora početne točke, možemo smatrati nezavisnim pokusom, a za minimume pretpostavljamo da su distribuirani po normalnom ili log. normalnom zakonu distribucije.

Neka svojstva r -separabilnih funkcija:

Pogledajmo koje se komponente u (6.5.1) promjene kada predjemo od jednog na drugi lokalni minimum. Neka je u R^n zadana ortonormirana baza e_i . Onda je

$$\text{grad } F(x) = \frac{\partial F}{\partial x_1} e_1 + \dots + \frac{\partial F}{\partial x_n} e_n. \quad (6.5.4)$$

No, za proizvoljni "i" (na osnovi 6.5.2) od varijable x_i zavisi $(r+1)$ - sumanada u (6.5.1). Na osnovi (6.5.1) i (6.5.4) imamo

$$\begin{aligned} \text{grad} F(x) = & \underbrace{\left(\frac{\partial F_1}{\partial x_1} + \frac{\partial F_{n-r+1}}{\partial x_1} + \dots + \frac{\partial F_r}{\partial x_1} \right)}_{\phi_1} e_1 + \\ & + \underbrace{\left(\frac{\partial F_1}{\partial x_2} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_{n-r+2}}{\partial x_2} + \dots + \frac{\partial F_n}{\partial x_2} \right)}_{\phi_2} e_2 + \dots \\ & \dots + \underbrace{\left(\frac{\partial F_{n-r+1}}{\partial x_{n-1}} + \dots + \frac{\partial F_{n-1}}{\partial x_{n-1}} \right)}_{\phi_{n-1}} e_{n-1} + \underbrace{\left(\frac{\partial F_{n-r}}{\partial x_n} + \dots + \frac{\partial F_r}{\partial x_n} \right)}_{\phi_n} e_n \end{aligned}$$

$$\text{tj. } \text{grad } F(x) = \sum_{i=1}^n \phi_i e_i \quad (6.5.5)$$

gdje su:

$$\phi_i = \begin{cases} \frac{\partial F_1}{\partial x_i} + \dots + \frac{\partial F_i}{\partial x_i} + \frac{\partial F_{n-r+1}}{\partial x_i} + \dots + \frac{\partial F_n}{\partial x_i} & \text{za } i \leq r \\ \frac{\partial F_{i-r}}{\partial x_i} + \dots + \frac{\partial F_i}{\partial x_i} & \text{za } i > r \end{cases} \quad (6.5.6)$$

Zavisnost komponentata vektora ϕ od komponentata vektora X dana je izrazom

$$\begin{aligned}
\phi_1(X) &= \psi_1(x_1, \dots, x_{r+1}, x_{n-r+1}, \dots, x_n) \\
&= \dots = \dots = \dots = \dots \\
\phi_r(X) &= \psi_r(x_1, \dots, x_{2r}, x_n) \\
&= \dots = \dots = \dots = \dots \\
\phi_{n-r}(X) &= \psi_{n-r}(x_{n-2r}, \dots, x_n) \\
&= \dots = \dots = \dots = \dots \\
\phi_n(X) &= \psi_n(x_1, \dots, x_r, x_{n-r}, \dots, x_n)
\end{aligned} \tag{6.5.7}$$

Iz (6.5.7) vidimo da svaka komponenta x_i ulazi u ne više od $(2r+1)$ komponentata vektora ϕ .

U (6.5.7) se pretpostavlja da vrijedi

$$2r+1 < n.$$

Naime, ako to nije ispunjeno, onda svaka funkcija zavisi od svih komponentata vektora X .

Neka $\tilde{X} \in D$ predstavlja lokalni minimum. Onda je $\text{grad}F(\tilde{X}) = 0$, pa je svaka $\phi_i(X) = 0$. Slučajnim postupkom izaberemo α koordinata vektora X , i promijenimo im vrijednosti, ali tako da novi tako dobiven $\tilde{X}^\alpha \in D$. Ako je $\text{grad}F(\tilde{X}^\alpha) = 0$, onda je \tilde{X}^α - lokalna minimala ili maksimala ili pak sedlasta točka, a komponente ϕ_i se nisu promijenile, tj. ostale su jednake nuli.

Međutim, općenito je

$$\text{grad} F(\tilde{X}^\alpha) \neq 0. \tag{6.5.9}$$

No, to onda znači da je \tilde{X}^α najviše

$$\alpha(2r+1) \tag{6.5.10}$$

koordinata vektora ϕ različito od nule. Ostale se komponente podudaraju s komponentama $\phi(\tilde{X})$.

Ocjena (6.5.10) je obično previsoka. Naime, ako dolazi do promijena dviju bliskih koordinata x_i i x_j , onda mnogo komponentata vektora ϕ zavisi od x_i i x_j pa je ukupni broj komponentata koje su se promijenile manji od $\alpha(2r+1)$. Pod blizinom se podrazumijeva odnos između $|i - j|$ i r .

Ako imamo (6.5.9), onda $\exists d_1$ - takav da $\exists \tilde{X}^{\alpha 1} \in D$ da vrijedi

$$\tilde{X}^{\alpha 1} = X^\alpha + d_1 \text{ grad } F(\tilde{X}^\alpha) \quad (6.5.11)$$

i imamo

$$F(\tilde{X}^{\alpha 1}) < F(\tilde{X}^\alpha).$$

Iz prednjih razmatranja slijedi da se bar $n - \alpha (2r+1)$ - koordinata u $\tilde{X}^{\alpha 1}$ i \tilde{X}^α podudaraju. Gradijentnim postupkom formiramo niz točaka $\tilde{X}^{\alpha j}$ po pravilu da je:

$$\tilde{X}^{\alpha j} = \tilde{X}^{\alpha, j-1} + d_j \text{ grad } F(\tilde{X}^{\alpha, j-1}) \quad (6.5.12)$$

s tim da korake izaberemo tako da vrijedi

$$F(\tilde{X}^{\alpha j}) < F(\tilde{X}^{\alpha, j-1}).$$

U [41] je dokazano konvergencija prednjeg procesa. Znači da za proizvoljni $\varepsilon > 0$ postoji takav broj koraka β , gdje $\beta = \beta(\varepsilon)$, da vrijedi

$$|F(\tilde{X}^{\alpha \beta}) - F(\bar{X})| < \varepsilon \quad (6.5.13)$$

gdje \bar{X} - lokalna minimala u čijoj "zoni privlačnosti" leži \tilde{X}^α . Prema tome ako se kod prijelaza iz \tilde{X}^α u $\tilde{X}^{\alpha \beta}$ promijeni α - koordinata, onda u \tilde{X} i $\tilde{X}^{\alpha \beta}$ različitih može biti samo δ - koordinata, gdje je

$$\delta \leq \alpha \beta (2r+1). \quad (6.5.14)$$

Iz (6.5.2) vidimo da svaka komponenta $F_i(x)$ zavisi od $(r+1)$ - koordinata, pa broj komponenta $F_i(x)$ - koje imaju različite vrijednosti u \tilde{X} i $\tilde{X}^{\alpha \beta}$ neće biti veći od

$$\gamma = \alpha \beta (2r+1)(r+1). \quad (6.5.15)$$

Na osnovi toga vrijedi slijedeći teorem.

Teorem 6.1.

Ako se u točki \tilde{X} promijeni α - koordinata, onda prikazi r - separabilne funkcije $F(x)$ pomoću (6.5.1) u \tilde{X} i \bar{X} imaju ne više od γ - različitih sumanada.

Gdje je: r - faktor separabilnosti, β - broj nužnih koraka gradijentnog spuštanja da se dođe u okolinu ε točke \bar{X} .

Ponašanje matematičkog očekivanja

Nazovimo izbor proizvoljnog $X \in D$ pokusom, a vrijednost $F(X)$ događajem. Znači da prostor događaja čine sve moguće vrijednosti $F(X)$, s tim da X prolazi kompaktnom D. Pošto je $F(X)$ sada slučajna veličina, možemo govoriti o matematičkom očekivanju $M[F(X)]$, o disperziji $D[F(X)]$ i o drugim vjerojatnosnim karakteristikama [48] [49].

Neka je zadana r-separabilna funkcija $F(x)$ čije je

$$M[F(X)] = m_1.$$

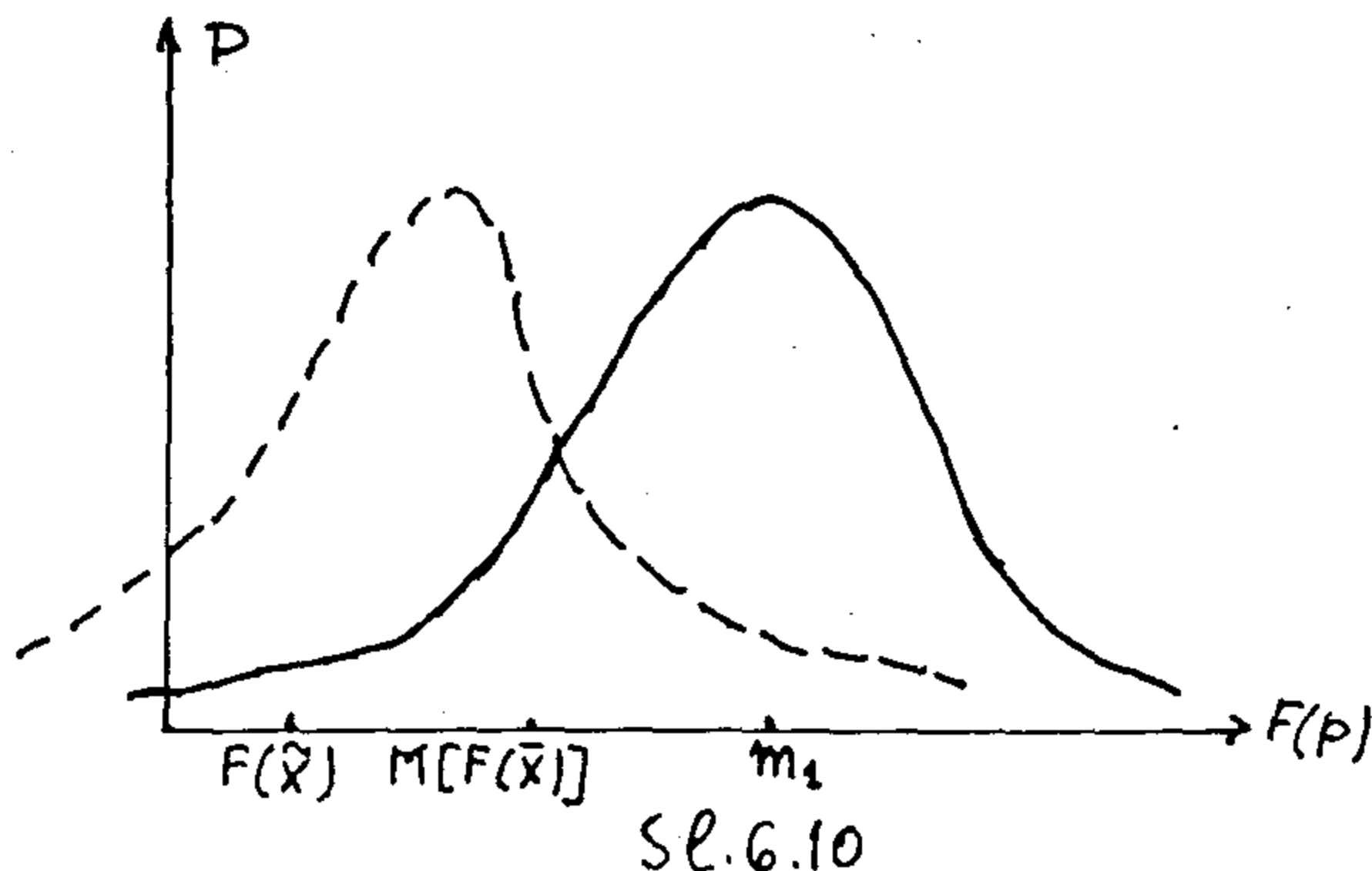
Želimo pokazati da uz neke uvjete iz nejednadžbe

$$F(\tilde{X}) < m_1$$

slijedi nejednadžba

$$M[F(\tilde{X}')] < m_1$$

gdje je \tilde{X} neka fiksirana lokalna minimala funkcije $F(x)$, a \tilde{X}' - različite lokalne minime koje se dobiju iz \tilde{X} - promjenom α - koordinata a zatim gradijentnog spuštanja. Sl.6.10.



Koordinate α koje mijenjamo u \tilde{X} biraju se slučajnim postupkom, pa otuda slijedi da su i \tilde{X}^α slučajne točke, a onda i točke $\tilde{X}^{\alpha\beta}$, tj. \bar{X} . Ako je \tilde{X} čak fiksna vrijednost $F(\tilde{X})$, možemo uzeti da je slučajna veličina i promatrati njeno matematičko očekivanje.

Uzmemo dvije proizvoljne minimale X^0 i X^1 i u njima promijenimo α koordinata, zatim se iz dobivenih točaka $X^{0\alpha}$ i $X^{1\alpha}$ pomaknemo gradijentnim postupkom u točke $X^{0\alpha\beta}$, ($X^{1\alpha\beta}$) koje su u okolini novih minimala \bar{X}^0 (\bar{X}^1) da vrijedi $|F(X^0) - F(X^{0\alpha\beta})| < \epsilon_1$.

Kako se radi o r -separabilnoj funkciji $F(x)$ koja je prikazana sa (6.5.1) pa kod prijelaza na bilo koju točku \bar{X}^0 (\bar{X}^1) svoje vrijednosti promjeni najviše γ - sumanada. (γ -dan sa (6.5.15).

Ako grupiramo sumande u (6.5.1) na one koji se promijene kod prijelaza iz X^0 u \bar{X}^0 (od X^1 u \bar{X}^1), onda vrijednosti funkcija u \bar{X}^0 (\bar{X}^1) možemo prikazati

$$F(\bar{X}^0) = \sum_{p=1}^{n-\gamma} F_{ip}(\bar{X}^0) + \sum_{p=n-\gamma-1}^n F_{ip}(\bar{X}^0) \quad (6.5.17)$$

$$F(\bar{X}^1) = \sum_{p=1}^{n-\gamma} F_{rp}(\bar{X}^1) + \sum_{p=n-\gamma-1}^n F_{rp}(\bar{X}^1). \quad (6.5.18)$$

Budući da smo u prve sumande stavili one komponente koje promijenile kod prijelaza iz X^0 u \bar{X}^0 (6.5.17), odnosno iz X^1 u \bar{X}^1 u (6.5.18), to možemo zapisati da je:

$$F(\bar{X}^0) = \sum_{p=1}^{n-\gamma} F_{ip}(X^0) + \sum_{p=n-\gamma+1}^n F_{ip}(\bar{X}^0) \quad (6.5.17a)$$

$$F(\bar{X}^1) = \sum_{p=1}^{n-\gamma} F_{rp}(X^1) + \sum_{p=n-\gamma+1}^n F_{rp}(\bar{X}^1). \quad (6.5.18a)$$

Iz (6.5.17a) i (6.5.18a) imamo: $F(\bar{X}^0) - F(\bar{X}^1) = F(X^0) - F(X^1) +$

$$+ \sum_{p=n-\gamma+1}^n [F_{ip}(\bar{X}^0) - F_{ip}(X^0) + F_{rp}(X^1) - F_{rp}(\bar{X}^1)]. \quad (6.5.19)$$

Pošto X^0 i X^1 nisu slučajne već fiksne pa ako mjesto (6.5.19) promatramo srednje vrijednosti, imamo:

$$M[F(\bar{X}^0)] - M[F(\bar{X}^1)] = F(X^0) - F(X^1) + \sum_{p=n-\gamma+1}^n \{ M[F_{ip}(\bar{X}^0)] - F_{ip}(X^0) + F_{rp}(X^1) - M[F_{ip}(\bar{X}^1)] \} \quad (6.5.20)$$

Neka su

$$F_i(\bar{X}^{0Z1i}) = \max_i |F_i(\bar{X}^0)| \quad i = (1, \dots, n)$$

$$F_r(\bar{X}^{1Z2r}) = \max_r |F_r(\bar{X}^1)|, \quad r = (1, \dots, n) \quad (6.5.21)$$

tada vrijedi:

$$|M[F_i(\bar{X}^0)]| < |F_i(\bar{X}^{0Z1i})| \quad i=1, n \quad (6.5.22)$$

$$|M[F_r(\bar{X}^1)]| < |F_r(\bar{X}^{1Z2r})| \quad r=1, n. \quad (6.5.23)$$

Pošto je modul sume \leq od sume modula, imamo slijedeću ocjenu:

$$\left| \sum_{p=n-\gamma+1}^n \{ M[F_{ip}(\bar{X}^0)] - F_{ip}(X^0) + F_{rp}(X^1) - M[F_{ip}(\bar{X}^1)] \} \right| <$$

$$< \sum_{p=n-\gamma+1}^n \left\{ |F_{ip}(\bar{X}^{0Z1i})| + |F_{ip}(X^0)| + |F_{rp}(X^1)| + |F_{rp}(\bar{X}^{1Z2r})| \right\} \quad (6.5.24)$$

Pomoću (6.5.3), svaki sumand na desnoj strani (6.5.24) možemo ocijeniti. Pošto tih sumanada ima 4 p, nalazimo ocjenu

$$\left| \sum_{p=n-\gamma+1}^n \{ M[F_{ip}(\bar{X}^0)] - F_{ip}(X^0) + F_{rp}(X^1) - M[F_{ip}(\bar{X}^1)] \} \right| < \frac{4\gamma A}{n} \quad (6.5.25)$$

Iz (6.5.20) i (6.5.25) imamo da je

$$M[F(\bar{X}^0)] - M[F(\bar{X}^1)] = F(X^0) - F(X^1) + \epsilon \quad (6.5.26)$$

Veličina ε se ocijenjuje sa:

$$|\varepsilon| < \frac{4\gamma A}{n}.$$

S obzirom na teorem 6.1. možemo napisati

$$|\varepsilon| < \frac{4\alpha\beta(2r+1)(r+1)A}{n} \quad (6.5.27)$$

Teorem 6.2.

Ako $n \rightarrow \infty$ i $\lim_{n \rightarrow \infty} \frac{r^2 \alpha}{n} = 0$,

onda $\text{Sign}\{M[F(\bar{X}^0)] - M[F(\bar{X}^1)]\} = \text{Sign}\{F(X^0) - F(X^1)\}$ (6.5.28)

Naime, s obzirom na (6.5.27) i činjenicu da $\lim_{n \rightarrow \infty} \frac{r^2 \alpha}{n} = 0$ slijedi da je:

$$\lim_{n \rightarrow \infty} |\varepsilon| = 0 \quad (6.5.29)$$

Pošto je $F(X^0) - F(X^1) = \text{const}$, tj. nezavisi od n , to iz (6.5.26) i (6.5.29), slijedi (6.5.28).

Opisno rečeno teorem kaže da je matematičko očekivanje vrijednosti r -separabilne funkcije u točkama koje se dobiju iz polazne minimale X promjenom α koordinata, a zatim gradijentim spuštanjem za β koraka, tim manje što je manja vrijednost funkcije cilja u polaznoj minimali X , uz pretpostavku da $r^2 \alpha$ raste sporije od n , kad $n \rightarrow \infty$. U [39] je pokazano da kod nekih problema optimalnog projektiranja imamo situaciju da između faktora separabilnosti " r " i broja varijabli " n " postoji takva veza da vrijedi:

$$\lim_{n \rightarrow \infty} \frac{r}{(n)^{1/3}} = 0 \quad (6.5.30)$$

Kao posljedicu teorema 6.2. i (6.5.30) imamo slijedeći teorem:

Teorem 6.3.

Ako $n \rightarrow \infty$ i vrijedi (6.5.30) a

$$\lim_{n \rightarrow \infty} \frac{\alpha}{(n)^{1/3}} = 0, \quad (6.5.31)$$

tada vrijedi (6.5.28).

Znači da ograničeni rast broja α koordinata koje se mijenjaju u polaznoj minimali, i neograničeni rast n - dimenzije prostora, osiguravaju korelaciju između početnih točaka i matematičkog očekivanja vrijednosti funkcije cilja u točkama koje su nadjene iz odgovarajućih početnih točaka.

Vidimo da se kod teorema 6.2. i 6.3. zahtijeva teženje dimenzije prostora $n \rightarrow \infty$, čega u praksi nema bez obzira što n - može biti dovoljno veliko. Međutim, pokazuje se da korelacija između početnih točaka i matematičkog očekivanja vrijednosti funkcije cilja vrijedi i za konačne n , za dovoljno malene α .

Teorem 6.4.

$$\text{Ako je } F(X^0) - F(X^1) = E > 0 \quad (6.5.32)$$

i ako broj α komponenata koje se mijenjaju zadovoljava

$$\alpha < \frac{En}{4\alpha\beta(2r+1)(r+1)A}, \quad (6.5.33)$$

onda je

$$M[F(\bar{X}^0)] > M[F(\bar{X}^1)]. \quad (6.5.34)$$

Dokaz:

IZ (6.5.26) i (6.5.22) slijedi da je

$$M[F(\bar{X}^0)] - M[F(\bar{X}^1)] > E - |\epsilon| \quad (6.5.35)$$

Iz (6.5.27) i (6.5.33) slijedi uvjet

$$|\epsilon| < E. \quad (6.5.36)$$

Znači da je (6.5.34) posljedica (6.5.35) i (6.5.36).

Dobivanje točnih ocijena za konstantu A u (6.5.3) kao i za faktor separabilnosti r , kao i broj gradijentnih koraka β , često je teško ili nemoguće. U tim slučajevima prednji teorem se ne može koristiti za izbor α , iako i tada (6.5.33) omogućava neke važne zaključke.

1. Korelacija između vrijednosti funkcije cilja $F(X^0)$ u početnoj točki i $M[F(\bar{x})]$ je to jača što je α manji.

2. Ako je $E = F(X^0) - F(X^1)$ premalen, ne mora doći do pomaka matematičkog očekivanja.

3. Ocjena (6.5.33) za α vrijedi i za znatno veće α , zato što je ocjena (6.5.15) pretjerana. Znatnu pogrešku u (6.5.33) unosi i gruba ocjena (6.5.24).

Bitno je to da smo do ocjene (6.5.15) došli primjenom konkretne metode traženja lokalnog minimuma i to gradijentne, zbog čega se u (6.5.33) pojavljuje broj koraka β . Broj β je karakteristika same gradijentne metode, a nije karakteristika problema. Lokalna minimala ne zavisi od toga kakve se metode primijenjuju da bi se je našlo, pa korelacija između vrijednosti $F(X^0)$ i $M[F(\bar{X}^0)]$ postoji ako umjesto gradijentne primijenimo bilo koju drugu metodu.

Važnot teorema 6.4. je u tome što on daje teorijsku osnovu za postojanje klase r -separabilnih funkcija za koje je matematičko očekivanje $M[F(\bar{X}^0)]$ korelirano s vrijednostima funkcije $F(X^0)$ u polaznoj točki.

Pošto je teško unaprijed ocijeniti da li konkretna funkcija pripada navedenoj klasi, pitanje pomaka matematičkog očekivanja rješava se eksperimentalno pomoću elektronskog računskog stroja. Pošto je za $\alpha = 1$ korelacija najveća, obično se uzima [48] da je $\alpha = 1$.

Naime, proces se sastoji u slijedećem. Nadjemo neki lokalni minimum funkcije $F(X^0)$. U točki X^0 slučajno promijenimo jednu koordinatu pa iz tako dobivene točke gradijentnom metodom nadjemo bližu lokalnu minimalu \bar{X}^{01} . Ako je $F(X^0) < F(\bar{X}^{01})$, vratimo se u X^0 , inače za polaznu točku uzimamo \bar{X}^{01} i ponavljamo proces.

U ovom radu su uvedene modifikacija u vezi s izborom broja α , jer pored pomaka matematičkog očekivanja imamo promjene i drugih karakteristika zakona distribucije.

Treba uočiti da u dosadašnjim dokazima ovog paragrafa nigdje nismo spominjali neki konkretni zakon distribucije, znači da pomak matematičkog očekivanja imamo kod svakog zakona distribucije minimuma r -separabilnih funkcija.

Promjena disperzije

Ovdje će biti pokazano kako se mijenja disperzija distribucije vrijednosti funkcije cilja u lokalnim minimalama u zavisnosti od broja koordinata α koje se promijene u polaznoj minimali [48]. Neka je $I = \{1, 2, 3, \dots, n\}$, a zadana je polazna minimala X^0 . Ako u X^0 promijenimo vrijednosti koordinata x_{j_k} ($j_k \in I, k = 1, \nu$) i vrijednosti x_{i_k} ($i_k \in I, k = 1, \nu$), onda dodjemo u točke $X^{(1)}$ odnosno $X^{(2)}$. Ako istovremeno promijenimo x_{j_k} i x_{i_k} , onda dodjemo u točku $X^{(1,2)}$.

Neka u okolini dobivenih točaka $X^{(1)}, X^{(2)}, X^{(1,2)}$ postoje pripadne lokalne minimale $X^1, X^2, X^{1,2}$ kojima se možemo približiti dovoljno blizu, koristeći gradijentnu metodu, a za β koraka. Neka su se kod prelaza iz X^0 u X^1 promijenile koordinate x_{a_p} ($a_p \in I; p = 1, P$) a kod prelaza iz X^0 u X^2 koordinate x_{b_s} ($b_s \in I; s = 1, S$). Žbog (6.5.14) imamo

$$P \leq \alpha \beta (2r+1) \quad (6.5.37)$$

$$S \leq \alpha \beta (2r+1).$$

Uzet ćemo da su vrijednosti funkcije $F(X^1)$, odnosno $F(X^2)$, rezultat pokusa koji se sastoji u tome da u X^0 slučajnim postupkom promijenimo vrijednosti α - koordinata. Isto tako vrijednost $F(X^{1,2})$ smatramo rezultatom promjene ne više od 2α koordinata.

Naime, kod izbora $\{x_{j_k}\}; \{x_{i_k}\}$, budući je izbor slučajan to $\{j_k\} \cap \{i_k\}$, ne mora biti prazan skup.

Disperziju slučajne veličine $F(X^1)$ i $F(X^2)$ označimo sa σ_α^2 a od $F(X^{1,2})$ sa $\sigma_{2\alpha}^2$. Kada se α smanjuje možemo imati slijedeće slučajeve

$$\sigma_\alpha^2 \leq \sigma_{2\alpha}^2$$

Kasnije će biti pokazano da povećanje disperzije kod smanjenja α ima za posljedicu da raste vjerojatnost dobivanja vrijednosti

funkcije cilja bliskih k globalnom minimumu. Zato treba disperziju pobliže ispitati.

Razbijemo sumande u (6.5.1) u 3 grupe:

- $F_i^{(1)}(x)$ - oni članovi u (6.5.1) koji ne zavise od $\{x_{a_p}\}$ i $\{x_{b_s}\}$
 $F_i^{(2)}(x)$ - ne zavise od $\{x_{b_s}\}$, ali zavise bar od jednog $\{x_{a_p}\}$;
 $F_i^{(3)}(x)$ - ne zavisi od $\{x_{a_p}\}$, ali zavisi od bar jednog $\{x_{b_s}\}$.

Prema tome za bilo koji $X \in D$ imamo:

$$F(x) = \sum_i F_i^{(1)}(x) + \sum_i F_i^{(2)}(x) + \sum_i F_i^{(3)}(x). \quad (6.5.39)$$

S obzirom na to kako smo formirali $F_i^{(m)}(x)$ ($m=1,2,3$), te kako smo došli do točaka $X^1, X^2, X^{1,2}$ imamo

$$F(X^0) = \sum_i F_i^{(1)}(X^0) + \sum_i F_i^{(2)}(X^0) + \sum_i F_i^{(3)}(X^0) \quad (6.5.40)$$

$$F(X^1) = \sum_i F_i^{(1)}(X^0) + \sum_i F_i^{(2)}(X^1) + \sum_i F_i^{(3)}(X^0) \quad (6.5.41)$$

$$F(X^2) = \sum_i F_i^{(1)}(X^0) + \sum_i F_i^{(2)}(X^0) + \sum_i F_i^{(3)}(X^2) \quad (6.5.42)$$

$$F(X^{1,2}) = \sum_i F_i^{(1)}(X^0) + \sum_i F_i^{(2)}(X^1) + \sum_i F_i^{(3)}(X^2) \quad (6.5.43)$$

Prema tome slijedi da je:

$$F(X^{1,2}) - F(X^0) = F(X^1) - F(X^0) + F(X^2) - F(X^0). \quad (6.5.44)$$

Uz prikrate:

$$\begin{aligned} F(X^{1,2}) - F(X^0) &= \Delta F^{12}; & F(X^1) - F(X^0) &= \Delta F^1 \\ F(X^2) - F(X^0) &= \Delta F^2. \end{aligned} \quad (6.5.45)$$

imamo

$$\Delta F^{12} = \Delta F^1 + \Delta F^2.$$

Na osnovi teorema o disperziji sume i (6.5.45) imamo

$$D(\Delta F^{12}) = D(\Delta F^1) + D(\Delta F^2) + 2\text{Cov}(\Delta F^1, \Delta F^2) \quad (6.5.46)$$

Zbog toga

$$\text{Cov}(\Delta F^1, \Delta F^2) = M \left[(\Delta F^1 - M[\Delta F^1]) \cdot (\Delta F^2 - M[\Delta F^2]) \right] = M[\Delta F^1 \cdot \Delta F^2] - M[\Delta F^1] \cdot M[\Delta F^2]$$

Kako se $F(X^1)$ i $F(X^2)$ formiraju na isti način, one imaju isti zakon distribucije pa je:

$$D(\Delta F^1) = D(\Delta F^2) = \sigma_\alpha^2. \quad (6.5.47)$$

Znamo da je koeficijent korelacije dan izrazom

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \cdot \sigma_y}.$$

Budući da je kod nas $\sigma_x = \sigma_y$, imamo:

$$\text{cov}(\Delta F^1, \Delta F^2) = \rho(\Delta F^1, \Delta F^2) \cdot \sigma_\alpha^2.$$

Prema tome imamo:

$$D(\Delta F^{12}) = 2(1+\rho)\sigma_\alpha^2 \quad (6.5.48)$$

gdje:

$$\rho = \rho(\Delta F^1, \Delta F^2).$$

Na osnovi ovih razmatranja vrijedi slijedeći teorem:

Teorem 6.5.

Neka su komponente $F_i(x)$ u (6.5.1) koje zavise od $\{x_{a_p}\}$ disjunktne s komponentama $F_i(x)$ koje zavise od $\{x_{b_s}\}$ i neka je $\rho > -0,5$; tada je $D(\Delta F^{12}) > D(\Delta F^1) = D(\Delta F^2)$.

Dokaz se svodi na promatranje jednadžbe (6.5.48). Slobodno rečeno, to znači da ako u (6.5.1) nema sumanada koji istovremeno zavise od $\{x_{a_p}\}$ i $\{x_{b_s}\}$, a koeficijent korelacije je veći od minus 0,5, onda se disperzija distribucije vrijednosti funkcije

cilja smanjuje kada se smanjuje broj koordinata α čija se vrijednost mijenja.

Što se tiče ograničenja $\rho > -0,5$, u praksi se može uzeti gotovo kao i da ga nema, jer je taj ^{gotovo uvijek} uvjet ispunjen. Medjutim, kako se izbor $\{j_k\}$ i $\{i_k\}$ vrši slučajno, teško je govoriti o disjunktosti koordinata $\{x_{j_k}\}$ i $\{x_{i_k}\}$ ali je ipak jasno da će ta vjerojatnost biti to veća što je odnos $n : \alpha$ veći.

Dakle, za manje vrijednosti α - imamo veću vjerojatnost da će se disperzija smanjiti. Budući da se nigdje nismo pozivali na konkretni zakon distribucije, znači da prednji zaključci ne zavise od tipa zakona distribucije minimuma r-separabilnih funkcija.

Vjerojatnost dobivanja boljih vrijednosti funkcije cilja

Vidjeli smo šta se dešava s matematičkim očekivanjem i disperzijom kada smo čisto slučajan izbor $X \in D$ zamijenili posebnim postupkom. Međutim, mi želimo doći do karakteristike koja bi nam govorila o vjerojatnosti dobivanja boljih vrijednosti funkcije cilja. Recimo da je t - najbolja dobivena vrijednost funkcije cilja.

Da li je metoda dobra, govori vjerojatnost

$$P\{F(\bar{X}) < t\} \quad (6.5.49)$$

gdje $F(\bar{X})$ vrijednost funkcije cilja koja se dobije korištenjem navedene metode, $P\{.\}$ - vjerojatnost događaja u $\{.\}$

Da bismo izračunali (6.5.49), moramo poći od neke pretpostavke o zakonu distribucije vrijednosti funkcije cilja u lokalnim ekstremima.

Recimo da imamo dvije hipoteze:

A) da imamo normalni zakon distribucije

$$G(Z) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^Z \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] dx, \quad (6.5.50)$$

B) log. normalni zakon distribucije

$$H(Z) = \begin{cases} \frac{1}{\sqrt{2\pi}s} \int_{\eta}^Z \frac{1}{x-\eta} \exp\left[-\frac{(\ln(x-\eta)-\mu)^2}{2s^2}\right] dx & \text{za } Z \geq \eta \\ 0 & \text{za } Z < \eta \end{cases} \quad (6.5.51)$$

gdje (η - donja granica veličine $F(x)$, $\mu = M[\ln(F(x)-\eta)]$
 $s^2 = D[\ln(F(x)-\eta)]$.

Vidjeli smo da kod zamjene metode čisto slučajnog traženja s metodom promjene α - koordinata u lokalnim minimalama, dolazi do smanjenja matematičkog očekivanja. Sada pogledajmo kako se to odražava na dobivanje boljih vrijednosti funkcije cilja.

Recimo da imamo dva normalna zakona distribucije sa pripadnim mat. očekivanjima $m_1 < m_2$ a $\sigma_1^2 = \sigma_2^2 = \sigma^2$.

$$G_1(z) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^z \exp\left[-\left(\frac{x-m_1}{\sqrt{2}\sigma}\right)^2\right] dx \quad (6.5.52)$$

$$G_2(z) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^z \exp\left[-\left(\frac{x-m_2}{\sqrt{2}\sigma}\right)^2\right] dx \quad (6.5.53)$$

Neka je

$$t < m_1 < m_2 \quad (6.5.54)$$

a treba izračunati $P = P\{F(z) < t\}$ (6.5.55)

Za računanje se koristimo poznatom funkcijom [19]

$$E(u) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^u \exp(-t^2) dt \quad (6.5.56)$$

ili sa

$$E(u) = \frac{1}{2}[1 + \operatorname{erf}(u)] \quad (6.5.57)$$

Za slučaj (6.5.52) imamo

$$P_1 = E\left(\frac{t-m_1}{\sqrt{2}\sigma}\right) \quad (6.5.58)$$

Za izraz (6.5.53) imamo

$$P_2 = E\left(\frac{t-m_2}{\sqrt{2}\sigma}\right) \quad (6.5.59)$$

Budući da je funkcija $E(u)$ monotono rastuća iz (6.5.54) te (6.5.58) i (6.5.59), imamo da je

$$\boxed{P_1 > P_2}$$

Tvrdnja 6.1.

Ako vrijedi (6.5.54), onda je vjerojatnost dana sa (6.5.55) veća za zakon dan sa (6.5.52) od onog danog sa (6.5.53).

Pogledajmo sada slučaj kada imamo dva log. normalna zakona $H_1(z)$ i $H_2(z)$ dana sa (6.5.51).

Znamo da je matematičko očekivanje veličine $F(z)$ distribuirane po (6.5.51) dano izrazom

$$M[F] = \eta + \exp\left(\frac{s^2}{2} + \mu\right). \quad (6.5.60)$$

Pretpostavimo da je $t > \eta$, te izračunajmo za $H_1(z)$ i $H_2(z)$ vrijednosti (6.5.55). Dobijemo:

$$P_1 = E\left[\frac{\ln(t-\eta) - \mu_1}{\sqrt{2} s}\right], \quad (6.5.61)$$

odnosno:

$$P_2 = E\left[\frac{\ln(t-\eta) - \mu_2}{\sqrt{2} s}\right]. \quad (6.5.62)$$

Iz monotonosti funkcije $E(u)$ te (6.5.61) i (6.5.62) slijedi, što je manji parametar μ , to je veća vjerojatnost boljih vrijednosti. Parametar μ monotono raste s rastom matematičkog očekivanja, što se vidi iz (6.5.60) pa zato možemo izreći ovu tvrdnju:

Tvrdnja 6.2.

Ako imamo dva log. normalna zakona, onda je vjerojatnost dobivanja boljih vrijednosti funkcije cilja veća kod onog zakona koji ima manje matematičko očekivanje.

Pogledajmo slučaj dvaju normalnih zakona distribucije sa $m_1 = m_2 = m$, a $\sigma_1^2 \neq \sigma_2^2$.

Kod ovog slučaja dobijemo pripadne izraze za (6.5.55):

$$P_1 = \left(\frac{t-m_1}{\sqrt{2}\sigma_1}\right) \quad (6.5.63)$$

$$P_2 = \left(\frac{t-m_2}{\sqrt{2}\sigma_2}\right) \quad (6.5.64)$$

Tvrđnja 6.3.

Ako je $t < m_1 = m_2 = m$, a $\sigma_1 > \sigma_2$, onda je

$$P_1 > P_2.$$

Naime, pošto je $t < m_1 = m_2$, brojnici su u (6.5.63) i (6.5.64) negativni, a budući da je $E(u)$ monotona, tvrdnja 6.3 vrijedi.

Pogledajmo do kakvih odnosa vjerojatnosti dobivanja boljih vrijednosti dolazi kod log. normalnog zakona kada se mijenja disperzija.

Disperzija vrijednosti slučajne veličine F koja je distribuirana po (6.5.51) dana je izrazom

$$D(F) = \exp(2\mu + s^2) [\exp(s^2) - 1]. \quad (6.5.65)$$

Iz (6.5.65) slijedi da je $D(F)$ - monotona rastuća funkcija parametra s . Prema tome, ako je μ - fiksno, onda zakonu distribucije s većom disperzijom odgovara veći s i obratno.

Ako su zadana dva zakona $H_1(z)$ i $H_2(z)$ sa (6.5.51), onda se pripadne vjerojatnosti (6.5.55) dobiju pomoću:

$$P_1 = E \left[\frac{\ln(t-n) - \mu}{\sqrt{2} s_1} \right] \quad (6.5.66)$$

$$P_2 = E \left[\frac{\ln(t-n) - \mu}{\sqrt{2} s_2} \right]. \quad (6.5.67)$$

Tvrđnja 6.4.

Ako je $s_1 > s_2$ a $t < n + \exp(\mu)$, onda je $P_1 > P_2$ [P_1, P_2 dobiveni pomoću (6.5.66) odnosno (6.5.67)]. Naime, u brojnicima izraza (6.5.66) i (6.5.67) imamo negativne veličine. Zbog monotonosti funkcije $E(u)$ i zbog $s_1 > s_2$ slijedi $P_1 > P_2$.

Na osnovi prednjih tvrdnji možemo izreći slijedeću tvrdnju

Tvrđnja 6.5.

Ako je zakon distribucije slučajne veličine F normalan ili log. normalan, onda smanjenje matematičkog očekivanja za $\sigma = \text{const.}$ ima za posljedicu da se povećava vjerojatnost dobivanja boljih vrijednosti funkcije cilja. Smanjenje disperzije kod konstantnog matematičkog očekivanja vodi k smanjenju vjerojatnosti dobivanja boljih vrijednosti.

Pokazano je da u koliko promijenimo vrijednosti malom broju koordinata, da dolazi do istovremene promjene m i σ .

Slučaj A:

m i σ se smanjuju a imamo normalni zakon distribucije. Vjerojatnosti poboljšanja su dane izrazima

$$P_1 = E \left(\frac{t-m_1}{\sqrt{2}\sigma_1} \right); \quad P_2 = E \left(\frac{t-m_2}{\sqrt{2}\sigma_2} \right). \quad (6.5.68)$$

Neka je $t < m_1 < m_2$ a $\sigma_1 < \sigma_2$.

Zbog monotonosti funkcije $E(u)$, uvjet $P_1 > P_2$ ekvivalentan je uvjetu

$$\frac{t-m_1}{\sigma_1} > \frac{t-m_2}{\sigma_2} \quad \text{odnosno}$$

$$\frac{m_1\sigma_2 - m_2\sigma_1}{\sigma_1\sigma_2} < t. \quad (6.5.69)$$

Slučaj B:

Za log. normalne zakone imamo pripadne vjerojatnosti

$$P_1 = E \left[\frac{\ln(t-n) - \mu_1}{\sqrt{2} s_1} \right]; \quad P_2 = E \left[\frac{\ln(t-n) - \mu_2}{\sqrt{2} s_2} \right]. \quad (6.5.70)$$

Ako imamo $\mu_1 < \mu_2$ i $s_1 < s_2$, onda je zbog monotonosti $E(u)$ uvjet

$$P_1 > P_2$$

ekvivalentan uvjetu

$$\frac{\ell_n(t-n)-\mu_1}{s_1} > \frac{\ell_n(t-n)-\mu_2}{s_2},$$

ili drugačije zapisano

$$\exp \left[\frac{\mu_1 s_2 - \mu_2 s_1}{s_2 - s_1} \right] < t - n. \quad (6.5.71)$$

Iz do sada izloženog slijedi da zamjena metode čisto slučajnog traženja boljih vrijednosti naprijed izloženom metodom promjene vrijednosti α - koordinata u lokalnim minimalama ima za posljedicu da se to dvojako reflektira na dobivanje boljih vrijednosti od onih koje su do tog momenta postignute.

Pošto je upravo navedena vjerojatnost najbolja karakteristika kvalitete traženja, nju ćemo uzeti kao kriterij za prijelaz od jednog na drugi režim traženja.

O jednoj adaptivnoj metodi (ADME)

Podvucimo kriterije koje mora zadovoljiti $F(x)$ da bi prednje nalaženje globalnog ekstrema funkcioniralo [48].

1. $F(x)$ mora biti r -separabilna.
2. $F(x)$ mora biti mnogoekstremalna.
3. Pretpostavlja se da je poznat algoritam za nalaženje lokalnih minimuma.
4. Broj lokalnih minimuma je takav da možemo govoriti o zakonu njihove distribucije, smatrajući svaki od njih realizacijom slučajne veličine.
5. Lokalni minimumi su distribuirani po normalnom ili log. normalnom zakonu.

ADME - se sastoji od nekoliko etapa.

Prva etapa:

Slučajno se biraju $X \in D$ i onda se iz njih vrši spuštanje u bližu lokalnu minimalu u kojoj se izračuna vrijednost funkcije. To se uradi odredjen broj puta i nadje takav $X^* \in D$ da je $F^*(X^*)$ do tada najmanja postignuta vrijednost.

Sve dobivene lokalne minimume uzimamo kao uzorak i izračunamo matematičko očekivanje \bar{m} i disperziju $\bar{\sigma}^2$. Na osnovi prihvaćene hipoteze o zakonu distribucije računamo vjerojatnosti dobivanja boljih vrijednosti funkcije od najbolje $F^*(X^*)$ koju imamo. Ako je to normalni zakon, onda se to izračuna pomoću

$$P \{F < F^*\} = E \left(\frac{F^*(X^*) - \bar{m}}{\sqrt{2} \bar{\sigma}} \right) \quad (6.5.72)$$

Ova vjerojatnost se uspoređi s onom dobivenom na prethodnom koraku (prije prvog prolaza stavlja se da je $P = 0$).

Ako je nova vjerojatnost veća od prethodne, ponavlja se ista etapa, u obrnutom slučaju prelazi se na novu etapu.

Druga etapa:

U X^* promijenimo slučajnim postupkom vrijednosti α koordinata s tim da je novo dobivena točka $X^{*\alpha} \in D$ i iz te točke se vrši

spuštanje u najbližu minimalu \bar{X} . Ako se pokaže da je $F(\bar{X}) < F^*(X)$ onda po završetku druge etape umjesto X^* uzimamo \bar{X} . Nakon određenog broja koraka vrši se statistička obrada. Sve do tada do vjerojatnosti dobivanja boljih rješenja rastu, radimo u istom režimu, tj. s istim brojem α i istom točkom \bar{X} .

Ako se vjerojatnost smanji, prelazimo na novi režim, tj. po odbranom pravilu smanjujemo α . U početku druge etape je $\alpha = n$.

Proces po ADME se završava kada je $\alpha = 1$ a metoda zahtijeva da se α smanji.

ADME - je jedna vjerojatnosna metoda pa ima prednosti i nedosttke takvih metoda. Naime, mi nigdje ne tvrdimo da ta metoda osigurava nalaženje globalnog ekstrema. Pod konvergencijom ove metode podrazumijeva se premještanje k nekoj fiksnoj točki iz skupa svih minimala, tj. dovoljno je učiniti konačan broj "bacanja" da dodjemo u tu fiksnu točku. Pod "bacanjem" podrazumijevamo izbor $X \in D$ a zatim spuštanje u bližu minimalu. Konačnost broja "bacanja" osigurana je time što je prijelaz od etape k etapi određen veličinom (6.5.72).

Naime, budući da imamo nova poboljšanja to se F^* smanjuje i skladno tome, vjerojatnost dana sa (6.5.72) se smanjuje zbog čega dolazi do prijelaza od etape na etapu. Pokazuje se da ADME omogućava da znatno brže nego pomoću metode Monte-Karlo nadjemo dobre lokalne ekstreme, pod pretpostavkom da se radi o funkcijama koje imaju naprijed navedenih pet svojstava.

6. Induciranje diskretne strukture kod problema neregularnog smještanja geometrijskih objekata

Vidjeli smo da se kod neregularnog smještanja geometrijskih objekata radi o minimizaciji funkcije u R^n , da je broj lokalnih ekstrema reda $> n!$ i da je njihovo traženje determinističkim metodama, koliko se meni čini, bezizgledno. Kod korištenja metode SPS za smještaj objekata, analog početne točke traženja je u stvari redoslijed objekata kako ih slažemo. Kada je redoslijed zadan, onda je jednoznačno određena vrijednost funkcije cilja. Dakle, uspostavljena je veza između permutacije od n - simbola i racionalne vrijednosti funkcije cilja (te su vrijednosti jednake približno vrijednostima u lokalnim minimumima ili pak samim njima). Proces smještanja se realizira u dvije etape.

Prva etapa:

Određivanje lokalnog minimuma - za što se koristi metoda SPS koju realiziramo pomoću HKF.

Druga etapa:

Ispitivanje ekstrema, za što se koriste metode optimizacije funkcionala zadanih na skupu permutacija.

Dakle, naš problem je u svojoj formulaciji kontinualan problem minimizacije funkcije u R^n , ali u procesu računanja on poprima diskretnu strukturu. Diskretnu strukturu je inducirala metoda SPS, a način realizacije metode pomoću HKF, inducira dvije bitne osobine:

- lokalne minimale obavezno leže na granici oblasti dopustivih rješenja, tj. na HKF,
- specifičnu vezi s nalaženjem točnih lokalnih minimuma.

Naime, vrijednosti funkcije cilja se jako razlikuju u različitim lokalnim minimumima. Prema tome u početku nije ni važno točno određivanje tih minimuma. Kada nadjemo ekstrem tj. kada je već diskretna struktura rješavanja definirana, tek tada je nužno što točnije odrediti lokalne minimume. Ovo se kod smještanja geometrijskih objekata ogleda u slijedećem.

Objekte složene strukture grubo aproksimiramo s jednostavnim poligonima i za njih pomoću SPS nađemo najbolje rješenje, tj. optimalni raspored, odnosno optimalnu permutaciju. Nakon toga idemo na finiju aproksimaciju objekata, tj. na aproksimaciju s praktički dozvoljenom točnošću. Sada te objekte, kao takve, smještamo. Dakle, naš problem nalaženja minimuma funkcije u R^n sveden je na traženje minimuma funkcionala na skupu permutacija. Samim prijelazom iz kontinualnog u diskretni problem, bez obzira što je to velik i važan korak, on ipak nije dovoljan. Naime, pretpostavimo da treba smijestiti 20 objekata. Po Stirlingovoj formuli imamo približno 10^{18} permutacija, tj. mogućih razmještaja. Ako bismo morali ispitati sve permutacije da bi našli globalni ekstrem, za to bi trebalo elektronskim strojevima koji imaju brzine oko 10^6 operacija u sekundi, i ukoliko bi se izvršavala samo po jedna operacija kod svake permutacije, trebalo bi utrošiti oko 3 tisuće godina neprekidnog rada. Međutim, mi nećemo među svim permutacijama $p \in \Pi$ tražiti onu za koju vrijedi

$$F(p_0) < F(p); \quad p \in \Pi, \quad (6.6.1)$$

već ćemo promatrati nekakav podskup Π_{ν} tražiti

$$F(p_0) < F(p) \quad p \in \Pi_{\nu}. \quad (6.6.2)$$

Da bismo mogli vršiti ispitivanja, u prostoru permutacija Π su uvedene razne metrike: lančana, inverziona, transpoziciona, leksikografska i alfabetska [30]- [36].

Recimo da funkcional F ima jedinstven globalni minimum $F(p_0)$. Neka u okolini oko p_0 radijusa R_1 ima m_1 permutacija, odnosno u okolini R_2 ima m_2 permutacija. Kod toga nije važno o kojoj se metrici radi.

Definicija:

Za funkcional definiran na skupu permutacija Π kažemo da je monoton u prosjeku ako on ima jedinstven minimum $F(p_0)$ i ako iz $R_1 < R_2$ slijedi nejednakost

$$\frac{1}{m_1} \sum_{(R_1)} F(p) < \frac{1}{m_2} \sum_{(R_2)} F(p). \quad (6.6.3)$$

Stohastički (6.6.3) znači da je matematičko očekivanje funkcionala $F(p)$ to veće što je veći radijus okoline točke permutacije p_0 .

Iz monotonosti u srednjem slijedi da za dovoljno mali radijus v sve vrijednosti $F(p_v)$ padaju u interval $a_v < F(p_v) < b_v$ gdje $p_v \in \Pi$ za koju vrijedi

$$\rho(p_0, p_v) < v.$$

Sam funkcional $F(p)$ je determiniran, međjutim pošto ćemo $p \in \Pi$ birati slučajnim postupkom, vrijednosti $F(p)$ bit će slučajne veličine, pa će se govoriti o njihovoj distribuciji te o parametrima m i σ^2 te distribucije.

Pretpostavimo da se radi o normalnom ili log. normalnom zakonu. Ovi zakoni su potpuno karakterizirani pomoću m i σ , a što je u konkretnom slučaju najvažnije eksperimentalno sam utvrdio da su pretpostavke točne za slučaj smještaja geometrijskih objekata. Pretpostavlja se da je poznat zakon distribucije za $F(p)$ kada se ne promatra $p \in \Pi$ već $p \in \Pi, c \Pi$. Kada je poznato da kod $p \in \Pi$ vrijednosti $F(p)$ leže na intervalu $a < F(p) < b$, onda se umjesto (6.5.50), (6.5.51) promatraju pripadni odsječeni normalni ili log. normalni zakoni

$$G^*(z) = \begin{cases} 0 & \text{za } y < a \\ \frac{1}{\sqrt{2\pi}\sigma} \int_a^y \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] dx & \text{za } a \leq y < b \\ 1 & \text{za } y \geq b \end{cases} \quad (6.6.4)$$

gdje je:

$$P = \frac{1}{\sqrt{2\pi}} \int_a^b \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] dx \quad (6.6.5)$$

odnosno:

$$H^*(y) = \begin{cases} 0 & \text{za } y < a \\ \frac{1}{\sqrt{2\pi} s R} \int_a^y \frac{1}{x-\eta} \exp\left[-\left(\frac{\ln(x-\eta)-\mu}{\sqrt{2} s}\right)^2\right] dx & \text{za } a \leq y < b \quad (6.6.6) \\ 1 & \text{za } y \geq b \end{cases}$$

gdje je:

$$R = \frac{1}{\sqrt{2\pi} s} \int_a^b \frac{1}{x-\eta} \exp\left\{-\left[\frac{\ln(x-\eta)-\mu}{\sqrt{2} s}\right]^2\right\} dx \quad (6.6.7)$$

Korištenje odsječenih zakona distribucije je moguće kad znamo interval (a, b) promjena $F(p)$. Ako granice nisu poznate onda se koriste obični zakoni distribucije (6.5.50) i (6.5.51). Međutim, parametri m i σ odnosno (μ, s) u praksi se određuju statistički upravo izborom $p \in \Pi$.

Smanjenje matematičkog očekivanja i ponašanje disperzije

Promatrajmo takve funkcionalne $F(p)$ čije su vrijednosti koncentrirane na segmentu $[a, b]$ za one p koji zadovoljavaju

$$\rho(p, p_0) \leq v,$$

a p_0 - permutacija na kojoj se postiže globalni minimum

Neka je $\Pi_v \subset \Pi$ koji zadovoljava nejednadžbu

$$\rho(p_v, p_0) \leq v.$$

Pretpostavimo da su vrijednosti $F(p)$ distribuirane po normalnom zakonu (6.5.50) s parametrima m i σ , kada se biraju $p \in \Pi$.

Ako se pak biraju $p \in \Pi_v \subset \Pi$, da su vrijednosti distribuirane po odsječenom zakonu s matematičkim očekivanjem m_1 i devijacijom σ_1 . Vrijede sljedeće tvrdnje:

Tvrnja 6.5.

Da bi bilo matematičko očekivanje $m_1 < m$, nužno je da vrijedi

$$\frac{a+b}{2} < m.$$

Znači, ako su vrijednosti $F(p)$ za $p \in \Pi$ distribuirane po (6.5.50), onda će matematičko očekivanje vrijednosti $F(p)$ za $p \in \Pi_v$ biti manje od matematičkog očekivanja svih vrijednosti $F(p)$ ako je

$$m > \frac{a_v + b_v}{2}. \quad (6.6.8)$$

Tvrnja 6.6.

Da bi matematičko očekivanje M - log. normalnog zakona (6.5.51) bilo manje od matematičkog očekivanja distribucije (6.6.6), dovoljno je da vrijedi

$$b < n + \exp(u). \quad (6.6.9)$$

Tvrđnja 6.7.

Vrijednosti disperzije slučajne veličine distribuirane po zakonu (6.6.4) ne prelazi disperziju slučajne veličine distribuirane po (6.5.50). Intuitivno je jasno a eksperimentalno je pokazano još i više nego se u tvrdnji 6.7. kaže. Naime, pokazuje se da ne dolazi do smanjenja disperzije kod prijelaza od osnovnog zakona na pripadni odsječeni zakon već i da u slučaju kad se promatraju segmenti

$$[a_k, b_k] \subset [a_{k-1}, b_{k-1}], \dots, [a_2, b_2] \subset [a_1, b_1] \quad \text{imamo}$$

$$\sigma_k < \sigma_{k-1} < \dots < \sigma_2 < \sigma_1 .$$

Na osnovi navedenog ovdje se metoda ADME sastoji u slijedećem: ako je $F(p)$ nadjena vrijednost funkcionala, a imamo normalni zakon distribucije, onda je vjerojatnost da se može naći bolja vrijednost dana sa

$$P_{F(p)} = E \left(\frac{F(p) - m}{\sqrt{2} \sigma} \right) .$$

Pokazali smo da se promjenom radijusa okolina iz kojih se vrši izbor $p \in \Pi$ dolazi do promjene parametara m i σ , a time i do promjene $P_{F(p)}$.

Mi imamo nekoliko režima traženja:

I režim: Slučajnim postupkom biramo $p \in \Pi$ i računamo $F(p)$. To uradimo određeni broj puta i nadjemo onu $p^* \in \Pi$ na kojoj je funkcional $P^*(p^*)$ minimalan. Dobivene vrijednosti $F_i(p_i)$ tretiramo kao slučajne veličine za koje izračunamo \bar{m} i $\bar{\sigma}^2$. Sada na osnovi hipoteze o zakonu distribucije računamo vjerojatnosti poboljšanja:

za normalni zakon

$$P_{F^*(p^*)} = E \left(\frac{F^*(p^*) - m}{\sqrt{2} \sigma} \right) \quad (6.6.11)$$

a za log. normalni

$$P_{F^*(p^*)} = E \left(\frac{\ln F^*(p^*) - \eta}{\sqrt{2} s} \right) . \quad (6.6.12)$$

U zavisnosti od vrijednosti $P_{F^*(p^*)}$ i ranijeg P odlučujemo o prijelazu na novi režim.

Drugi, treći i svi ostali režimi: se međusobno razlikuju samo u tome što su različiti radijusi okolina iz kojih se biraju $p \in \Pi$. Dakle, režim se sastoji u tome da iz kruga radijusa v u čijem je centru najbolja permutacija p^* slučajno biramo $p \in \Pi$. Izračuna se m i σ^2 i izračunamo vjerojatnost poboljšanja rezultata. Ako ta vjerojatnost raste, ostajemo u istom režimu, ako pada, prelazimo u novi režim. Ako je dobivena \bar{p} za koju je $F(\bar{p}) < F^*(p)$, onda uzimamo \bar{p} za centar kruga i proces nastavljamo tako da duljina serije bacanja $N \text{ const}$, a radijus v se smanjuje na način koji će biti objašnjen kasnije.

Proces se prekida kad je $v = 1$ a potrebno je smanjiti radijus. Znači da nam metoda omogućava da upravljamo sa m i σ , a to znači da se kod prijelaza iz jednog režima u drugi smanjuje vjerojatnost dobivanja loših rezultata. Ovo ne mora značiti da se vjerojatnost dobivanja boljeg rezultata povećava. Naime, nije mi jasno kako dokazati konvergenciju k globalnom minimumu, ali eksperimenti pokazuju da su rezultati zadovoljavajući.

Izbor duljine serije

Duljina serije se određuje tako da se uzme hipoteza o distribuciji vrijednosti $F(p)$. Za duljinu serije uzme se minimalni broj "bacanja" koji osigurava odsustvo signifikantnih odstupanja dobivenih rezultata od hipoteze. Kod toga se broj "bacanja" ograniči nekom gornjom granicom, i ako se do tog broja ne postigne zadovoljavajući rezultat, hipoteza se odbacuje.

Ako se radi o normalnom zakonu distribucije, onda nakon što se izvrši N bacanja (duljina serije) iz dobivenih vrijednosti $F_i(p_i)$ izračunamo

$$m = \frac{1}{N} \sum_{i=1}^n F_i(p_i); \quad \sigma^2 = \frac{1}{N-1} \sum_{i=1}^n [F_i(p_i) - m]^2. \quad (6.6.13)$$

Odredimo $\alpha = \min F_i(p_i)$; $\beta = \max F_i(p_i)$ i segment $[\alpha, \beta]$ podijelimo na r -intervala T_1, T_2, \dots, T_r . Izračunaju se vjerojatnosti da padaju $F_1(p)$ na pojedine T_j

$$P_j = \frac{1}{\sqrt{2\pi}\sigma} \int_{x_j}^{x_{j+1}} \exp\left[-\frac{(z-m)^2}{2\sigma^2}\right] dz \quad j=1, 2, \dots, r \quad (6.6.14)$$

Izračuna se δ_j - broj slučajeva pada $F_1(p)$ na interval T_j . Za mjeru odstupanja distribucije uzorka od pretpostavljene distribucije koristimo se sa χ^2 testom.

$$\chi^2 = \sum_{j=1}^r \frac{(\delta_j - NP_j)^2}{NP_j} \quad (6.6.15)$$

Na osnovi teorema Pirsona promatra se

$$P = P(\chi^2 > \chi_p^2)$$

gdje χ_p^2 označava - p - procentnu vrijednost χ^2 kod $r-1$ stepena slobode.

Uzimao sam da je $p = 1\%$. Promatranje sam vršio kod $r-3$ stepena slobode. Ako se ustanovi da u dotičnom uzorku imamo $\chi^2 > \chi_p^2$, smatra se da uzorak ima signifikantno odstupanje od hipoteze pa se hipoteza odbacuje. Pokazalo se da dobre rezultate dobivamo

kod $Np_j \geq 10$, zato kad je za neki interval $Np_j < 10$, vršimo spajanje tog intervala sa slijedećim, osim u slučaju zadnjeg intervala koji se spaja s prethodnim. Na osnovu prednjeg napravljeni su programi pomoću kojih se određuje duljina serije [35] [36]. Metoda ADME [30] - [36] pretpostavlja postupno smanjenje radijusa okolina iz kojih se vrši biranje $p \in \Pi$, i to po jednom od slijedećih pravila

$$1. \quad v_i = N \left[\frac{v_{i-1}}{2} \right] \quad (6.6.16)$$

$$2. \quad v_i = v_{i-1} - 1 \quad (6.6.17)$$

3. Pomoću brojeva Fibonaći

$$F_i = F_{i-1} + F_{i-2} \quad (6.6.18)$$

Za najveći radijus kruga v_1 uzimamo najveći član u nizu Fibonaći koji ne prelazi vrijednost v_0 . Recimo da je to F_s . Tada se svi daljnji radijusi formiraju po pravilu

$$v_i = F_{s-i+1} \quad (6.6.19)$$

Kod transpozicione, lančane i alfabetske metrike stavljamo da je $v_0 = n-1$, za inverzionu $v_0 = \frac{n(n-1)}{2}$,

a kod leksikografske takodjer $v_0 = \frac{n(n-1)}{2}$, jer inače bi trebali staviti $v_0 = n!$ što se pokazalo kao trošenje vremena, a bez rezultata.

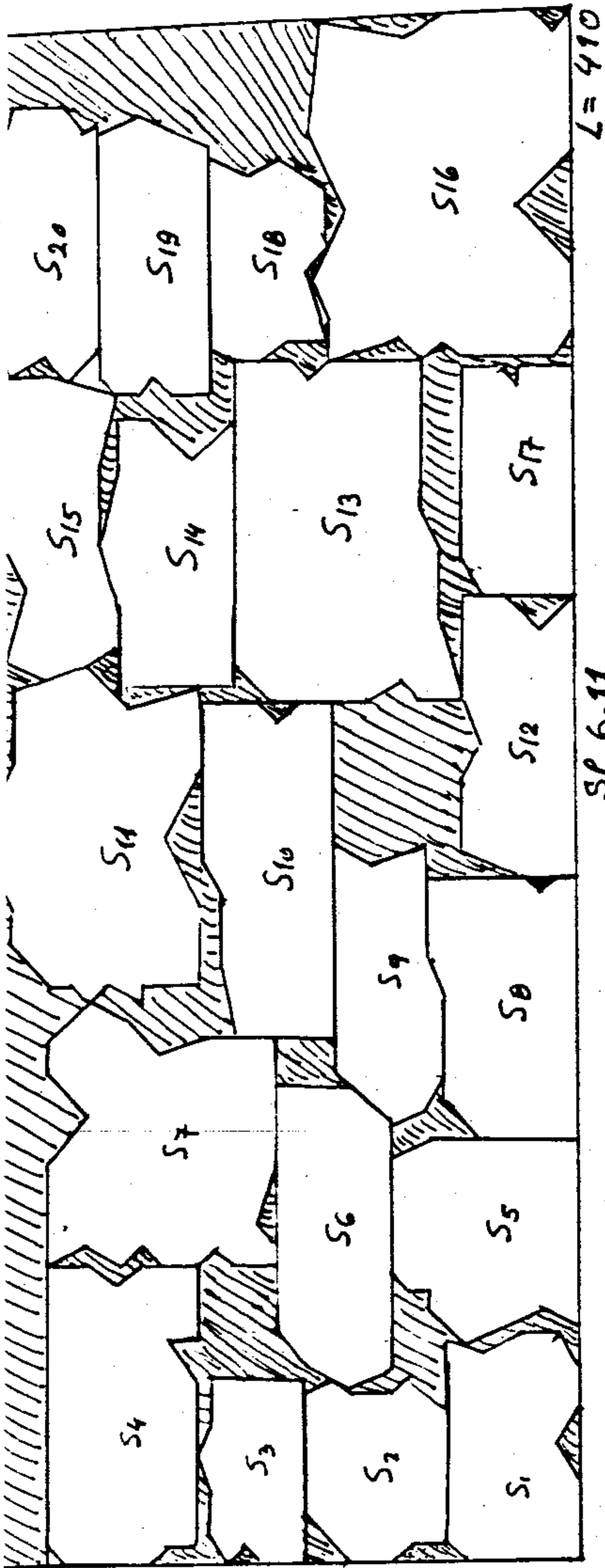
Za sva tri načina smanjenja radijusa i svih pet metrika napravljeni su programi za računar. Za nalaženje minimuma funkcionala $F(p)$ gdje $p \in \Pi$ razvijen je kompleks programa ADME. Pomoću njega se vrši neregularno smještanje geometrijskih objekata u zadanu oblast i dobivaju se vrlo dobri rezultati. Međutim, njegova primjena je daleko šira, jer se mnogi problemi izravno pojavljuju kao problemi računanja vrijednosti funkcionala na skupu permutacija.

Pomoću paketa programa "NERESM" riješio sam veći broj problema neregularnog smještaja u oblasti s pomičnim granicama. Jedan od njih je slijedeći. Zadan je skup od 20 međusobno različiti orijentiranih objekata složena oblika. Objekte treba smjestiti u polubeskonačnu traku širine $H = 155$ tako da se utroši minimalna duljina trake. Početni smještaj vidimo na sl.6.11. kome odgovara $L = 410$. Na sl.6.12. vidimo smještaj koji je dobiven nakon 292 iteracije a kome odgovara $L = 350$. Duljina serije je bila fiksna. Korištena je alfabetska metrika, a radius okolina se smanjivao po brojevima Fibonaći. Pretpostavljeno je da su vrijednosti funkcionala distribuirane po normalnom zakonu što je i potvrđeno pomoću χ^2 -testa.

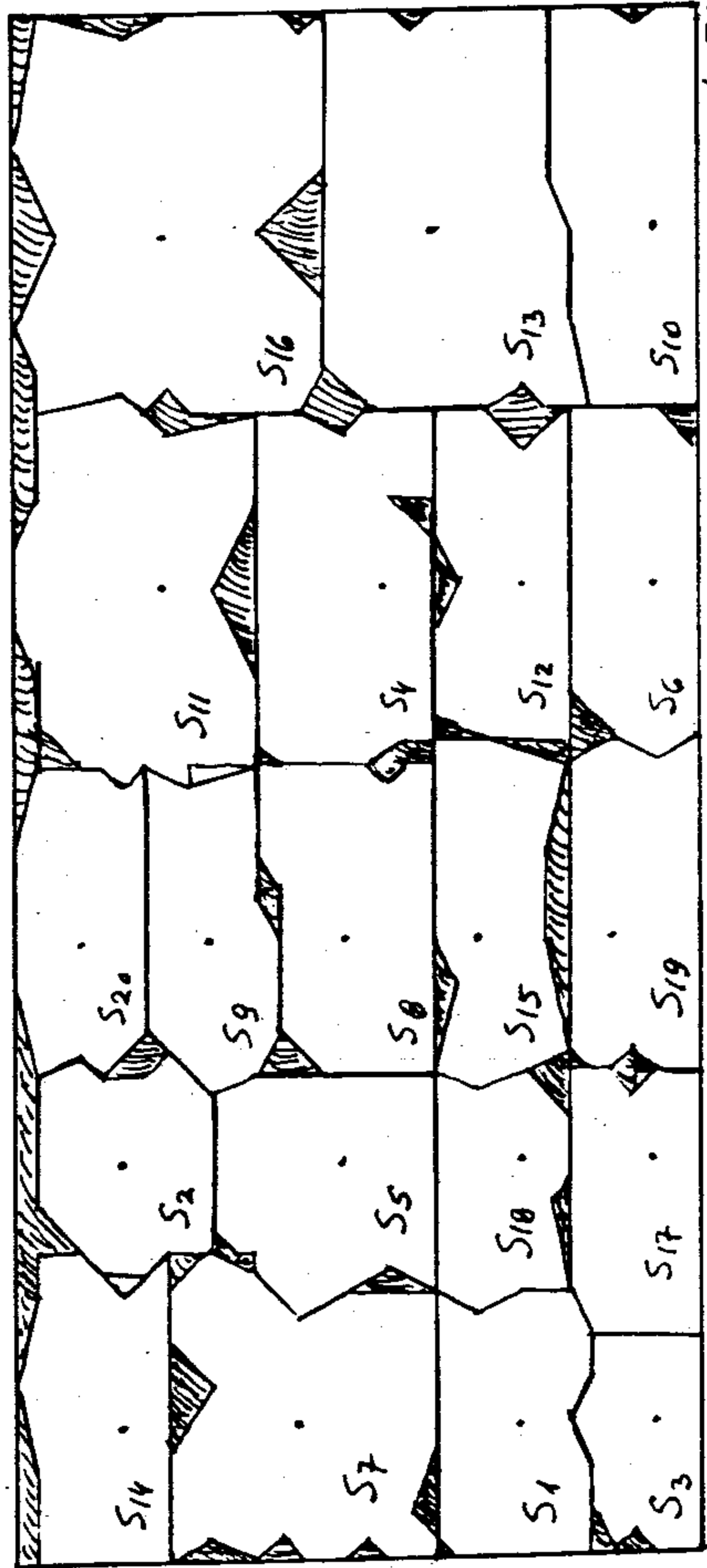
Izvršeno je 12 pokusa s istim brojem iteracija ali pomoću metode Monte-Karlo i dobiveni su rezultati od 11-19% slabiji od onih dobivenih pomoću paketa "NERESM".

Napomena:

Budući da nisam potpuno zadovoljan sa svim programima koji ulaze u paket "NERESM", u pogledu duljine vremena izvodjenja, to iste ne smatram definitivnim i zato ih dajem izdvojene kao prilog glavi VI.



SL. 6.11



SL. 6.12

L=350

L I T E R A T U R A

- [1] Adamovicz, M., The optimum two dimensional allocation of irregular, multiple-connected shapes with linear, logical and geometric constraints, Ph. Thesis, Dept. of Elect, Eng., New York University, 1969.
- [2] Adamovicz, M., Albano Nesting two- dimensional shapes in rectangular moduls Computer Aided Design Vol.8 n^o1 January 1976.
- [3] Adamovicz, M., A solution of the rectangular cutting stock problem, IEEE Transactions on System Science and Cybernetics. 1977.
- [4] Balanter, B.J., Optimalnij raskvoj sortovogo prokuta. Vestnik mašinstroenija 1967 N^o30.
- [5] Beljakova, L.B., Voprosi optimalnog raspoloženija kongruentnih figur na ploskosti. Autoreferat Kand. diss. Gorki 1970.
- [6] Bondarenko, N.A., Algoritmi racionalnogo razmešćenija ploskih geometričeskikh objektov. Autoreferat Kand diss. Harkov 1972.
- [7] Cabot, A.V., An enumeration algorithm for knapsack problems. - Operates Res. 1970. 18,2
- [7*] Dyson, R.G. and Gregory A.S., The cutting stock problem in the flat glass industry. Operations Research Quarterly. Vol.25, N^o1 pp 41-54. 1974.
- [8] Čejbišev, P.L., O Krojke odeždi UMN. 1946, 1.2
- [9] Čerepahin, V.M., Nekatorie zadaći optimalnogo razmešćenija geometričeskikh objektov i algoritmih rešenja, Autoreferat Kand. disertaciji 1970.
- [9*] Fary, I., Sur la densite des reseaux de domaines convexes, Bull. Soc. Math. France 78, 152-161 (1950)
- [10] Fedorov, E.C., Simetrija i struktura kristalov, Izd-vo AN SSSR M, 1949
- [11] Fedorov, E.C., Primenenie funkcii plotnogo razmešćenija v zadačah optimalnogo razmešćenija geometričeskikh objektov. ING. Teh. kibernetiki Akademii nauk BSSR 1972.
- [12] Gil, N.I., Stojan, Ju.G., Funkcija šiljnogo razmišenija ta ii osobivosti DAN USSR 1973, 4 350-353
- [13] Gil, N.I., Stojan, Ju.G., Formalizacija zadać razmešćenija

- geometričeskikh objektov i algoritmiih rešenija. Autoreferat Kand.diss.Kiev 1973.
- [14] Gilmore,P.C., The cutting stock problem. Cand.Math.Bull, 1966
- [15] Gilmore,P.C., Gomory,R.E., Multistage cutting stock problem of two and more dimensiones. Operat.Res.1965.13.1
- [16] Guter,R.S., Ovčinski B.V., Elementi čislenogo analiza i mat.obrabotka rezultatov opita Izd.Nauka 1970.
- [17] Gluško,A.G., Nekatrorie zadaći optimalnogo raskvoja v slučaje vikrojek složnoj formi i ih avtomatičeskoe rešenje Autoreferat Kand.diss.Harkov 1967. god.
- [18] Gluško,A.G., Rvačev,V.L.,Stojan,Ju.G., O najbolje plotnom zapolenii polosí figurami složnoj geometričesnoj formi I Kibernetika N^o 4 1968
- [19] Gluško,A.G., Rvačev,V.L.,Stojan,Ju.G., O najbolje plotnom zaposlenii polosí figurami složnoj geometričesnoj formi II,Kibernetika N^o3 1969.
- [20] Han,S.G., On the optimal cutting of defective sheets. Operat.Res.1968,16,6
- [21] Heesch,H., Regulares Parkettierungs problem. Wiss.Abhandl. Arbeitsgemeinsch.Forseh.Lande Nordheim-Westfalna Koln-Oplaafn, 1968, N^o172.
- [22] Heesch,H., Parkettierungsprobleme. Matehmatikunterrrichb, 1968,14.4
- [23] Kantorovič,L.V., Matematičesnie metodi v organizaciji i planirovanii proizvodstva. Izd.-Vo LGU, L,1939.
- [24] Kantorovič,L.V., Ob odnon effektivnom metode rešenija nekatorih klasov ekstremalnih problem DAN SSSR, 1940,28
- [25] Kantorovič,L.V., Zalgaller,V.A., Racionalnij raskroj promišljenih materijalov. Nauka SO AN SSSR Novosibirsk 1971.god.
- [26] Kobilin,A.M., Algoritmi rešenija zadaći troh - rjadnoj štapovki zagotovok. Ins.Teh.Kibernetiki AN BSSR 1972.
- [27] Kobilin,A.M., Stojan,Ju.G., Šiškin,V.N., Rešenije odnorjadnoj štapovki s učutom kompletnosti. Ekonomika i mat.metodi 1973 1.6.
- [28] Kolmogorov,A.N., Novikov,S.P., Problemi kombinatornog analiza Izd.MIR Moskva 1980.
- [29] Krasilnikov,Ja.I., Racionalnii raskroj metala Mašgi M.1961.

- [29*] Lončar, J., Zakrajšek, E., Program "Optima" Jugoslavenska autorska agencija 1974, Zagreb
- [30] Lončar, J., O jednoj metodi proračuna optimalnog balansa Informatica N^o3 1981.
- [31] Lončar, J., Primjena inverzione metrike kod optimalnog balansiranja Informatica N^o3 1981.
- [32] Lončar, J., Primjena leksikografske metrike za proračun balansa lopatica zrakoplovnih turbina, Informatica, N^o4 1981.
- [33] Lončar, J., Primjena transpozicione metrike za proračun optimalnog balansa, Informatica N^o4 1981.
- [34] Lončar, J., Računanje balansa lopatica zrakoplovnih turbina koristeći alfabetsku metriku. Neke druge metrike u prostoru permutacija. Informatica N^o 1982.
- [35] Lončar, J., Proračun optimalnog balansa kod zrakoplovnih i drugih turbina Sixteenth International Symposium on Computer Technology and Problems of Informatics, Ljubljana 1982.
- [36] Lončar, J., Izračunavanje debalansa lopatica zrakoplova i drugih turbina u različitim metrikama. Sixteenth International Symposium on Computer Technology and Problems of Informatics, Ljubljana 1982.
- [37] Mašbic, G. Ja., Automatičeskoje sastavljenije kart raskroja dlja sudokorpusnih detalij. Upravljajućie mašini i sistemi, Izd. IN-Ta Kibernetiki AN USSR, K, 1966.
- [38] Mazur, V. V., Interpolirovanie kontura ploskoj figuri lineinimi funkciami s pomoću metoda najmenših kvadrata, Minsk 1972.
- [39] Minieka, E., Optimisation Algorithms for Networks and Graphs Narcek Dekner Inc New York And Basel 1978.
- [39*] Mockys, I. B., Mnogoekstremalnie zadaći v proektivovanii. Nauka M 1967.
- [40] Panasenko, A. A., K voprosu predstavlenija oblasti opredelenija funkcii celi n-rjadnogo periodičeskog razmešćenija Viša škola Kiev 1976.
- [41] Pšenični, B. N., Danilin, Ju. N., Čisknie metodi v ekstremalnih zadaćah M. Nauka 1975.
- [42] Rogers, C. A., Packing and Covering Camrage University Press 1964.

- [43] Rvačev, V.L., Geometričeskie prilazhenija algebir logiki. Tehnika K. 1967.
- [44] Rvačev, V.L., Stojan, Ju.G., K voprosu ob optimalnom raskoje materijala. Vaprosi teoritičeskoj kibernetiki. Izd. IN-TA Kibernetiki AN USSR 1966.
- [45] Shapiro, J.F., Dinamic programing algorithms for the integer programing problem Operat. Res 1968. 16.6
- [46] Shapiro, J.F., Wagner A finite remeal algorithm for the knapsack and turnpike models. Operat. Res. 1967 15.
- [47] Stojan, Ju.G., Metodi R - funkcii v zadačah optimalnogo raskroja Autoreferat. Kand. diss. Kiev 1966.
- [48] " Ob odnom sposobe poiska najlučšego rešenija dlja odnogo klasa mnogoekstremalnih zadač. SO AN SSSR Novosibirsk 1969.
- [49] " Autoreferat doktorskoj disertaciji MTIPP, M, 1970.
- [50] Stojan, Ju.G., Gil, N.I., Metodi asimptotičeskogo perebora lokalnih ekstremov Izd. IN-TA Problem mašinostroenia AN. USSR 1974.
- [51] Tot, L.F., Raspoloženiija na ploskosti na sfere i v prostvanstve FIZ. MAT. GIZ 1958.
- [52] Watson, G.J., Determining cutting schedules by computer. Ind Woodwork 1969, 21, 5.
- [53] Zalgaller, V.A., Ob odnom neophodiom priznake plotnejšego raspoloženiija figur. UMN 1953. 8, 4
- [54] Žirkov, V.F., Primenenije elektronij vičislitelnoj tehniki dlja rešenija zadači raskroja listovogo polirovanogo stekla vipuskaemogo na konvejere. Autoreferat Kand. diss. M. 1969.

A U T O B I O G R A F I J A

Rodjen sam 27.02.1927. u Majsnim Poljanama, Glina, SRH, od oca Pavla i majke Andjelije. Do 1941. godine pohađao sam školu u Glini, a tada iz poznatih razloga prekidam školovanje. Učesnik sam NOB od 1941. Nakon oslobodjenja školujem se u Vojnoj akademiji u Moskvi sve do 1948. godine. Kao aktivni oficir JNA službujem po čitavoj Jugoslaviji na raznim dužnostima. Za zasluge u ratu i u miru odlikovan sam pet puta. Po vlastitoj želji napuštam JNA i radim u Meteorološkoj službi SRH. Upisujem 1958. godine studij matematike koji završavam 1962., te iste godine nastavljam studij na III stupnju koga završam 1965. godine. Od 1966. do 1973. godine radim ponovo u JNA na Visokoj tehničkoj školi JNA u Zagrebu u svojstvu docenta i rukovodioca elektronskog računskog centra. U tom vremenu bio sam voditelj i predavač na trećem stupnju studija iz operacionih istraživanja. Nakon toga prelazim u Institut za naftu, Zagreb gdje se bavim problemom matematičkog modeliranja.

Od 1977. godine do danas radim kao nastavnik u Višoj zrakoplovnoj školi Sveučilišta u Zagrebu.

Član sam SKOJ-a od 1941. godine a član SKJ postao sam 1945. godine.

Zagreb, 10. srpnja 1982.

Univerzitet u Beogradu
 Prirodno-matematički fakultet
 MATEMATIČKI FAKULTET
 BIBLIOTEKA

Jovan Lončar

Broj _____ Datum _____

POPIS ZNANSTVENIH I STRUČNIH RADOVA

1. Ispitivanje stabilnosti pomoću spektralne teorije diferencijalnih operatora. *Automatika* br. 1/1967.
2. Neka razmatranja u vezi s korelacionim momentom i koeficijentom korelacije. Zbornik meteoroloških i hidroloških radova br. 3/1969.
3. O jednom nelinearnom problemu distribucije homogenih sredstava. *Međunarodni Simpozij FSIP 1968.*
4. Metode dinamičkog programiranja za rješavanje nekih nelinearnih problema optimalnog korištenja ograničenih resursa. *Međunarodni Simpozij FFCIP 1968.*
5. O problemu korektnosti i u matematičkom programiranju. *Međunarodni Simpozij FCIP 1970.*
6. Uvodni pojmovi iz Laplasovih transformacija. Skripta.
7. Numerička analiza. Skripta, Geodetski fakultet Sveučilišta u Zagrebu, 1972.
8. Prethodna studija protivgradnog raketnog sistema VTŠ KOV JNA Zagreb 1971.
9. Optimalno krojenje - matematičke metode. Šumarski fakultet-
ZAGREB - 1978 god.
10. Numeričko rješavanje diferencijalnih jednačini metodom konačnih razlika i primjene. Magistarski rad 1966.
11. Uvodni pojmovi iz linearnog programiranja. TŠC 1968. (lekcija)
12. Uvodni pojmovi iz nelinearnog programiranja. TŠC 1968. (lekcija)
13. Elementarni pojmovi iz teorije igara i neke mogućnosti primjene u vojsci, TŠC 1968. (lekcija)
14. O jednoj metodi proračuna optimalnog balansa. *Informatica* 3/1981.
15. Primjena inverzione metrike kod optimalnog balansiranja. *Informatica* 3/1981.
16. Primjena transpozicione metrike za proračun optimalnog balansa. *Informatica* 4/1981.
17. Primjena leksikografske metrike za proračun debalansa zrakovnih turbina. *Informatica* 4/1981.

18. Izračunavanje debalansa lopatica zrakoplovnih i drugih turbina u različitim metrikama, 16th Yugoslav International Symposium on Computer Technology and Problems of Informatics Ljubljana, 1982.
19. Proračun optimalnog balansa kod zrakoplovnih i drugih turbina, 16th Yugoslav International Symposium on Computer Technology and Problems of Informatics, Ljubljana 1982.

P U B L I K A C I J E

1. Linearno programiranje* (lekcije)
2. Teorija igara* (lekcije)
3. Nelinearno programiranje* (lekcije)
4. Dinamičko programiranje* (lekcije)
5. Programski jezik "ALGOL" (lekcije)
6. Značajne recenzije. Recenzija projekta Ujedinjenih naroda, a za potrebe jugoslavenske vlade. Regulacija rijeke Save i njenih pritoka. (Autori projekta stručnjaci Instituta iz Rima i iz Praga 1969-1972).

Napomena: Radovi označeni () napisani su i umnoženi u TŠC za studente III stupnja studija 1969.

PREDAVAO SAM KOLEGIJE:

1. Linearno programiranje
2. Nelinearno programiranje
3. Dinamičko programiranje
4. Teorija igara
5. Numeričke metode i programiranje
6. Operaciona istraživanja
7. Programiranje u FORTRANU

Osim toga bio sam voditelj III stupnja, kao i voditelj triju magistarskih radova (Diskretno programiranje, Primjena operativnih istraživanja u vazduhoplovstvu i primjena operacionih istraživanja u mornarici).

OSNOVNI PROGRAMI IZ PAKETA
" N E R E S M " UZ GLAVU VI

ITM=KORR(1).SHESTI/TRI

1 (* SLUCAJNO SHESTANJE POLIGONA. POLACI
2
3 TROCIFRENI BROJ XYZ (DEBUG, DEBIG, DEBDG)
4 FAKTOR ZA PRELAZ U INTEGEN
5 BROJ PERMUTACIJA
6 OSNOVNI POLIGON <EOLN>
7 BROJ POJAVLJANJA PRVOG POLIGONA
8 PRVI POLIGON <EOLN>
9 BROJ POJAVLJANJA DRUGOG POLIGONA
10 DRUGI POLIGON <EOLN>
11
12
13 SVAKI POLIGON JE ZADAN KAO
14 BROJ CVOROVA
15 X1, Y1, X2, Y2,, XN, YN

18 PROGRAM SHESTI(INPUT,PRINT,OUTPUT);

20 LABEL 13;

21 CONST

22 MAXP = 179;
23 FCTR = 1.0E3; (* 6 = SQRT(FCTR * MAXCOOR) * MAXINT *)
24 SMXTHAX = 18500; (* = SQR(MAXINT*1) *)
25 LENGTH = 6; (* VISESTRUKOST ARITMETIK *)
26 MAXEL = 50;

28 TYPE

29 VPTR = 2 VPR;
30 CPTR = 2 CVOR;
31 VPR =

32 RECORD

33 X, Y : INTEGER;
34 NEXT : VPTR
35 END (* RECORD *)

36 CVOR

37 PACKED RECORD
38 V : VPTR;
39 NEXT, SIN : CPTR
40 END (* RECORD *)

41 INDEX = 0 .. MAXP;
42 TABLE = ARRAY INDEX OF

43 PACKED RECORD

44 X, Y : INTEGER;
45 S, H, K, L : INDEX;
46 GR, UN, C, PG : BOOLEAN
47 END (* RECORD *)

48 ODRS = (IDENT, AIN, BINA, CROSS, DISJUNT);

49 BSS = PACKED ARRAY 01 .. 205 OF CHAR;

50 MULT = ARRAY 01 .. LENGTHS OF INTEGER;

51

52

53 VAR

54 FREEV : VPTR;
55 FREEC : CPTR;
56 IUNT : INDEX;
57 TAB : TABLE;
58 PRINT : TEXT;
59 DEBUG, DEBIG : BOOLEAN;
60 FACT : REAL;
61 NUMC : INDEX;
62 NUMOR : INDEX;

```

63      XTAB          : ARRAY CINDEXS OF INDEX;
64      IIPERM, N, M, K, NI : INTEGER;
65      LPER          : INTEGER;
66      A             : CPTR;
67      B             : ARRAY C] .. MAXELS OF CPTR;
68      C             : ARRAY C] .. MAXELS OF ] .. MAXELS;
69      D             : INTEGER;
70      L, LMAX, LMIN  : INTEGER;
71      LL            : REAL;
72      DEBOG         : BOOLEAN;
73      RAND, RM, RA, START : INTEGER;
74      I             : INTEGER;
75
76      PROCEDURE RANSET(I : INTEGER);
77      BEGIN
78          START := 5;
79      END (= FUNCTION RANSET=);
80
81      FUNCTION RANDON : REAL;
82      VAR I : INTEGER;
83      BEGIN
84          IF START = 1 THEN
85              BEGIN
86                  START := 0;
87                  RM := 2; FOR I := 1 TO 19 DO RM := RM*2;
88                  RAND := 566387;
89                  RA := 2; FOR I := 1 TO 9 DO RA := RA*2; RA := RA + 3;
90              END;
91              RAND := (RA+RAND) MOD RM;
92              RANDON := RAND/RM;
93          END (= FUNCTION RANDON=);
94
95
96      FUNCTION NEWV : VPTR;
97
98      VAR
99          V          : VPTR;
100
101      BEGIN (= FUNCTION NEWV =);
102          IF FREEV = NIL THEN NEWV(V) ELSE
103              BEGIN V := FREEV; FREEV := V2,NEXT END;
104              NEWV := V;
105          END (= FUNCTION NEWV =);
106
107      FUNCTION NEWC : CPTR;
108
109      VAR
110          C          : CPTR;
111
112      BEGIN (= FUNCTION NEWC =);
113          IF FREEC = NIL THEN NEWC(C) ELSE
114              BEGIN C := FREEC; FREEC := C2,NEXT END;
115              NEWC := C;
116          END (= FUNCTION NEWC =);
117
118      PROCEDURE DISPV(V : VPTR);
119
120      VAR
121          W          : VPTR;
122
123      BEGIN (= PROCEDURE DISPV =);
124          IF V <> NIL THEN
125              BEGIN W := V2,NEXT; V2,NEXT := FREEV; FREEV := W END

```

```

124     END (= PROCEDURE DISPV =)
127
128     PROCEDURE DISPCIC : CPTRI
129
130     VAR
131     D           : CPTRI
132
133     BEGIN (= PROCEDURE DISPC =)
134     WHILE C <> NIL DO
135     WITH C2 DO
136     BEGIN DISPC(SIN); DISPV(V); D := NEXT; NEXT := FREEC;
137     FREEC := C; C := D
138     END
139     END (= PROCEDURE DISPC =)
140
141     FUNCTION COPYV(A : VPTR) : VPTR
142
143     VAR
144     P, Q, R, S   : VPTR
145
146     BEGIN (= FUNCTION COPYV =)
147     IF A <> NIL THEN
148     BEGIN P := NEWV; Q := P; PZ.X := AZ.X; PZ.Y := AZ.Y;
149     R := AZ.NEXT;
150     REPEAT S := NEWV; SZ.X := RZ.X; SZ.Y := RZ.Y; QZ.NEXT := S;
151     Q := S; R := RZ.NEXT
152     UNTIL R = A;
153     QZ.NEXT := P
154     END (= IF A <> NIL THEN =)
155     ELSE P := NIL; COPYV := P
156     END (= FUNCTION COPYV =)
157
158     FUNCTION COPY(A : CPTR) : CPTR
159
160     VAR
161     P, Q           : CPTR
162
163     BEGIN (= FUNCTION COPY =)
164     P := NIL;
165     WHILE A <> NIL DO
166     WITH A2 DO
167     BEGIN W := NEWC;
168     WITH W2 DO
169     BEGIN V := COPYV(AZ.V); SIN := COPY(AZ.SIN); NEXT := P
170     END;
171     P := W; A := NEXT
172     END (= WITH A2 DO =)
173     COPY := P
174     END (= FUNCTION COPY =)
175
176     FUNCTION CONC(A, B : CPTR) : CPTR
177
178     VAR
179     P           : CPTR
180
181     BEGIN (= FUNCTION CONC =)
182     IF A = NIL THEN CONC := B ELSE
183     BEGIN P := A;
184     WHILE PZ.NEXT <> NIL DO P := PZ.NEXT; PZ.NEXT := B;
185     CONC := A
186     END
187     END (= FUNCTION CONC =)
188

```

```

189 FUNCTION SQUEEZE(V : VPTR) : VPTR
190
191 VAR
192   Q, QO, R, S : VPTR
193   DELETE      : BOOLEAN
194
195 BEGIN (= FUNCTION SQUEEZE =)
196   Q := V; QO := V
197   IF Q <> NIL THEN
198     BEGIN
199       REPEAT N := Q2.NEXT; S := R2.NEXT;
200         DELETE := Q2.X * (R2.Y - S2.Y) +
201           R2.X * (S2.Y - Q2.Y) +
202           S2.X * (Q2.Y - R2.Y) = 0;
203       IF DELETE THEN
204         BEGIN Q2.NEXT := S; QO := Q; R2.NEXT := FREEV;
205           FREEV := R;
206         END
207       ELSE U := R
208         UNTIL (U = QO) AND NOT DELETE OR (Q2.NEXT = Q);
209       IF Q2.NEXT = Q THEN
210         BEGIN Q2.NEXT := FREEV; FREEV := Q; U := NIL; END
211       END;
212     SQUEEZE := Q
213   END (= FUNCTION SQUEEZE =)
214
215 FUNCTION SQUEEZE(A : CPTR) : CPTR
216
217 VAR
218   P, P1, X : CPTR
219
220 BEGIN (= FUNCTION SQUEEZE =)
221   P := A;
222   WHILE P <> NIL DO
223     WITH P2 DO
224       BEGIN V := SQUEEZE(V);
225         IF V = NIL THEN
226           BEGIN X := P; P := X2.NEXT;
227             X2.NEXT := NIL;
228             IF X = A THEN A := P ELSE P12.NEXT := P; DISPC(A)
229           END
230         ELSE
231           BEGIN S1N := SQUEEZE(S1N); P1 := P; P := NEXT
232         END
233       END (= WITH P2 DO =)
234     SQUEEZE := A
235   END (= FUNCTION SQUEEZE =)
236
237 FUNCTION POSITIVE(P : VPTR) : BOOLEAN
238
239 VAR
240   A, B, C : VPTR
241   S : INTEGER
242
243 BEGIN (= FUNCTION POSITIVE =)
244   A := P; B := P2.NEXT; S := 0;
245   REPEAT C := B2.NEXT; S := S + B2.X * (C2.Y - A2.Y);
246     A := B; B := C
247   UNTIL A = P;
248   POSITIVE := S > 0
249 END (= FUNCTION POSITIVE =)
250
251 PROCEDURE OUTTAB;

```

```

252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314

```

```

VAR
IA, IB      : INDEXI
W           : INTEGERI

FUNCTION NEXT(XA, YY : INTEGER; SS : INDEXI; GG : BOOLEAN) : INDEXI
BEGIN (*- FUNCTION NEXT -*)
IF NUMT >= MAXP THEN
BEGIN WRITELN(OUTPUT, ' TABLE FULL');
WRITELN(PRINT, ' TABLE FULL');
OUTTAB; GOTO 13
END;
NUMT := NUMT + 1;
WITH TABCNUMTS DO
BEGIN X := XI; Y := YY; S := SS; GR := GG; UN := FALSE;
K := 0; L := 0; AC := FALSE; PG := FALSE
END;
NEXT := NUMT
END (*- FUNCTION NEXT -*);

PROCEDURE ENTERP(A : VPTR; C : BOOLEAN; VAR IA : INDEXI);

VAR
I, IB      : INDEXI
P          : VPTR;

BEGIN (*- PROCEDURE ENTERP -*)
P := A;
REPEAT
WITH PZ DO
BEGIN
I := NEXT(X, Y, 0, FALSE);
IF P = A THEN
BEGIN IA := I; IB := I END
ELSE
BEGIN
IF C THEN TABCIAS.N := IB
ELSE TABCIBS.N := I;
IB := I
END;
P := NEXT
END
UNTIL P = A;
IF C THEN TABCIAS.N := IB
ELSE TABCIBS.N := IA;

I := IA;
REPEAT
WITH TABCIS DO
BEGIN N := I; L := N; I := N END
UNTIL I = IA
END (*- PROCEDURE ENTERP -*);

PROCEDURE MOVL;

BEGIN (*- PROCEDURE MOVL -*)
WRITELN(OUTPUT, ' MULTI INTEGER OVERFLOW');
WRITELN(PRINT, ' MULTI INTEGER OVERFLOW');
GOTO 13
END (*- PROCEDURE MOVL -*);

PROCEDURE CHS(VAR A : MULTI);

```

```

315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

```

```

VAR
  I, P          : INTEGER
BEGIN (== PROCEDURE CHS ==)
  P := 11
  FOR I := 0 TO LENGTH DO
    BEGIN P := SURTMAX - I + P - ACIS
          ACIS := P MOD SURTMAX; P := P DIV SURTMAX
    END
  END (== PROCEDURE CHS ==)

PROCEDURE HADD(A, B : MULTI; VAR C : MULTI);

VAR
  I, P          : INTEGER
BEGIN (== PROCEDURE HADD ==)
  P := 01
  FOR I := 0 TO LENGTH DO
    BEGIN P := ACIS + BCIS + PI
          CCIS := P MOD SURTMAX; P := P DIV SURTMAX
    END;
    IF (CLENGTHS > 0) AND (CLENGTHS < SURTMAX - 1)
      THEN MOVL
    END (== PROCEDURE HADD ==)

PROCEDURE HSUB(A, B : MULTI; VAR C : MULTI);

BEGIN (== PROCEDURE HSUB ==)
  CHS(B); HADD(A, B, C)
END (== PROCEDURE HSUB ==)

PROCEDURE HMUL(A, B : MULTI; VAR C : MULTI);

VAR
  I, J, P      : INTEGER
  SA, SB       : BOOLEAN
BEGIN (== PROCEDURE HMUL ==)
  SA := ALENGTHS > 0; SB := BLENGTHS > 0
  IF SA THEN CHS(A); IF SB THEN CHS(B);
  FOR I := 0 TO LENGTH DO CCIS := 0;
  FOR I := 0 TO LENGTH DO
    BEGIN P := 01
          FOR J := 0 TO LENGTH DO
            BEGIN P := ACIS + BCJS + PI
                  IF I + J < LENGTH THEN
                    BEGIN P := P + CCI + JSI
                          CCJ + JS := P MOD SURTMAX; P := P DIV SURTMAX
                    END
                  ELSE IF P > 0 THEN P := 1
                    END;
            IF P > 0 THEN MOVL
            END;
          IF SA <> SB THEN CHS(C)
        END (== PROCEDURE HMUL ==)

PROCEDURE HSETVAR A : MULTI; X : INTEGER);

VAR
  I, P          : INTEGER

```



```

378 BEGIN (*= PROCEDURE HSET *)
379 P := ABS(X)
380 FOR I := 0 TO LENGTH DO
381 BEGIN
382 ACIS := P MOD SQRTMAX; P := P DIV SQRTMAX
383 END;
384 IF X < 0 THEN CHS(A)
385 END (*= PROCEDURE HSET *)
386
387 FUNCTION SIGN(VAR A : MULTI); INTEGER;
388
389 VAR
390 I : INTEGER;
391
392 BEGIN (*= FUNCTION SIGN *)
393 IF ACLENGTHS > 0 THEN SIGN := -1 ELSE
394 BEGIN I := LENGTH;
395 WHILE (I > 0) AND (ACIS = 0) DO I := I - 1;
396 SIGN := ORD(ACIS > 0)
397 END
398 END (*= FUNCTION SIGN *)
399
400 FUNCTION QL(I, J, K : INDEX) : INTEGER;
401
402 BEGIN (*= FUNCTION QL *)
403 QL := TABCS,X * (TARCJ,Y - TABCKS,Y) + TABCJS,X * (TABCKS,Y -
404 TABCS,Y) + TABCKS,X * (TABCS,Y - TARCJ,Y)
405 END (*= FUNCTION QL *)
406
407 PROCEDURE CUT(A, IB : INDEX);
408
409 VAR
410 I, I1, I2, I3, J, J1, J2, L, M : INDEX;
411 S1, S2, S3, S4, XA, YA,
412 XB, YB, P1, P2, Q1, Q2 : INTEGER;
413 CR : BOOLEAN;
414
415 PROCEDURE SORT(L, R : INDEX);
416
417 VAR
418 I, J : INDEX;
419 X, W : INDEX;
420
421 FUNCTION LESS(VAR I, J : INDEX) : BOOLEAN;
422
423 TYPE
424 M = ARRAY CO .. 35 OF INTEGER;
425
426 VAR
427 C, D : M;
428
429 PROCEDURE MUL(AA, BB : INTEGER; VAR C : M);
430
431 VAR
432 I, J : 0 .. 3;
433 A, B : ARRAY CO .. 35 OF INTEGER;
434 X, P : INTEGER;
435
436 BEGIN (*= PROCEDURE MUL *)
437 ACIS := AA DIV SQRTMAX; ACCS := AA MOD SQRTMAX;
438 BCIS := BB DIV SQRTMAX; BCJS := BB MOD SQRTMAX;
439 FOR I := 0 TO 3 DO CCIS := 0;
440 FOR J := 0 TO 3 DO

```

```

441 BEGIN P := 0;
442 FOR I := 0 TO 1 DO
443 BEGIN
444 P := P + ACIS + BCJS + CC1+JS;
445 CC1+JS := P MOD SURTHAX; P := P DIV SURTHAX
446 END;
447 CCJ+2S := P
448 END
449 END (= PROCEDURE MUL =);
450
451 BEGIN (= FUNCTION LESS =)
452 IF TABCIS.K = TABCJS.K THEN
453 BEGIN
454 MUL(TABCIS.X, TABCJS.Y, C1) MUL(TABCIS.Y, TABCJS.X, D1)
455 IF CC3S <> DC3S THEN LESS := CC3S < DC3S ELSE
456 IF CC2S <> DC2S THEN LESS := CC2S < DC2S ELSE
457 IF CC1S <> DC1S THEN LESS := CC1S < DC1S ELSE
458 LESS := CCUS < DCUS
459 END
460 ELSE LESS := TABCIS.K < TABCJS.K
461 END (= FUNCTION LESS =);
462
463 BEGIN (= PROCEDURE SORT =)
464 I := L; J := R; X := XTABC(L+R) DIV 2;
465 REPEAT
466 WHILE LESS(X, XTABCIS) DO I := I + 1;
467 WHILE LESS(X, XTABCJS) DO J := J - 1;
468 IF I <= J THEN
469 BEGIN W := XTABCIS; XTABCIS := XTABCJS; XTABCJS := W;
470 I := I + 1; J := J - 1;
471 END;
472 UNTIL I > J;
473 IF L < J THEN SORT(I, J);
474 IF I < K THEN SORT(I, R);
475 END (= PROCEDURE SORT =);
476
477 BEGIN (= PROCEDURE CUT =)
478 I := IA;
479 REPEAT I1 := TABCIS.I1; J := I;
480 REPEAT J1 := TABCJS.H1; CR := FALSE;
481 S1 := QL(I, I1, J); S3 := QL(J, J1, I);
482 S2 := QL(I, I1, J); S4 := QL(J, J1, I);
483 IF S1 = 0 THEN
484 WITH TABCIS DO
485 BEGIN
486 XA := TABCIS.X - X; YA := TABCIS.Y - Y;
487 XB := TABCJS.X - X; YB := TABCJS.Y - Y;
488 P1 := XA + XB + YA + YB;
489 Q1 := SQR(XA) + SQR(YA);
490 IF (P1 >= 0) AND (P1 < Q1) THEN
491 BEGIN CR := TRUE; P2 := 0; Q2 := 1; END
492 END
493 ELSE IF S3 = 0 THEN
494 WITH TABCJS DO
495 BEGIN
496 XA := TABCJS.X - X; YA := TABCJS.Y - Y;
497 XB := TABCIS.X - X; YB := TABCIS.Y - Y;
498 P2 := XA + XB + YA + YB;
499 Q2 := SQR(XA) + SQR(YA);
500 IF (P2 >= 0) AND (P2 < Q2) THEN
501 BEGIN CR := TRUE; P1 := 0; Q1 := 1; END
502 END
503 ELSE

```

```

504 IF (S1 > 0) AND (S2 < 0) AND (S3 < 0) AND (S4 > 0) OR
505 (S1 < 0) AND (S2 > 0) AND (S3 > 0) AND (S4 < 0) THEN
506 BEGIN CR := TRUE;
507 P1 := S3; Q1 := S3 - S4;
508 IF Q1 < 0 THEN BEGIN P1 := - P1; Q1 := - Q1 END;
509 P2 := S1; Q2 := S1 - S2;
510 IF Q2 < 0 THEN BEGIN P2 := - P2; Q2 := - Q2 END;
511 END;
512 IF CR THEN
513 BEGIN
514 DO := CROSS;
515 IF P1 = 0 THEN I2 := I ELSE
516 BEGIN I2 := NEXT(P1, Q1, 0, FALSE);
517 TABCI2S,K := I; TABCI2S,L := I;
518 NUMC := NUMC + 1; XTABCNUMCS := I2;
519 END;
520 IF P2 = 0 THEN J2 := J ELSE
521 BEGIN J2 := NEXT(P2, Q2, 0, FALSE);
522 TABCJ2S,K := J; TABCJ2S,L := J;
523 NUMC := NUMC + 1; XTABCNUMCS := J2;
524 END;
525 IF TABCI2S,GR OR TABCJ2S,GR THEN
526 BEGIN
527 WRITELN(OUTPUT, ' ZAPETLJAN CUT, I, J = ', I; 5, J; 5);
528 WRITELN(PRINT, ' ZAPETLJAN CUT, I, J = ', I; 5, J; 5);
529 OUTTAB; GOTO 13;
530 END;
531
532 WITH TABCI2S DO
533 BEGIN GR := TRUE; S := J2;
534 IF (S1 <> 0) AND (S2 <> 0) AND (S4 <> 0) THEN
535 BEGIN AC := TRUE; UN := S4 > 0 END;
536 END;
537 WITH TABCJ2S DO
538 BEGIN GR := TRUE; S := I2;
539 IF (S2 <> 0) AND (S3 <> 0) AND (S4 <> 0) THEN
540 BEGIN AC := TRUE; UN := S2 > 0 END;
541 END;
542 END (- IF CR THEN -);
543 J := J1;
544 UNTIL J = 18;
545 I := I1;
546 UNTIL I = 18;
547
548 IF NUMC > J THEN
549 BEGIN
550 SORT(I, NUMC);
551
552 IF DEBUG THEN
553 BEGIN
554 WRITELN(PRINT, ' AFTER SORT');
555 OUTTAB;
556 END;
557
558 TABCJS,K := 0; XTABCJS := 0; L := NUMC; H := XTABCLS;
559 WHILE TABCHS,K <> 0 DO
560 BEGIN I := TABCHS,K; I1 := TABCI2S,I; I3 := I1;
561 REPEAT
562 WITH TABCHS DO
563 BEGIN I2 := H;
564 P1 := X; X := Y - X; Y := P1;
565 H := I3; I3 := I2;

```

```

567         END;
568         L := L - 1; M := XTABCLB;
569         UNTIL TABCLS,A <> 1;
570         TABCLS,H := 12;
571     END
572 END;
573
574 IF DEBUG THEN
575     BEGIN
576         WRITELN(PRINT, ' END OF CUT');
577         OUTTAB
578     END
579 END (*= PROCEDURE CUT =*)
580
581 FUNCTION INPOL(I, IB: INDEX): INTEGER;
582
583 LABEL
584 I;
585
586 VAR
587     J, J1      : INDEX;
588     DY0, DY1, DX1, DX2, P, W;
589     XX, YY, NS : INTEGER;
590
591 BEGIN (*= FUNCTION INPOL =*)
592     WITH TABCLS DO
593         BEGIN XX := TABCLS,X + X; YY := TABCLS,Y + Y END;
594     REPEAT
595         IB := TABCLS,H; DY0 := 2 * TABCLS,Y - YY;
596     UNTIL DY0 <> 0;
597     J := IB; INPOL := 0; NS := 0;
598     REPEAT
599         J1 := TABCLS,H; DY1 := 2 * TABCLS,Y - YY;
600     IF DY1 = 0 THEN
601         BEGIN DX1 := 2 * TABCLS,X - XX;
602         IF DX1 = 0 THEN GOTO I;
603         REPEAT
604             J1 := TABCLS,H; DY1 := 2 * TABCLS,Y - YY;
605         IF DY1 = 0 THEN
606             BEGIN DX2 := 2 * TABCLS,X - XX;
607             IF (DX2 = 0) OR ((DX1 < 0) <> (DX2 < 0)) THEN GOTO I;
608         END;
609     UNTIL DY1 <> 0;
610     IF DX1 > 0 THEN
611         IF (DY0 < 0) <> (DY1 < 0) THEN NS := NS + 1;
612     END;
613 ELSE
614     IF (DY0 < 0) <> (DY1 < 0) THEN
615         BEGIN
616             P := TABCLS,X + DY1 - TABCLS,X + DY0;
617             Q := TABCLS,Y - TABCLS,Y1;
618             IF W < 0 THEN
619                 BEGIN P := -P; Q := -Q END;
620             IF P = Q * XX THEN GOTO I;
621             IF P > Q * XX THEN NS := NS + 1;
622         END;
623     J := J1; DY0 := DY1;
624     UNTIL J = IB;
625     INPOL := 2 * ORD(ODD(NS)) - 1;
626 I;
627 END (*= FUNCTION INPOL =*)
628
629 PROCEDURE DEFPR(I, IB: INDEX);

```

```

630
631 VAR
632 I, I1 : INDEX;
633 W : INTEGER;
634 XX, YY : REAL;
635 DF, PR : BOOLEAN;
636
637 FUNCTION INPOL(I, IB: INDEX; INTEGER)
638
639 LABEL
640 I1
641
642 VAR
643 J, J1, I1, I2 : INDEX;
644 P1, P2, Q1, Q2, NS : INTEGER;
645 FCT, P, Q, W, XX, YY, DY0, DY1, DX1, DX2 : MULTI;
646
647 BEGIN (*- FUNCTION INPOL -*)
648 WITH TABCIS DO
649 BEGIN I1 := I; I2 := I;
650 IF I > NHOR THEN BEGIN P1 := X; Q1 := Y END
651 ELSE BEGIN P1 := I; Q1 := 0 END;
652 WITH TABCS DO
653 IF TABCS.I > NHOR THEN BEGIN P2 := X; Q2 := Y END
654 ELSE BEGIN P2 := 0; Q2 := I END;
655 END;
656
657 MSET(FCT, P1 + P2 + Q1 + Q2);
658 MSET(P, P1 + P2); MSET(Q, Q1 + Q2);
659 MMUL(X, TABCIS.X); MMUL(W, TABCIS.W);
660 MMUL(X, P, XX); MMUL(Q, W); MADD(X, W, XX);
661 MSET(YY, TABCIS.Y); MSET(W, TABCIS.W);
662 MMUL(YY, P, YY); MMUL(W, Q, L); MADD(YY, W, YY);
663
664 REPEAT
665 IB := TABCIS.L;
666 MSET(DY0, TABCIS.Y); MMUL(FCT, DY0, DY0);
667 MSUB(DY0, YY, DY0);
668 UNTIL SIGN(DY0) <> 0;
669 J := IB; INPOL := 0; NS := 0;
670 REPEAT
671 J1 := TABCS.L;
672 MSET(DY1, TABCS.Y); MMUL(FCT, DY1, DY1);
673 MSUB(DY1, YY, DY1);
674 IF SIGN(DY1) = 0 THEN
675 BEGIN
676 MSET(DX1, TABCS.X); MMUL(FCT, DX1, DX1);
677 MSUB(DX1, XX, DX1);
678 IF SIGN(DX1) = 0 THEN GOTO I;
679 REPEAT
680 J1 := TABCS.L;
681 MSET(DY1, TABCS.Y); MMUL(FCT, DY1, DY1);
682 MSUB(DY1, YY, DY1);
683 IF SIGN(DY1) = 0 THEN
684 BEGIN MSET(DX2, TABCS.X);
685 MMUL(FCT, DX2, DX2); MSUB(DX2, XX, DX2);
686 IF (SIGN(DX2) = 0) OR
687 ((SIGN(DX1) < 0) <> (SIGN(DX2) < 0)) THEN GOTO I;
688 END;
689 UNTIL SIGN(DY1) <> 0;
690 IF SIGN(DX1) > 0 THEN
691 IF (SIGN(DY1) < 0) <> (SIGN(DY1) < 0) THEN NS := NS + 1;
692 END;

```

```

693 ELSE
694 IF (SIGN(DYU) < 0) <> (SIGN(DYI) < 0) THEN
695 BEGIN
696 MSETIP, TABCJS, XI; MSETIW, TABCJIS, XI;
697 MMULIP, DYI, P; MMULIW, DYD, W;
698 IF TABCJIS, Y >= TABCJS, Y THEN MSUBIP, L, P;
699 ELSE MSUBIW, P, P;
700 MSETIQ, ABS(TABCJIS, Y - TABCJS, Y);
701 MMULIQ, XX, Q; MSUBIP, Q, P;
702 IF SIGN(P) = 0 THEN GOTO 1;
703 IF SIGN(P) > 0 THEN NS := NS + 1;
704 END;
705
706 IF DEBUG THEN
707 WRITELN(PRINT, ' INQOL, I, J, NS = ', I, I, 5, J, 5, NS, 5);
708 J := J; DYU := DYI;
709 UNTIL J = IB;
710 INQOL := 2 * ORD(ODD(NS)) - 1;
711 1;
712 END (* FUNCTION INQOL *)
713
714 BEGIN (* PROCEDURE DEFPN *)
715 I := IA; DF := FALSE;
716 REPEAT
717 WITH TABCIS DO
718 IF AC THEN DF := TRUE ELSE
719 IF GR THEN
720 BEGIN W := INQOL(I, IB);
721 IF W <> 0 THEN
722 BEGIN AC := TRUE; UN := (W < 0) * CB; DF := TRUE END;
723 END;
724 I := TABCIS, I;
725 UNTIL I = IA;
726 IF DEBUG THEN
727 BEGIN
728 WRITELN(PRINT, ' AFTER INPOL');
729 OUTTAB;
730 END;
731
732 IF DF THEN
733 BEGIN
734 WHILE NOT TABCIAS, AC DO IA := TABCIAS, NI;
735 I := IA;
736 REPEAT
737 WITH TABCIS DO
738 IF AC THEN PR := UN ELSE UN := PR;
739 I := TABCIS, I;
740 UNTIL I = IA;
741
742 REPEAT II := TABCIS, II;
743 WITH TABCIS DO
744 BEGIN
745 AC := UN <> TABCIS, UN;
746 IF AC AND NOT GR THEN
747 BEGIN
748 WRITELN(OUTPUT, ' DEFPN, II = ', II, 5);
749 WRITELN(PRINT, ' DEFPN, II = ', II, 5);
750 OUTTAB; GOTO 13;
751 END;
752 END;
753 I := II;
754 UNTIL I = IA;
755

```

```

754      I := IB;
755      REPEAT I := TABCIS,N UNTIL (I = IB) OR TABCIS.AC;
756      IF TABCIS.AC THEN PR := TABCIS,UN ELSE
757      BEGIN
758          REPEAT I := TABCIS,N; W := INPOL(I, IA)
759          UNTIL (W <> 0) OR (I = IB);
760          IF W = 0 THEN
761              BEGIN
762                  WRITELN(OUTPUT, ' DEFPR, IB = ', IB : 5);
763                  WRITELN(PRINT, ' DEFPR, IB = ', IB : 5);
764                  OUTTAB: GOTO 13
765              END;
766          PR := (W < 0) = CA
767          END;
768      IB := I;
769      REPEAT TABCIS,UN := PN; I := TABCIS,N;
770      WITH TABCIS DO
771      IF GR THEN
772      IF TABCSS.AC THEN PR := NOT PR
773      UNTIL I = IB;
774
775      FOR I := 1 TO NUMC DO
776      WITH TABCXTABCIS DO
777      BEGIN
778          XX := X / (X + Y); YY := Y / (X + Y);
779          X := ROUND((XX * TABCKS,X + YY * TABCLS,X));
780          Y := ROUND((XX * TABCKS,Y + YY * TABCLS,Y));
781      END
782      END (== IF DF THEN ==)
783      ELSE OD := IDENT
784      END (== PROCEDURE DEFPR ==);
785
786      PROCEDURE EVICT(IA; INDEX);
787
788      VAR
789      I, II : INDEX;
790
791      BEGIN (== PROCEDURE EVICT ==)
792      I := IA;
793      REPEAT
794      II := TABCIS,N;
795      IF TABCIS.GR AND TABCIS.GR THEN
796      IF TABCTABCIS.SS,N = TABCIS,S THEN
797      IF TABCIS,UN = TABCTABCIS.SS,UN THEN
798      BEGIN TABCIS,PG := TRUE; TABCTABCIS.SS,PG := TRUE END;
799      I := II;
800      UNTIL I = IA
801      END (== PROCEDURE EVICT ==);
802
803      BEGIN (== PROCEDURE FILL ==)
804      NUMT := 0; NUMC := 0;
805      ENTERP(A, CA, IA); ENTERP(B, CB, IB); NUMON := NUMT;
806      OD := DISJUNT; CUT(IA, IB);
807
808      IF OD = CROSS THEN
809      BEGIN DEFPR(IA, IB); EVICT(IA) END
810      ELSE
811      BEGIN J := INPOL(IA, IB);
812      IF W = 0 THEN
813      BEGIN
814          WRITELN(OUTPUT, ' BAD INPOL, IA, IB = ', IA : 5, IB : 5);
815          WRITELN(PRINT, ' BAD INPOL, IA, IB = ', IA : 5, IB : 5);
816          OUTTAB: GOTO 13
817      END
818

```

```

819     ENDI
820     IF W > U THEN OD := AING ELSE
821     BEGIN W := INPOL(IB, IA);
822     IF U = D THEN
823     BEGIN
824     WRITELN(OUTPUT, ' BAD INPOL, IB, IA = ', IB : 5, IA : 5);
825     WRITELN(PRINT, ' BAD INPOL, IB, IA = ', IB : 5, IA : 5);
826     OUTTAB; GOTO I3
827     ENDI
828     IF W > D THEN OD := BINA
829     END
830     ENDI
831
832     IF DEBUG THEN
833     BEGIN
834     WRITELN(PRINT, ' AFTER FILL, IA, IB = ', IA : 5, IB : 5);
835     OUTTAB
836     END
837     END (*- PROCEDURE FILL -*)
838
839     FUNCTION EXCP(PR : BOOLEAN) : CPTR;
840
841     VAR
842     C, D      : CPTR;
843     W, U      : VPTR;
844     FOUND     : BOOLEAN;
845
846     FUNCTION FOLLOW(PR : BOOLEAN) : VPTR;
847
848     VAR
849     I, IA     : INDEX;
850     FOUND, ERROR : BOOLEAN;
851     LAST, W   : VPTR;
852
853     BEGIN (*- FUNCTION FOLLOW -*)
854     IA := J;
855     REPEAT IA := IA + 1;
856     WITH TABCIS DO FOUND := NOT PG AND (UN = PR);
857     UNTIL FOUND OR (IA = NUNT);
858     IF DEBUG THEN
859     BEGIN
860     WRITELN(PRINT, ' START OF FOLLOW, IA = ', IA : 5);
861     OUTTAB
862     ENDI
863     LAST := NIL;
864     IF FOUND THEN
865     BEGIN I := IA;
866     REPEAT
867     WITH TABCIS DO
868     BEGIN
869     W := NEWV; WZ.X := X; WZ.Y := Y;
870     IF LAST = NIL THEN
871     BEGIN WZ.NEXT := W; LAST := W END
872     ELSE
873     BEGIN WZ.NEXT := LASTZ.NEXT; LASTZ.NEXT := W; LAST := W
874     ENDI
875     I := N
876     END (*- WITH TABCIS DO -*)
877     ERROR := FALSE;
878     WITH TABCIS DO
879     IF PG OR (UN <> PR) THEN
880     IF UR THEN
881     BEGIN I := S;

```



```

882         WITH TABCIS DO
883             IF PG OR (UN <> PR) THEN ERROR := TRUE
884         END
885         ELSE ERROR := TRUE;
886     WITH TABCIS DO
887         IF ERROR THEN
888             BEGIN
889                 WRITELN(OUTPUT, ' ZAPETLJAH FOLLOW, IA, I = ', IA ; S, I ; S);
890                 WRITELN(PRINT, ' ZAPETLJAH FOLLOW, IA, I = ', IA ; S, I ; S);
891                 OUTTAB: GOTO J3
892             END
893         ELSE PG := TRUE
894         UNTIL I = IA
895         END (*- IF FOUND THEN *-);
896     FOLLOW := LAST
897     END (*- FUNCTION FOLLOW *-);
898
899 FUNCTION ONEPIECE : VPTR;
900
901     VAR
902         P, Q, R          : VPTR;
903
904     BEGIN (*- FUNCTION ONEPIECE *-)
905         P := W;
906         REPEAT P := P2.NEXT; Q := W;
907             WITH P2 DO
908                 WHILE (Q2.X <> X) OR (Q2.Y <> Y) DO Q := Q2.NEXT
909             UNTIL (Q <> P) OR (P = W);
910             IF Q <> P THEN
911                 BEGIN R := P2.NEXT; P2.NEXT := Q2.NEXT; Q2.NEXT := R;
912                     ONEPIECE := P
913                 END
914             ELSE
915                 BEGIN ONEPIECE := W; W := NIL END
916             END (*- FUNCTION ONEPIECE *-);
917
918 FUNCTION SQUEEZER(V : VPTR) : VPTR;
919
920     VAR
921         Q, Q0, R, S      : VPTR;
922         DELETE           : BOOLEAN;
923
924     BEGIN (*- FUNCTION SQUEEZER *-)
925         W := V; Q0 := V;
926         IF Q <> NIL THEN
927             BEGIN
928                 REPEAT R := Q2.NEXT; S := R2.NEXT;
929                     DELETE := (R2.X = S2.X) AND (R2.Y = S2.Y);
930                 IF DELETE THEN
931                     BEGIN Q2.NEXT := S; Q0 := Q; R2.NEXT := FREEV;
932                         FREEV := R
933                     END
934                 ELSE W := R
935                 UNTIL (Q = Q0) AND NOT DELETE OR (Q2.NEXT = Q);
936                 IF Q2.NEXT = Q THEN
937                     BEGIN Q2.NEXT := FREEV; FREEV := Q; Q := NIL END
938                 END;
939                 SQUEEZER := Q
940             END (*- FUNCTION SQUEEZER *-);
941
942 FUNCTION RELATEIA : CPTR; CPTR;
943
944     VAR

```

```

945      NP, NN      : INTEGNI
946      Q, R, S      : CPTR:
947      X, Y, Z      : VFTR:
948
949      BEGIN (*- FUNCTION RELATE *-)
950      UP := Q; NH := Q; Q := A;
951      WHILE Q <> NIL DO
952      WITH QZ DO
953      BEGIN
954      IF POSITIVE(V) THEN
955      BEGIN NP := NP + 1; R := Q END
956      ELSE NH := NH + 1; Q := NEXT
957      END (*- WITH QZ DO *-);
958      IF NH > Q THEN
959      BEGIN
960      IF NP <> 1 THEN
961      BEGIN WRITELN(OUTPUT, ' NP,NN =', NP : 5, NN : 5);
962      WRITELN(PRINT, ' NP,NN =', NP : 5, NN : 5); GOTO 13
963      END;
964      Q := A; A := R;
965      WHILE Q <> NIL DO
966      WITH QZ DO
967      BEGIN S := NEXT;
968      IF Q <> R THEN
969      BEGIN NEXT := RZ.SIN; RZ.SIN := Q; X := V; Y := XZ.NEXT;
970      REPEAT Z := YZ.NEXT; YZ.NEXT := X; X := Y; Y := Z
971      UNTIL X = Y
972      END (*- IF Q <> R THEN *-);
973      Q := S
974      END (*- WITH QZ DO *-);
975      AZ.NEXT := NIL
976      END (*- IF NH > Q THEN *-);
977      RELATE := A
978      END (*- FUNCTION RELATE *-);
979
980      BEGIN (*- FUNCTION EXCP *-)
981      C := NIL;
982      REPEAT W := SQUEEZER(FOLLOW(PW)); FOUND := W <> NIL;
983      WHILE W <> NIL DO
984      BEGIN U := SQUEEZEPIONEPIECE;
985      IF U <> NIL THEN
986      BEGIN D := NEUC;
987      WITH DZ DO
988      BEGIN V := U; NEXT := C; SIN := NIL END;
989      C := D
990      END
991      END (*- WHILE W <> NIL DO *-);
992      UNTIL NOT FOUND;
993      EXCP := RELATE(C)
994      END (*- FUNCTION EXCP *-);
995
996      FUNCTION PRE(A, B : CPTR) : CPTR; FORWARD;
997
998      FUNCTION UNIA(A, B : CPTR) : CPTR; FORWARD;
999
1000     FUNCTION DIF(A, B : CPTR) : CPTR; FORWARD;
1001
1002     FUNCTION PRE;
1003
1004     LABEL
1005     ;;
1006
1007     VAR

```

```

1071      X := Q1 Q := X2.NEXT1 X2.NEXT := NIL1
1072      IF Q1 = NIL THEN B := Q ELSE Q12.NEXT := Q1 DISPC(X1)
1073      GOTO 1
1074      END;
1075      AINB :
1076      BEGIN X := P1 P := X2.NEXT1 X2.NEXT := NIL1
1077      Y := DIF(Q2,SIN, X1) DISPC(Q2,SIN1) Q2,SIN := Y1
1078      IF P1 = NIL THEN A := P ELSE P12.NEXT := P1 DISPC(X1)
1079      GOTO 2
1080      END;
1081      BINA :
1082      BEGIN X := Q1 Q := X2.NEXT1 X2.NEXT := NIL1
1083      Y := DIF(P2,SIN, X1) DISPC(P2,SIN1) P2,SIN := Y1
1084      IF Q1 = NIL THEN B := Q ELSE Q12.NEXT := Q1 DISPC(A)
1085      END;
1086      CROSS :
1087      BEGIN R := EXCP(FALSE); X := P1 P := X2.NEXT1
1088      X2.NEXT := NIL1 A1 := X2,SIN1 X2,SIN := NIL1 Y := Q1
1089      Q := Y2.NEXT1 Y2.NEXT := NIL1 B1 := Y2,SIN1
1090      Y2,SIN := NIL1 S := DIF(A1, Y1) S := CONCDIF(B1, X1, S1)
1091      S := CONCPRE(A1, B1), S1 DISPC(A1) DISPC(B1);
1092      Z := DIF(R, S1) DISPC(S1) DISPC(X1);
1093      IF P1 = NIL THEN A := P ELSE P12.NEXT := P1
1094      IF Q1 = NIL THEN B := Q ELSE Q12.NEXT := Q1
1095      IF Z <> NIL THEN
1096      BEGIN S := Z;
1097      WHILE S2.NEXT <> NIL DO S := S2.NEXT1 S2.NEXT := B1
1098      IF Q1 = NIL THEN Q1 := S1 B := Z
1099      END;
1100      DISPC(X1) DISPC(Y1) GOTO 2
1101      END;
1102      DISJUNKT :
1103      BEGIN Q1 := Q1 Q := Q2.NEXT END
1104      END (*- CASE OD OF -*)
1105      END (*- WHILE Q <> NIL DO -*)
1106      I :
1107      P1 := P1 P := P2.NEXT1
1108      Z :
1109      END (*- WHILE P <> NIL DO -*)
1110      UNI := CONCA(A, B)
1111      END (*- FUNCTION UNI -*)
1112
1113      FUNCTION DIF;
1114
1115      VAR
1116      S, Q, C, X, Y, D, R : CPTR1;
1117      OD : UDNOS1;
1118
1119      BEGIN (*- FUNCTION DIF -*)
1120      IF B = NIL THEN DIF := COPY(A) ELSE
1121      BEGIN S := NIL;
1122      WHILE A <> NIL DO
1123      BEGIN Q := NEWC1 Q2,V := COPY(A2,V);
1124      Q2,NEXT := NIL; Q2,SIN := NIL;
1125      D := UNIA2,SIN, B1; C := D1 R := NIL;
1126      WHILE C <> NIL DO
1127      BEGIN FILL(Q2,V, C2,V, OD, FALSE, TRUE);
1128      CASE OD OF
1129      IDENT :
1130      BEGIN S := CONCD(COPY(C2,SIN), S1) DISPC(W1);
1131      W := NIL; C := NIL
1132      END;
1133      AINB :

```

```

1008      S, Q, Q1, R, X, Y : CPTR1
1009      OD                : ODNOS1
1010
1011      BEGIN (= FUNCTION PRE =)
1012      B := COPY(B); S := NIL
1013      WHILE A <> NIL DO
1014      BEGIN Q := B; Q1 := NIL
1015      WHILE Q <> NIL DO
1016      BEGIN FILL(AZ.V, QZ.V, OD, FALSE, FALSE);
1017      CASE OD OF
1018      IDENT :
1019      BEGIN R := NEWCI
1020      WITH RZ DO
1021      BEGIN V := COPYV(AZ.V); NEXT := S;
1022      SIN := UN1(AZ.SIN, RZ.SIN)
1023      END;
1024      S := R; X := Q; Q := XZ.NEXT; XZ.NEXT := NIL
1025      IF Q1 = NIL THEN W := Q ELSE Q1Z.NEXT := Q; DISPC(X);
1026      GOTO 1
1027      END;
1028      AINB :
1029      BEGIN X := AZ.NEXT; AZ.NEXT := NIL; Y := DIF(A, QZ.SIN);
1030      S := CONCI(Y, S); AZ.NEXT := X; Q1 := Q; W := QZ.NEXT;
1031      GOTO 1
1032      END;
1033      BINJA :
1034      BEGIN X := Q; Q := XZ.NEXT; XZ.NEXT := NIL;
1035      Y := DIF(X, AZ.SIN); S := CONCI(Y, S);
1036      IF Q1 = NIL THEN B := Q ELSE Q1Z.NEXT := Q; DISPC(X)
1037      END;
1038      CROSS :
1039      BEGIN R := EXCP(TRUE); X := UN1(AZ.SIN, QZ.SIN);
1040      Y := DIF(R, X); DISPC(R); DISPC(X); S := CONCI(Y, S);
1041      Q1 := Q; Q := QZ.NEXT
1042      END;
1043      DISJUNKT :
1044      BEGIN Q1 := Q; Q := QZ.NEXT END
1045      END (= CASE OD OF =)
1046      END (= WHILE Q <> NIL DO =);
1047      1 :
1048      A := AZ.NEXT
1049      END (= WHILE A <> NIL DO =);
1050      DISPC(B); PRE := S
1051      END (= FUNCTION PRE =);
1052
1053      FUNCTION UN1;
1054
1055      LABEL
1056      1, 2;
1057
1058      VAR
1059      P, P1, Q, Q1, R, S, X, Y, Z, A1, B1 : CPTR1
1060      OD                : ODNOS1
1061
1062      BEGIN (= FUNCTION UN1 =)
1063      A := COPY(A); B := COPY(B); P := A; P1 := NIL;
1064      WHILE P <> NIL DO
1065      BEGIN Z := B; Q1 := NIL;
1066      WHILE Q <> NIL DO
1067      BEGIN FILL(PZ.V, QZ.V, OD, FALSE, FALSE);
1068      CASE OD OF
1069      IDENT :
1070      BEGIN X := PRE(PZ.SIN, QZ.SIN); DISPC(PZ.SIN); PZ.SIN := X;

```

```

1134 BEGIN
1135 S := CONC(PREIQ, C2.SIN), S); DISPC(Q);
1136 Q := NIL; C := NIL
1137 END;
1138 BINA :
1139 BEGIN X := C2.NEXT; C2.NEXT := NIL;
1140 R := CONC(COPYIC), R); C2.NEXT := X; C := X
1141 END;
1142 CROSS :
1143 BEGIN X := EXCP(TRUE);
1144 IF X = NIL THEN
1145 BEGIN S := CONC(PREIQ, C2.SIN), S); DISPC(Q);
1146 Q := NIL; C := NIL
1147 END
1148 ELSE
1149 IF X2.NEXT = NIL THEN
1150 BEGIN DISPC(Q); Q := X; R := CONC(C2.SIN, R);
1151 Q2.SIN := NIL; C := C2.NEXT
1152 END
1153 ELSE
1154 BEGIN R := CONC(R, COPY(C2.NEXT)); Y := DIF(X, R);
1155 DISPC(X); DISPC(R); S := CONC(Y, S);
1156 DISPC(Q); Q := NIL; C := NIL
1157 END (== ELSE IF X2.NEXT = NIL THEN ==)
1158 END;
1159 DISJUNKT:
1160 C := C2.NEXT
1161 END (== CASE OF OF ==)
1162 END (== WHILE C <> NIL DO ==)
1163 DISPC(D);
1164 IF J <> NIL THEN
1165 BEGIN Q2.SIN := R; S := CONC(Q, S) END;
1166 A := A2.NEXT
1167 END (== WHILE A <> NIL DO ==);
1168 DIF := S
1169 END (== ELSE IF B = NIL THEN ==);
1170 END (== FUNCTION DIF ==);
1171
1172 FUNCTION PRESEK(A, B : CPTR) : CPTR; FORWARD;
1173
1174 FUNCTION UNIJA(A, B : CPTR) : CPTR; FORWARD;
1175
1176 FUNCTION DIFERENCA(A, B : CPTR) : CPTR; FORWARD;
1177
1178
1179 FUNCTION COMPL(A : CPTR) : CPTR;
1180
1181 VAR
1182 P : CPTR;
1183
1184 BEGIN (== FUNCTION COMPL ==)
1185 IF A = NIL THEN
1186 BEGIN P := NEWC;
1187 WITH P2 DO
1188 BEGIN V := NIL; NEXT := NIL; SIN := NIL END
1189 END (== IF A = NIL THEN ==)
1190 ELSE
1191 WITH A2 DO
1192 IF V = NIL THEN P := COPY(SIN) ELSE
1193 BEGIN P := NEWC;
1194 WITH P2 DO
1195 BEGIN V := NIL; NEXT := NIL; SIN := COPY(A) END
1196 END (== ELSE IF V = NIL THEN ==);

```

```

1197      COMPL := P
1198      END (*- FUNCTION COMPL -*)
1199
1200  PROCEDURE OUT(VAR F : TEXT; A : CPTR)
1201
1202      VAR
1203          G          : INTEGER
1204
1205  PROCEDURE OUT1(A : CPTR; OTAC, BRAT : INTEGER)
1206
1207      VAR
1208          ID          : INTEGER
1209          Q           : VPTR
1210
1211  BEGIN (*- PROCEDURE OUT1 -*)
1212      IF A <> NIL THEN
1213          BEGIN G := G + 1; ID := G;
1214              WRITELN(F, ' : 25; ID ', ID : 5, ' : 5; 'BRAT ', BRAT : 5,
1215                  ' : 5; 'OTAC ', OTAC : 5; Q := AZ,V;
1216              IF Q <> NIL THEN
1217                  REPEAT
1218                      WITH Q2 DO
1219                          BEGIN WRITELN(F, ' : 30; X / FACT : 10 : 5,
1220                              Y / FACT : 10 : 5);
1221                              Q := NEXT
1222                          END
1223                      UNTIL Q = AZ,V
1224                      ELSE WRITELN(F, ' : 30; 'NIL');
1225                      WITH AZ DO
1226                          BEGIN OUT1(NEXT, OTAC, ID); OUT1(SIN, ID, Q) END
1227                      END (*- IF A <> NIL THEN -*)
1228                  END (*- PROCEDURE OUT1 -*)
1229
1230  BEGIN (*- PROCEDURE OUT -*)
1231      WRITELN(F); WRITELN(F); G := 0; OUT1(A, Q, Q);
1232      WRITELN(F); WRITELN(F); WRITE(F, ' ');
1233      END (*- PROCEDURE OUT -*)
1234
1235  FUNCTION PRESEK:
1236
1237      VAR
1238          C          : CPTR;
1239
1240  BEGIN (*- FUNCTION PRESEK -*)
1241      IF (A = NIL) OR (B = NIL) THEN C := NIL ELSE
1242          IF AZ,V = NIL THEN C := DIFERENCA(B, AZ,SIN) ELSE
1243              IF BZ,V = NIL THEN C := DIFERENCA(A, BZ,SIN) ELSE C := PNL(A, B);
1244      PRESEK := C
1245      END (*- FUNCTION PRESEK -*)
1246
1247  FUNCTION UNIJA:
1248
1249      VAR
1250          C          : CPTR;
1251
1252  BEGIN (*- FUNCTION UNIJA -*)
1253      IF A = NIL THEN C := COPY(B) ELSE
1254          IF B = NIL THEN C := COPY(A) ELSE
1255              IF AZ,V = NIL THEN C := DIFERENCA(B, AZ,SIN) ELSE
1256                  IF BZ,V = NIL THEN C := DIFERENCA(A, BZ,SIN) ELSE
1257                      C := UNIJA, B; UNIJA := C
1258                  END (*- FUNCTION UNIJA -*)
1259

```

```

1260 FUNCTION DIFERENCAI
1261
1262 VAR
1263     C           : CPTR;
1264
1265 BEGIN (= FUNCTION DIFERENCA =)
1266     IF B = NIL THEN C := COPY(A) ELSE
1267     IF A = NIL THEN C := NIL ELSE
1268     IF BZ,V = NIL THEN C := PRESEN(A, BZ,SIN) ELSE
1269     IF A2,V = NIL THEN
1270     BEGIN C := NEWCT
1271     WITH C2 DO
1272     BEGIN V := NIL; NEXT := NIL; SIN := UNIAZ,SIN; B)
1273     END
1274     END (= IF A2,V = NIL THEN =)
1275     ELSE C := DIF(A, B); DIFERENCA := C
1276     END (= FUNCTION DIFERENCA =);
1277
1278 FUNCTION INCLUSION(AA, B : CPTR) : CPTR;
1279
1280 VAR
1281     S, D, A      : CPTR;
1282     PH, P        : VPTR;
1283     MII, MINI, DX, DY, PLB : INTEGER;
1284
1285 PROCEDURE DEBPRIN : HESS; A : CPTR;
1286
1287 BEGIN (= PROCEDURE DEBPR =)
1288     WRITELN(PRINT); WRITELN(PRINT, MII); WRITELN(PRINT);
1289     OUT(PRINT, A)
1290     END (= PROCEDURE DEBPR =);
1291
1292 PROCEDURE ROUNDA : CPTR;
1293
1294 PROCEDURE ROUNDP(A, B : VPTR; VAR S : CPTR);
1295
1296 VAR
1297     P, Q, R, T, P1, P2, P3, P4 : VPTR;
1298     C, S1, D, D1 : CPTR;
1299     UX, UY, VX, VY : INTEGER;
1300
1301 BEGIN (= PROCEDURE ROUNDP =)
1302     P := A;
1303     REPEAT Q := P2,NEXT; UX := D1; UY := D1; R := B; D := NIL;
1304     REPEAT T := R2,NEXT; VX := UX + R2,X - T2,X;
1305     VY := UY + R2,Y - T2,Y; P1 := NEWV; P2 := NEWV;
1306     P3 := NEWV; P4 := NEWV;
1307     WITH R12 DO
1308     BEGIN X := P2,X + UX; Y := P2,Y + UY; NEXT := P2 END;
1309     WITH P22 DO
1310     BEGIN X := Q2,X + VX; Y := Q2,Y + VY; NEXT := P3 END;
1311     WITH P32 DO
1312     BEGIN X := Q2,X + VX; Y := Q2,Y + VY; NEXT := P4 END;
1313     WITH P42 DO
1314     BEGIN X := P2,X + VX; Y := P2,Y + VY; NEXT := P1 END;
1315     C := NEWCT;
1316     WITH C2 DO
1317     BEGIN V := P1; NEXT := NIL; SIN := NIL END;
1318     C := SQUEEZE(C);
1319     IF C <> NIL THEN
1320     BEGIN P1 := C2,V;
1321     IF NOT POSITIVE(P1) THEN
1322     BEGIN P2 := P1; P3 := P22,NEXT;

```

```

1323         REPEAT P4 := P32,NEXT; P32,NEXT := P2; P2 := P4;
1324         P3 := P4;
1325         UNTIL P2 = P1;
1326         END (* IF NOT POSITIVE(P1) THEN **);
1327         D1 := UN(D, C); DISPC(D); DISPC(C); D := D1;
1328         END (* IF C <> NIL THEN **);
1329         UX := VX; UY := VY; R := T;
1330         UNTIL R = B;
1331         IF DEBIG THEN DEBPR(' ODUZETA LJUSKA ', D);
1332         S1 := DIF(S, D); DISPC(S); DISPC(D); S := S1;
1333         P := Q;
1334         UNTIL P = A;
1335         END (* PROCEDURE ROUNDPP **);
1336
1337     BEGIN (* PROCEDURE ROUND **);
1338     WHILE A <> NIL DO
1339     BEGIN ROUNDPP(AZ,V, BZ,V, S); ROUND(AZ,S); A := AZ,NEXT;
1340     END;
1341     END (* PROCEDURE ROUND **);
1342
1343     PROCEDURE TRANSLATE(A : CPTR);
1344
1345     VAR
1346     P          : VPTR;
1347
1348     BEGIN (* PROCEDURE TRANSLATE **);
1349     WHILE A <> NIL DO
1350     WITH AZ DO
1351     BEGIN P := V;
1352     REPEAT
1353     WITH PZ DO
1354     BEGIN X := X + DX; Y := Y + DY; P := NEXT END;
1355     UNTIL P = V;
1356     TRANSLATE(S); A := NEXT;
1357     END (* WITH AZ DO **);
1358     END (* PROCEDURE TRANSLATE **);
1359
1360     FUNCTION PL(P : VPTR) : INTEGER;
1361
1362     VAR
1363     A, B, C          : VPTR;
1364     S                : INTEGER;
1365
1366     BEGIN (* FUNCTION PL **);
1367     A := P; B := PZ,NEXT; S := D;
1368     REPEAT C := BZ,NEXT; S := S + BZ,X + (CZ,Y - AZ,Y);
1369     A := B; B := C;
1370     UNTIL A = P;
1371     PL := S;
1372     END (* FUNCTION PL **);
1373
1374     FUNCTION REDCOPY(A : CPTR) : CPTR;
1375
1376     VAR
1377     P, Q             : CPTR;
1378
1379     BEGIN (* FUNCTION REDCOPY **);
1380     P := NIL;
1381     WHILE A <> NIL DO
1382     WITH AZ DO
1383     BEGIN
1384     IF PL(V) >= PLB THEN
1385     BEGIN Q := NEWC;

```



```

1386         WITH QZ DO
1387             BEGIN V := COPYV(A2,V); S11 := COPY(A2,S11); NEXT := P
1388             ENDI
1389             P := Q
1390             ENDI
1391             A := NEXT
1392             END (*- WITH A2 DO *-*)
1393             REDCOPY := P
1394             END (*- FUNCTION REDCOPY *-*)
1395
1396 BEGIN (*- FUNCTION INCLUSION *-*)
1397 IF B = NIL THEN INCLUSION := NIL ELSE
1398 WITH B2 DO
1399     IF NEXT = NIL THEN
1400         BEGIN PLB := PL(B2,V); A := REDCOPY(A);
1401             S := COPY(A); ROUND(A); DISPC(A);
1402             IF DEB1G THEN DEBPR(' NAKON ROUND', S);
1403             IF S = NIL THEN
1404                 BEGIN WRITELN(OUTPUT, ' NEHA MJESTA');
1405                     WRITELN(PRINT, ' NEHA MJESTA'); GOTO 13
1406                 ENDI
1407             WITH SZ, VZ DO
1408                 BEGIN PH := V; MIN := X; MINI := Y ENDI
1409                 D := S;
1410                 REPEAT
1411                     WITH DZ DO
1412                         BEGIN P := V;
1413                             REPEAT
1414                                 WITH PZ DO
1415                                     BEGIN
1416                                         IF (X < MIN) OR (X = MIN) AND (Y < MINI) THEN
1417                                             BEGIN MIN := X; MINI := Y; PH := P ENDI
1418                                         P := NEXT
1419                                         END (*- WITH PZ DO *-*)
1420                                     UNTIL P = V;
1421                                     D := NEXT
1422                                 END (*- WITH DZ DO *-*)
1423                             UNTIL D = NIL;
1424                             DX := PH2.X - VZ.X; DY := PH2.Y - VZ.Y; DISPC(S);
1425                             IF DEB1G THEN
1426                                 WRITELN(PRINT, ' DX,DY =', DX; 10, DY; 10);
1427                             S := COPY(S); TRANSLATE(S); INCLUSION := S
1428                             END (*- IF NEXT = NIL THEN *-*)
1429                             ELSE HALT
1430                             END (*- FUNCTION INCLUSION *-*)
1431
1432 PROCEDURE ERROR1 (I : INTEGER);
1433
1434 PROCEDURE ERROR1VAR F : TEXT;
1435
1436 BEGIN (*- PROCEDURE ERROR1 *-*)
1437     WRITELN(F); WRITELN(F); WRITE(F, ' GRESKA: ');
1438     CASE I OF
1439         1: WRITE(F, ' OČEKUJE SE '' I '' ILI '' , '' );
1440         2: WRITE(F, ' NEDOZVOLJENO IME');
1441         3: WRITE(F, ' OČEKUJE SE ZNAHENKA');
1442         4: WRITE(F, ' OČEKUJE SE '' ) '' );
1443         5: WRITE(F, ' OČEKUJE SE '' ( '' );
1444         6: WRITE(F, ' OČEKUJE SE '' , '' );
1445         7: WRITE(F, ' OČEKUJE SE '' I '' ILI '' ) '' );
1446         8: WRITE(F, ' POLIGON JE KRIVO ORIJENTIRAN');
1447         9: WRITE(F, ' OČEKUJE SE BROJ');
1448         10: WRITE(F, ' OČEKUJE SE '' = '' );

```

```

1449      11: WRITE(F, 'OCEKUJE SE FAKTOR')
1450      END (*- CASE I OF *)
1451      WRITEL(F)
1452      END (*- PROCEDURE ERRORI *)
1453
1454      BEGIN (*- PROCEDURE ERROR *)
1455      ERROR(OUTPUT); ERROR(PRINT); GOTO 13
1456      END (*- PROCEDURE ERROR *)
1457
1458      PROCEDURE NEXTCH
1459
1460      PROCEDURE GETCH(VAR F : TEXT)
1461
1462      LABEL
1463      11
1464
1465      BEGIN (*- PROCEDURE GETCH *)
1466      IF NOT FIRSTCALL THEN GET(F); FIRSTCALL := FALSE
1467      WHILE EOLN(F) DO
1468      BEGIN
1469      IF SPOREDNI THEN WRITELN(OUTPUT) ELSE WRITELN(OUTPUT, '---');
1470      WRITELN(PRINT); WRITE(PRINT, ' '); READLN(F);
1471      WRITE(OUTPUT, ' ');
1472      IF EOF(F) THEN
1473      IF SPOREDNI THEN
1474      BEGIN SPOREDNI := FALSE; CH := ' '; GOTO 1 END
1475      ELSE HALT
1476      END (*- WHILE EOLN(F) DO *)
1477      CH := F2; WRITE(PRINT, CH);
1478      IF SPOREDNI THEN WRITE(OUTPUT, CH);
1479      1 :
1480      END (*- PROCEDURE GETCH *)
1481
1482      BEGIN (*- PROCEDURE NEXTCH *)
1483      REPEAT
1484      IF SPOREDNI THEN GETCH(DATA) ELSE GETCH(INPUT)
1485      UNTIL CH <> ' '
1486      END (*- PROCEDURE NEXTCH *)
1487
1488      PROCEDURE INSYMBOL
1489
1490      VAR
1491      U          : ALFA;
1492      K, J       : INTEGER;
1493      Z, F      : REAL;
1494
1495      BEGIN (*- PROCEDURE INSYMBOL *)
1496      IF CH IN LETTERS THEN
1497      BEGIN W := ' '; K := 0;
1498      REPEAT K := K + 1;
1499      IF K <= ALFALENG THEN WCK := CH; NEXTCH
1500      UNTIL NOT (CH IN LETTERS + DIGITS);
1501      RTABCS,ID := W; J := RESWORDS;
1502      WHILE RTABCS,ID <> W DO J := J + 1;
1503      IF J = 0 THEN
1504      IF K = 1 THEN
1505      BEGIN SY := SKUP; VARCH := WCK END
1506      ELSE ERROR(2) ELSE
1507      WITH RTABCS DO
1508      BEGIN SY := SYN;
1509      CASE SY OF
1510      CONSTSY : CONSTPTR := S;
1511      FUNCSY, SPECSY : SELECT := 1

```

```

1512         END (*- CASE SY OF *-)
1513     END
1514     END (*- IF CH IN LETTERS THEN *-)
1515 ELSE
1516     IF CH IN DIGITS THEN
1517         BEGIN Z := 0;
1518             REPEAT Z := 10 * Z + ORD(CH) - ORD('0'); NEXTCH UNTIL
1519             NOT (CH IN DIGITS);
1520             IF CH = '.' THEN
1521                 BEGIN NEXTCH; F := 1;
1522                     IF NOT (CH IN DIGITS) THEN ERROR(3);
1523                     REPEAT Z := 10 * Z + ORD(CH) - ORD('0'); F := 10 * F;
1524                     NEXTCH
1525                     UNTIL NOT (CH IN DIGITS);
1526                     Z := Z / F;
1527             END (*- IF CH = '.' THEN *-);
1528             SY := BRUJ; XY := Z;
1529         END (*- IF CH IN DIGITS THEN *-)
1530     ELSE
1531         BEGIN SY := SYNSECCHS;
1532             IF SY <> PERIOD THEN NEXTCH;
1533         END
1534     END (*- PROCEDURE INSYMBOL *-);
1535
1536 PROCEDURE TEST(A : SYMBOL; E : INTEGER);
1537
1538 BEGIN (*- PROCEDURE TEST *-)
1539     IF SY = A THEN INSYMBOL ELSE ERROR(E);
1540 END (*- PROCEDURE TEST *-);
1541
1542 PROCEDURE STATEMENT;
1543
1544 PROCEDURE EXPRESS(VAR U : CPTR; VAR D : BOOLEAN);
1545
1546     VAR
1547         B, C          : CPTR;
1548         DB            : BOOLEAN;
1549
1550 PROCEDURE SIMPLEXPRESS(VAR U : CPTR; VAR D : BOOLEAN);
1551
1552     VAR
1553         B, C          : CPTR;
1554         DB            : BOOLEAN;
1555         OP            : SYMBOL;
1556
1557 PROCEDURE TERM(VAR U : CPTR; VAR D : BOOLEAN);
1558
1559     VAR
1560         B, C          : CPTR;
1561         DB            : BOOLEAN;
1562
1563 PROCEDURE FACTOR(VAR U : CPTR; VAR D : BOOLEAN);
1564
1565     VAR
1566         B              : CPTR;
1567         COMPLEMENT     : BOOLEAN;
1568         I              : INTEGER;
1569
1570 FUNCTION POLIGON : CPTR;
1571
1572     VAR
1573         A, W          : VPTR;
1574         B              : CPTR;

```

```

1575          N          : INTEGER;
1576
1577 FUNCTION RBROJ : REAL;
1578
1579 VAR
1580     Z          : REAL;
1581
1582 BEGIN (= FUNCTION RBROJ =)
1583     Z := 1;
1584     IF SY IN CPLUS, MINUS THEN
1585         BEGIN
1586             IF SY = MINUS THEN Z := - 1; INSYMBOL
1587         END;
1588     RBROJ := Z * XY; TEST(BROJ, 9);
1589 END (= FUNCTION RBROJ =);
1590
1591 BEGIN (= FUNCTION POLIGON =)
1592     TEST(LEFT, 5); A := NIL; N := 0;
1593     REPEAT W := NEW;
1594         WITH W2 DO
1595             BEGIN X := ROUND(FACT * RBROJ);
1596                 TEST(DIHA, 6);
1597                 Y := ROUND(FACT * HBROJ);
1598             END;
1599         IF A = NIL THEN
1600             BEGIN W2.NEXT := W; A := W END;
1601         ELSE
1602             BEGIN W2.NEXT := AZ.NEXT; AZ.NEXT := W; A := W END;
1603         N := N + 1;
1604         IF SY <> RIGHT THEN TEST(SEMICOLON, 7);
1605     UNTIL SY = RIGHT;
1606     INSYMBOL;
1607     IF N < 3 THEN
1608         BEGIN DISV(A); A := NIL END;
1609     IF A <> NIL THEN
1610         IF NOT POSITIVE(A) THEN ERROR(B); B := NEW;
1611     WITH B2 DO
1612         BEGIN V := A; NEXT := NIL; SIN := NIL END;
1613     POLIGON := B
1614 END (= FUNCTION POLIGON =);
1615
1616 BEGIN (= PROCEDURE FACTOR =)
1617     COMPLIMENT := SY = MINUS;
1618     IF COMPLIMENT THEN INSYMBOL;
1619     IF SY IN CCONSTSY, SKUP, LEFT, FUNCSY THEN
1620     CASE SY OF
1621     CONSTSY:
1622         BEGIN U := CONSTPTR; D := FALSE; INSYMBOL END;
1623     SKUP:
1624         BEGIN U := SETSCVARCHS; D := FALSE; INSYMBOL END;
1625     LEFT:
1626         BEGIN INSYMBOL; EXPRESS(U, D); TEST(RIGHT, 9) END;
1627     FUNCSY:
1628         BEGIN I := SELECT; INSYMBOL;
1629             CASE I OF
1630             I:
1631                 BEGIN U := POLIGON; D := TRUE END
1632             END (= CASE I OF =);
1633         END;
1634     END (= CASE SY OF =);
1635     ELSE ERROR(1);
1636     IF COMPLIMENT THEN
1637         BEGIN B := COMPL(U);

```

```

1638         IF D THEN DISPC(U); U := B; D := TRUE
1639     END
1640     END (= PROCEDURE FACTOR =);
1641
1642     BEGIN (= PROCEDURE TERM =)
1643     FACTOR(U, D);
1644     WHILE SY = STAR DO
1645     BEGIN INSYMBOL; FACTOR(B, DB); C := PRESEK(U, B);
1646     IF D THEN DISPC(U);
1647     IF DB THEN DISPC(B); U := C; D := TRUE
1648     END
1649     END (= PROCEDURE TERM =);
1650
1651     BEGIN (= PROCEDURE SIMPLEEXPRESS =)
1652     TERM(U, D);
1653     WHILE SY IN CPLUS, MINUSS DO
1654     BEGIN UP := SY; INSYMBOL; TERM(B, DB);
1655     IF OP = MINUS THEN C := DIFERENCA(U, B) ELSE C := UNIJAI(U, B);
1656     IF D THEN DISPC(U);
1657     IF DB THEN DISPC(B); U := C; D := TRUE
1658     END
1659     END (= PROCEDURE SIMPLEEXPRESS =);
1660
1661     BEGIN (= PROCEDURE EXPRESS =)
1662     SIMPLEEXPRESS(U, D);
1663     IF SY = INCL THEN
1664     BEGIN INSYMBOL; SIMPLEEXPRESS(B, DB); C := INCLUSION(U, B);
1665     IF D THEN DISPC(U);
1666     IF DB THEN DISPC(B); D := TRUE; U := C
1667     END
1668     END (= PROCEDURE EXPRESS =);
1669
1670     PROCEDURE ASSIGN;
1671
1672     VAR
1673     A           : LETTER;
1674     B           : CPTR;
1675     D           : BOOLEAN;
1676
1677     BEGIN (= PROCEDURE ASSIGN =)
1678     A := VARCH; INSYMBOL; TEST(ELSY, 10); EXPRESS(B, D);
1679     DISPC(SETSCAS);
1680     IF D THEN SETSCAS := B ELSE SETSCAS := COPY(B)
1681     END (= PROCEDURE ASSIGN =);
1682
1683     PROCEDURE SPECIAL; : INTEGER;
1684
1685     VAR
1686     K           : INTEGER;
1687     A           : CPTR;
1688     DA          : BOOLEAN;
1689
1690     BEGIN (= PROCEDURE SPECIAL =)
1691     CASE 1 OF
1692     1:
1693     BEGIN INSYMBOL; TEST(LEFT, 5); EXPRESS(A, DA); OUT(PRINT, A);
1694     OUT(OUTPUT, A);
1695     IF DA THEN DISPC(A); TEST(RIGHT, 4)
1696     END;
1697     2:
1698     BEGIN WRITELN(OUTPUT); WRITE(OUTPUT, ' '); SPONEDNI := TRUE;
1699     WRITELN(PRINT); WRITE(PRINT, ' '); FIRSTCALL := TRUE;
1700     INSYMBOL

```

```

1701     ENDI
1702     3:
1703     BEGIN INSYMBOL: TEST(ENSY, 10)
1704     IF SY = BROJ THEN
1705         BEGIN K := ROUND(XY); DEBUG := ODD(K); K := K DIV 2;
1706         DEBIG := ODD(K);
1707     END
1708     ELSE ERROR(9); INSYMBOL;
1709     END
1710     END (*- CASE 1 OF *-)
1711     END (*- PROCEDURE SPECIAL *-);
1712
1713     BEGIN (*- PROCEDURE STATEMENT *-)
1714     IF SY = SKUP THEN ASSIGN ELSE
1715     IF SY = SPECSY THEN SPECIAL(SELECT)
1716     END (*- PROCEDURE STATEMENT *-);
1717
1718     BEGIN (*- PROGRAM INTERPRETER *-)
1719     NEW(R2);
1720     WITH R2 DO
1721         BEGIN V := NIL; NEXT := NIL; SIN := NIL; END;
1722     EMPTY := NIL;
1723     WITH RTABC1S DO
1724         BEGIN ID := 'NIL' ; SYM := CONSTSY; S := EMPTY
1725     END;
1726     WITH RTABC2S DO
1727         BEGIN ID := 'R2' ; SYM := CONSTSY; S := R2
1728     END;
1729     WITH RTABC3S DO
1730         BEGIN ID := 'POLIGON' ; SYM := FUNCSY; I := 1; END;
1731     WITH RTABC4S DO
1732         BEGIN ID := 'WRITE' ; SYM := SPECSY; I := 1; END;
1733     WITH RTABC5S DO
1734         BEGIN ID := 'READ' ; SYM := SPECSY; I := 2; END;
1735     WITH RTABC6S DO
1736         BEGIN ID := 'TEST' ; SYM := SPECSY; I := 3; END;
1737     FOR I := FIRSTCH TO LASTCH DO SYMSCHR(I) := OT,ENSY;
1738     SYMSC1'S := SEMICOLON; SYMSC2'S := PERIOD; SYMSC3'S := ENSY;
1739     SYMSC4'S := PLUS; SYMSC5'S := MINUS; SYMSC6'S := STAR;
1740     SYMSC7'S := LEFT; SYMSC8'S := RIGHT; SYMSC9'S := COMMA;
1741     SYMSC10'S := INCL; LETLWS := 'A' .. 'Z';
1742     DIGITS := '0' .. '9';
1743     FREEV := NIL; FREEC := NIL; SPUREDNI := FALSE;
1744     FOR C := 'A' TO 'Z' DO SETSCCS := EMPTY; DEBUG := FALSE;
1745     DEBIG := FALSE; RESET(INPUT); REWRITE(OUTPUT); RESET(DATA);
1746     REWRITE(PRINT); FACT := FCTR;
1747     URITE(OUTPUT, ' '); WRITE(PRINT, ' ');
1748     FIRSTCALL := TRUE; NEXTCH := INSYMBOL;
1749     REPEAT STATEMENT;
1750     IF NOT (SY IN CSEMICOLON, PERIODS) THEN ERROR(1);
1751     IF SY = SEMICOLON THEN INSYMBOL;
1752     UNTIL SY = PERIOD;
1753     13:
1754     END (*- PROGRAM INTERPRETER *-);

```

0, RKPT PRINTS

```

ZTM*WORK(1).INTERPRETER
1  PROGRAM INTERPRETER(INPUT, OUTPUT, DATA, PRINT);
2
3  LABEL
4  13;
5
6  CONST
7  MAXP          = 179;
8  FCTR          = 1.0E3; (* & * SQRT(FCTR * MAXCOOR) < MAXINT *)
9  SQRTMAX       = 18500; (* = SQRT(MAXINT) *)
10 LENGTH        = 6; (* VISESTRUKOST ANITHETINE *)
11 RESWORDS     = 6;
12 FIRSTCH      = 0;
13 LASTCH       = 63;
14 QUOTE        = ''';
15 ALFALENG     = 10;
16
17 TYPE
18 VPTR          = 2 VRH;
19 CPTR          = 2 CVOR;
20 VRH           =
21 RECORD
22   X, Y : INTEGER;
23   NEXT : VPTR;
24 END (* RECORD -*);
25 CVOR         =
26 PACKED RECORD
27   V : VPTR;
28   NEXT, SIN : CPTR;
29 END (* RECORD -*);
30 INDEX        = 0 .. MAXP;
31 TABLE       = ARRAY CINDEXS OF
32 PACKED RECORD
33   X, Y : INTEGER;
34   S, H, K, L : INDEX;
35   GR, OR, AC, PG : BOOLEAN;
36 END (* RECORD -*);
37 ODNOS        = (IDENT, AINB, BINA, CROSS, DISJUNCT);
38 NISS         = PACKED ARRAY C1 .. 205 OF CHAR;
39 MULTI        = ARRAY C0 .. LENGTHS OF INTEGER;
40 SYMBOL       =
41 (SKIP, BRUJ, SEMICOLON, PERIOD, EQSY, PLUS, MINUS, STAR, INCL, LEFT,
42  RIGHT, COMMA, CONSTSY, FUNCSY, SPECSY, OTHERSY);
43 LETTER       = 'A' .. 'Z';
44 ALFA         = PACKED ARRAY C1 .. ALFALENGS OF CHAR;
45 ENTRY        =
46 RECORD
47   ID : ALFA;
48   CASE SYM : SYMBOL OF
49   CONSTSY :
50     (S : CPTR);
51   FUNCSY, SPECSY :
52     (I : INTEGER);
53 END (* RECORD -*);
54
55 VAR
56 FREEV        : VPTR;
57 FREEC        : CPTR;
58 NUMT         : INDEX;
59 TAB          : TABLE;
60 PRINT        : TEXT;
61 DEBUG, DEBIG : BOOLEAN;
62 FACT         : REAL;

```

```

63      NUMC          : INDEX;
64      NUMOR        : INDEX;
65      XTAB         : ARRAY CINDEXS OF INDEX;
66      VARCH        : LETTER;
67      XY           : REAL;
68      CONSTPTR     : CPTR;
69      SELECT       : INTEGER;
70      SY           : SYMBOL;
71      CH           : CHAR;
72      RTAB         : ARRAY CO .. RESLORDS OF ENTRY;
73      SYMS         : ARRAY CCHARS OF SYMBOL;
74      LETTERS, DIGITS : SET OF 'A' .. '9';
75      R2, EMPTY    : CPTR;
76      SETS         : ARRAY CLETTERS OF CPTR;
77      SPUREDN1     : BOOLEAN;
78      DATA        : TEXT;
79      I            : INTEGER;
80      C            : CHAR;
81      FIRSTCALL    : BOOLEAN;
82
83      FUNCTION NEWV : VPTR;
84
85      VAR
86      V            : VPTR;
87
88      BEGIN (= FUNCTION NEWV =)
89      IF FREEV = NIL THEN NEW(V) ELSE
90      BEGIN V := FREEV; FREEV := V2, NEXT END;
91      NEWV := V;
92      END (= FUNCTION NEWV =);
93
94      FUNCTION NEWC : CPTR;
95
96      VAR
97      C            : CPTR;
98
99      BEGIN (= FUNCTION NEWC =)
100     IF FREEC = NIL THEN NEW(C) ELSE
101     BEGIN C := FREEC; FREEC := C2, NEXT END;
102     NEWC := C;
103     END (= FUNCTION NEWC =);
104
105     PROCEDURE DISPV(V : VPTR);
106
107     VAR
108     W            : VPTR;
109
110     BEGIN (= PROCEDURE DISPV =)
111     IF V <> NIL THEN
112     BEGIN W := V2, NEXT; V2, NEXT := FREEV; FREEV := W END;
113     END (= PROCEDURE DISPV =);
114
115     PROCEDURE DISPC(C : CPTR);
116
117     VAR
118     D            : CPTR;
119
120     BEGIN (= PROCEDURE DISPC =)
121     WHILE C <> NIL DO
122     WITH C2 DO
123     BEGIN DISPC(SINH); DISPV(V); D := NEXT; NEXT := FREEC;
124     FREEC := C; C := D;
125     END

```



```

126      END (= PROCEDURE DISPC =)
127
128      FUNCTION COPYV(A : VPTR) : VPTR
129
130      VAR
131      P, Q, R, S      : VPTR
132
133      BEGIN (= FUNCTION COPYV =)
134      IF A <> NIL THEN
135      BEGIN P := NEWV; Q := P; PZ,X := AZ,X; PZ,Y := AZ,Y;
136      R := AZ,NEXT;
137      REPEAT S := NEWV; SZ,X := RZ,X; SZ,Y := RZ,Y; QZ,NEXT := S;
138      Q := S; R := RZ,NEXT;
139      UNTIL R = A;
140      QZ,NEXT := P;
141      END (= IF A <> NIL THEN =)
142      ELSE P := NIL; COPYV := P;
143      END (= FUNCTION COPYV =);
144
145      FUNCTION COPY(A : CPTR) : CPTR
146
147      VAR
148      P, Q              : CPTR
149
150      BEGIN (= FUNCTION COPY =)
151      P := NIL;
152      WHILE A <> NIL DO
153      WITH AZ DO
154      BEGIN Q := NEWV;
155      WITH QZ DO
156      BEGIN V := COPYV(AZ,V); S := COPY(AZ,S); NEXT := P;
157      END;
158      P := Q; A := NEXT;
159      END (= WITH AZ DO =);
160      COPY := P;
161      END (= FUNCTION COPY =);
162
163      FUNCTION CONC(A, B : CPTR) : CPTR
164
165      VAR
166      P              : CPTR
167
168      BEGIN (= FUNCTION CONC =)
169      IF A = NIL THEN CONC := B ELSE
170      BEGIN P := A;
171      WHILE PZ,NEXT <> NIL DO P := PZ,NEXT; PZ,NEXT := B;
172      CONC := A;
173      END;
174      END (= FUNCTION CONC =);
175
176      FUNCTION SQUEEZEP(V : VPTR) : VPTR
177
178      VAR
179      Q, QZ, R, S      : VPTR
180      DELETE           : BOOLEAN
181
182      BEGIN (= FUNCTION SQUEEZEP =)
183      Q := V; QZ := V;
184      IF Q <> NIL THEN
185      BEGIN
186      REPEAT R := QZ,NEXT; S := RZ,NEXT;
187      DELETE := QZ,X * (RZ,Y - SZ,Y) +
188      RZ,X * (SZ,Y - QZ,Y) +

```

```

189           S2,X * (Q2,Y - R2,Y) = 01
190       IF DELETE THEN
191           BEGIN Q2,NEXT := S; Q2 := Q; R2,NEXT := FREEV;
192           FREEV := R
193           END
194       ELSE Q := R
195       UNTIL (Q = Q2) AND NOT DELETE OR (Q2,NEXT = Q);
196       IF Q2,NEXT = Q THEN
197           BEGIN Q2,NEXT := FREEV; FREEV := Q; Q := NIL; END
198       END;
199       SQUEEZEP := Q
200   END (= FUNCTION SQUEEZEP =);
201
202   FUNCTION SQUEEZE(A : CPTR) : CPTR;
203
204   VAR
205       P, P1, X      : CPTR;
206
207   BEGIN (= FUNCTION SQUEEZE =)
208       P := A;
209       WHILE P <> NIL DO
210           WITH P2 DO
211               BEGIN V := SQUEEZEP(V);
212                   IF V = NIL THEN
213                       BEGIN X := P; P := X2,NEXT;
214                           X2,NEXT := NIL;
215                           IF X = A THEN A := P ELSE P2,NEXT := P; DISPC(X)
216                           END
217                       ELSE
218                           BEGIN S1N := SQUEEZE(S1N); P1 := P; P := NEXT
219                           END
220                       END (= WITH P2 DO =);
221                   SQUEEZE := A
222               END (= FUNCTION SQUEEZE =);
223
224   FUNCTION POSITIVE(P : VPTR) : BOOLEAN;
225
226   VAR
227       A, B, C      : VPTR;
228       S            : INTEGER;
229
230   BEGIN (= FUNCTION POSITIVE =)
231       A := P; B := P2,NEXT; S := 0;
232       REPEAT C := B2,NEXT; S := S + B2,X * (C2,Y - A2,Y);
233           A := B; B := C
234       UNTIL A = P;
235       POSITIVE := S > 0
236   END (= FUNCTION POSITIVE =);
237
238   PROCEDURE OUTTAB;
239
240   VAR
241       I      : INDEX;
242
243   BEGIN (= PROCEDURE OUTTAB =)
244       FOR I := 1 TO NUNT DO
245           WITH TABCIS DO
246               WRITELN(PRINT, I : 4, K : 4, L : 4, X : 16, Y : 16,
247                   U : 4, S : 4, GR : 6, UN : 6, AC : 6, PG : 6);
248               WRITELN(PRINT); WRITELN(PRINT); WRITE(PRINT, ' ');
249           END (= PROCEDURE OUTTAB =);
250
251   PROCEDURE FILL(A, B : VPTR; VAR UD : ODNO; CA, CB : BOOLEAN);

```

```

252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314

```

```

VAR
  I          : INDEX;

BEGIN (= PROCEDURE OUTTAB =)
  FOR I := 1 TO NUHT DO
    WITH TABCIS DO
      WRITELN(PRINT, I : 4, K : 4, L : 4, X : 10, Y : 10,
        N : 4, S : 4, GR : 6, UN : 6, AC : 6, PG : 6);
      WRITELN(PRINT); WRITELN(PRINT); WRITE(PRINT, ' ');
    END (= PROCEDURE OUTTAB =);

PROCEDURE FILL(A, B : VPTR; VAR OD : ODNO; CA, CB : BOOLEAN);

VAR
  IA, IB     : INDEX;
  W         : INTEGER;

FUNCTION NEWTIX, YY : INTEGER; SS : INDEX; GG : BOOLEAN) : INDEX;

BEGIN (= FUNCTION NEWT =)
  IF NUHT >= MAXP THEN
    BEGIN WRITELN(OUTPUT, ' TABLE FULL');
      WRITELN(PRINT, ' TABLE FULL');
      OUTTAB; GOTO I;
    END;
  NUHT := NUHT + 1;
  WITH TABCIS DO
    BEGIN X := XX; Y := YY; S := SS; GR := GG; UN := FALSE;
      K := 0; L := 0; AC := FALSE; PG := FALSE;
    END;
  NEWT := NUHT;
END (= FUNCTION NEWT =);

PROCEDURE ENTER(A : VPTR; C : BOOLEAN; VAR IA : INDEX);

VAR
  I, IB     : INDEX;
  P         : VPTR;

BEGIN (= PROCEDURE ENTER =)
  P := A;
  REPEAT
    WITH P2 DO
      BEGIN
        I := NEWTIX, Y, U, FALSE;
        IF P = A THEN
          BEGIN IA := I; IB := I; END;
        ELSE
          BEGIN
            IF C THEN TABCIS.N := IB;
              ELSE TABCIS.N := I;
            IB := I;
          END;
        P := NEXT;
      END;
  UNTIL P = A;
  IF C THEN TABCIAS.N := IB;
    ELSE TABCIBS.N := IA;
  I := IA;
  REPEAT
    WITH TABCIS DO

```

```

315     BEGIN K := 1; L := N; I := 4; END
316 UNTIL I = 1;
317 END (*- PROCEDURE ENTERP *-);
318
319 PROCEDURE HOVL;
320
321     BEGIN (*- PROCEDURE HOVL *-)
322     WRITELN(OUTPUT, 'MULTI INTEGER OVERFLOW');
323     WRITELN(PRINT, 'MULTI INTEGER OVERFLOW');
324     GOTO 13;
325     END (*- PROCEDURE HOVL *-);
326
327 PROCEDURE CHS(VAR A : MULTI);
328
329     VAR
330     I, P      : INTEGER;
331
332     BEGIN (*- PROCEDURE CHS *-)
333     P := 1;
334     FOR I := 0 TO LENGTH DO
335     BEGIN P := SQRTMAX - 1 + P - ACIS;
336     ACIS := P MOD SQRTMAX; P := P DIV SQRTMAX;
337     END;
338     END (*- PROCEDURE CHS *-);
339
340 PROCEDURE HADD(A, B : MULTI; VAR C : MULTI);
341
342     VAR
343     I, P      : INTEGER;
344
345     BEGIN (*- PROCEDURE HADD *-)
346     P := 0;
347     FOR I := 0 TO LENGTH DO
348     BEGIN P := ACIS + BCIS + PI;
349     CCIS := P MOD SQRTMAX; P := P DIV SQRTMAX;
350     END;
351     IF (CLENGTHS > 0) AND (CLENGTHS < SQRTMAX - 1)
352     THEN HOVL;
353     END (*- PROCEDURE HADD *-);
354
355 PROCEDURE HSUB(A, B : MULTI; VAR C : MULTI);
356
357     BEGIN (*- PROCEDURE HSUB *-)
358     CHS(B); HADD(A, B, C);
359     END (*- PROCEDURE HSUB *-);
360
361 PROCEDURE HMUL(A, B : MULTI; VAR C : MULTI);
362
363     VAR
364     I, J, P    : INTEGER;
365     SA, SB     : BOOLEAN;
366
367     BEGIN (*- PROCEDURE HMUL *-)
368     SA := ACLENGTHS > 0; SB := BCLENGTHS > 0;
369     IF SA THEN CHS(A); IF SB THEN CHS(B);
370     FOR I := 0 TO LENGTH DO CCIS := 0;
371     FOR J := 0 TO LENGTH DO
372     BEGIN P := 0;
373     FOR J := 0 TO LENGTH DO
374     BEGIN P := ACIS + BCJS + PI;
375     IF I + J < LENGTH THEN
376     BEGIN P := P + CCJ + JS;
377     CCJ + JS := P MOD SQRTMAX; P := P DIV SQRTMAX;

```

```

378         END
379         ELSE IF P > 0 THEN p := i
380         ENDI
381         IF P > 0 THEN MOVL
382         ENDI
383         IF SA <> SB THEN CHS(C)
384         END (*= PROCEDURE MMUL *)
385
386 PROCEDURE HSET(VAR A : MULTI; X : INTEGER)
387
388     VAR
389     I, P          : INTEGER;
390
391     BEGIN (*= PROCEDURE HSET *)
392     P := ABS(X);
393     FOR I := 0 TO LENGTH DO
394     BEGIN
395     ACIS := P MOD SQRTMAX; P := P DIV SQRTMAX
396     ENDI
397     IF X < 0 THEN CHS(A)
398     END (*= PROCEDURE HSET *)
399
400 FUNCTION SIGN(VAR A : MULTI): INTEGER
401
402     VAR
403     I          : INTEGER;
404
405     BEGIN (*= FUNCTION SIGN *)
406     IF ACLENGTHS > 0 THEN SIGN := - 1 ELSE
407     BEGIN I := LENGTH;
408     WHILE (I > 0) AND (ACIS = 0) DO I := I - 1;
409     SIGN := ORD(ACIS > 0)
410     END
411     END (*= FUNCTION SIGN *)
412
413 FUNCTION QL(I, J, K : INDEX) : INTEGER
414
415     BEGIN (*= FUNCTION QL *)
416     QL := TABCIS.X * (TABCJS.Y - TABCKS.Y) + TABCJB.X * (TABCKS.Y -
417     TABCIS.Y) + TABCKS.X * (TABCIS.Y - TABCJS.Y)
418     END (*= FUNCTION QL *)
419
420 PROCEDURE CUT(A, B: INDEX)
421
422     VAR
423     I, J1, J2, J3, J4, J5, J6, L, M: INDEX;
424     S1, S2, S3, S4, XA, YA,
425     XB, YB, P1, P2, Q1, Q2: INTEGER;
426     CN          : BOOLEAN;
427
428     PROCEDURE SORT(L, R: INDEX)
429
430     VAR
431     I, J          : INDEX;
432     X, M          : INDEX;
433
434     FUNCTION LESS(VAR I, J: INDEX): BOOLEAN
435
436     TYPE
437     M          = ARRAY CO .. JS OF INTEGER;
438
439     VAR
440     C, D          : M;

```

```

441
442 PROCEDURE MUL(AA, BB: INTEGER; VAR C: M);
443
444 VAR
445 I, J      : 0 .. 3;
446 A, B     : ARRAY CO .. IS OF INTEGER;
447 X, P     : INTEGER;
448
449 BEGIN (= PROCEDURE MUL =)
450 ACIS := AA DIV SURTHAX; ACUS := AA MOD SURTHAX;
451 BCIS := BB DIV SURTHAX; BCUS := BB MOD SURTHAX;
452 FOR I := 0 TO 3 DO CCIS := 0;
453 FOR J := 0 TO 3 DO
454 BEGIN P := 0;
455 FOR I := 0 TO 3 DO
456 BEGIN
457 P := P + ACIS * BCJS + CCJ * JSI;
458 CCJ * JSI := P MOD SURTHAX; P := P DIV SURTHAX;
459 END;
460 CCJ * JSI := P;
461 END;
462 END (= PROCEDURE MUL =);
463
464 BEGIN (= FUNCTION LESS =)
465 IF TABCIS.K = TABCJS.A THEN
466 BEGIN
467 MUL(TABCIS.X, TABCJS.Y, C); MUL(TABCIS.Y, TABCJS.A, D);
468 IF CCJS <> DCJS THEN LESS := CCJS < DCJS ELSE
469 IF CC2S <> DC2S THEN LESS := CC2S < DC2S ELSE
470 IF CCIS <> DCIS THEN LESS := CCIS < DCIS ELSE
471 LESS := CCOS < DCOS;
472 END;
473 ELSE LESS := TABCIS.A < TABCJS.K;
474 END (= FUNCTION LESS =);
475
476 BEGIN (= PROCEDURE SORT =)
477 I := L; J := R; X := XTABC(L+R) DIV 2S;
478 REPEAT
479 WHILE LESS(XTABCIS, X) DO I := I + 1;
480 WHILE LESS(X, XTABCJS) DO J := J - 1;
481 IF I < J THEN
482 BEGIN W := XTABCIS; XTABCIS := XTABCJS; XTABCJS := W;
483 I := I + 1; J := J - 1;
484 END;
485 UNTIL I > J;
486 IF L < J THEN SORT(L, J);
487 IF I < K THEN SORT(I, R);
488 END (= PROCEDURE SORT =);
489
490 BEGIN (= PROCEDURE CUT =)
491 I := IA;
492 REPEAT J1 := TABCIS.H; J := IB;
493 REPEAT J1 := TABCJS.H; CR := FALSE;
494 S1 := GL(I, I1, J1); S3 := GL(J, J1, I);
495 S2 := GL(I, I1, J); S4 := GL(J, J1, I);
496 IF S1 = 0 THEN
497 WITH TABCIS DO
498 BEGIN
499 XA := TABCIS.X - XI; YA := TABCIS.Y - YI;
500 XB := TABCJS.X - XI; YB := TABCJS.Y - YI;
501 PI := XA * XB + YA * YB;
502 QI := SQH(XA) + SQH(YA);
503 IF (PI >= 0) AND (PI < QI) THEN

```

```

504         BEGIN CR := TRUE; P2 := 0; Q2 := 1 END
505     END
506 ELSE IF S3 = 0 THEN
507     WITH TABCJS DO
508     BEGIN
509         XA := TABCJS.X - XI; YA := TABCJS.Y - YI
510         XB := TABCIS.X - XI; YB := TABCIS.Y - YI
511         P2 := XA * XB + YA * YB;
512         Q2 := SQR(XA) + SQR(YA);
513         IF (P2 >= 0) AND (P2 < Q2) THEN
514             BEGIN CR := TRUE; P1 := 0; Q1 := 1 END
515         END
516     ELSE
517     IF (S1 > 0) AND (S2 < 0) AND (S3 < 0) AND (S4 > 0) OR
518     (S1 < 0) AND (S2 > 0) AND (S3 > 0) AND (S4 < 0) THEN
519     BEGIN CR := TRUE;
520     P1 := S3; Q1 := S3 - S4;
521     IF Q1 < 0 THEN BEGIN P1 := - P1; Q1 := - Q1 END;
522     P2 := S1; Q2 := S1 - S2;
523     IF Q2 < 0 THEN BEGIN P2 := - P2; Q2 := - Q2 END
524     END;
525     IF CR THEN
526     BEGIN
527     OD := CROSS;
528     IF P1 = 0 THEN I2 := I ELSE
529     BEGIN I2 := NEXT(P1, Q1, 0, FALSE);
530     TABCIS.K := I; TABCIS.L := I;
531     NUMC := NUMC + 1; XTABNUMCS := I2
532     END;
533     IF P2 = 0 THEN J2 := J ELSE
534     BEGIN J2 := NEXT(P2, Q2, 0, FALSE);
535     TABCJS.K := J; TABCJS.L := J;
536     NUMC := NUMC + 1; XTABNUMCS := J2
537     END;
538
539     IF TABCIS.GR OR TABCJS.GH THEN
540     BEGIN
541     WRITELN(OUTPUT, ' ZAPETLJAH CUT, I, J = ', I, ' : ', S, J, ' : ', S);
542     WRITELN(PRINT, ' ZAPETLJAH CUT, I, J = ', I, ' : ', S, J, ' : ', S);
543     OUTTAB; GOTO 13
544     END;
545
546     WITH TABCIS DO
547     BEGIN GR := TRUE; S := J2;
548     IF (S1 <> 0) AND (S2 <> 0) AND (S4 <> 0) THEN
549     BEGIN AC := TRUE; UN := S4 > 0 END
550     END;
551     WITH TABCJS DO
552     BEGIN GR := TRUE; S := I2;
553     IF (S2 <> 0) AND (S3 <> 0) AND (S4 <> 0) THEN
554     BEGIN AC := TRUE; UN := S2 > 0 END
555     END
556     END IF CR THEN
557     J := J;
558     UNTIL J = IB;
559     I := I;
560     UNTIL I = IA;
561
562     IF NUMC > 0 THEN
563     BEGIN
564     SORT(I, NUMC);
565
566     IF DEBUG THEN

```

```

567 BEGIN
568 WRITELN(PRINT, ' AFTER SORT' );
569 OUTTAB
570 END;
571
572 TABCJS,K := 0; XTABCS := 0; L := NUMC; H := XTABCL;
573 WHILE TABCS,K <> 0 DO
574 BEGIN I := TABCS,K; I1 := TABCS,H; I2 := I;
575 REPEAT
576 WITH TABCS DO
577 BEGIN I2 := H;
578 P1 := X; X := Y - X; Y := P1;
579 H := I1; I3 := I2;
580 END;
581 L := L - 1; H := XTABCL;
582 UNTIL TABCS,K <> 0;
583 TABCS,H := I2;
584 END;
585 END;
586
587 IF DEBUG THEN
588 BEGIN
589 WRITELN(PRINT, ' END OF CUT' );
590 OUTTAB
591 END;
592 END (= PROCEDURE CUT =);
593
594 FUNCTION IMPOL(I, IB: INDEX): INTEGER;
595 LABEL
596 I;
597
598
599 VAR
600 J, J1 : INDEX;
601 DY0, DY1, DX1, DX2, P, U,
602 XX, YY, NS : INTEGER;
603
604 BEGIN (= FUNCTION IMPOL =)
605 WITH TABCS DO
606 BEGIN XX := TABCS,X + X; YY := TABCS,Y + Y END;
607 REPEAT
608 IB := TABCS,N; DY0 := 2 * TABCS,Y - YY;
609 UNTIL DY0 <> 0;
610 J := INT IMPOL := 0; NS := 0;
611 REPEAT
612 J1 := TABCS,N; DY1 := 2 * TABCS,Y - YY;
613 IF DY1 = 0 THEN
614 BEGIN DX1 := 2 * TABCS,X - XX;
615 IF DX1 = 0 THEN GOTO 1;
616 REPEAT
617 J1 := TABCS,N; DY1 := 2 * TABCS,Y - YY;
618 IF DY1 = 0 THEN
619 BEGIN DX2 := 2 * TABCS,X - XX;
620 IF (DX2 = 0) OR ((DX1 < 0) <> (DX2 < 0)) THEN GOTO 1;
621 END;
622 UNTIL DY1 <> 0;
623 IF DX1 > 0 THEN
624 IF (DY0 < 0) <> (DY1 < 0) THEN NS := NS + 1;
625 END;
626 ELSE
627 IF (DY0 < 0) <> (DY1 < 0) THEN
628 BEGIN
629 P := TABCS,X + DY1 - TABCS,X + DY0;

```



```

630         Q := TABCJIS.Y - TABCJS.YI
631         IF Q < 0 THEN
632             BEGIN P := - P; Q := - Q ENDI
633         IF P = Q * XX THEN GOTO I1
634         IF P > Q * XX THEN NS := NS + 1
635         ENDI
636         J := J1; DYO := DYI
637         UNTIL J = IB;
638         INPOL := 2 * OKD(ODD(NS)) - 1;
639         I;
640     END (* FUNCTION INPOL *)
641
642 PROCEDURE DEFPRIA; IB: INDEX;
643
644     VAR
645         I, I1          : INDEX;
646         W              : INTEGER;
647         XX, YY        : REAL;
648         DF, PR        : BOOLEAN;
649
650     FUNCTION INPOL(I, IB: INDEX): INTEGER;
651
652     LABEL
653     I;
654
655     VAR
656         J, J1, I1, I2 : INDEX;
657         P1, P2, Q1, Q2, NS : INTEGER;
658         FCT, P, Q, W, XX, YY, DYO, DY1, DX1, DX2 : MULT;
659
660     BEGIN (* FUNCTION INPOL *)
661         WITH TABCIS DO
662             BEGIN I1 := I; I2 := I;
663                 IF I > NUNOR THEN BEGIN P1 := X; Q1 := Y END;
664                     ELSE BEGIN P1 := 1; Q1 := 0 END;
665             WITH TABCIS DO
666                 IF TABCIS.I > NUNOR THEN BEGIN P2 := X; Q2 := Y END;
667                     ELSE BEGIN P2 := 0; Q2 := 1 END;
668             END;
669
670             HSET(FCT, P1 + P2 + Q1 + Q2);
671             HSET(P, P1 + P2); HSET(Q, Q1 + Q2);
672             HSET(XX, TABCIS.X); HSET(W, TABCIS.W);
673             MMUL(XX, P, XX); MMUL(W, W, W); MADD(XX, W, XX);
674             HSET(YY, TABCIS.Y); HSET(W, TABCIS.W);
675             MMUL(YY, P, YY); MMUL(W, W, W); MADD(YY, W, YY);
676
677             REPEAT
678                 IB := TABCIS.L;
679                 HSET(DYO, TABCIS.Y); MMUL(FCT, DYO, DYO);
680                 HSUB(DYO, YY, DYO);
681             UNTIL SIGN(DYO) <> 0;
682             J := IB; INPOL := 0; NS := 0;
683             REPEAT
684                 J1 := TABCIS.L;
685                 HSET(DY1, TABCIS.Y); MMUL(FCT, DY1, DY1);
686                 HSUB(DY1, YY, DY1);
687             IF SIGN(DY1) = 0 THEN
688                 BEGIN
689                     HSET(DX1, TABCIS.X); MMUL(FCT, DX1, DX1);
690                     HSUB(DX1, XX, DX1);
691                     IF SIGN(DX1) = 0 THEN GOTO I1;
692                     REPEAT

```

```

693      J1 := TABCJIS.L1;
694      NSET(DY1, TABCJIS.Y1); MMUL(FCT, DY1, DY1);
695      MSUB(DY1, YY, :Y1);
696      IF SIGN(DY1) = 0 THEN
697      BEGIN NSET(DX2, TABCJIS.X1);
698      MMUL(FCT, DX2, DX2); MSUB(DX2, XX, DX2);
699      IF (SIGN(DX2) = 0) OR
700      ((SIGN(DX1) < 0) <> (SIGN(DX2) < 0)) THEN GOTO 1
701      END
702      UNTIL SIGN(DY1) <> 0;
703      IF SIGN(DX1) > 0 THEN
704      IF (SIGN(DY0) < 0) <> (SIGN(DY1) < 0) THEN NS := NS + 1
705      END
706      ELSE
707      IF (SIGN(DY0) < 0) <> (SIGN(DY1) < 0) THEN
708      BEGIN
709      NSET(P, TABCJIS.X1); NSET(Q, TABCJIS.X1);
710      MMUL(P, DY1, P); MMUL(Q, DY0, Q);
711      IF TABCJIS.Y >= TABCJIS.Y THEN MSUB(P, L, P);
712      ELSE MSUB(Q, P, Q);
713      NSET(Q, ABS(TABCJIS.Y - TABCJIS.Y));
714      MMUL(Q, XX, Q); MSUB(P, L, P);
715      IF SIGN(P) = 0 THEN GOTO 1;
716      IF SIGN(P) > 0 THEN NS := NS + 1
717      END;
718
719      IF DEBUG THEN
720      WRITELN(PRINT, ' INPOL: I, J, NS = ', I : 5, J : 5, NS : 5);
721      J := J1; DY0 := DY1;
722      UNTIL J = IB;
723      INPOL := 2 * JRD(ODD(NS)) - 1;
724      I;
725      END (*- FUNCTION INPOL *-);
726
727      BEGIN (*- PROCEDURE DEFN *-)
728      I := IA; DF := FALSE;
729      REPEAT
730      WITH TABCIS DO
731      IF AC THEN DF := TRUE; ELSE
732      IF GR THEN
733      BEGIN W := INPOL(I, IB);
734      IF W <> 0 THEN
735      BEGIN AC := TRUE; UN := (W < 0) = CB; DF := TRUE; END;
736      END;
737      I := TABCIS.N;
738      UNTIL I = IA;
739      IF DEBUG THEN
740      BEGIN
741      WRITELN(PRINT, ' AFTER INPOL');
742      OUTTAB
743      END;
744
745      IF DF THEN
746      BEGIN
747      WHILE NOT TABCIS.AC DO IA := TABCIS.N;
748      I := IA;
749      REPEAT
750      WITH TABCIS DO
751      IF AC THEN PR := UN; ELSE UN := PR;
752      I := TABCIS.N;
753      UNTIL I = IA;
754
755      REPEAT II := TABCIS.N;

```

```

756      WITH TABCIS DO
757      BEGIN
758      AC := UN <> TARCIS.UNI
759      IF AC AND NOT GR THEN
760      BEGIN
761      WRITELN(OUTPUT, ' DEFPR, II = ', II : 5);
762      WRITELN(PRINT, ' DEFPR, II = ', II : 5);
763      OUTTAB: GOTO 13
764      END
765      END;
766      I := II
767      UNTIL I = IAI
768
769      I := IBI
770      REPEAT I := TABCIS.W UNTIL (I = IBI) OR TABCIS.AC;
771      IF TABCIS.AC THEN PR := TARCIS.UN ELSE
772      BEGIN
773      REPEAT I := TABCIS.NI; W := INQOL(I, IAI)
774      UNTIL (W <> 0) OR (I = IBI);
775      IF W = 0 THEN
776      BEGIN
777      WRITELN(OUTPUT, ' DEFPR, IB = ', IB : 5);
778      WRITELN(PRINT, ' DEFPR, IB = ', IB : 5);
779      OUTTAB: GOTO 13
780      END;
781      PR := (W < 0) = CA
782      END;
783      IB := I;
784      REPEAT TABCIS.UN := PR; I := TABCIS.NI
785      WITH TABCIS DO
786      IF GR THEN
787      IF TABCIS.AC THEN PR := NOT PR
788      UNTIL I = IBI;
789
790      FOR I := 1 TO NUMC DO
791      WITH TABXTABCIS DO
792      BEGIN
793      XX := X / (X + Y); YY := Y / (X + Y);
794      X := ROUND((XX * TABCIS.X + YY * TABCIS.X));
795      Y := ROUND((XX * TABCIS.Y + YY * TABCIS.Y));
796      END
797      END (* IF DF THEN *)
798      ELSE OD := IDENT
799      END (* PROCEDURE DEFPR *)
800
801      PROCEDURE EVICT(A: INDEX);
802
803      VAR
804      I: II : INDEX;
805
806      BEGIN (* PROCEDURE EVICT *)
807      I := IAI
808      REPEAT
809      II := TABCIS.NI
810      IF TABCIS.GR AND TABCIS.GR THEN
811      IF TABXTABCIS.SS.N = TABCIS.S THEN
812      IF TABCIS.UN = TABXTABCIS.SS.UN THEN
813      BEGIN TABCIS.PG := TRUE; TABXTABCIS.SS.PG := TRUE END;
814      I := II
815      UNTIL I = IAI
816      END (* PROCEDURE EVICT *)
817
818      BEGIN (* PROCEDURE FILL *)

```

```

819      NUMT := 0; NUMC := 0;
820      ENTERP(A, CA, IA); ENTERP(B, CB, IB); NUMOR := NUMT;
821      OD := DISJUNKT; CUT(A, B);
822
823      IF OD = CROSS THEN
824      BEGIN DEPR(A, B); EVICT(A); END
825      ELSE
826      BEGIN W := INPOL(A, B);
827      IF W = 0 THEN
828      BEGIN
829      WRITELN(OUTPUT, ' BAD INPOL, IA, IB = ', IA, ' ', IB, ' ');
830      WRITELN(PRINT, ' BAD INPOL, IA, IB = ', IA, ' ', IB, ' ');
831      OUTTAB; GOTO 13;
832      END;
833      IF W > 0 THEN OD := A AND B ELSE
834      BEGIN W := INPOL(B, A);
835      IF W = 0 THEN
836      BEGIN
837      WRITELN(OUTPUT, ' BAD INPOL, IB, IA = ', IB, ' ', IA, ' ');
838      WRITELN(PRINT, ' BAD INPOL, IB, IA = ', IB, ' ', IA, ' ');
839      OUTTAB; GOTO 13;
840      END;
841      IF W > 0 THEN OD := B AND A;
842      END;
843      END;
844
845      IF DEBUG THEN
846      BEGIN
847      WRITELN(PRINT, ' AFTER FILL, IA, IB = ', IA, ' ', IB, ' ');
848      OUTTAB;
849      END;
850      END (*- PROCEDURE FILL -*);
851
852      FUNCTION EXCP(PR : BOOLEAN) : CPTR;
853
854      VAR
855      C, D          : CPTR;
856      W, U          : VPTR;
857      FOUND        : BOOLEAN;
858
859      FUNCTION FOLLOW(PR : BOOLEAN) : VPTR;
860
861      VAR
862      I, JA        : INDEX;
863      FOUND, ERROR : BOOLEAN;
864      LAST, J      : VPTR;
865
866      BEGIN (*- FUNCTION FOLLOW -*);
867      IA := 0;
868      REPEAT I := IA + 1
869      WITH TABCIAS DO FOUND := NOT PG AND (UN = PR)
870      UNTIL FOUND OR (IA = NUMT);
871      IF DEBUG THEN
872      BEGIN
873      WRITELN(PRINT, ' START OF FOLLOW, IA = ', IA, ' ');
874      OUTTAB;
875      END;
876      LAST := NIL;
877      IF FOUND THEN
878      BEGIN I := IA;
879      REPEAT
880      WITH TABCIAS DO
881      BEGIN

```

```

882      W := NEWV; WZ,X := X; WZ,Y := Y;
883      IF LAST = NIL THEN
884      BEGIN WZ,NEXT := W; LAST := W END
885      ELSE
886      BEGIN WZ,NEXT := LASTZ,NEXT; LASTZ,NEXT := W; LAST := W
887      END;
888      I := J
889      END (=- WITH TABCIS DO =-)
890      ERROR := FALSE;
891      WITH TABCIS DO
892      IF PG OR (UJ <> PR) THEN
893      IF OR THEN
894      BEGIN I := S;
895      WITH TABCIS DO
896      IF PG OR (UJ <> PR) THEN ERROR := TRUE
897      END
898      ELSE ERROR := TRUE;
899      WITH TABCIS DO
900      IF ERROR THEN
901      BEGIN
902      WRITELN(OUTPUT, ' ZAPETLJAN FOLLOW, IA, I = ', IA : 5, I : 5);
903      WRITELN(PRINT, ' ZAPETLJAN FOLLOW, IA, I = ', IA : 5, I : 5);
904      OUTTAB; GOTO 13
905      END
906      ELSE PG := TRUE
907      UNTIL I = IA
908      END (=- IF FOUND THEN =-);
909      FOLLOW := LAST
910      END (=- FUNCTION FOLLOW =-);
911
912      FUNCTION ONEPIECE : VPTR;
913
914      VAR
915      P, Q, R      : VPTR;
916
917      BEGIN (=- FUNCTION ONEPIECE =-)
918      P := W;
919      REPEAT P := PZ,NEXT; Q := W;
920      WITH PZ DO
921      WHILE (QZ,X <> X) OR (QZ,Y <> Y) DO Q := QZ,NEXT
922      UNTIL (Q <> P) OR (P = W);
923      IF Q <> P THEN
924      BEGIN R := PZ,NEXT; PZ,NEXT := QZ,NEXT; QZ,NEXT := R;
925      ONEPIECE := P
926      END
927      ELSE
928      BEGIN ONEPIECE := W; W := NIL END
929      END (=- FUNCTION ONEPIECE =-);
930
931      FUNCTION SQUEEZER(V : VPTR) : VPTR;
932
933      VAR
934      Q, QQ, R, S  : VPTR;
935      DELETE      : BOOLEAN;
936
937      BEGIN (=- FUNCTION SQUEEZER =-)
938      Q := V; QQ := V;
939      IF Q <> NIL THEN
940      BEGIN
941      REPEAT R := QZ,NEXT; S := RZ,NEXT;
942      DELETE := (RZ,X = SZ,X) AND (RZ,Y = SZ,Y);
943      IF DELETE THEN
944      BEGIN QZ,NEXT := S; QQ := Q; RZ,NEXT := FREEV;

```

```

945         FREEV := R
946     END
947     ELSE U := R
948     UNTIL (U = QQ) AND NOT DELETE OR (QZ.NEXT = Q)
949     IF QZ.NEXT = Q THEN
950         BEGIN QZ.NEXT := FREEV; FREEV := Q; Q := NIL END
951     END;
952     SQUEEZER := Q
953     END (* FUNCTION SQUEEZER *)
954
955 FUNCTION RELATE(A : CPTR) : CPTR
956
957     VAR
958         NP, NN      : INTEGER;
959         Q, R, S      : CPTR;
960         X, Y, Z      : VPTR;
961
962     BEGIN (* FUNCTION RELATE *)
963         NP := 0; NN := 0; Q := A;
964         WHILE Q <> NIL DO
965             WITH QZ DO
966                 BEGIN
967                     IF POSITIVE(V) THEN
968                         BEGIN NP := NP + 1; R := L END
969                     ELSE NN := NN + 1; Q := NEXT
970                     END (* WITH QZ DO *)
971                 IF NN > NP THEN
972                     BEGIN
973                         IF NP <> 1 THEN
974                             BEGIN WRITELN(OUTPUT, ' NP,NN = ', NP, S, NN, S);
975                                 WRITELN(PRINT, ' NP,NN = ', NP, S, NN, S); GOTO 13
976                             END;
977                         Q := A; A := R;
978                         WHILE Q <> NIL DO
979                             WITH QZ DO
980                                 BEGIN S := NEXT;
981                                     IF Q <> R THEN
982                                         BEGIN NEXT := RZ.SIN; RZ.SIN := Q; X := V; Y := AZ.NEXT;
983                                             REPEAT Z := YZ.NEXT; YZ.NEXT := X; X := Y; Y := Z
984                                                 UNTIL X = V
985                                         END (* IF Q <> R THEN *)
986                                     Q := S
987                                 END (* WITH QZ DO *)
988                                 AZ.NEXT := NIL
989                             END (* IF NN > NP THEN *)
990                             RELATE := A
991                         END (* FUNCTION RELATE *)
992
993     BEGIN (* FUNCTION EXCP *)
994         C := NIL;
995         REPEAT U := SQUEEZER(FOLLOWUPR); FOUND := U <> NIL;
996             WHILE U <> NIL DO
997                 BEGIN U := SQUEEZER(ONEPIECE);
998                     IF U <> NIL THEN
999                         BEGIN D := NEXT;
1000                             WITH DZ DO
1001                                 BEGIN V := U; NEXT := C; SIN := NIL END;
1002                                 C := D
1003                             END
1004                         END (* WHILE U <> NIL DO *)
1005                     UNTIL NOT FOUND;
1006                     EXCP := RELATE(C)
1007                 END (* FUNCTION EXCP *)

```

```

1008
1009 FUNCTION PRE(A, B : CPTR) : CPTR FORWARD;
1010
1011 FUNCTION UNI(A, B : CPTR) : CPTR FORWARD;
1012
1013 FUNCTION DIF(A, B : CPTR) : CPTR FORWARD;
1014
1015 FUNCTION PRE(A, B : CPTR) : CPTR;
1016
1017 LABEL
1018 1;
1019
1020 VAR
1021 S, Q, U1, R, X, Y : CPTR;
1022 QD : ODHOS;
1023
1024 BEGIN (= FUNCTION PRE =)
1025 B := COPY(B); S := NIL;
1026 WHILE A <> NIL DO
1027 BEGIN Q := B; Q1 := NIL;
1028 WHILE Q <> NIL DO
1029 BEGIN FILL(AZ, V; QZ, V, QD, FALSE, FALSE);
1030 CASE QD OF
1031 IDENT :
1032 BEGIN R := NEWC;
1033 WITH R2 DO
1034 BEGIN V := COPYV(AZ, V); NEXT := S;
1035 S1 := UNI(AZ, S1, QZ, S1);
1036 END;
1037 S := R; X := Q; Q := X2, NEXT; X2, NEXT := NIL;
1038 IF Q1 = NIL THEN B := Q ELSE Q12, NEXT := Q1; DISPC(X);
1039 GOTO 1;
1040 END;
1041 ALJ :
1042 BEGIN X := AZ, NEXT; AZ, NEXT := NIL; Y := DIF(A, QZ, S1);
1043 S := CONC(Y, S); AZ, NEXT := X; Q1 := Q1; Q := QZ, NEXT;
1044 GOTO 1;
1045 END;
1046 B1A :
1047 BEGIN X := Q; Q := X2, NEXT; X2, NEXT := NIL;
1048 Y := DIF(X, AZ, S1); S := CONC(Y, S);
1049 IF Q1 = NIL THEN B := Q ELSE Q12, NEXT := Q; DISPC(X);
1050 END;
1051 CROSS :
1052 BEGIN R := EXCP(THQ); X := UNI(AZ, S1, BZ, S1);
1053 Y := DIF(R, X); DISPC(R); DISPC(X); S := CONC(Y, S);
1054 Q1 := Q; Q := QZ, NEXT;
1055 END;
1056 DISJUNKT :
1057 BEGIN Q1 := C; Q := QZ, NEXT END;
1058 END (= CASE QD OF =)
1059 END (= WHILE Q <> NIL DO =);
1060 1 :
1061 A := AZ, NEXT;
1062 END (= WHILE A <> NIL DO =);
1063 DISPC(B); PRE := S;
1064 END (= FUNCTION PRE =);
1065
1066 FUNCTION UNI(A, B : CPTR) : CPTR;
1067
1068 LABEL
1069 1, 2;
1070

```

```

1071 VAR
1072 P, P1, Q, Q1, R, S, X, Y, Z, A1, B1 : CPTH;
1073 OD
1074 : ODNOSI
1075 BEGIN (= FUNCTION UNI =)
1076 A := COPY(A); B := COPY(B); P := A; P1 := NIL;
1077 WHILE P <> NIL DO
1078 BEGIN Q := B; Q1 := NIL;
1079 WHILE Q <> NIL DO
1080 BEGIN FILLIPZ.V: QZ.V, OD, FALSE, FALSE;
1081 CASE OD OF
1082 IDENT :
1083 BEGIN X := PRE(PZ.SIN, QZ.SIN); DISPC(PZ.SIN); PZ.SIN := X;
1084 X := Q; Q := XZ.NEXT; XZ.NEXT := NIL;
1085 IF Q1 = NIL THEN B := Q ELSE Q1Z.NEXT := Q; DISPC(X);
1086 GOTO 1
1087 END;
1088 A1B :
1089 BEGIN X := P; P := XZ.NEXT; XZ.NEXT := NIL;
1090 Y := DIF(QZ.SIN, X); DISPC(QZ.SIN); QZ.SIN := Y;
1091 IF P1 = NIL THEN A := P ELSE P1Z.NEXT := P; DISPC(X);
1092 GOTO 2
1093 END;
1094 B1A :
1095 BEGIN X := Q; Q := XZ.NEXT; XZ.NEXT := NIL;
1096 Y := DIF(PZ.SIN, X); DISPC(PZ.SIN); PZ.SIN := Y;
1097 IF Q1 = NIL THEN B := Q ELSE Q1Z.NEXT := Q; DISPC(A);
1098 END;
1099 CROSS :
1100 BEGIN R := EXC(FALSE); X := P; P := XZ.NEXT;
1101 XZ.NEXT := NIL; A1 := XZ.SIN; XZ.SIN := NIL; Y := Q;
1102 Q := YZ.NEXT; YZ.NEXT := NIL; B1 := YZ.SIN;
1103 YZ.SIN := NIL; S := DIF(A1, Y); S := CONC(DIF(B1, X), S);
1104 S := CONC(PRE(A1, B1), S); DISPC(A1); DISPC(B1);
1105 Z := DIF(R, S); DISPC(S); DISPC(R);
1106 IF P1 = NIL THEN A := P ELSE P1Z.NEXT := P;
1107 IF Q1 = NIL THEN B := Q ELSE Q1Z.NEXT := Q;
1108 IF Z <> NIL THEN
1109 BEGIN S := Z;
1110 WHILE SZ.NEXT <> NIL DO S := SZ.NEXT; SZ.NEXT := B;
1111 IF Q1 = NIL THEN Q1 := S; B := Z
1112 END;
1113 DISPC(X); DISPC(Y); GOTO 2
1114 END;
1115 DISJUNKT :
1116 BEGIN Q1 := Q; Q := QZ.NEXT END
1117 END (= CASE OD OF =)
1118 END (= WHILE Q <> NIL DO =);
1119 1 :
1120 P1 := P; P := PZ.NEXT;
1121 2 :
1122 END (= WHILE P <> NIL DO =);
1123 UNI := CONC(A, B);
1124 END (= FUNCTION UNI =);
1125
1126 FUNCTION DIF(A, B : CPTH) : CPTH;
1127
1128 VAR
1129 S, Q, C, X, Y, D, R : CPTH;
1130 OD
1131 : ODNOSI
1132 BEGIN (= FUNCTION DIF =)
1133 IF B = NIL THEN DIF := COPY(A) ELSE

```



```

1134 BEGIN S := NIL;
1135 WHILE A <> NIL DO
1136 BEGIN Q := NEWC1 Q2,V := COPYV(A2,V);
1137 Q2,NEXT := NIL; Q2,SIN := NIL;
1138 D := UNIAZ,SIN, B; C := D; R := NIL;
1139 WHILE C <> NIL DO
1140 BEGIN FILL(Q2,V, C2,V, D, FALSE, TRUE);
1141 CASE OD OF
1142 IDENT :
1143 BEGIN S := CONC(COPY(C2,SIN), S); DISPC(Q);
1144 U := NIL; C := NIL;
1145 END;
1146 AINB :
1147 BEGIN
1148 S := CONC(PRE(Q, C2,SIN), S); DISPC(Q);
1149 U := NIL; C := NIL;
1150 END;
1151 BINA :
1152 BEGIN X := C2,NEXT; C2,NEXT := NIL;
1153 R := CONC(COPY(C), R); C2,NEXT := X; C := X;
1154 END;
1155 CROSS :
1156 BEGIN X := EXCP(TRUE);
1157 IF X = NIL THEN
1158 BEGIN S := CONC(PRE(Q, C2,SIN), S); DISPC(Q);
1159 Q := NIL; C := NIL;
1160 END;
1161 ELSE
1162 IF X2,NEXT = NIL THEN
1163 BEGIN DISPC(Q); Q := X; R := CONC(Q2,SIN, R);
1164 Q2,SIN := NIL; C := C2,NEXT;
1165 END;
1166 ELSE
1167 BEGIN R := CONC(R, COPY(C2,NEXT)); Y := DIF(X, R);
1168 DISPC(X); DISPC(R); S := CONC(Y, S);
1169 DISPC(Q); U := NIL; C := NIL;
1170 END (:= ELSE IF X2,NEXT = NIL THEN :=);
1171 END;
1172 DISJUNT :
1173 C := C2,NEXT;
1174 END (:= CASE OD OF :=);
1175 END (:= WHILE C <> NIL DO :=);
1176 DISPC(D);
1177 IF Q <> NIL THEN
1178 BEGIN Q2,SIN := R; S := CONC(Q, S); END;
1179 A := A2,NEXT;
1180 END (:= WHILE A <> NIL DO :=);
1181 DIF := S;
1182 END (:= ELSE IF B = NIL THEN :=);
1183 END (:= FUNCTION DIF :=);
1184
1185 FUNCTION PRESEK(A, B : CPTR) : CPTR FORWARD;
1186
1187 FUNCTION UNIA(A, B : CPTR) : CPTR FORWARD;
1188
1189 FUNCTION DIFERENCA(A, B : CPTR) : CPTR FORWARD;
1190
1191
1192 FUNCTION COMPLIA : CPTR : CPTR
1193
1194 VAR
1195 P : CPTR;
1196

```

```

1197 BEGIN (= FUNCTION COMPL (=)
1198 IF A = NIL THEN
1199 BEGIN P := NEMCI
1200 WITH PZ DO
1201 BEGIN V := NIL; NEXT := NIL; SIN := NIL END
1202 END (= IF A = NIL THEN =)
1203 ELSE
1204 WITH AZ DO
1205 IF V = NIL THEN P := COPY(SIN) ELSE
1206 BEGIN P := NEMCI
1207 WITH PZ DO
1208 BEGIN V := NIL; NEXT := NIL; SIN := COPY(A) END
1209 END (= ELSE IF V = NIL THEN =)
1210 COMPL := P
1211 END (= FUNCTION COMPL =)
1212
1213 PROCEDURE OUT1(VAR F : TEXT; A : CPTR)
1214
1215 VAR
1216 G : INTEGER;
1217
1218 PROCEDURE OUT1A : CPTR; OTAC, BRAT : INTEGER;
1219
1220 VAR
1221 ID : INTEGER;
1222 Q : VPTR;
1223
1224 BEGIN (= PROCEDURE OUT1 =)
1225 IF A <> NIL THEN
1226 BEGIN G := G + 1; ID := G;
1227 WRITELN(F, ' : 25, 'ID =', ID : 5, ' : 5, 'BRAT =', BRAT : 5,
1228 ' : 5, 'OTAC =', OTAC : 5; Q := AZ.V;
1229 IF Q <> NIL THEN
1230 REPEAT
1231 WITH QZ DO
1232 BEGIN WRITELN(F, ' : 30, X / FACT : 10 : 5,
1233 Y / FACT : 10 : 5);
1234 Q := NEXT
1235 E.ID
1236 UNTIL W = AZ.V
1237 ELSE WRITELN(F, ' : 30, 'NIL')
1238 WITH AZ DO
1239 BEGIN OUT1(NEXT, OTAC, ID); OUT1(SIN, ID, Q) END
1240 END (= IF A <> NIL THEN =)
1241 END (= PROCEDURE OUT1 =)
1242
1243 BEGIN (= PROCEDURE OUT =)
1244 WRITELN(F); WRITELN(F); G := 0; OUT1A, Q, Q;
1245 WRITELN(F); WRITELN(F); WRITE(F, ' ')
1246 END (= PROCEDURE OUT =)
1247
1248 FUNCTION PRESEK(A, B : CPTR) : CPTR;
1249
1250 VAR
1251 C : CPTR;
1252
1253 BEGIN (= FUNCTION PRESEK =)
1254 IF (A = NIL) OR (B = NIL) THEN C := NIL ELSE
1255 IF AZ.V = NIL THEN C := DIFERENCA(B, AZ.SIN) ELSE
1256 IF BZ.V = NIL THEN C := DIFERENCA(A, BZ.SIN) ELSE C := PRE(A, B);
1257 PRESEK := C
1258 END (= FUNCTION PRESEK =)
1259

```

```

1260 FUNCTION UNIJA(A, B : CPTR) : CPTR;
1261
1262 VAR
1263     C          : CPTR;
1264
1265 BEGIN (*- FUNCTION UNIJA *-)
1266     IF A = NIL THEN C := COPY(B) ELSE
1267         IF B = NIL THEN C := COPY(A) ELSE
1268             IF AZ,V = NIL THEN C := DIFERENCA(B, AZ,SIN) ELSE
1269                 IF BZ,V = NIL THEN C := DIFERENCA(A, BZ,SIN) ELSE
1270                     C := UNIJA(A, B); UNIJA := C
1271             END (*- FUNCTION UNIJA *-);
1272
1273 FUNCTION DIFERENCA(A, B : CPTR) : CPTR;
1274
1275 VAR
1276     C          : CPTR;
1277
1278 BEGIN (*- FUNCTION DIFERENCA *-)
1279     IF B = NIL THEN C := COPY(A) ELSE
1280         IF A = NIL THEN C := NIL ELSE
1281             IF BZ,V = NIL THEN C := PRESEN(A, BZ,SIN) ELSE
1282                 IF AZ,V = NIL THEN
1283                     BEGIN C := NEWC;
1284                         WITH C2 DO
1285                             BEGIN V := NIL; NEXT := NIL; SIN := UNI(AZ,SIN, B)
1286                             END
1287                         END (*- IF AZ,V = NIL THEN *-);
1288                 ELSE C := DIF(A, B); DIFERENCA := C
1289             END (*- FUNCTION DIFERENCA *-);
1290
1291 FUNCTION INCLUSIUN(A, B : CPTR) : CPTR;
1292
1293 VAR
1294     S, D, A          : CPTR;
1295     PH, P            : VPTR;
1296     NI1, NI2, DX, DY, PLB : INTEGER;
1297
1298 PROCEDURE DEBPR(H : MESS; A : CPTR);
1299
1300 BEGIN (*- PROCEDURE DEBPR *-)
1301     WRITELN(PRINT); WRITELN(PRINT, H); WRITELN(PRINT);
1302     OUT(PRINT, A)
1303     END (*- PROCEDURE DEBPR *-);
1304
1305 PROCEDURE ROUND(A : CPTR);
1306
1307 PROCEDURE ROUNDPP(A, B : VPTR; VAR S : CPTR);
1308
1309 VAR
1310     P, Q, R, T, P1, P2, P3, P4 : VPTR;
1311     C, SI, U, D : CPTR;
1312     UX, UY, VX, VY : INTEGER;
1313
1314 BEGIN (*- PROCEDURE ROUNDPP *-)
1315     P := A;
1316     REPEAT Q := P2.NEXT; UX := Q1; UY := Q1; H := B; D := NIL;
1317     REPEAT T := R2.NEXT; VX := UX + R2.X - T2.X;
1318         VY := UY + R2.Y - T2.Y; P1 := NEWV; P2 := NEWV;
1319         P3 := NEWV; P4 := NEWV;
1320     WITH P12 DO
1321         BEGIN X := P2.X + UX; Y := P2.Y + UY; NEXT := P2 END;
1322     WITH P22 DO

```

```

1323     BEGIN X := Q2.X + UX; Y := Q2.Y + UY; NEXT := P3 END;
1324 WITH P32 DO
1325     BEGIN X := Q2.X + VX; Y := Q2.Y + VY; NEXT := P4 END;
1326 WITH P42 DO
1327     BEGIN X := P2.X + VX; Y := P2.Y + VY; NEXT := P1 END;
1328 C := NEWCI
1329 WITH C2 DO
1330     BEGIN V := P1; NEXT := NIL; SIN := NIL END;
1331 C := SQUEEZE(C);
1332 IF C <> NIL THEN
1333     BEGIN P1 := C2.V;
1334     IF NOT POSITIVE(P1) THEN
1335         BEGIN P2 := P1; P3 := P22.NEXT;
1336             REPEAT P4 := P32.NEXT; P32.NEXT := P2; P2 := P3;
1337                 P3 := P4
1338             UNTIL P2 = P1
1339             END (* IF NOT POSITIVE(P1) THEN **)
1340             D1 := UNID(C); DISPC(D1); DISPC(C); D := D1
1341             END (* IF C <> NIL THEN **)
1342             UX := VX; UY := VY; R := T
1343             UNTIL R = B1
1344             IF DEBIG THEN DEBPR(' ODUZETA LJUSKA ', D);
1345             S1 := DIF(S, D); DISPC(S1); DISPC(D); S := S1;
1346             P := Q
1347             UNTIL P = A
1348             END (* PROCEDURE ROUNDPP **)
1349
1350 BEGIN (* PROCEDURE ROUND **)
1351 WHILE A <> NIL DO
1352     BEGIN ROUNDPP(A2.V, B2.V, S1); ROUND(A2,S1); A := A2.NEXT
1353     END
1354 END (* PROCEDURE ROUND **)
1355
1356 PROCEDURE TRANSLATE(A : CPTR);
1357
1358 VAR
1359     P : VPTR;
1360
1361 BEGIN (* PROCEDURE TRANSLATE **)
1362 WHILE A <> NIL DO
1363     WITH A2 DO
1364         BEGIN P := V1
1365             REPEAT
1366                 WITH P2 DO
1367                     BEGIN X := X + DX; Y := Y + DY; P := NEXT END
1368                 UNTIL P = V1
1369                 TRANSLATE(SIN); A := NEXT
1370             END (* WITH A2 DO **)
1371         END (* PROCEDURE TRANSLATE **)
1372
1373 FUNCTION PL(P : VPTR) : INTEGER;
1374
1375 VAR
1376     A, B, C : VPTR;
1377     S : INTEGER;
1378
1379 BEGIN (* FUNCTION PL **)
1380     A := P; B := P2.NEXT; S := D1
1381     REPEAT C := B2.NEXT; S := S + B2.X * (C2.Y - A2.Y);
1382             A := B; B := C
1383     UNTIL A = P;
1384     PL := S
1385 END (* FUNCTION PL **)

```

```

1386
1387 FUNCTION REDCOPY(A : CPTR) : CPTR;
1388
1389 VAR
1390     P, Q : CPTR;
1391
1392 BEGIN (*- FUNCTION REDCOPY -*)
1393     P := NIL;
1394     WHILE A <> NIL DO
1395         WITH AZ DO
1396             BEGIN
1397                 IF PL(V) >= PLB THEN
1398                     BEGIN Q := HELC;
1399                         WITH QZ DO
1400                             BEGIN V := COPYV(AZ,V); SIN := COPY(AZ,SIN); NEXT := P
1401                                 END;
1402                             P := Q
1403                         END;
1404                     A := NEXT
1405                 END (*- WITH AZ DO -*);
1406             REDCOPY := P
1407         END (*- FUNCTION REDCOPY -*);
1408
1409 BEGIN (*- FUNCTION INCLUSION -*)
1410     IF B = NIL THEN INCLUSION := NIL ELSE
1411         WITH DZ DO
1412             IF NEXT = NIL THEN
1413                 BEGIN PLB := PL(DZ,V); A := REDCOPY(AA);
1414                     S := COPY(A); ROUND(A); DISPC(A);
1415                     IF DEBIG THEN DEBPR('NAKON ROUND', S);
1416                     IF S = NIL THEN
1417                         BEGIN WRITELN(OUTPUT, ' NEHA MJESTA');
1418                             WRITELN(PRINT, ' NEHA MJESTA'); GOTO 13
1419                         END;
1420                     WITH SZ, VZ DO
1421                         BEGIN PH := V; MIN := X; MINI := Y END;
1422                     D := S;
1423                     REPEAT
1424                         WITH DZ DO
1425                             BEGIN P := V;
1426                                 REPEAT
1427                                     WITH PZ DO
1428                                         BEGIN
1429                                             IF (X < MIN) OR (X = MIN) AND (Y < MINI) THEN
1430                                                 BEGIN MIN := X; MINI := Y; PH := P END;
1431                                             P := NEXT
1432                                         END (*- WITH PZ DO -*);
1433                                         UNTIL P = V;
1434                                         D := NEXT
1435                                     END (*- WITH DZ DO -*);
1436                                 UNTIL D = NIL;
1437                                 DX := PH2,X - VZ,X; DY := PH2,Y - VZ,Y; DISPC(S);
1438                                 IF DEBIG THEN
1439                                     WRITELN(PRINT, ' DX,DY =', DX, ' 10, DY : 10);
1440                                 S := COPY(B); TRANSLATE(S); INCLUSION := S
1441                             END (*- IF NEXT = NIL THEN -*);
1442                         ELSE HALT
1443                     END (*- FUNCTION INCLUSION -*);
1444
1445 FUNCTION HEADP : CPTR;
1446
1447 VAR
1448     N: I : INTEGER;

```

```

1449      P, Q          : VPTR;
1450      C            : CPTR;
1451      XX, YY       : REAL;
1452
1453      BEGIN (= FUNCTION READP =)
1454      READ(INPUT, N); IF N < 3 THEN HALT;
1455      READ(INPUT, XX, YY); P := NEWV;
1456      WITH P2 DO
1457      BEGIN X := ROUND(XX * FACT); Y := ROUND(YY * FACT);
1458      NEXT := P
1459      END;
1460      FOR I := 2 TO N DO
1461      BEGIN READ(INPUT, XX, YY); Q := NEWV;
1462      WITH Q2 DO
1463      BEGIN X := ROUND(XX * FACT); Y := ROUND(YY * FACT);
1464      NEXT := P2, NEXT; P2, NEXT := Q;
1465      P := Q
1466      END;
1467      END (= FOR I := 2 TO N =);
1468      IF NOT POSITIVE(P) THEN HALT;
1469      C := NEWC;
1470      WITH C2 DO BEGIN V := P; NEXT := NIL; SIN := NIL END;
1471      READLN(INPUT); READP := C
1472      END (= FUNCTION READP =);
1473
1474      PROCEDURE PERM;
1475
1476      VAR
1477      I, J, K      : 0 .. MAXEL;
1478
1479      BEGIN (= PROCEDURE PERM =)
1480      FOR I := N DOWNTO 1 DO
1481      BEGIN J := TRUNC(I * RANDOM) + 1;
1482      K := CCIS; CCIS := CCJS; CCJS := K
1483      END
1484      END (= PROCEDURE PERM =);
1485      FUNCTION SLOZI(PRT: BOOLEAN): INTEGER;
1486
1487      VAR
1488      X, G, H, A1, P : CPTR;
1489      Q              : VPTR;
1490      XMAX          : INTEGER;
1491      K             : INTEGER;
1492      TD, T1       : ARRAY[1 .. 35] OF REAL;
1493
1494      BEGIN (= PROCEDURE SLOZI =)
1495      X := COPY(A); XMAX := -MAXINT;
1496      FOR K := 1 TO N DO
1497      BEGIN
1498      TOC1S := ROUND(CPTIME(0));
1499      TOC2S := ROUND(CPTIME(1));
1500      TOC3S := ROUND(CPTIME(3));
1501      G := BCCASS; H := INCLUSION(X, G);
1502      IF PRT OR DEBUG THEN
1503      BEGIN WRITELN(PRINT); WRITELN(PRINT);
1504      WRITELN(PRINT, ' POLIGON BROJ ', CCAS);
1505      WRITELN(PRINT); WRITELN(PRINT);
1506      OUT(PRINT, H)
1507      END;
1508      P := H;
1509      WHILE P <> NIL DO
1510      WITH P2 DO
1511      BEGIN V := V;

```

```

1512         REPEAT
1513         WITH QZ DO
1514         BEGIN IF X > XMAX THEN XMAX := X; Q := NEXT END
1515         UNTIL Q = V;
1516         P := NEXT
1517         END;
1518         X1 := DIFERENCA(X, H); DISPC(H); DISPC(X); X := X1;
1519         TIC15 := ROUND(CPTIME(0));
1520         TIC25 := ROUND(CPTIME(1));
1521         TIC35 := ROUND(CPTIME(3));
1522         WRITELN(OUTPUT, ' ULOZEN POLIGON BR. ', K; 5; ' T.J. ', CCNS; 5);
1523         WRITE(OUTPUT, ' SUP = ', (TIC15-TIC15)/1000; 8; 3);
1524         WRITE(OUTPUT, ' TOT = ', (TIC25-TIC25)/1000; 8; 3);
1525         WRITE(OUTPUT, ' CPU = ', (TIC35-TIC35)/1000; 8; 3);
1526         WRITELN(OUTPUT);
1527         END (= FOR K := 1 TO N =);
1528         DISPC(X); SLOZI := XMAX;
1529         END (= FUNCTION SLOZI =);
1530
1531
1532 BEGIN (= PROGRAM SNESTI =)
1533 $FASTCALL
1534 RESET(INPUT); REWRITE(OUTPUT); REWRITE(PRINT);
1535 FREEV := NIL; FREEC := NIL;
1536 READ(INPUT, N, LPER, LL);
1537 DEBUG := (N MOD 10 > 0); N := N DIV 10;
1538 DEBIG := (N MOD 10 > 0); N := N DIV 10;
1539 DEBUG := (N MOD 10 > 0); N := N DIV 10;
1540 READ(INPUT, FACT);
1541 RANSET(); (* DA BI PROGRAM BIO REPRODUCIBILAN *)
1542 READ(INPUT, NPERM); A := READP; N := 0; M := 0;
1543 REPEAT READ(INPUT, M); M := M + 1; BCNS := READP;
1544 REPEAT N := N + 1; CCNS := M; M := M - 1 UNTIL M = 0;
1545 UNTIL EOF(INPUT);
1546
1547 FOR K := 1 TO LPER DO PERM;
1548 LHAX := -MAXINT; LMIN := MAXINT;
1549 D := 0;
1550
1551 FOR K := 1 TO NPERM DO
1552 BEGIN PERM;
1553 FOR K := 1 TO LPER DO PERM;
1554 WRITELN(OUTPUT, ' PERMUTACIJA BR. ', K; 5);
1555 WRITELN(OUTPUT, ' REDOSLIJED SLAGANJA JE :');
1556 PAGE(PRINT);
1557 WRITELN(PRINT, ' PERMUTACIJA BR. ', K; 5);
1558 WRITELN(PRINT);
1559 FOR I := 1 TO N DO
1560 BEGIN
1561 IF ((I DIV 20) * 20 = I - 0) THEN WRITELN(OUTPUT);
1562 WRITE(OUTPUT, CCNS; 4);
1563 END;
1564 WRITELN(OUTPUT);
1565 L := SLOZI(TRUE);
1566 WRITELN(OUTPUT);
1567 WRITELN(OUTPUT, ' POSTIGNUTA DUZINA ', L/FACT; 10; 5);
1568 IF L < LMIN THEN BEGIN LMIN := L; D := K; END;
1569 IF L > LHAX THEN LHAX := L;
1570 END;
1571 WRITELN(PRINT); WRITELN(PRINT, 'SITUACIJA NAKON ', NPERM; 4; ' PERMUTACIJA ');
1572 WRITELN(PRINT); WRITELN(PRINT);
1573 WRITELN(PRINT); WRITELN(PRINT, ' LMIN = ', LMIN/FACT; 10; 5);
1574 WRITELN(PRINT, ' LHAX = ', LHAX/FACT; 10; 5);

```

```
1575 WRITELN(PRINT, ' MINIMUM POSTIGNUT NA PERMUTACIJE BR. ', D ; 5)
1576 13:
1577 END (*- PROGRAM SHESTI *-).
```

```
DIRT.S W.SNTEST
```