KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Laura Grinkevičiūtė

Finansinių fondų dinamikos miglotųjų prognozavimo algoritmų sudarymas ir tyrimas

Magistro darbas

Darbo vadovas
prof. dr. R. Jasinevičius

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

Laura Grinkevičiūtė

Finansinių fondų dinamikos miglotųjų prognozavimo algoritmų sudarymas ir tyrimas

Magistro darbas

Recenzentas

doc. Vytautas Petrauskaus

Vadovas

prof.dr.R.Jasinevičius

Atliko

IFM-4/1 gr.stud.

Laura Grinkevičiūtė

Kaunas, 2010

Turinys

SANTRUMPŲ IR TERMINŲ ŽODYNAS

Agentas – Siauroje specializuotoje aplinkoje veikianti kompiuterinė programa, kuri, naudodamasi sukauptomis žiniomis, ateinančia iš aplinkos informacija ir siekdama agentui keliamų tikslų, autonomiškai ir lanksčiai formuoja agento išėjimo veiksmus.

DAS – daugiaagentinės sistemos, kuriose abstraktūs agentai sąveikauja tarpusavyje ir su juos supančia aplinka.

MPP – miglotieji pažintiniai planai

AR- Autoregresinis metodas

SANTRAUKA

Prognozuojant finansinių fondų dinamiką, dažniausiai sunku pasirinkti vieną iš daugelio esamų prognozavimo metodų, nes kiekvienas jų turi pranašumų ir trūkumų. Atlikta plačiai naudojamų prognozavimo metodų Soros, Bayeso, AR, Wiener analizė. Plačiai prognozavimui naudojami matematinės statistikos metodai, tačiau dažnai jie nagrinėja funkcijas su dideliais kintamųjų skaičiais. Tai daugelio diskretinių optimizavimo problemų atvejai. Norint apriboti klaidų skaičių reikia daugiau skaičiavimo bandymų. Dažnai bandymų skaičius turi didėti eksponentiškai. Gali būti kad įvykių statistikoje nėra duomenų kaip elgtis pagal tam tikrą situaciją. Dėl šių ir kitų priežasčių ieškoti kiti prognozavimo metodai. Šiame darbe finansinių fondų dinamikai prognozuoti naudojama miglotoji logika. Sukurta sistema prognozuoja įvykių grupių poveikius fondų kreivei, o ne remiasi statistika. Tai ekspertinis prognozavimo metodas. Miglotoji logika suteikia galimybę ekspertams analizuoti ir nusakyti probleminę sritį miglotomis taisyklėmis. Miglotosios taisyklės sudaromos iš žmogui suprantamų teiginių. Jomis ekspertai gali nustatyti sistemos įėjimo duomenis. Tokių įėjimo duomenų paklaida priklauso nuo specialistų patirties ir intuicijos įvertinti įvykių grupių poveikį fondų kreivei. Šiame darbe analizuoti fondų kreivės aproksimavimo būdai naudojant interpoliavimo ir suglodinimo metodus. Taip pat analizuotas Sklansky ir Gonzalez aproksimavimo algoritmas, kuris buvo realizuotas jį pamodifikavus. Bei pateikti kreivės pokyčio prognozavimo realizacijos proceso algoritmai. Pasiūlytas prognozavimo metodas paremtas miglotąja logika.

SANTRAUKA ANGLŲ KALBA (SUMMARY)

Forecasting the dynamics of financial funds, often are difficult to choose one of the many existing prediction methods, because each of them has advantages and disadvantages. Performed widely used forecasting techniques Soros, Bayeso, AR, Wiener analysis. It is widely used in prediction methods of mathematical statistics. Often mathematical statistics methods examine the functions of the large number of variables. Large numbers of variables are the main problems in many cases for discreet optimization methods. To limit the number of errors in calculating it is need for more testing. Often number of experiments should increase exponentially. It is possible that the statistics of events do not behave according to the data of a certain situation. For these and other reasons there was seek for other forecasting methods. In this work, fuzzy logic used to predict the dynamics of financial funds. The developed system predicts event’s groups effects of funds curve, and are not based on statistics. This is expert prediction method. Fuzzy logic makes possible for experts to analyze and define the problematic area of vague rules. Fuzzy rules are made from statements which are easy to understand. Experts can detect the input data. Such input data error depends on the expert experience and intuition to assess the impact of events groups funds curve. In this work where analyzed approximation methods as interpolation and smoothing the curve. Approximation algorithm of Sklansky and Gonzalez was modified and realized. There are algorithms to predict changes of the graph . There was proposed prediction method which is based on fuzzy logic.

.

ĮVADAS

Dauguma atliekamų teorinių ir praktinių intelektinio valdymo srities tyrimų siejami su miglotosios logikos ir neuroninių tinklų metodikomis bei jų deriniais. Egzistuoja nemažai finansinių fondų dinamikos prognozavimo sistemų „Rosetta Stone Secret“ [1],“ Ski gold stoks prediction“ [2], ECON [3], TIMESTAT [4], TSE [5] ir t.t. Dauguma jų sukurtos naudojant matematinės statistikos metodus. Populiarūs prognozavimo metodai yra Wiener [6,22], AR[7, 22] , Soros [8, 22], Bayesian [9, 22]ir t.t. Šiame darbe siūloma susipažinti su miglotąja logika [1]ir panaudoti ją finansinių fondų dinamikos prognozavimui.

Problemos aktualumas. Prognozavimo svarba turi didelę įtaką. Galima parengti verslo planą, kuris apibrėžti kiek reikės finansinių išteklių, prognozavimas padeda nuspėti finansinių fondų dinamiką, investicijų, kreditų saugumą.bankuose taikomi statistiniai ir ekonominiai modeliai atlieka veiksmus su finansiniais duomenimis. Gyvendami nuolat kintančiame pasaulyje svarbu numatyti ateities tendencijas ir būsimą įvykių eigą .

Darbo tikslas yra sudaryti ir tyrinėti algoritmus, paremtus miglotąja logika, prognozuoti finansinių fondų dinamiką. Realizuoti sistemą naujam kreivės taškui prognozuoti. Tam, kad šie tikslai būtų pasiekti, reikėjo atlikti tokius uždavinius:

- Atlikti bendrąją miglotosios logikos analizę.
- Išanalizuoti metodus, sukurti algoritmus ir realizuoti sistemos kūrimo proceso etapus.

Mokslinis naujumas. Miglotosios logikos teorija buvo pradėta vystyti 1965m. Berklio Universitete (JAV). Prieš tai valdymo logika buvo kuriama Aristotelio mokymo pagrindu, kad bet kuris elementas arba yra, arba nėra aibės narys. Tačiau realiai dažnai elementai negali būti griežtai apibrėžti ir suklasifikuoti pagal kažkokį kriterijų. Tokio griežto klasifikavimo nereikalauja miglotoji logika ("fuzzy" ang. reiškia neaiškus, neraiškus). Ir nors miglotoji logika yra griežtai apibrėžta matematiškai, ji leidžia sistemų valdymui naudoti kalbines sąvokas. Kadangi lingvistinis sistemos apibrėžimas yra lengvesnis nei matematinis modelis, miglotoji logika yra patrauklesnė sudėtingose valdymo sistemose. Šiuo metu miglotoji logika yra populiari [17]. Ji leidžia matematiškai apibrėžti dalinį teisingumą, netikslumą panaudojant lingvistinius kintamuosius. Pasitelkiant miglotųjų aibių teoriją, kasdieninės kalbos informaciją galima perteikti kompiuteriui, o tai leidžia lengviau ir suprantamiau formuoti

[1] Fuzzy logic

įvairias valdymo ir kitas užduotis. Netikslius, neaiškius reiškinius aprašyti aibių teorija pagrįsta matematika yra sudėtinga, o be tikslaus matematinio modelio valdyti neaiškius procesus yra sudėtinga arba visai neįmanoma. Tokiu atveju išeitis gali būti rasta panaudojant miglotąją logiką.

Darbo struktūra. *Pirmajame darbo skyriuje* susipažinta su dinaminėmis sistemomis, jų pavyzdžiais, išskirtos ekonominių matematinių modelių ir prognozavimo metodų grupės, pateikta bendroji miglotosios logikos, plačiai naudojamų prognozavimo metodų teorijos apžvalga. *Antrajame skyriuje* sudaryta fondų kreivės prognozavimo sistemos architektūra. Pasiūlytas prognozavimo metodas paremtas miglotąja logika. *Trečiajame skyriuje* analizuojama sistemos veiksena, pasiūlyti egzistuojantys arba sudaryti nauji algoritmai iškilusiems uždaviniams spręsti.

1 TEORIJA

Dinaminės sistemos

Šiame darbe bus nagrinėjama finansinių fondų dinamika, tai dinaminė sistema.

$\vec{K} = \phi(\vec{k}, \vec{x})$ k-vidinė sistemos būsena; K-kita sistemos būsena; x-sistemą veikiantys veiksniai. Sistemą veikia išoriniai veiksniai, jie keičia sistemos vidinę būseną ir sistema pereina į kitą būseną. Apibrėžiamos dinaminės sistemos savybės, bei pateikiami dinaminių sistemų pavyzdžiai.

Dinaminės sistemos savybės

- Įvykių grandinės yra priežastinės ir pasekminės
- Pasekmės inercija priežasties arba inercijos atžvilgiui

Analizuojant sistemą keliami trys uždaviniai

- Pasiekiamumo uždavinys – ar galima pasiekti norimą sistemos rezultatą K esant sistemos būsenai k ir veikiant veiksniais x.
- Stebimumo uždavinys – jei žinom rezultatą ar galim pasakyti būseną.
- Stabilumo sistemos uždavinys – ar sistema stabili, ar ji gali gyvuoti. Ar maži pokyčiai įėjimų ir parametrų iššaukia didelius sistemos išėjimų pokyčius.

Sistemos yra veikiamos teigiamais arba neigiamais poveikiais, paveiktų sistemų būsenos pakinta. Sistemą veikiantys poveikiai sukelia skirtingą sistemos rezultatų vėlinimą.

Agentai ir dirbtinis intelektas

Tai dinaminės sistemos. Intelektualių agentų sąvoka kilo iš dirbtinio inte1eko (DI). Vienas iš būdų apibūdinti DI yra intelektualaus agento kūrimo problema. Svarbu atskirti inte1ektą,

kuris yra DI tikslas, ir inte1ektą, kurtį norima realizuoti agentuose. Agento intelektui keliamas vienas reikalavimas — priimti teisingą spendimą laiku ir kad šis sprendimas būtų naudingas. Kiti agentams keliami intelekto reikalavimai priklauso nuo aplinkos, kurioje jie veikia (pvz., ne visiems agentams būtinas sugebėjimas mokytis). DI metodai dažniausiai tinka programiniams agentams kurti. Finansinių fondų dinamikos prognozavimas bus atliekamas kuriant panašų agentą.

Neuroniniai tinklai

Intelektinė sistema (ar mašina) turi sugebėti atlikti protingą (ar tinkamą) sprendimą ar veiksmą esant kintančiai ar netiksliai aplinkai. Tam turi būti panaudotos visos lygiagretaus informacijos apdorojimo, mašinų mokymosi, asociatyvinės atminties, apibendrinimo ir kūrybos teikiamos galimybės. Su šia sfera susietų tyrimų rezultatai įtakojo neuroninių tinklų technologijų sėkmingą taikymą vaizdų atpažinime, paveikslų ir natūralios kalbos apdorojime ir robotų technikoje ir t.t. Neuroniniai tinklai (dinaminė sistema kaip galingas įrankis) plačiai naudojami intelektinėse valdymo sistemose, ypač netiesinių sistemų modeliavime. Svarbiausi neuroninių tinklų vystymosi aspektai yra noras suprojektuoti taip vadinamą topologiją (architektūrą), mokymosi algoritmus bei apmokymo šablonų parinkimas.

Prognozavimo metodai

Naudojami prognozavimo metodai

Ekonominių matematinių modelių grupės

- Funkciniai modeliai (naudojami, tuomet kai nagrinėjamos tiesioginės priklausomybės tarp kintamųjų);
- Balansiniai modeliai (aprašomi pasitelkus lygčių sistemą);
- Optimizaciniai modeliai (naudojami, kai norima nustatyti maksimalią ar minimalią kriterijaus reikšmę);
- Imitaciniai modeliai (naudojami tuomet, kai negalima taikyti kitų metodų, kriterijai pateikiami algoritmo forma)
- Kompleksiniai modeliai.

Prognozavimo metodai

- Prognozavimas grindžiamas nuomonėmis. Prognozės, atliekamos šiais metodais, pamatas yra subjektyvi, surinkta iš įvairių šaltinių informacija. Gana dažnai kitais būdais gauti informacijos iš šių šaltinių neįmanoma.

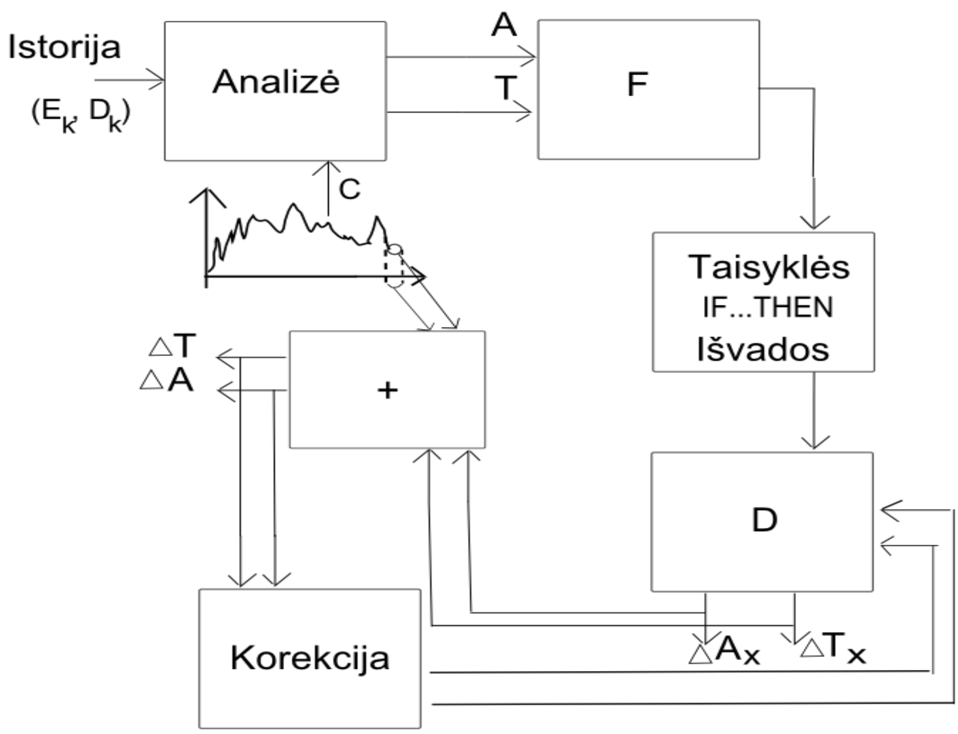
- Prognozavimas, remiantis praėjusių laikotarpiu, duomenimis (duomenų sekų analizė). Juo numatomos kintamojo dydžio reikšmės, darant prielaidą, kad kiti dydžiai nekis. Dabarties dėsningumai „perkeliami“ į ateitį. Šiais metodais neieškoma kintamųjų dydžių tarpusavio ryšių, o paprasčiausiai remiamasi praėjusių laikotarpių (istoriniais) duomenimis.

- Tarpusavio ryšių radimo metodai (asociatyvusis prognozavimas). Jais nustatomi vienas ar daugiau kintamųjų

Prognozavimo metodo tinkamumo kriterijai

- prognozės tikslumo laipsnis;
- prognozės laikotarpis;
- reikiamas pradinių duomenų kiekis;
- prognozavimo kaštai;
- rezultatų įdiegimo ir pritaikymo lygis.

Prognozavimo metodas remiantis miglotąja logika



E_k - paduodami į sistemą įvykiai

D_k - Įvykio poveikio vėlinimo reikšmė.

A- Kreivės kampas

T- laiko momentas

ΔA_x – prognozuojamas kampas kokiu pasikeis fondų kreivės grafikas

ΔT_x – prognozuojamasis laiko momentas kada pasikeis kreivės kampas

ΔA – gautą kreivės pokyčio kampą sudėjus su esamu kreivės pokyčio kampu gaunamas prognozuojamas kreivės kitimo kampas.

ΔT – gautas prognozuojamo kampo vėlinimas pridėtas prie vėliausios kreivės datos.

k=1,2,…

F- tai transformacija kuri gautiems duomenims randa miglotosios logikos atitikmenis.

Sudaromos miglotosios logikos taisyklės, surandamos prognozuojamo kreivės pokyčio kampas ir jo vėlinimas.
D- miglotoje rezultatai paverčiami į prognozuojamo kreivės pokyčio kampą ir vėlinimą.

Korekcija- prognozuojamo kreivės pokyčio kampo ir vėlimo reikšmė pakoreguojamos jei reikia pagal nustatytus [min;max] intervalus. Miglotosios taisyklės šioje iteracijoje gali būti keičiamos, papildomos naujomis.
+- operacija prie turimų kreivės kampo ir datos reikšmių prideda gautą vėlinimą ir kampą.

Sekantį kartą prognozuojant naudojama kreivė su nauju prognozuotu tašku.

Plačiai naudojami prognozavimo metodai

Soros metodas

Pasižymėjimai:

s- klientas

t- laiko momentas

i- pirkėjų[2] skaičius vienai akcijai i=1,2.

$z(t)$ - akcijų kursas laiku t

$z^{buy}(t)$ - slenkstis kada pradėti pirkti akcijas laiko momentu t

---

[2] Akcininkų

$z^{sell}(t)$ - slenkstis kada pradėti parduoti akcijas laiko momentu t

$z_i(t+1)$ - akcijų kursas laiku *(t+1)*prognozuotas naudojant AR metodą

d(t+1) – tikėtini dividendai momentu *(t+1)*

a(t+1)- pajamos laiko momentu *(t+1)*

$\Delta_i(t+1)$ - akcininko *i* pelno dydis laiko momentu *(t+1)*

$\Delta_i(t)$ - akcininko *i* tikėtinas pelno dydis laiko momentu *(t)*

$\Delta_j(t)$ - akcininko *j* tikėtinas pelno dydis laiko momentu *(t)*

$k_i^{buy} > 0$ - pirkimo slenkstis

$k_i^{sell} < 0$ - pardavimo slenkstis

- Slenkstis kada pradėti pirkti

  $z_i^{\min}(t) = k_{buy} \cdot \Delta_i(t), k_{buy} > 1$

- Slenkstis kada pradėti parduoti

  $z_i^{\max}(t) = k_{sell} \cdot \Delta_i(t), k_{sell} < 1$

- Akcininkas *i* perka akcijas laiko momentu *t* jeigu

  $\Delta_i(t) \geq k_i^{buy}$

- Akcininkas i parduoda akcijas laiko momentu *t* jeigu

  $\Delta_i(t) \leq k_i^{sell}$

- Papildomas akcininkas *s* perka akcijas laiku *t* jeigu

  $z(t) \leq z^{buy}(t)$

- Papildomas akcininkas *s* parduoda akcijas laiku *t* jeigu

  $z(t) \geq z^{sell}(t)$

- Akcijų kursas laiku *(t+1)* nustatomas pagal akcijų kursą laiku *t* [8]

$$z(t+1) = \begin{cases} z(t) + \varepsilon(t+1), & if\ k_{\mathrm{i}}^{\mathrm{sell}} < \Delta_i(t) < k^{buy}\ for\ all\ i \\ z(t), & if\ \Delta_{\mathrm{i}}(t) \leq k^{sell}\ for\ some\ i \\ z(t), & if\ \Delta_{\mathrm{j}}(t) \geq k^{buy}\ for\ some\ j \\ z^{sell}(t), & if\ z(t) \geq z^{sell}(t) \\ z^{buy}(t), & if\ z(t) \leq z^{buy}(t) \end{cases} \quad ,\text{kur}$$

- Tikėtinas pelnas akcininko *i* momentu *(t+1)*

$\Delta_i(t+1) = (\beta_i(t+1) + \delta(t+1) - \alpha(t+1))$, kur

$\beta_i(t+1) = (z_i(t+1) - z(t))/z(t)$

$\delta(t+1) = d(t+1)/z(t)$

$\alpha(t+1) = a(t+1)/z(t)$

AR metodas

Pasižymėjimai:

t- būsimas laiko momentas

w_t - spėjimas busimam laiko momentui

w_{t-1} - reikšmė esamu laiku

ε_t - generuota atsitiktinė reikšmė busimam laiko momentui

a_i - koeficientas apibrėžtas minimizuojant tokią formulę

$$f(x) = \sum_{t=1}^{T} |\varepsilon_t|$$

Minimizacijai naudojamas tiesinis programavimas

$\min_{a,u} \sum_{t=1}^{T} u_t$, kur

$u_t \geq \varepsilon_t$, t=1,...,T

$u_t \geq -\varepsilon_t$, t=1,...,T

$a_i = a_i^1 - a_i^2$, $a_i^k \geq 0$, k=1,2, i=1,...,p

- *AR* lygties išraiška [7]

$$w_t = \sum_{i=1}^{p} a_i \cdot w_{t-i} + \varepsilon_t$$

Bayeso metodas

Bayeso euristinis metodas yra naudojamais optimizuoti funkcijoms su dideliais kintamųjų skaičiais [9]. Tiesioginio Bayeso algoritmo tikslas, naudojamas daugeliu atvejų, yra pateikti kiek įmanomai mažą paklaidą laikantis konvergavimo sąlygų.

Fiksuojant ankstesnio išsibarstymo P funkcijų f(x) rinkiniui ir minimizuojant Bayeso rizikos funkciją. Rizikos funkcija R(x) yra planuojamas skirtumas iš globalaus minimumo fiksuotame taške x. P išsibarstymas yra apgalvotas kaip stochastinis modelis funkcijai f(x), $x \in R$, kur f(x) gali būti determinuota arba stochastinė funkcija. Gauso atveju, laikant kad (n+1) stebėjimas yra paskutinis

$$R(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot s_n(x)}} \cdot x_{-\infty}^{+\infty} \cdot \min(c_n, z) \cdot e^{\frac{1}{2} \cdot \frac{y - m_n(x)^2}{s_n(x)} dz}$$,kur

$c_n = \min_i \cdot z_i$,

$z_i = f(x_i) \cdot m_n(x)$ yra sąlyginė tikimybė atsižvelgiant į stebėjimų reikšmes z_i

i=1,...,n

$s_n^2(x)$ yra sąlyginis pasiskirstymas ir $\varepsilon > 0$ yra taisymo parametras.

Rizikos funkcijos R(x) minimumas yra surandamas taške

$$x_{n+1} = \arg\max_x \frac{s_n(x)}{m_n(x) - c_n}$$

Wiener metodas

Wiener metodas atspindi grįžtamojo ryšio teorijos principus [6]. Tai mechanizmas kurį teoriškai gali simuliuoti mašinos. Šis atradimas įtakojo didžiąją dalį to meto dirbtinio intelekto tyrimų.

Miglotoji logika

Miglotosios logikos požiūris taikomas kuriant augančius (MPP) miglotuosius pažintinius planus, išreiškiančius skirtingus filosofinius ir loginius tinklo augimo aspektus. Metodai leidžiantys analizuoti procesus ir sintezuoti įvairiausias sistemas [22], funkcionuojančias apibrėžtumo sąlygomis remiasi miglotąja logika, miglotosiomis aibėmis, bei miglotaisiais pažintiniais planais (MPP). (Kosko, 1997). Daugumą sudėtingų sistemų ir situacijų galima pavaizduoti kaip daugiaagentines sistemas (DAS), kuriose abstraktūs agentai sąveikauja tarpusavyje ir su juos supančia aplinka (Ferber, 1999). Tokių abstrakčių agentų savybes, elgseną ir sąveiką dažniausiai galima aprašyti tik labai miglotai. Miglotoji logika tampa gana įprastu formalizavimo mechanizmu. Statiniai agentų sąryšiai charakterizuojami tam tikru miglotu tikrumo laipsniu $\mu_R(x, y)$. Čia R- sąryšis, x ir y- nagrinėjamieji agentai, o $\mu_R(*)$ - tikrumo laipsnis intervale [0,1]. DAS elementų sąveikos dinamiką, gerai atspindi miglotieji pažintiniai planai (MPP). MPP sudaromas iš bivalentinių mazgų (i=1,2,…,j,k,…..,N), vaizduojančių sąveikaujančius agentus, kurių vidinę būseną nusako koncepciniai dydžiai C_i iš [0,1]. Mazgai esti susieti priežasties ir pasekmės santykiais. Tokio sąryšio tarp C_i ir C_k stiprumas reiškiamas skaičiumi iš [-1,1], atitinkančiu tikrumo funkciją . MPP koncepcija remiasi galimybiniais sąveikos tikrumo įverčiais. Finansinio kapitalo rinkoje prognozuoti investicijų riziką ir pelną ilgą laiką naudota tradicinė taip/ne dvejetainė algebra, besiremianti if/then/else sąlygos sakiniais. Arba naudotos matricos, kuriose sulyginami kintamieji poromis. Šiuo metu paplito neryškių aibių sistemos ir miglotoji logika, kuri leidžia logines taisykles išreikšti naudojant „dalines“ tiesas, pavyzdžiui, if then might-be-true arba if then could-possibly-be-true(Craven, Shavlik,1995). Sukurtų taisyklių kokybė priklauso nuo jų tikslumo, tinkamumo ir suprantamumo(Diederich, 1994). Miglotoji logika turi daug bendrumų su tolydžiąja (sinkretine) logika. Ją lyginant su kategoriškąja logika, akivaizdžiai matomas nekategoriškumas – trečiosios būsenos galimybė, ko negali būti kategoriškojoje Bulio logikoje. Vertinant situacijas, tinklų elementai (agentai) savybes vertina tolydžiaisiais norminiais dydžiais μ arba **p**. Čia **p**, gali būti tikimybinė išraiška, tik maksimali reikšmė prilyginta maksimaliam dydžiui, pvz., vienetui (1), arba kitokiam norminiam dydžiui M, kurio prasmė – nekategoriška išraiška: ***maža, daug, lygu*** ir panašiai. Miglotosios logikos atvejais priimama, kad svoriniai koeficientai yra lygūs tik priešingų ženklų. Miglotosios logikos yra patraukli metodika vystant intelektines valdymo sistemas. Apskritai kalbant, žmonių sprendimai ar valdymo veiksmai priimami netikslios ir/arba nepilnos sistemos aplinkos informacijos, žinių ir apytikrių skaičiavimų pagrindu. Miglotoje logikoje naudojami taisyklių rinkiniai [14]. Tiek būsenos, tiek išvados čia apibrėžiamos miglotaisiais kintamaisiais, o tiksliau tų kintamųjų aibe (tokios aibės pavyzdys: ypač žemas, labai žemas, žemas, vidutinis, aukštas, labai aukštas, ypač aukštas). Be to miglotųjų taisyklių teisingumas, kitaip vadinamas miglotaisiais teiginiais, taip pat gali būti aprašomas įvairiais miglotaisiais terminais (pavyzdžiui: galbūt, tikėtina, beveik neįmanoma ir pan.). Apskritai sistemų funkcionalumui vertinti ir atlikti skaičiavimus (t.y. procesų modeliavimui, įvertinimui, valdymui bei optimizavimui) reikalingi skaitiniai sprendimai. Taigi miglotosios taisyklės šiuo atžvilgiu nėra tinkamos. Tačiau reikia išskirti tai, kad miglotos logikos teorija leido sukurti efektyvią ir realią metodiką, kuri lingvistinius tvirtinimus ir netikslią informaciją bei žinias paverčia skaitmenine struktūra.

Yra daug įvairių būdų aprašančių miglotųjų kintamųjų aibės priklausomybės funkcijas. Naudojama pasiskirstymo funkcija. Miglotosios aibės elementai gali būti aprašyti atitinkamomis skaitinėmis reikšmėmis. Matematiškai tos ribos išreikštos ribiniu intervalu [0,1]. Tuo pačiu galima išskirti savitą miglotosios logikos bruožą, kuris skiriasi nuo klasikinės logikos ir klasikinių aibių. Čia egzistuoja persidengiančios sritys tarp ribų. Ši savybė parodo, kad tikimybė $P(A \cap A^C)$ nelygi nuliui, tuo tarpu klasikinėje logikoje tokia aibių tikimybė lygi nuliui

Miglotieji valdymo metodai

- Mamdani
- Larsen
- Tsukamoto- šis metodas sutaupo laiko atverčiant reikšmes iš miglotųjų reikšmių.
- Sugeno- generuoja taisykles iš įėjimo ir išėjimo duomenų aibių.

[13]

Miglotosios aibės

Panašiai kaip tradicinėje aibių teorijoje, miglotosios logikos aibėms yra taikomi tokie veiksmai, kaip aibių pjūvis, suma ir paneigimas. Visi veiksmai atliekami ne su pagrindiniais aibės porų elementais, o su priklausomumo aibei reikšmėmis [10].
Miglotose (fuzzy) aibėse taip pat naudojamos panašios operacijos kaip ir klasikinėse aibėse. Tai pagrindinės palyginimo, sąjungos, sankirtos, papildymo ir apjungimo operacijos.

Laikome, kad A ir B yra poaibiai X, kur X = {xi, i=1,n}. Tuomet miglotųjų (fuzzy) aibių operacijos išreiškiamos taip:

Palyginimas: A ir B yra lygios tada ir tik tada, jei jų priklausomybės funkcijos yra lygios:

$m_A(x) = m_B(x),\ x \subset X.$

Apjungimas: B priklauso A, t.y. $A \subset B$, kai $m_A(x) \le m_B(x),\ x \subset X.$

Sąjunga: $A \cup B$ (loginis arba) aprašoma kaip $m_{A \cup B}(x) = \max(m_A(x), m_B(x)),\ x \subset X.$

Sankirta: $A \cap B$ (loginis ir) aprašoma kaip $m_{A \cap B}(x) = \min(m_A(x), m_B(x)),\ x \subset X.$

Papildymas: A papildymas žymimas A^C ir aprašomas: $A^C = \{1 - m_A(x)\},\ x \subset X.$

$$\tilde{A} \cap \tilde{B} = \min(\tilde{A}, \tilde{B})$$

$$\tilde{A} \cup \tilde{B} = \max(\tilde{A}, \tilde{B})$$

Pagrindiniai dėsniai

Komutatyvumo	A (+) B = B (+) A	A (.) B = B (.) A
Asociatyvumo	(A (+) B) (+)C= A (+)(B(+)C)	(A (.) B) (.)C= A (.)(B(.)C)
Neutralumo	A (+) 0 = 0 (+) A = A	A (.) 1 = 1 (.) A= A
Inversijos	A (+) Ā = Ā (+) A ≠0	A (.)A-1 = A-1 (.) A ≠ 1

A (+) B = [a1, a2] (+) [b1,b 2] = [a1+b1, a2 +b 2].

A (-) B = [a1, a2] (-) [b1,b 2] = [a1-b2, a2 -b 1].

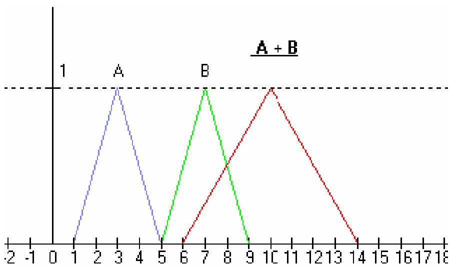
A (v) B = [a1, a2] (v) [b1,b 2] = [a1vb1, a2 vb 2].

A (^) B = [a1, a2] (^) [b1,b 2] = [a1^b1, a2 ^b 2].

A (+) B = [1, 5] (+) [5,9] = [1+5, 5 +9] = [6,14].

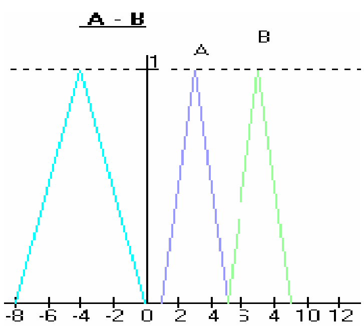
Equivalence ↔(binary), defined as

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$$



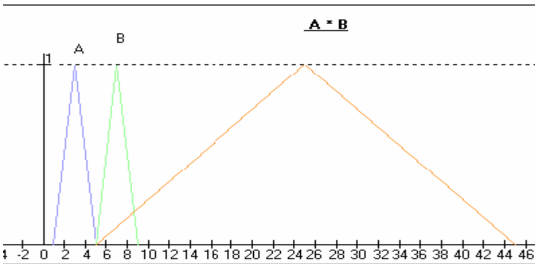
1 pav. Aibių sudėtis

A (+) B = [1, 5] (+) [5,9] = [1+5, 5 +9] = [6,14].



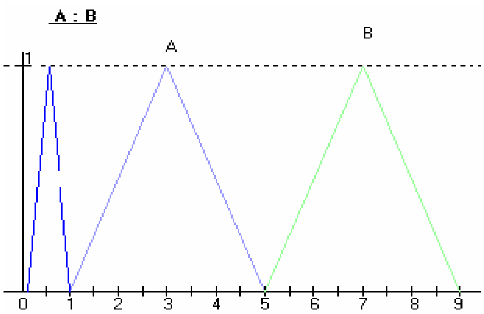
2 pav. Aibių atimtis

A (-) B = [1, 5] (-) [5,9] = [1-9, 5 -5] = [-8,0].



3 pav. Aibių sandauga

A (.) B = [1, 5] (.) [5,9] = [1.5, 5 .9] = [5, 45],

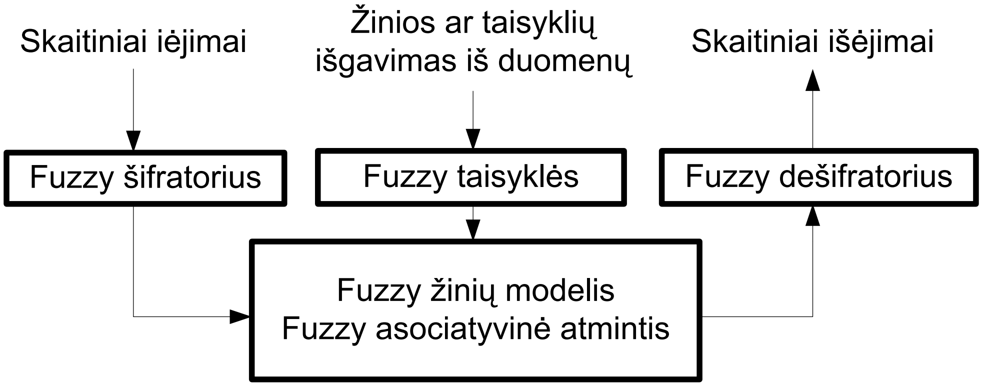


4 pav. Aibių dalyba

A (:) B = [1, 5] (:) [5,9] = [1:9, 5 :5] = [0.1, 1],

Miglotųjų žinių modelis

Miglotųjų žinių modelis gali būti naudojamas tiek sistemos įėjimų ir išėjimų elgsenos aprašymui, tiek sistemos valdymui ir sprendimų priėmimui atlikti. Bendroji miglotųjų žinių modelio architektūra pateikta 6 paveiksle.



5 pav. „Miglotųjų žinių modelio bendroji architektūra“

Miglotos logikos sistemose skaitiniai įėjimų signalai pirmiausiai turi būti paverčiami atitinkamais miglotųjų aibių kintamaisiais. Tai atlieka „miglotieji šifratoriai“. Tokiu būdu gauti miglotieji įėjimai naudojami miglotame (fuzzy) modelyje kur atliekami įvairiausi skaičiavimai ir galiausiai gaunami miglotieji išėjimo signalai. Jie pakeičiami skaitiniais naudojant „miglotuosius dešifratorius“. Miglotame (fuzzy) modelyje įvairiems skaičiavimams atlikti naudojami miglotieji taisyklių rinkiniai. Jie pagrinde kuriami remiantis žmonių žiniomis arba gali būti papildomi naudojant neuroninių tinklų mokymąsi [11].

Visiems priimtina sugretinti „miglotumo“ ir „tikėtinumo“ (fuzziness and probability) sąvokas ir ypatybes naudojant miglotąją logiką. Tiek „miglotumas“ ir „tikėtinumas“ aprašo įvykio netikslumą, kurio reikšmė kinta intervale [0,1]. Tačiau tikėtinumas išreiškia atsitiktinio įvykio pasirodymo netikslumą remiantis dideliu statistinės informacijos kiekiu. Tuo tarpu miglotoji aibė aprašo netikslumą natūralia kalba. Tikėtinumas taip pat gali būti naudojamas miglotosios aibės įvykiams aprašyti. Yra keletas tokių skaičiavimų metodų. Pirmas miglotųjų aibių kintamųjų skaičiavimo tikėtinumais metodą pasiūlė Zedah (1968).

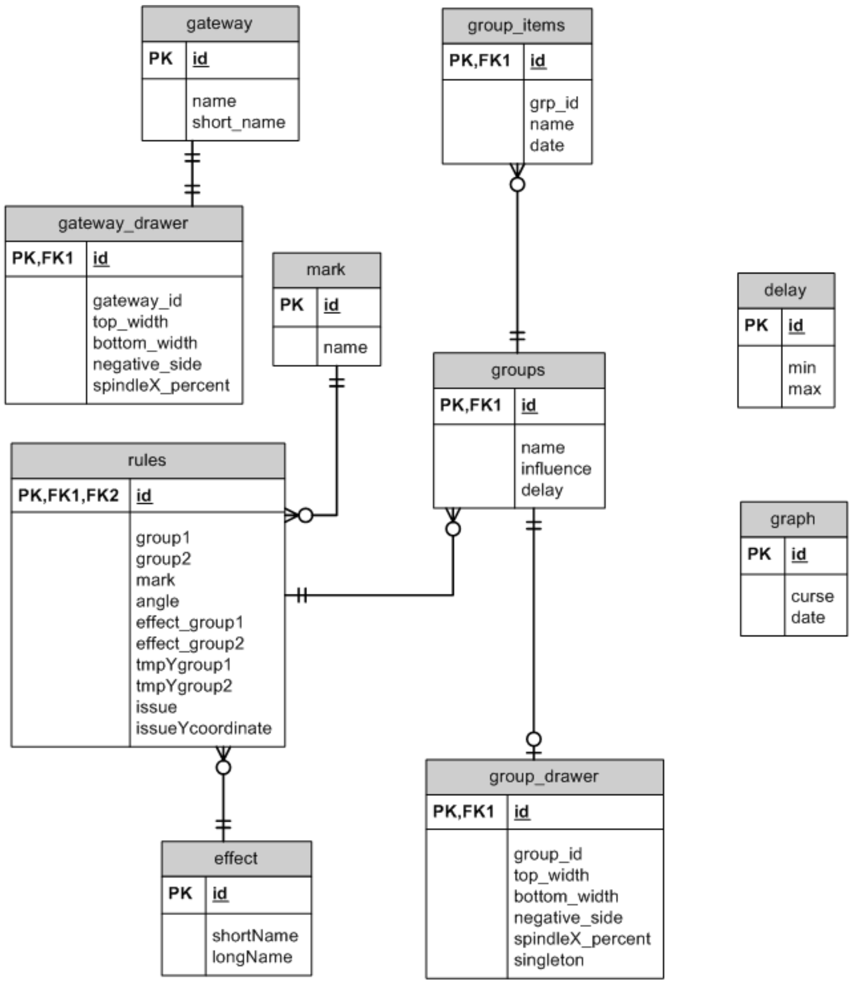
L.H. Tsoukalas ir R.E. Uhrig knygoje „Fuzzy and neural approaches in engineering“ iš esmės aprašomas pasiskirstymas sistemų, kurios pasiekia aukštą našumo lygį naudodamos miglotos logikos metodologijas. Sistemos, kurios naudoja miglotos logikos ir neuroskaičiavimų kombinacijas vadinamos neuromiglotomis (neurofuzzy) sistemomis. Tokias sistemas vieni iš pirmųjų aprašė japonai Takagi ir Hayashi (1988 metais). Šios sistemos, jų teoriniai modeliavimai ir panaudojimas sparčiai vystosi tiek kiekybiniu požiūriu, tiek aprėpiama sritimi, tiek naudojimo svarba. Platesniu požiūriu neuromiglotos sistemos sudaro programiniu principu paremtų sistemų poklasį. Pagrindiniai miglotosios logikos uždaviniai yra: elgsena atsižvelgiant į netikslumus, apytikslis pagrindimas, taisyklėmis paremtos sistemos.

Miglotoji(fuzzy) logika naudojama

- duomenų analizėje
- finansinėse sistemose

2 SISTEMOS ARCHITEKTŪRA

Duomenų bazės reliacinis modelis

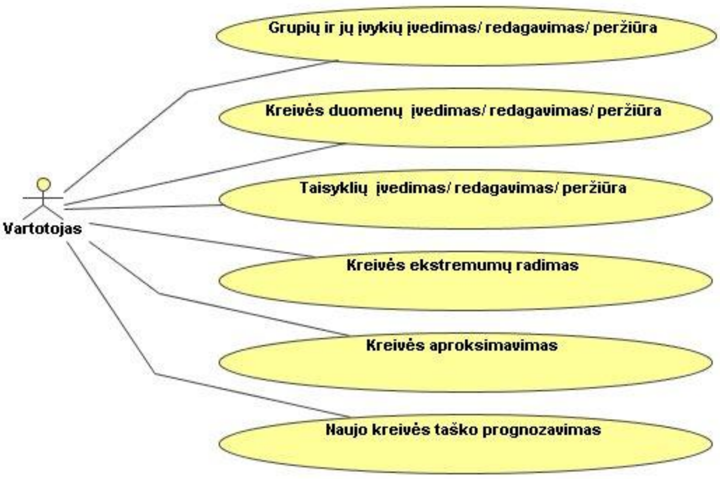


6 pav. Duomenų bazės reliacinis modelis

Sistemos duomenų bazės struktūra	
Duomenų lentelės pavadinimas	**Duomenų lentelės prasmė**
gateway	Saugomi sistemos išėjimai. Tai

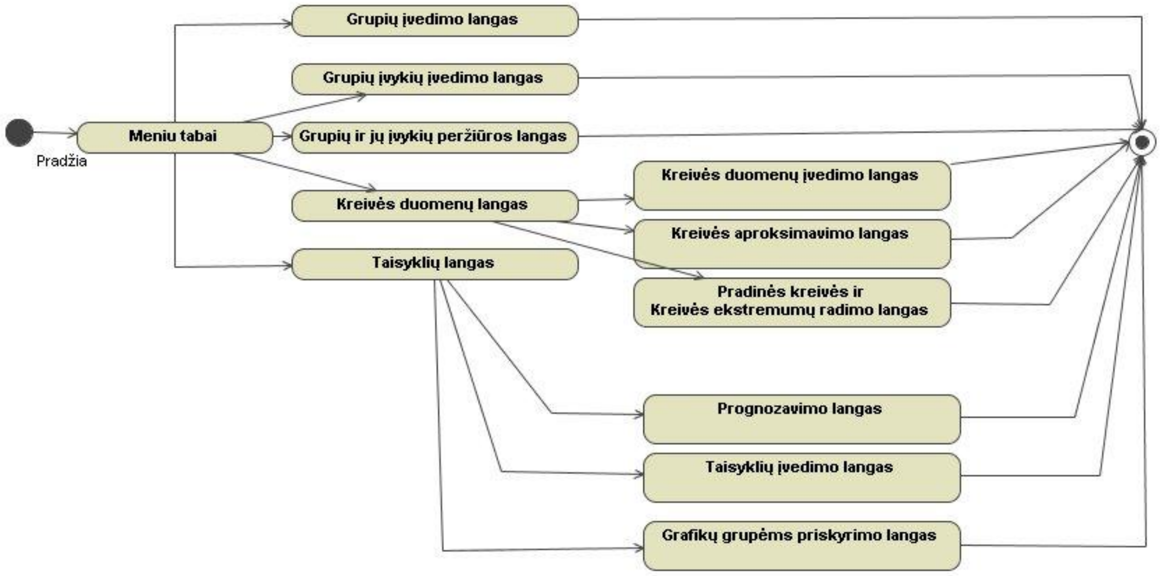
	prognozuojamo kreivės pokyčio kampas ir jo vėlinimas
gateway_drawer	Saugomi sistemos išėjimams priskirtų grafikų parametrai.
delay	Lentelėje saugomi prognozuojamo vėlinimo galimos reikšmės intervalas [min;max]
graph	Saugomos kreivės taškų koordinatės
Kitos sujungtos ryšiais duomenų lentelės saugo miglotųjų taisyklių informaciją: groups- saugomos grupės, group_drawer- grupėms priskirti grafikai, mark- Zadeh loginės operacijos, group_items- grupėms priskirti įvykiai, effect- išsisaugotos galimos miglotosios reikšmės (-D,-V,...,+V,+D)	

Sistemos panaudos atvejų diagrama



7 pav. Panaudos atvejų diagrama

Sistemos veiklos diagrama



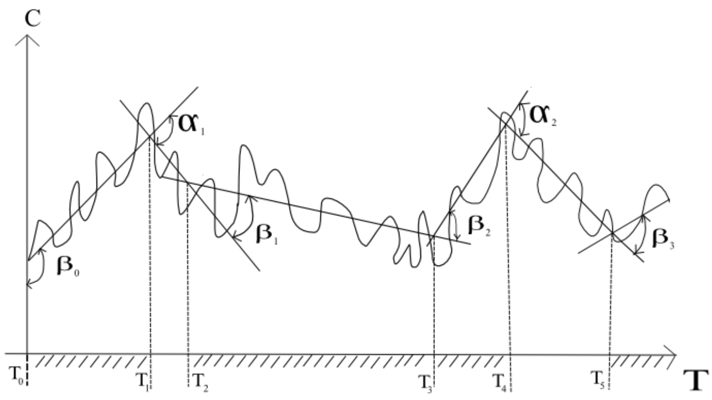
8 pav. Veiklos diagrama

3 SISTEMOS VEIKSENOS PROCESAI

Pirminės žinios

Įvykių istorija H

- Ekspertų numanomos įvykių įtakos $E_k^g(pavadinimas, T_k)$, kur g=1,2,3,..,G k=0,1,...
- Fondų raida A
- Vėlinimas $D_k^g(T_k)$
- Fondo kreivė C(T)



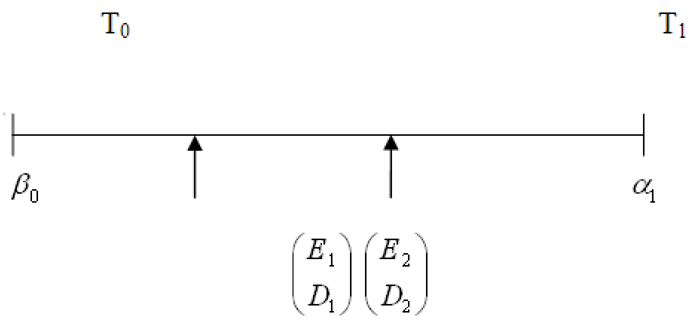
9 pav. Finansinių fondų kreivės kitimas

β - kampas kuriuo kyla kreivė

α - kampas kuriuo leidžiasi kreivė

T - laiko intervalai kai fondo kreivė kyla

Kreivės istorijos intervalai



10 pav. Intervalas nuo kreivės kilimo iki pradėjimo leistis

E_1 – šitam intervalui ekspertų numanomos įvykių įtakos

Sistemos duomenų įvedimas

Įvykių grupių įvedimas. Ekspertai įveda įvykių grupes, kurios teigiamai arba neigiamai veikia fondų kreivę.

- Įvykių grupėms priskyrimas. Įvesti įvykiai priskiriami pasirinktoms grupėms

Fondų kreivės aproksimavimas.

Funkcijų aproksimavimo uždavinių gausu įvairiose matematikos, fizikos ir technikos srityse. Literatūroje nagrinėjama daugybė funkcijų aproksimavimo uždavinio sprendimo metodų. Tai paaiškinama tuo, kad praktikoje susiduriama su daugeliu skirtingų šio uždavinio formuluočių.

Funkcijų aproksimavimo uždavinį galima suformuluoti taip:

turint funkcijos y = f(x) reikmių lentelę (x_i, y_i) (čia i = 0,..,n, $y_i = f(x_i)$

ir $x_i \neq x_j$, kai $i \neq j$), reikia rasti funkcijos f (x) reikšmę, kai x = t, o $t \in [x_0; x_n]$

Funkcijos f(x) analizinė išraiška paprastai yra nežinoma arba pernelyg sudėtinga.

Atsižvelgiant į y_i reikšmių tikslumą, taikomi du šio uždavinio sprendimo metodai:

• interpoliavimo metodas,

• suglodinimo metodas.

Taikomosios diskrečiosios matematikos metodai

Tai kreivės aproksimavimo metodai [16].

Interpoliavimas

Duota funkcijos y = f(x) reikšmių ente1ė (x_i, y_i); čia i = 0,..,n, y_i reikmės yra tikslios arba paklaidos tokios mažos. kad praktiškai jų galima nepaisyti. Reikia rasti aproksimuojančią funkciją y = F(x), priklausančią funkcijų klasei K ir tenkinančią sąlygas

$F(x_i)=y_i$, i=0..n.

Šios sąlygos vadinamos Lagranžo interpoliavimo sąlygomis, o pati funkcija y = F(x) Lagranžo interpoliacine funkcija, arba tiesiog interpoliacine funkcija.

Jei, be $f(x_i)$ reikmių, taške x_i, yra žinomos funkcijos f(x) išvestinių iki (m_i-1)-osios eilės imtinai reikmės, tai galima reikalauti, kad aproksimuojančioji funkcija Y = F(x) tenkintų sąlygas

$F^{(l)}(x_i) = f^{(l)}(x_i)$, i=0..n, l=0..$m_i$-1

Simbolis (l) žymi l-tosios eilės išvestinę. Šios sąlygos vadinamos Ermito interpoliavimo sąlygomis, o funkcija F(x) — Ermito interpoliacine funkcija.

Kaip matyti iš interpoliavimo uždavinio formuluotės, keičiant aproksimuojančiųjų funkcijų klasę K bei interpoliavimo sąlygas, galima rasti įvairias interpoliacines funkcijas.

Interpoliacinių polinomų konstravimas

Tarkime kad taškai x_i (i=0..n) yra skirtingi, t.y. $x_i \neq x_j$, kai $i \neq j$. Tada n-tosios eilės interpoliacinis polinomas egzistuoja ir yra vienintelis. Sakykime, $F(x)=a_n x_n + a_{n-1}x^{n-1}+..+a_1x+a_0$

Yra ieškomasis interpoliacinis polinomas. Jis turi tenkinti sąlygas $a_n x_i^n + a_{n-1}x_i^{n-1}+..+a_1x_i+a_0 =y_i$, i=0..n

Iš šios tiesinės lygčių sistemos apskaičiuojami interpoliacinio polinomo koeficientai. Kadangi sistemos determinantas yra Vandermondo determinantas, kuris nelygus nuliui, kai $x_i \neq x_j$ $(i \neq j)$, tai sistemos sprendinys egzistuoja ir yra vienintelis.

Lagrandžo interpoliacinė išraiška

$$F_i(x) = \frac{(x-x_0)...(x-x_{i-1})(x-x_{i+1})...(x-x_n)}{(x_i-x_0)...(x_i-x_{i-1})(x_i-x_{i+1})...(x_i-x_n)}$$, i=0..n

Ji turi tokią savybę:

$$F_i(x_j) = \begin{cases} 1, kai & i = j \\ 0, kai & i \neq j \end{cases}$$

Čia $j \in \{0,1,..,n\}$

Pasinaudodami polinomais $F_i(x)$, galime iš karto parašyti interpoliacinio polinomo išraišką.

$$F(x) = \sum_{i=0}^{n} F_i(x) \cdot y_i$$

Šis polinomas interpoliacinis, nes jis yra n-tosios eilės polinomas ir $F(x_i)=y_i$ i=0..n Interpoliacinio polinomo išraiška vadinama Lagrandžo interpoliacine išraiška $L_n(x)$

Aitkeno interpoliacinė išraiška

Aitkeno interpoliacinei išraiškai būdinga, tai kad interpoliacinis polinomas neužrašomas išreikštiniu pavidalu, o jo reikšmė apskaičiuojama naudojant antrosios eilės determinantus.

Suglodinimas

Nagrinėdami interpoliavimo uždavinį, funkciją f(x), apibrėžtą reikšmių lentele, stengėmės pakeisti tokia aproksimuojančiąja funkcija F(x) (n-tojo laipsnio polinomu, splainu ir pan.), kad taškuose x_i (i= 0..n) F(x_i) reikšmės būtų lygios f(x_i) reikmėms.

Labai dažnai f(x_i) reikšmės yra eksperimento rezultatai ir turi matavimo bei metodo paklaidų. Todėl reikalauti, kad aproksimuojančioji funkcija F(x) tenkintų sąlygą F(x_i) = f(x_i) (i= 0..n, butų neprotinga. Geriau rasti tokią funkciją F (x), kuri pagal pasirinktą kriterijų geriausiai aproksimuotų f(x). Toks funkcijos F(x) apskaičiavimo metodas vadinamas suglodinimu, Atsižvelgiant į suglodinimo kriterijų, galima gauti įvairias F(x) išraiškas.

Suglodinimo metodo formulavimas

Sakykime, funkcija y=f(x) nusakyta reikšmių lentele ($x_i , \tilde{y}_i$); čia i = 1..m. Simbolis $\tilde{y}_i$ rodo, kad f(x) reikmės taškuose x_i yra apytikslės. Taip pat žinoma aproksimuojančiosios funkcijos F(x) analizinė išraiška: F(x, a_0, ..., a_n); čia a_i (i= 0..n) - nežinomi parametrai ir n << m. Reiškia rasti tokias parametrų a_i reikšmes, su kuriomis F(x) geriausiai aproksimuotų funkciją f(x).

Sprendžiant šį uždavinį, taikomi įvairūs suglodinimo kriterijai, o kartu ir įvairūs suglodinimo metodai:

Pasirinktų taškų metodas

Taikant šį metodą, iš lentelės ($x_i ; \tilde{y}_i$) (čia i = 1..m) pasirenkamas n + 1 taškas ($x_{i_k} ; \tilde{y}_{i_k}$) (čia k = i..n), kurio $\tilde{y}_{i_k}$ reikšmės yra tiksliausios, ir parametrai a_k (k =0..n) apskaičiuojami atsižve1giant į sąlygą

F(x_{i_k}) = $\tilde{y}_{i_k}$ k = 0..n.

Aišku, kad taip apskaičiuota funkcija F(x) sutampa su interpoliuojančiąja funkcija, einančia per taškus ($x_{i_k} ; \tilde{y}_{i_k}$), k = 0..n.

Vidurkių metodas

Šuo metodu parametrai a_k (k=0..n) apskaičiuojami taip, kad $f(x_i)$ ir $F(x_i)$ skirtumų suma būtų lygi nuliui, t. y.

$$z = \sum_{i=1}^{m} (f(x_i - F(x_i)) = 0$$

Mažiausių kvadratų metodas

Naudojant mažiausių kvadratų metodą, kreivė aproksimuojama tiese.

Tai dažniausiai taikomas suglodinimo metodas. Jis formuluojamas taip: koeficientus a_k (k = 0..n) reikia apskaičiuoti taip, kad $f(x_i)$ ir $F(x_i)$ skirtumų kvadratų suma būtų, pati mažiausia, t.y. reikia minimizuoti $z = \sum_{i=1}^{m} (F(x_i, a_0, ..., a_n) - \tilde{y}_i)^2$. Šia formule nusakyta tikslo funkcija turi vienintelį ekstremumą, kuris apskaičiuojamas iš lygčių sistemos $\frac{\delta z}{\delta a_k} = 0$, k=0..n

Bendruoju atveju ši sistema yra netiesinė, taigi ją galima spręsti taikant netiesinių lygčių sistemų sprendimo metodus, išnagrinėtus trečiajame skyriuje. $\frac{\delta z}{\delta a_k} = 0$, k=0..n sistemos formavimą ir sprendimą, galima palengvinti dvejopai:

1) jei F(x, a_0, ..., a_n) yra n-tojo laipsnio polinomas, tai $\frac{\delta z}{\delta a_k} = 0$ lygčių sistema yra tiesinė ir jos formavimas bei sprendimas nesudaro sunkumų;

2) galima taikyti vadinamąjį ištiesinimo metodą, kurio esmė tokia: atitinkamai parinktoje koordinačių sistemoje (X,Y) takai ($x_i ; \tilde{y}_i$) (čia i = 1..m) apytiksliai tenkina tiesės Y = kX + b lygtį, tada mačiausių kvadratų metodas toje sistemoje realizuojamas labai paprastai.

Pavyzdžiui, tarkime, kad taškai ($x_i ; \tilde{y}_i$) tenkina dėsnį

$y = ae^{cx}$. Logoritmuodami abi šios lygybės puses, gauname: ln y = ln a + cx.

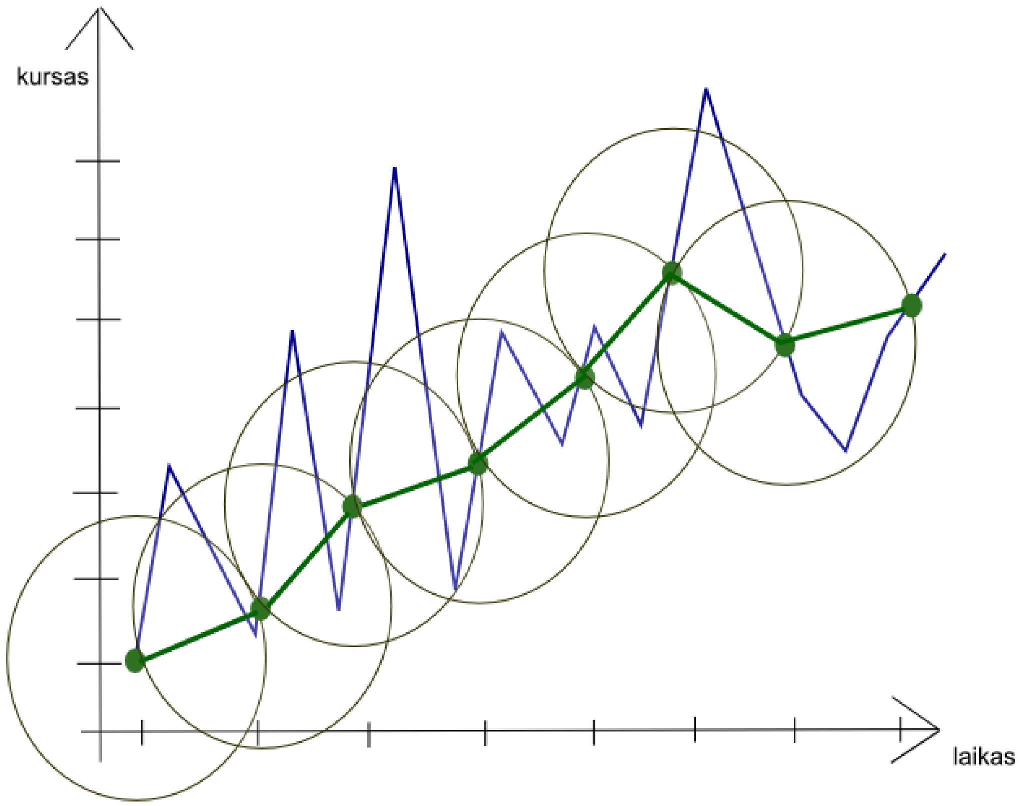
Pažymėkime: X = x, Y = ln y, b = ln a. Tada lygtį $y = ae^{cx}$ koordinačių sistemoje (X,Y) galėsime užrašyti taip: Y = cX + b.

Ištiesinimo metodas gali būti taikomas, norint nustatyti funkcijos

F(x, a_0, ..., $a_{,n}$) analizinę išraišką t. y. sužinoti, kokį dėsnį tenkina taškai ($x_i ; \tilde{y}_i$). Šio metodo esmė yra ta, kad, turėdami testinį koordinačių sistemos (X,Y) rinkinį, ieškome, kurioje koordinačių sistemoje taškai ($x_i ; \tilde{y}_i$) apytiksliai tenkina tiesės Y = kX + b 1ygtį.

Kreivėms aproksimavimas pagal Sklansky ir Gonzalez

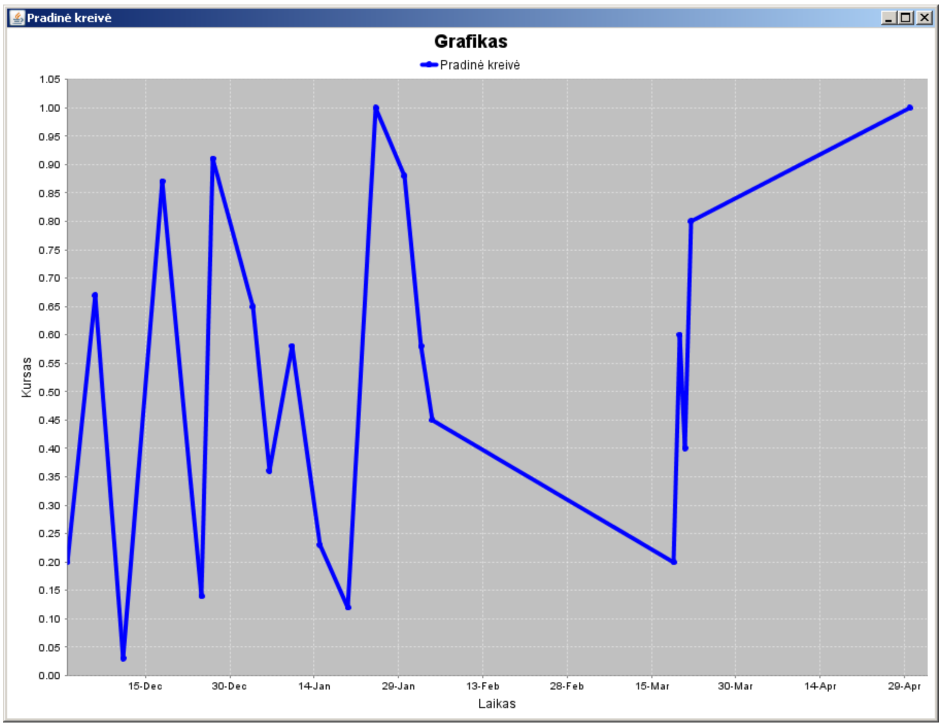
būdas (1). Šio būdo idėja yra brėžti pasirinkto spindulio skritulius, slenkančius linija. Kiekvienas sekantis skritulys prasideda ten kur baigėsi prieš tai buvęs skritulys. Tokiu būdu kreivė suaproksimuojama norimu spinduliu.



11 pav. Kreivės aproksimacija naudojant apskritimą

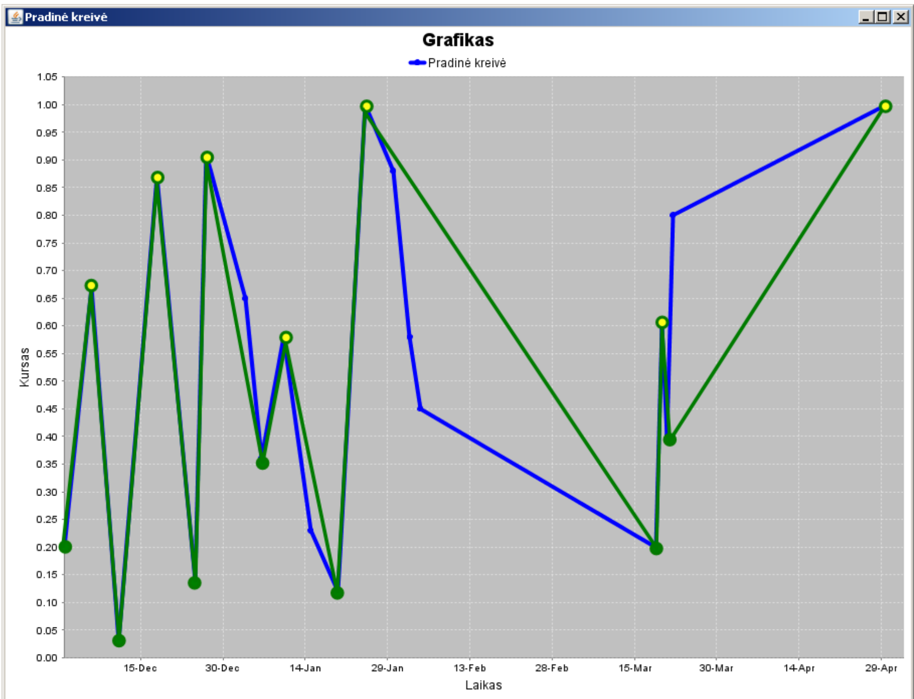
būdas (2). Šis būdas yra panašus į būdą (1). Realizuojant šį būdą pastebėta kad aproksimuotos kreivės taškai persislenka abscisių ašyje, todėl atlikta korekcija. Pagal pradinės kreivės ekstremumų taškus rastieji aproksimuotieji taškai yra pakoreguojami kad taškai nebūtų persislinkę abscisių ašyje.

Tarkime kad pradinė duomenų kreivė yra tokia:

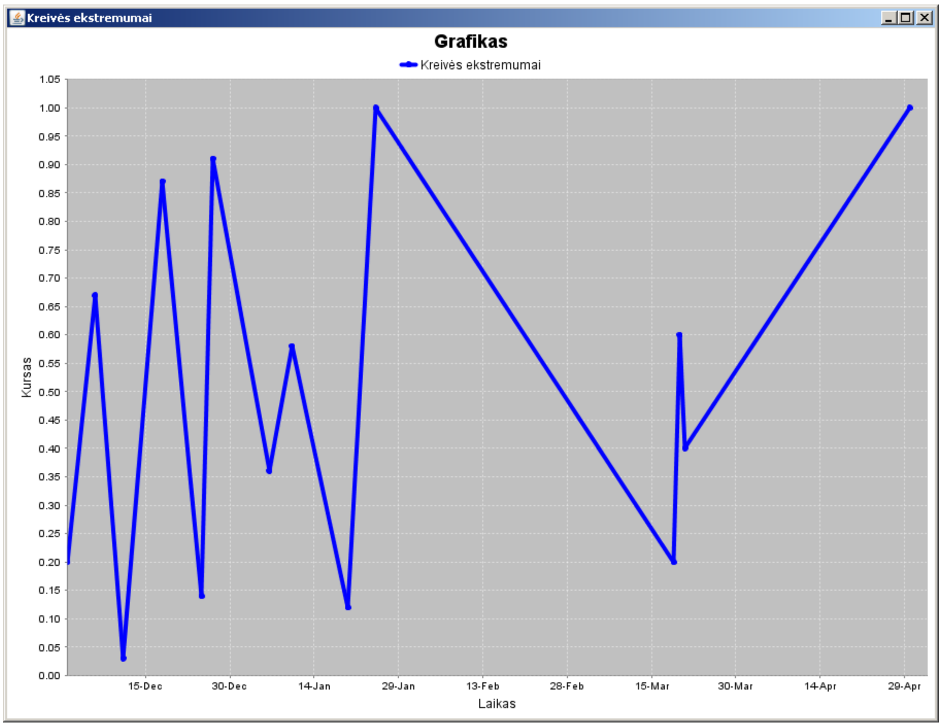


12 pav. Pradinės kreivės grafikas

Pradinės kreivės ekstremumo taškų radimo algoritmas



13 pav. Kreivės ekstremumo taškai



14 pav. Kreivė iš ekstremumo taškų

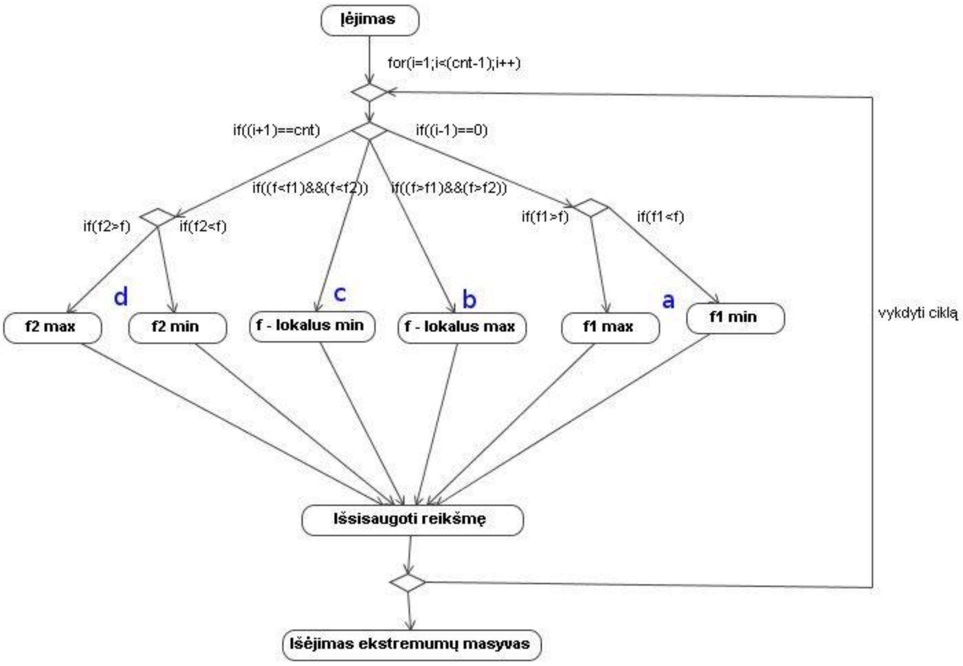
Pasižymėjimai:

cnt- kreivės taškų skaičius

f- (i)-tasis kreivės taškas

f1- (i-1)-masis kreivės taškas

f2- (i+1)-masis kreivės taškas

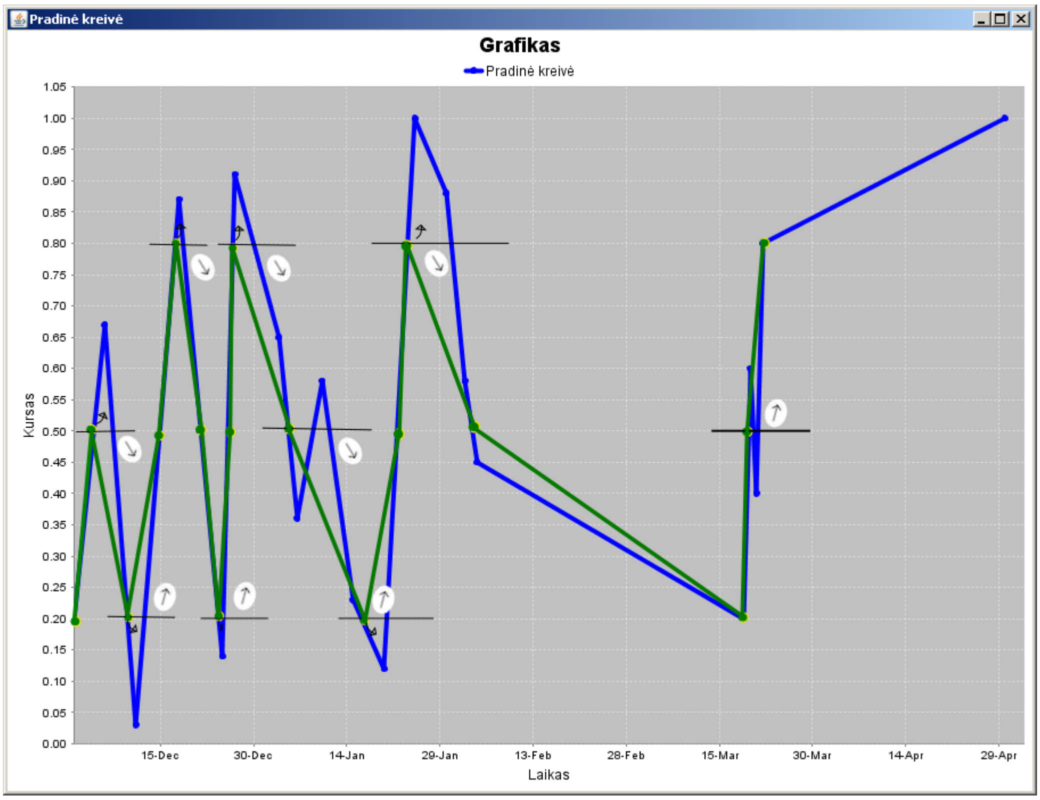


15 pav. Ekstremumų radimo algoritmas

Paaiškinimas:

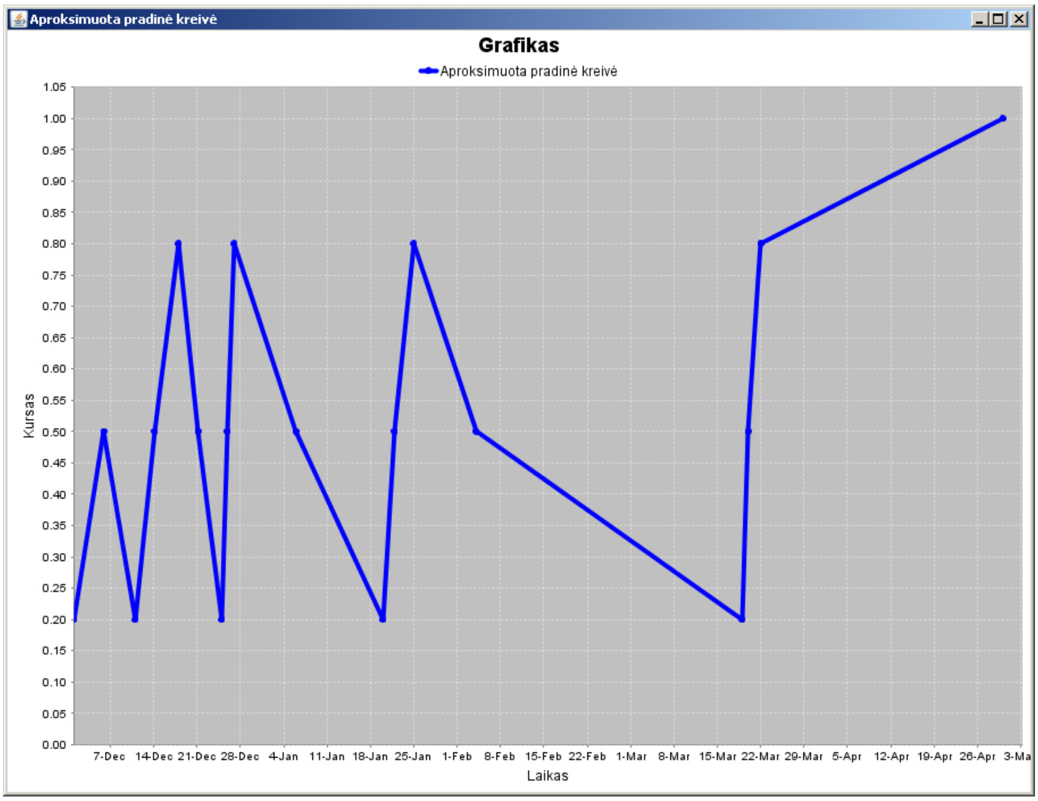
Ciklas vyksta nuo pirmo iki prieš paskutinio kreivės taško. Išskiriamos keturios tikrinimų situacijos (a,b,c,d). a) Jeigu tikrinamasis taškas yra pirmas, tai tikrinama ar nulinis taškas yra mažesnis už jį, jei taip tai nulinis taškas yra minimumas, kitaip nulinis taškas yra maksimumas. b) Vykdant ciklą tikrinama ar i-tasis taškas yra didesnis už (i-1)-ąjį tašką ir didesnis už (i+1)-ąjį tašką, tai rastas lokalus maksimumas. c) Jeigu i-tasis taškas yra mažesnis už (i-1)-ąjį tašką ir mažesnis už (i+1)-ąjį tašką, jeigu taip tai rastas lokalus minimumas. d) Jeigu sekantis i-tajam taškui yra paskutinis kreivės taškas, tikrinama ar tas paskutinis taškas yra daugiau už i-tajį tašką, jeigu taip, tai paskutinis taškas yra maksimumas, kitaip paskutinis taškas yra minimumas.

Pradinės kreivės aproksimavimo algoritmas. (Rasti ekstremumo taškai panaudojami pakoreguoti rastus aproksimuotos kreivės taškus)



16 pav. Kreivės aproksimacija naudojant žingsnį

Aproksimuota pradinė kreivė, kai slenksčiu 0,3. (Tarp žalių kreivės taškų ilgis yra 0,3).



17 pav. Aproksimuota pradinė kreivė kai slenkstis 0,3 rezultatas

Pasižymėjimai:

cnt- pradinės kreivės taškų skaičius

taškai- pradinės kreivės taškų (objektų) rinkinys, kur taškus sudaro kursas, data

K(i)- kursas i-ojo taško

D(i)- data i-ojo taško

K1- nauja apskaičiuota kurso reikšmė

D1- nauja apskaičiuota datos reikšmė

sill- slenkstis kuriuo aproksimuota kreivė

delim- skirtumas tarp dviejų kreivės taškų.

i=0,..,cnt

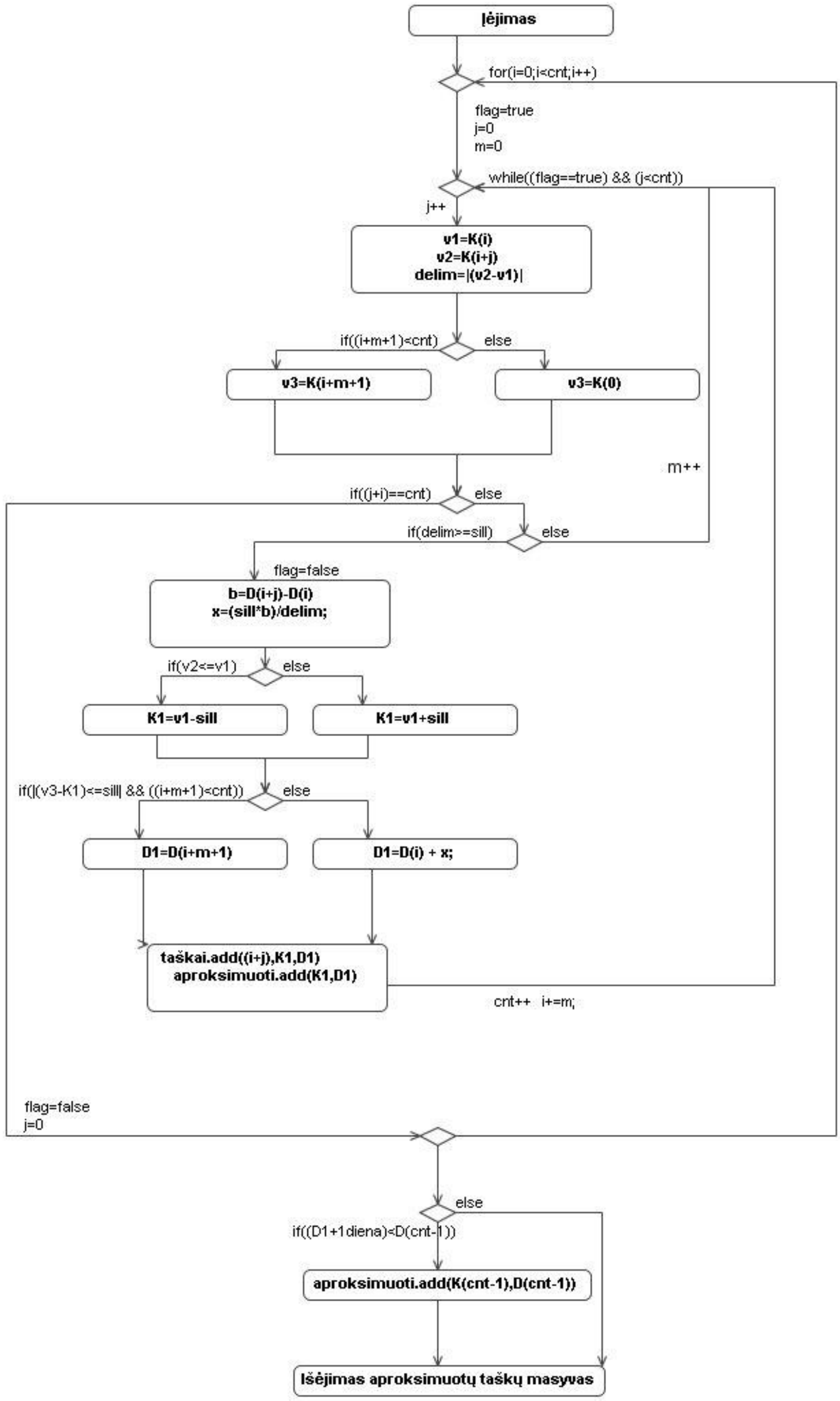
j=0,..,cnt

v1, v2, v3- pradinės kreivės taškų kurso reikšmės x- laiko tarpas kuri reikia pridėti prie D(i), kad gauti naują aproksimuotą kreivės tašką

m- taškų skaičius kurių nereikia aproksimuoti

(i) ir *(i+j)* tai taškai tarp kurių tikrinamas atstumas ar jis lygus slenksčiui ar didesnis už jį.

(i+m+1)- sekantis taškas naujai įdėtam taškui



18 pav. Aproksimacijos algoritmas su taškų korekcija pagal ekstremumus

Paaiškinimas:

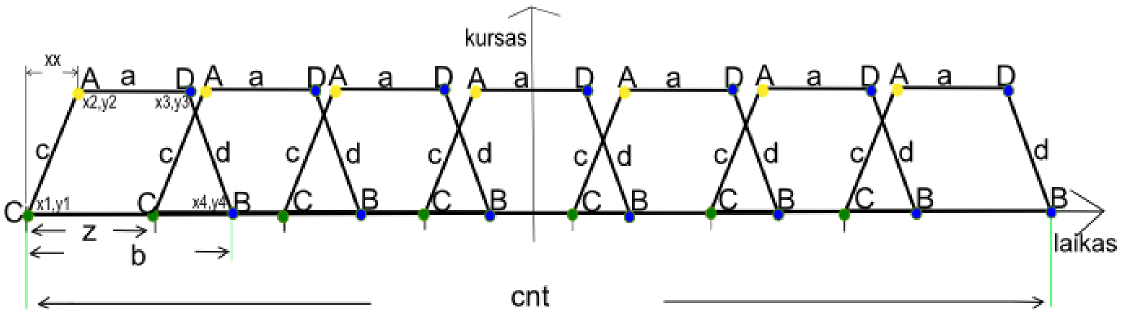
Ciklas vykdomas per visus pradinės kreivės taškus. Jo viduje vykdomas dar vienas ciklas, kuris nuo nagrinėjamo *(i)-tojo* taško ima vis sekantį *(i+j)* kreivės tašką. Randamas nagrinėjamų taškų kursų skirtumas *delim*. Patikrinama ar su *(i)-tuoju* tašku patikrinti visi likę kreivės taškai t.y. *(i+j)* , jeigu taip vidinis ciklas nutraukiamas ir pradedamas nagrinėti sekantis *(i)* taškas. Dar tikrinama ar taškų *(i)* ir *(i+j)* kursų skirtumas *delim* yra didesnis už slenksčio reikšmę. Jeigu skirtumas *delim* yra didesnis tada sukuriamas naujas taškas, kurio data *D1* yra *(i)- tojo* taško data pridėjus laiko tarpą atitinkanti *K(i)+sill* reikšmei. Tačiau jeigu atstumas *(v3)* tarp naujai įdėto taško ir sekančio jam taško *(i+m+1)* yra mažesnis nei slenkstis, tai apskaičiuotajam taškui pakeičiama data į sekančio jam datą *(i+m+1)*. Naujam taškui kursas apskaičiuojamas *(i)-tojo* taško kursą pridėti arba atimti slenksčio reikšmę. Pridėti ar atimti priklauso nuo to ar taško *(i+j)* kurso reikšmė yra didesnė ar mažesnė už taško *(i)* kurso reikšmę. Naujas taškas įdedamas į taškų masyvą *taškai*, kad sekančioje iteracijoje nuo jo galima būtų skaičiuoti, padidinamas taškų skaičius *cnt*, bei įdedamas į aproksimuotų taškų masyvą. Kiekvieną kartą baigus vidinį ciklą *(i)* padidinamas tiek kiek rasta taškų iki kurių atstumas yra mažesnis nei slenkstis t.y. tiek kiek taškų buvo atmesta *(m)*. Gale į aproksimuotų taškų masyvą įdedamas ir paskutinis kreivės taškas, jeigu jo data didesnė nei paskutinio aproksimuoto taško.

Įėjimų išėjimų grafikų parametrų priskyrimas

Miglotosios logikos figūrų dydžių nustatymas. Sistemos įėjimuose ir išėjimuose įvykių grupių poveikių stiprumui (-D, -V, -M, O, M, V, D) atvaizduoti naudojamos miglotosios logikos figūros. Kiekvienai įvykių grupei priskiriamas miglotosios logikos figūrų grafikas. Figūros viršaus a ir apačios b ilgį galima pasirinkti. Kad pasirinkto dydžio figūros tilptų grafiko lange skaičiuojamas žingsnelis z kuris parodo kiek paslinkti sekančia figūrą nuo prieš tai esančios figūros pradžios taško.

Žingsnelio z, kuris leidžia sutalpinti į grafiką pasirinkto dydžio, n figūrų skaičiavimo algoritmas.

Kur n yra 4 arba 7.



19 pav. Atstumo tarp trapecijų radimas

Pasižymėjimai:

c- trapecijos pirma kraštinė

a- trapecijos viršaus plotis ilgis pasirinktas (antra kraštinė)

d- trapecijos trečia kraštinė

b- trapecijos apačios plotis ilgis pasirinktas

z- žingsnelio reikšmė nuo kur pradėti piešti kitą trapeciją

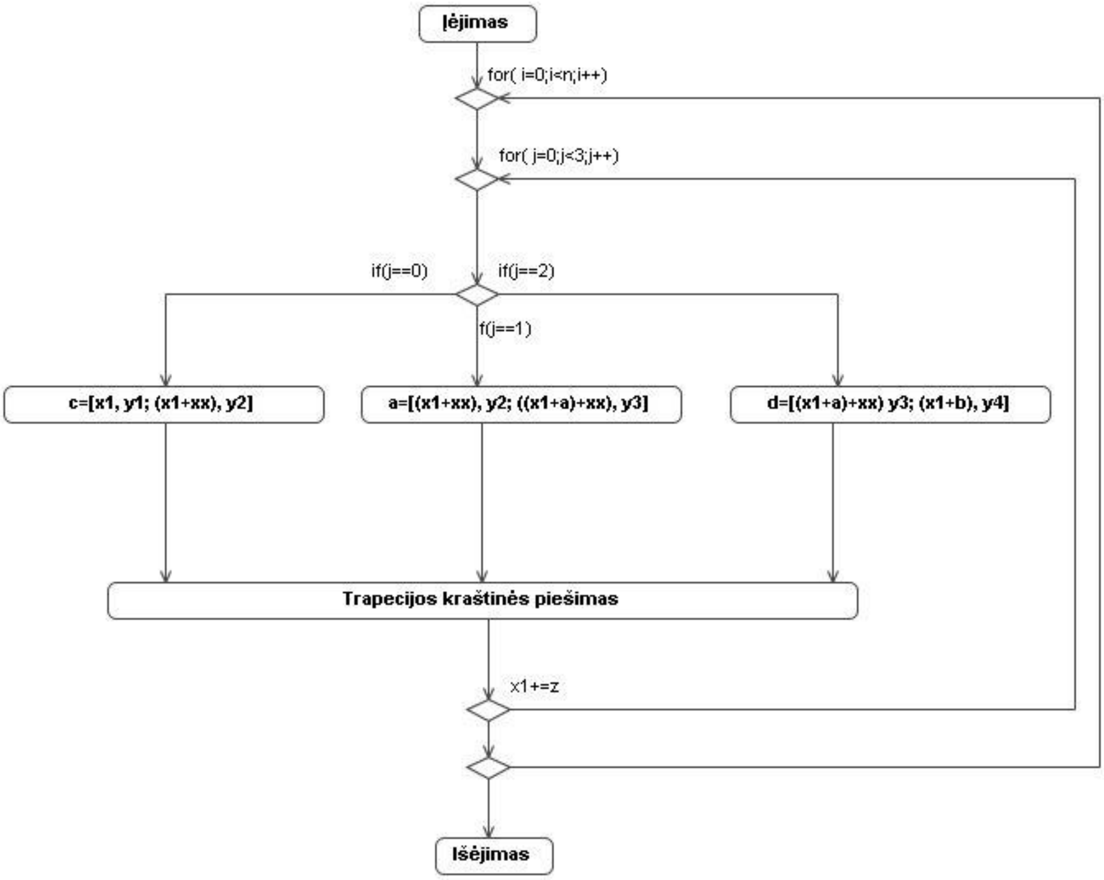
xx- trapecijos viršaus pradžios pozicijos reikšmė

cnt- grafiko abscisių ašies plotis

Turimos reikšmės: x1, y1, y2, y3, a, b, n

Rasti: z ir c, a, d sekančių figūrų koordinates pasislinkusių pirmosios atžvilgiu

- $t = b \cdot (n - 1)$
- $t2 = (cnt - b)/t$
- $z = b \cdot t2$
- $xx = (b - a)/2$



20 pav. n figūrų sutalpinimo į atstumą cnt algoritmas

Paaiškinimas:

Vykdomas ciklas per figūrų skaičių, jo viduje vykdomas ciklas per tris trapecijos kraštines c, a, d. Išskirtos trys sąlygos brėžti trapecijos kraštinėms. Pirmos kraštinės c koordinatės taško C(x1,y1), taško A((x1+xx),y1), nes taško A abscisės koordinatė yra pasislinkusi nuo taško C koordinatės apskaičiuotu ilgiu xx. Kraštinės a taško D abscisės koordinatė yra pasislinkusios nuo taško A abscisės koordinatės per kraštinės a ilgį. t.y. D((x1+a+xx),y2). Kraštinės d taško B koordinatės yra pasislinkę per kraštinės d ilgį nuo taško C abscisės koordinatės t.y. taško B koordinatės ((x1+b),y4). Prieš piešiant naują figūrą *i* pirmos kraštinės c taško C abscisės koordinatė yra perslenkama žingsniu z kuris yra paskaičiuotas prieš tai. Taip nupiešiamos visos n figūros.

- Pasirinkimas ar bus grafike neigiama dalis. Jei yra neigiama dalis miglotosios logikos figūros bus septynios (galimas ir neigiamas poveikis), kitaip keturios (galimas tik teigiamas poveikis). Taip pat jeigu grupę sudarančių įvykių poveikių negalima laipsniuoti t.y. poveikis yra arba nėra, tokioms grupėms priskiriamas grafikas -

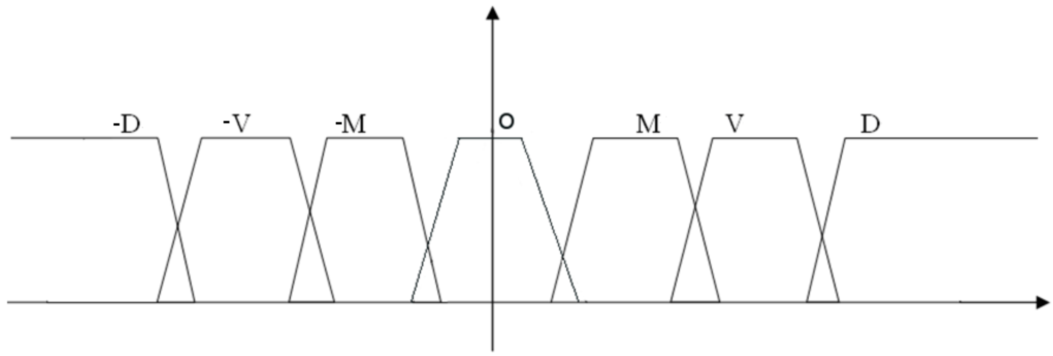
singletonas. Singletonu vadinamas grafikas kuriame yra viena miglotosios logikos figūra, kuri rodo kad poveikis fondų kreivei bus, jei priskirtas singletono grafikas ir figūros nėra reiškia poveikio nebus.

- Įvedamos minimali ir maksimali prognozuojamo taško vėlinimo reikšmės . Vėlinimas įvedamas dienomis. Norint įvesti reikšmes valandomis h, tada vedamas skaičius gautas atlikus veiksmą $h/24$ (valanda yra dalis paros). Jos naudojamos triškaitėje taisyklėje išreikšti atitikmenį intervale [min;max], rastai sistemos išėjime prognozuojamo taško vėlinimo reikšmei.

Taisyklių sudarymas.

Taisyklių sąrašas tai taisyklės, kurios sudarinėjamos stebint įvykių grupes veikiančius fondų kreivės kitimą. Taisyklių sąrašas nuolat papildomas norint gauti tikslesnius prognozavimo rezultatus.

Turint esamu laiku vykstančių įvykių sąrašą, pagal išsaugotas taisykles nustatomi tų įvykių poveikių įverčiai. Tai yra teigiamai ar neigiamai jie pakreiptų fondų kreivės dinamiką ir kiek turėtų įtakos tam pakreipimui: mažai (M), vidutiniškai (V) ar daug (D).



21 pav. Įvykio veikiančio fondų kreivės kitimą galimi įverčiai

-D - įtakoja daug neigiamai

-M - įtakoja mažai neigiamai

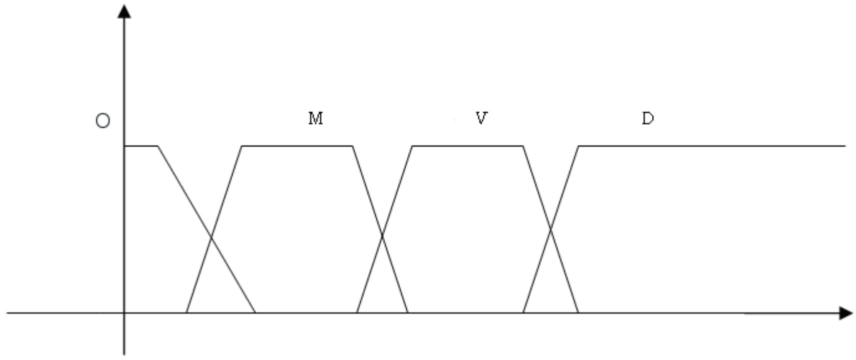
-V - įtakoja vidutiniškai neigiamai

O- neįtakoja

M - įtakoja mažai teigiamai

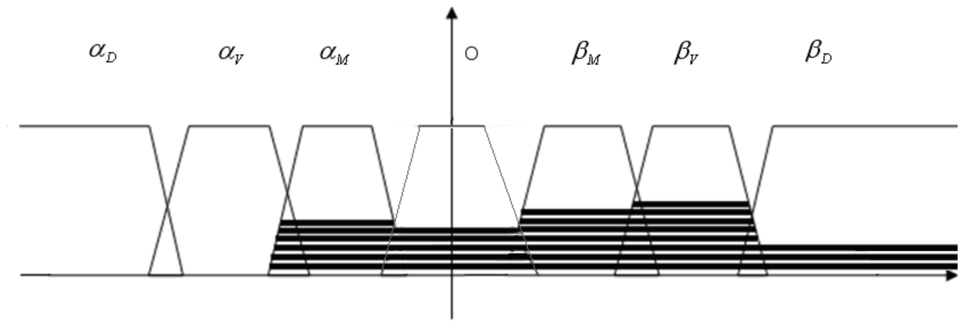
...

Įvykio poveikį charakterizuoja sistemos reakcijos vėlinimas reaguojant į jį.



22 pav. Įvykių veikiančių fondų kreivės kitimą poveikių vėlinimo galimos vertės

- Taisyklių formavimas. Pasirenkamos įvykių grupė1 ir miglotoji reikšmė1. Pasirenkama grupe2 ir miglotoji reikšmė 2. Pasirenkamas loginis operatorius *and* arba *or* sujungti taisyklės sąlygos dalis. Sudarius taisyklės sąlygą nurodomas taisyklės poveikis fondų kreivės kampui ir vėlinimui.
  reikšmė1, reikšmė2- tai įvykių grupės poveikio fondų kreivei dydis ė1 (-D, -V, -M, O, M, V ar D)
  Išraiška atrodo taip:
  IF [grupė1] = reikšmė1 or [grupe2] = reikšmė2 THEN (^)=poveikis kampui, (µ)=poveikis vėlinimui



23 pav. Taisyklių rezultatų rasti poveikiai fondų kreivei kampui (^)

Taisyklių gauti migloti įvertinimai rodo kiek keisis fondų kreivės kampas α_M - mažai neigiamai, o-nesikeis, β_M -mažai teigiamai, β_V - vidutiniškai teigiamai, β_D -daug teigiamai.

Taško prognozavimas

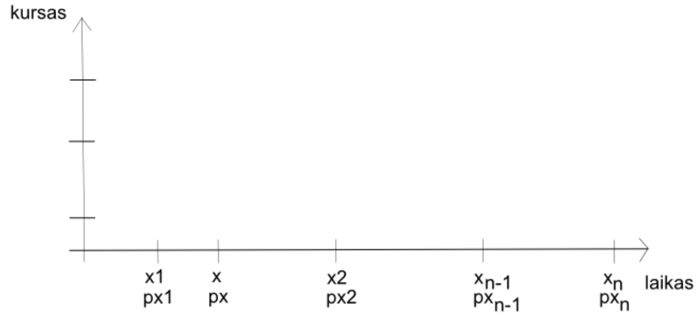
- Įvykių grupių pasirinkimas. Pagal šiandien vykstančius įvykius, iš įvestų grupių sąrašo, pasirenkamos įvykių grupės. Kiekvienai iš jų įvedamos reikšmės *x* kurios žymi poveikio dydį fondų kreivei.

- Taisyklių atranka. Sistema išrenka visas taisykles kuriose paminėtos pastarosios dvi įvykių grupės.

- Miglotųjų atitikmenų radimas iš įvestų duomenų prognozės metu.Pasirinktų grupių įvestoms reikšmėms randamos miglotosios reikšmės (miglotosios logikos figūros kurios yra kertamos ties tomis reikšmėmis ir kirtimo taškų ordinatės *y*).

Reikšmės intervale radimo algoritmas

Taško abscisės radimas kur įvesta reikšmė x kerta „miglotąją“ figūrą.



24 pav. dd

Duoti intervalai $[x_{n-1}, x_n]$, šių intervalų reikšmės pikseliais $[px_{n-1}, px_n]$. Įvesta taško x reikšmė. Norint rasti taško x reikšmę pikseliais *r* atliekami tokie veiksmai [20]:

- Intervalo [x1,x2] ilgis pikseliais h=px2-px1

- Intervalo [x1,x2] ilgis m=x2-x1

- Intervalo [x1,x] ilgis k=x-x1

- Sudaroma proporcija:

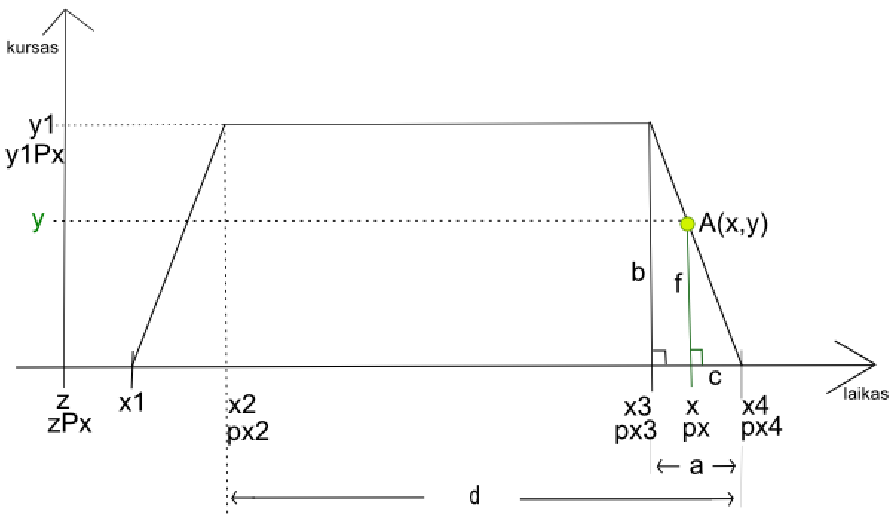
h – m

r – k

k=(h*k)/m

- Tada px=px1+r

Taško A(x,y) kertančio figūrą, y koordinatės radimas turint jo x koordinatės reikšmę algoritmas



25 pav. Taško kertančio figūrą ordinatės reikšmės paieška žinant abscisės reikšmę

Radus kuriam intervalui (Miglotosios logikos figūrai) priklauso įvesta reikšmė x, randama figūrą kertančio taško ordinatės reikšmė y

Jeigu duotos reikšmės x, x1, x2, x3, x4, y1ir pikselių reikšmės px, px2, px3, y1Px tada reikšmė y ieškoma taip:

- Intervalo [x2,x4] *ilgis* d=x4-x2
- Intervalo [x2,x4] *ilgis pikseliais* dPx=px4-px2
- Intervalo [x,x4] *ilgis* c=x4-x
- Taško *x reikšmė pikseliais* išreiškiama proporcija (*cPx*-intervalo [x,x4] ilgis pikseliais*)*:

  d – dPx

  c – cPx

  $cPx = (c \cdot dPx)/d$

  px=px4- cPx

- Statinio *b ilgis pikseliais* bPx=zPx- y1Px
- Statinio *a ilgis pikseliais* aPx=px4-px3
- Randamas statinio *f ilgis pikseliais* iš proporcijos:

$$\frac{f}{bPx} = \frac{\text{cPx}}{aPx}, \text{ tada } f = \frac{bPx \cdot \text{cPx}}{aPx}$$

- Randama taško A y koordinatės reikšmė iš proporcijos:

bPx – y1

f - y

tada $y = \frac{f \cdot y1}{bPx}$

- Atsakymas.: rasta „miglotoji“ figūra taško A koordinatės yra (x,y)

Taisyklių apskaičiavimas. Radus reikšmes ties kuriomis yra kertamos miglotosios logikos figūros, reikšmės įsistatomos į taisykles. Atliekamos loginės Zadeh operacijos taisyklių sąlygose ir gaunami taisyklių rezultatai. Taisyklių rezultatai yra prognozuojamo kreivės kampo pokyčio ir jo vėlinimo miglotosios reikšmės (-D,-M,-V,O,M,V ar D ir miglotosios logikos figūrų kirtimo taškų ordinatės).

- Zadeh operacijų atlikimas. Miglotųjų taisyklių veikimo principai

$$\mu(X \; and \; Y) = MIN(\mu(X), \mu(Y))$$

$$\mu(X \text{ or } Y) = MAX(\mu(X), \mu(Y))$$

$$\mu(not \; X) = 1.0 - \mu(X)$$

$$\mu(X \; and \; \theta) = \theta$$

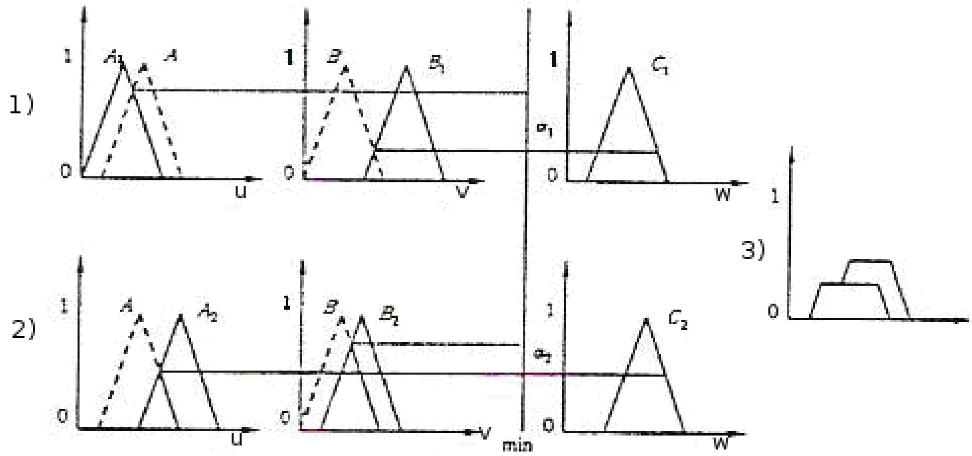
$$\mu(X \text{ or } \theta) = \mu(X)$$

- Veiksmai su loginiu operatorių *min* pagal Mamdani

Paaiškinimas:

Atvaizduotos dvi taisyklės numeriais 1) ir 2). Jų rezultatai pateikti numeriu 3). Abi taisyklės 1) ir 2) atlieka operacija AND t.y. gautas rezultatas yra dviejų reikšmių minimumas.

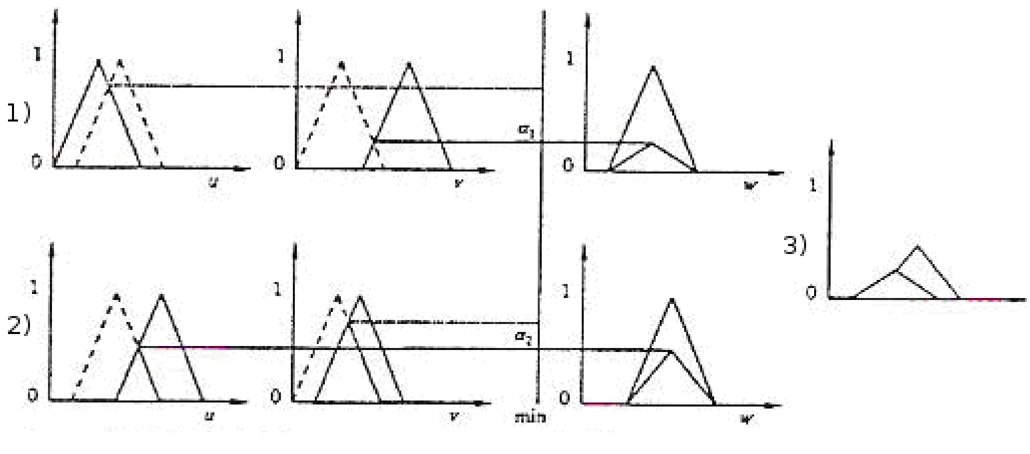
(paveiksle taisyklių rezultatai pavaizduoti raidėmis c_1 ir c_2). Numeriu 3) taisyklių 1) ir 2) rezultatai atvaizduoti viename grafike.



26 pav. Veiksmai su loginiu operatorių min pagal Mamdani

- Veiksmai su loginiu operatorių *min* pagal Larsen

Šis būdas realizuotas sistemoje. Jis panašus į Mamdani. Skiriasi tik tuo kad išlaikoma figūrų formos proporcija imant reikalingą figūros aukštį.



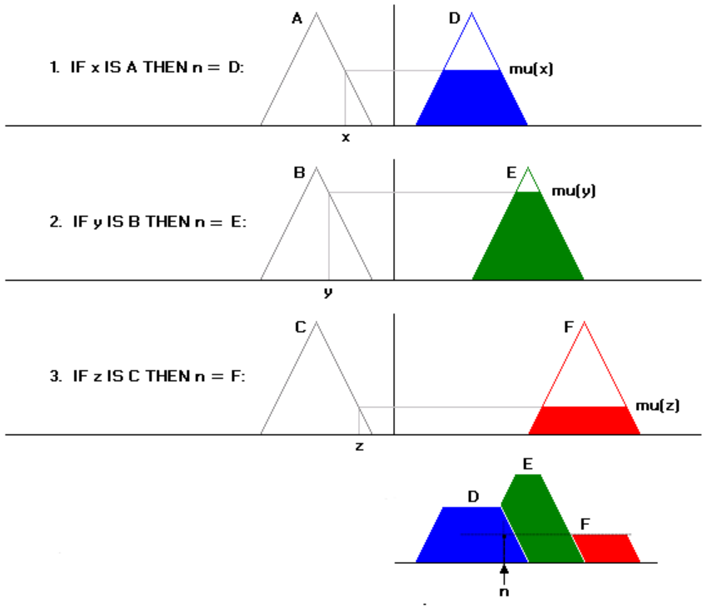
27 pav. Veiksmai su loginiu operatorių min pagal Larsen

- Miglotųjų taisyklių rezultatai

Jeigu taisyklių rezultatai rodo skirtingus miglotosios logiko figūrų aukščius, taip kaip atvaizduoti taisyklių rezultatai numeriu 3). Rezultatai apjungiami su loginiu operatoriumi *or*, kuris išrenka maksimalias reikšmes t.y. rezultatas yra didesnė figūra.

- Taisyklių rezultatų atvaizdavimas. Nubrėžiami prognozuojamo fondų kreivės pokyčio kampo ir vėlinimo grafikai. T.y. atvaizduojamos pagal taisyklių rezultatus atrinktos miglotosios reikšmės (-D,-M,...,D), jų aukštį paėmus pagal rastą kirtimosi *y* koordinatę.

Grafiko svorio centro radimas. Suskaičiuojami taisyklėse rastų prognozuojamo fondų kreivės kampo ir vėlinimo pokyčio miglotųjų reikšmių grafikų svorio centrai [19].



28 pav. Determinatorius , n- taisyklių rezultatų svorio centras[5]

- Randamas visas figūros DEF plotas *S*, atsižvelgiant kaip trapecijos susikirtę Norint rasti trapecijų kirtimosi taškus randamos trapecijų kraštinių lygtys. Paskui randami trapecijų kirtimosi taškus.

Tiesės lygties radimo, turint du jos taškus (x1,y1) ir (x2,y2) , algoritmas

- Jeigu tiesės lygties išraiška yra tokia [18]:
  $\mathrm{y} = \mathrm{sk} \cdot \mathrm{x} + \mathrm{sk2}$, kur sk, sk2 yra $(-\infty;\infty)$,

- tai iš tiesės lygties apibrėžimo $\frac{y-y1}{x-x1} = \frac{y2-y1}{x2-x1}$, kur $\frac{y2-y1}{x2-x1} = sk$, tada

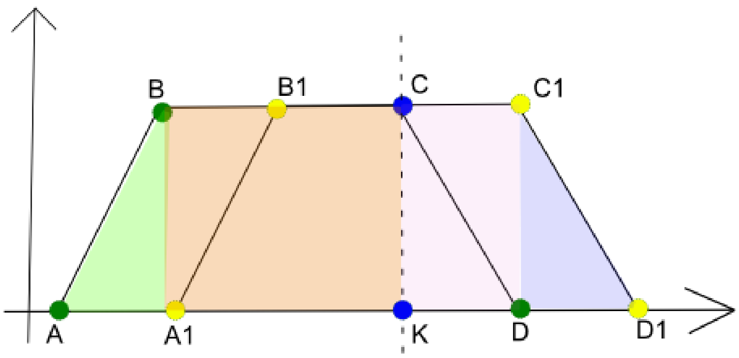
- $\frac{y-y1}{x-x1} = \frac{sk}{1}$, tada $(y-y1)\cdot 1 = (x-x1)\cdot sk$, iš čia $y - y1 = sk\cdot x - sk\cdot x1$,

- tada perkėlus kintamuosius į vieną pusę, skaičius į kitą gaunama $y - skx = y1 - sk \cdot x1$, čia $y1 - sk \cdot x1 = sk2$ t.y. sk2 yra skaičius(visi kintamieji žinomi)

- todėl tiesės lygtis $y = sk \cdot x + sk2$, kur $sk = \dfrac{y2 - y1}{x2 - x1}$, o $sk2 = y1 - sk \cdot x1$

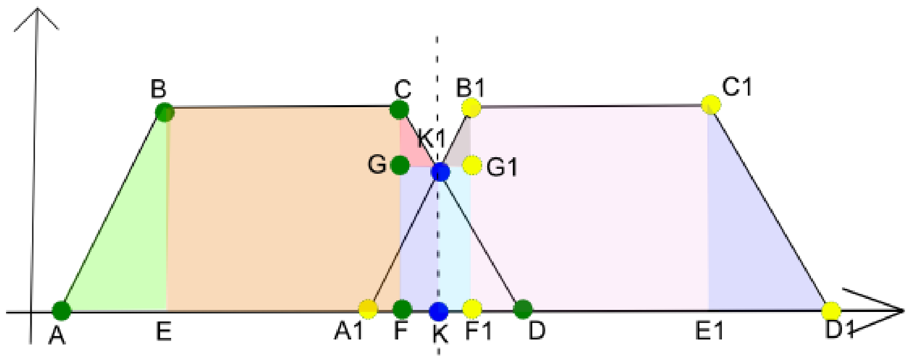
Trapecijų kirtimosi būdai. Skaičiuojant susikirtusių trapecijų plotą S [17] išskiriami keli kirtimosi variantai:

- Kai trapecijos vienodo aukščio ir pirmos trapecijos taškas C yra tarp antros trapecijos B1 ir C1 taškų $S = \Delta ABA1 + A1BCK + KCC1D + \Delta DC1D1$



29 pav. Trapecijų kirtimosi Būdas (1)

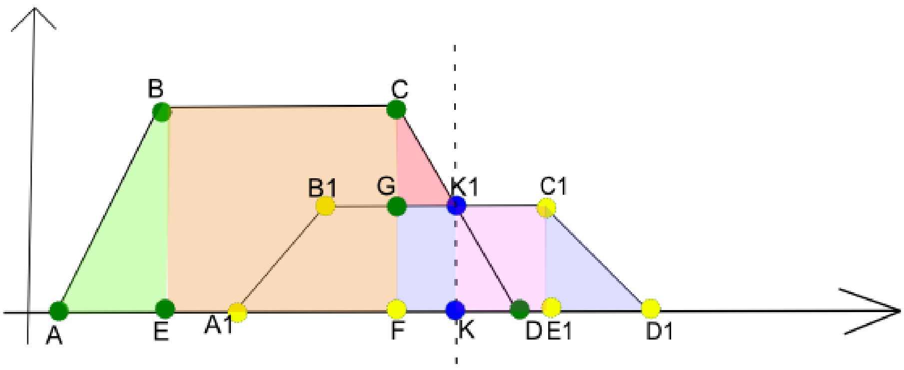
- Jeigu figūros kertasi šoninėmis kraštinėmis $S = \Delta ABE + EBCF + \Delta CGK1 + FGK1K + \Delta K1B1G1 + KK1G1F1 + F1B1C1E1 + E1C1D1$



30 pav. Trapecijų kirtimosi Būdas (2)

- Jeigu figūros kertasi šonine ir viršutine kraštinėmis

  $S = \Delta ABE + EBCF + \Delta GCK1 + FGK1K + KK1C1E1 + \Delta E1C1D1$

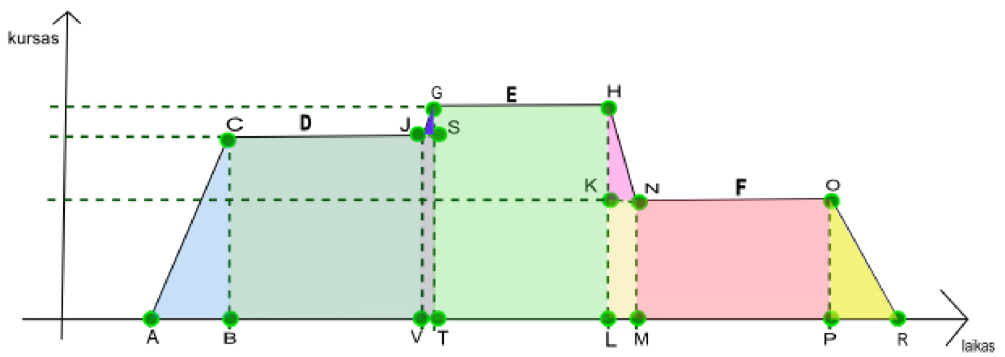


31 pav. Trapecijų kirtimosi būdas (3)

Figūros DEF ploto ir svorio centro radimas. Ši figūra skaidoma į trikampius ir stačiakampius, tada surandamas jos plotas.

- Δ - žymi trikampį

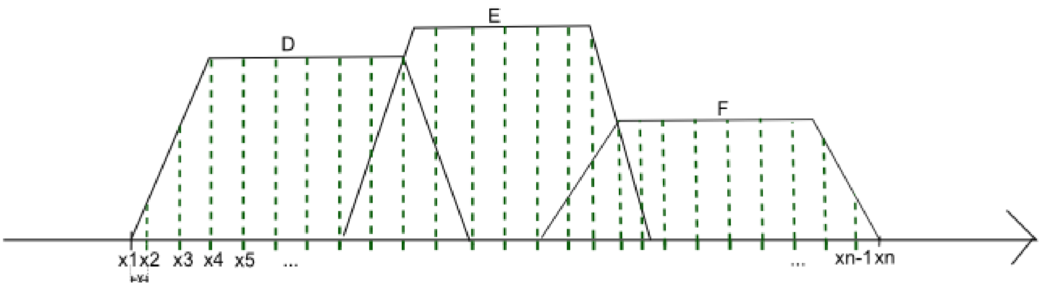
  $S = \Delta ACB + BCJV + \Delta JGS + VJST + TGHL + LKNM + \Delta KHN + MNOP + \Delta POR$



32 Figūros ploto skaičiavimas.

- Randama pusė figūros DEF ploto $p = \frac{S}{2}$, kad žinoti kokiomis dalimis pasiskirstęs svorio centras.

- Žingsneliu x skaičiuojamas ir sumuojamas trapecijų plotas (pagal tai kaip kertasi trapecijos), tol kol jis tampa lygus pusei figūros DEF ploto p. Taip randamas figūros svorio centras.



33 pav. Svorio centro paieška

Duoto pavyzdžio rezultatas yra figūros DEF svorio centras [5].

Vėlinimo atitikmens radimas įvertinus galimą intervalą [min;max].

Pasižymėjimai:

max- maksimali įvesta vėlinimo reikšmė

min- minimali įvesta vėlinimo reikšmė

t- vėlinimo grafiko (išėjime) maksimali abscisių ašies reikšmė

f- vėlinimo grafiko (išėjime) rasta vėlinimo reikšmė

sudaroma triškaitė taisyklė rasti prognozuojamo taško vėlinimo reikšmę x:

max – t

x – f

$x = (f \cdot \max) / t$

Jeigu x yra mažiau už minimalią vėlinimo reikšmę min, tada x=min.

Kampo atitikmens radimas įvertinus galimą intervalą [-90;90].

Pasižymėjimai:

t- prognozuojamo kampo grafiko (išėjime) maksimali abscisių ašies reikšmė

f- kampo grafiko (išėjime) rasta kampo reikšmė

sudaroma triškaitė taisyklė prognozuojamo kreivės pokyčio reikšmei x rasti:

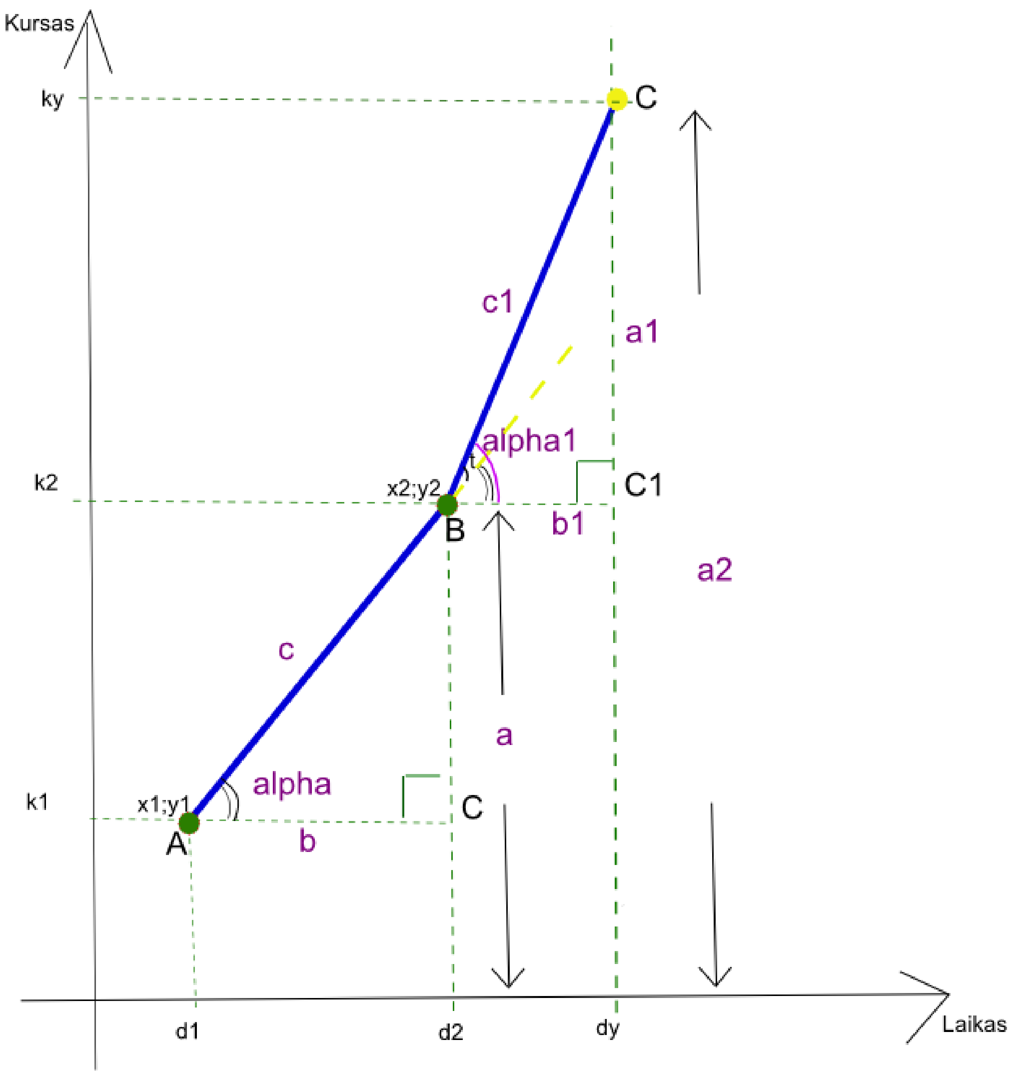
t - 90

f - x

$x = (f \cdot 90)/t$

Jeigu x< -90 tai x=-90

Prognozuojamo taško koordinačių skaičiavimas. Suskaičiavus prognozuojamo taško vėlinimo reikšmę ir kampo reikšmę kuriuo pasikeis kreivė randamos prognozuojamo taško koordinatės [21].



34 pav. Prognozuojamo taško C koordinačių skaičiavimas

Pasižymėjimai:

C- prognozuojamas kreivės taškas

(dy, ky)- prognozuojamo kreivės taško koordinatės

A, B- paskutiniai kreivės taškai

(x1,y1)- taško A koordinatės pikseliais

(d1,k1)- taško A koordinatės data ir kursas

(d2,k2)- taško B koordinatės data ir kursas

(x2,y2)- taško B koordinatės pikseliais

a,b,c trikampio ΔABC kraštinės

a1,b1,c1 trikampio ΔA1B1C1 kraštinės

alpha- paskutinis kreivės pasvirimo kampas

t- kampas kuriuo pasikeis kreivės pasvirimas

alpha1- prognozuojamas kreivės pasvirimo kampas

Algoritmas:

- Kraštinės *c ilgis pikseliais* iš kraštinės tarp dviejų taškų formulės $c = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$
- Statinio *b ilgis pikseliais* $b = x2 - x1$
- Statinio *b ilgis dienomis* $\mathrm{skirtumas} = \mathrm{d2} - \mathrm{d1}$
- Randamas atstumas *a (ilgis pikseliais)* nuo paskutinio kreivės taško B iki abscisių ašies. Jis yra iš grafiko lango aukščio pikseliais atėmus taško B ordinatės y2 reikšmę $a = 580 - y2$
- Randamas kampo *alpha dydis laipsniais* skaičiuojama taip kaip šioje nuorodoje http://www.easycalculation.com/trigonometry/triangle-angles.php $alpha = (ar\cos(b / c) * 180 / PI * 100) / 100$

- Randamas naujas prognozuojamas kreivės kampas *alpha1* (prie paskutinio kreivės pasvirimo pridedamas apskaičiuotas kreivės pokyčio kampas t) $alpha1 = alpha + t$

  alpha1 turėtų būti intervale [-90;90], gavus alpha1 reikšmę nepatenkančią į šį intervalą ji yra pakeičiama. Jeigu alpha1 daugiau nei 90, tai priskiriama alpha1=90, jeigu alpha1 mažiau nei -90, priskiriama alpha1=-90

- Randama naujo prognozuojamo taško *C apscisės reikšmė timestamp*. Prie paskutinio kreivės taško datos *d2* pridedant vėlinimo dienų skaičių.
  (Veiksmai su dienomis atliekami dienas verčiant į skaičius (*timestamp*).
  (1 *diena*=1*24*60*60=86400 *timestamp*))
- Naujo taško vėlinimas dienomis paverčiamas į *timestamp*:
  $\mathrm{timetamp1}(p) = p * 86400$

- Paskutinio kreivės taško data *d2* paverčiama į *timestamp*:
  $\mathrm{timetamp2}(d2) = d2 * 86400$
- Taško *C apscisės reikšmė timestamp*: $\text{vėlinimas} = \mathrm{timestamp2} + \mathrm{timestamp1}$
- Taško *C apscisės reikšmė konvertuota į (data)*: dy= vėlinimas/86400
- Kadangi viskas kreivės grafike skaičiuojama pikseliais, naujo prognozuojamo taško statinio *b1 ilgis išreiškiamas pikseliais*:
- Iš statinio b ilgio dienomis konvertuotomis į timestamp ir jo ilgio pikseliais, randama kiek *viena diena atitinka pikselių*.
  skirtumas=skirtumas*86400
  dienaPx= b/skirtumas
- Statinio *b1 ilgis pikseliais*: $b1 = p * dienaPx$
- Randamas naujo prognozuojamo taško atstumas *a2 pikseliais* iki abscisių ašies:
  Iš formulės $\mathrm{tg(alpha1)} = a1/b1$, išreiškus a1 yra $a1 = \mathrm{tg(alpha1)} * b1$, tada $a2 = a + a1$
- Turint statinio *a1 ilgį* pikseliais randama jį atitinkanti *kurso reikšmė*. Naudojama triskaitė taisyklė statiniams a, a1 pikseliais prilyginus jų kurso reikšmes.
  a – k2
  a1 - ky

  $ky = (a1 * k2)/a$
- Atsakymas prognozuojamo taško C koordinatės.: (dy;ky)

- Papildomo taško paskaičiavimas. Senoji kreivė (tiesė) tęsiama buvusiu kampu iki naujos datos (senoji+vėlinimas) ir tik tame taške atidedamas (pridedamas/atimamas) naujas kampo prieauglis.
- Kreivės aproksimavimas radus prognozuojamą tašką. Kreivės taškai ir naujas taškas aproksimuojami pasirinktu slenksčiu.
- Skaičiavimų rezultatas. Skaičiavimo proceso metu atlikus tarpinių rezultatų komponavimą ir agregavimą gautas naujas prognozuojamas fondų kreivės taškas, kuris parodo kaip kreivė keisis ateityje.

IŠVADOS

Šiame darbe analizuoti matematinės statistikos prognozavimo metodai AR, Wiener, Bayeso, Soros. Aptartos ekonominių matematinių modelių grupės ir kiti prognozavimo metodai nesiremiantys matematine statistika. Pasiūlytas prognozavimo metodas paremtas miglotąja logika. Analizuoti kreivės aproksimavimo būdai naudojant interpoliavimo ir suglodinimo metodus. Taip pat analizuotas Sklansky ir Gonzalez aproksimavimo algoritmas, kuris buvo realizuotas jį pamodifikavus. Sudaryti algoritmai, realizuoti visi sistemos, prognozuojančios naują fondų kreivės tašką, kūrimo proceso etapai:

- kreivės ekstremumų taškų radimas;
- taško A(x,y) kertančio figūrą, y koordinatės radimas turint jo x koordinatės reikšmę;
- žingsnelio z, kuris leidžia sutalpinti į grafiką pasirinkto dydžio, n figūrų apskaičiavimas;
- turint tiesės du taškus (x1,y1) ir (x2,y2) , jos lygties radimas;
- figūrų svorio centro skaičiavimas;
- figūrų kirtimosi būdų ir susikirtimo taškų radimas;
- veiksmų su Zadeh operacijomis OR, AND atlikimas;
- vėlinimo ir kampo atitikmenų radimas įvertinus galimą intervalą [min;max];
- prognozuojamo taško koordinačių radimas, atskaitos taškais laikant paskutinius du kreivės taškus.
- ir kiti

Pasiūlytas prognozavimo metodas remiasi miglotąja logika, tai leidžia prognozuoti kreivės pokyčius naudojant ekspertų žinias, patirtį ir intuiciją, kurias galima pateikti žmogui suprantamais teiginiais.

LITERATŪRA

1. Rosetta Stone Secret [žiūrėta 2009 gegužės 2 d.], prieiga internete http://www.timingsecrets.com/index.php

2. Ski gold stoks prediction [žiūrėta 2009 gegužės 6 d.], prieiga internete http://www.skigoldstocks.com/about.php

3. ECON [žiūrėta 2009 gegužės 7 d.], prieiga internete http://www.pricepatternprediction.com/ga01010.htm

4. TIMESTAT [žiūrėta 2009 gegužės 20 d.], prieiga internete http://www.pricepatternprediction.com/ga01011.htm

5. TSE [žiūrėta 2009 gegužės 21 d.], prieiga internete http://www.pricepatternprediction.com/ga01039.htm

6. Wiener metodas [žiūrėta 2009 gegužės 5 d.], prieiga internete http://en.wikipedia.org/wiki/Wiener_process

7. http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/ar/

8. http://www.dailyreckoning.co.uk/commodities-trading/george-soros-investment-philosophy.html

9. http://en.wikipedia.org/wiki/Bayesian_probability

10. Fuzzy Sets [žiūrėta 2009 gegužės 5 d.], prieiga internete http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/jp6/article2.html

11. L.H. Tsoukalas, R.E. Uhrif, Fuzzy and neural approaches in engineering, 1996

12. Neurotinklas [žiūrėta 2009 gegužės 6 d.], prieiga internete http://www.biofizika.gf.vu.lt/files/uploaded/neuroinformatika/skyrius8.html

13. Category:Fuzzy logic [žiūrėta 2009 gegužės 12 d.], prieiga internete http://www.infm.ulst.ac.uk/~siddique/CI/CI-Week2.pdf

14. Fuzzy Control System [žiūrėta 2009 gegužės 13 d.], prieiga internete http://en.wikipedia.org/wiki/Fuzzy_Control_System

15. _Zadeh, L.A. "Fuzzy algorithms. *Information and Control* , 1968, 12 (2): p.94–102.

16. K. Plukas Skaitiniai metodai ir algoritmai: vadovėlis. Naujasis Lankas, 2001

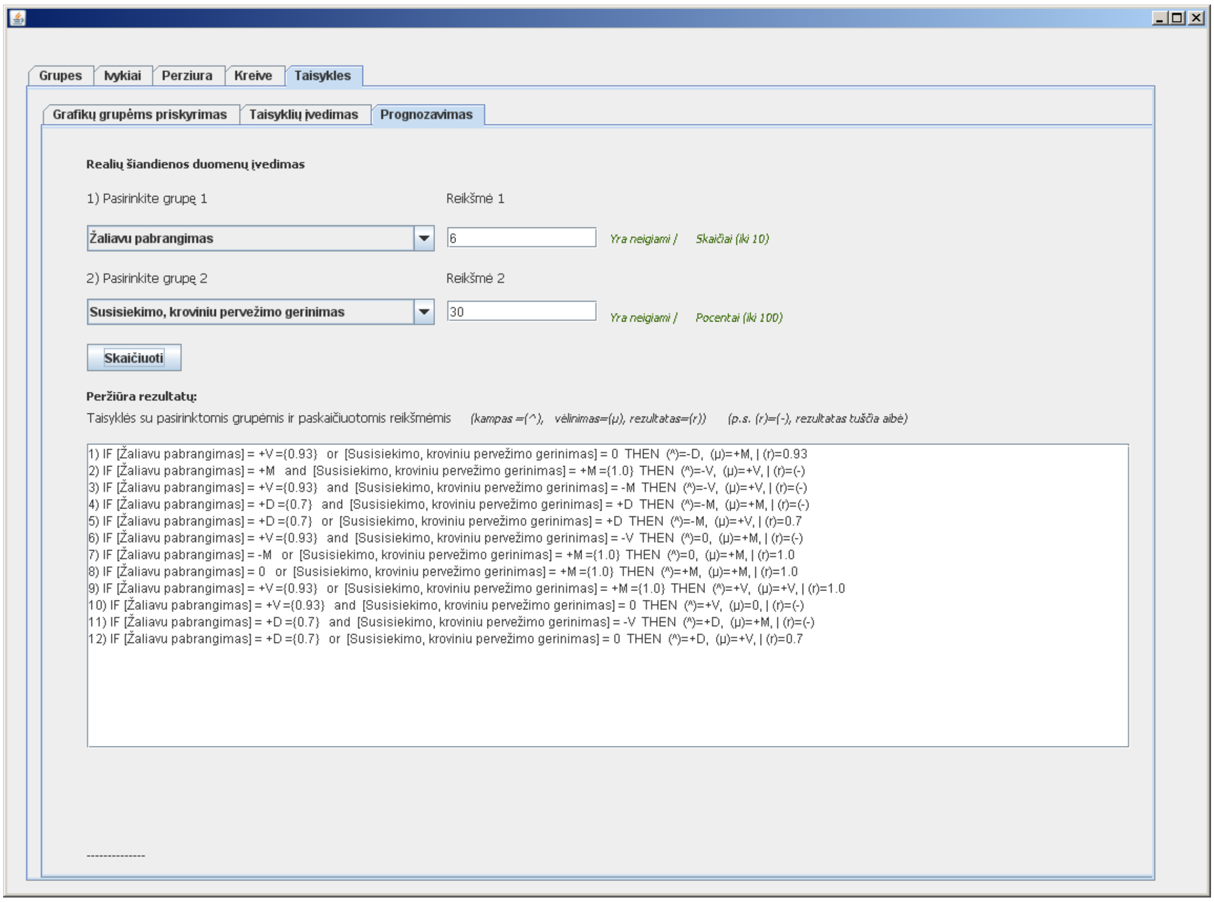
17. Trapecijos ploto skaičiavimas [žiūrėta 2010 kovo 13 d.], prieiga internete http://www.mathsteacher.com.au/year8/ch12_area/04_trap/trap.htm

18. Tiesės lygtis [žiūrėta 2010 kovo 17 d.], prieiga internete http://www.cut-the-knot.org/Curriculum/Calculus/StraightLine.shtml

19. Svorio centro skaičiavimas [žiūrėta 2010 kovo 20 d.], prieiga internete
http://www.fizika.lm.lt/content/view/604/
20. Atstumas tarp dviejų taškų [žiūrėta 2010 kovo 27 d.],
http://www.math24.info/90,atstumas-tarp-dvieju-tasku.html
21. Trikampio kampo radimas [žiūrėta 2010 balandžio 27 d.],
http://lt.wikipedia.org/wiki/Trikampis
22. Journal of Money, Credit, and Banking, magazine, April 2003,Vol. 35, No. 2
21. Miglotosios logikos valdymo metodai [žiūrėta 2010 kovo 28 d.], prieiga internete
http://www.google.lt/url?sa=t&source=web&ct=res&cd=10&ved=0CFIQFjAJ&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.129.6984%26rep%3Drep1%26type%3Dpdf&rct=j&q=Tsukamoto+fuzzy&ei=GyP_S5PNOZLp-AbXgrGkCg&usg=AFQjCNGRHyFxIxJ9nVt1JwJbdQjUFnisdw
22. Optimizavimo metodai [žiūrėta 2010 gegužės 28 d.], prieiga internete
http://mockus.us/optimum/docj/sliden.pdf

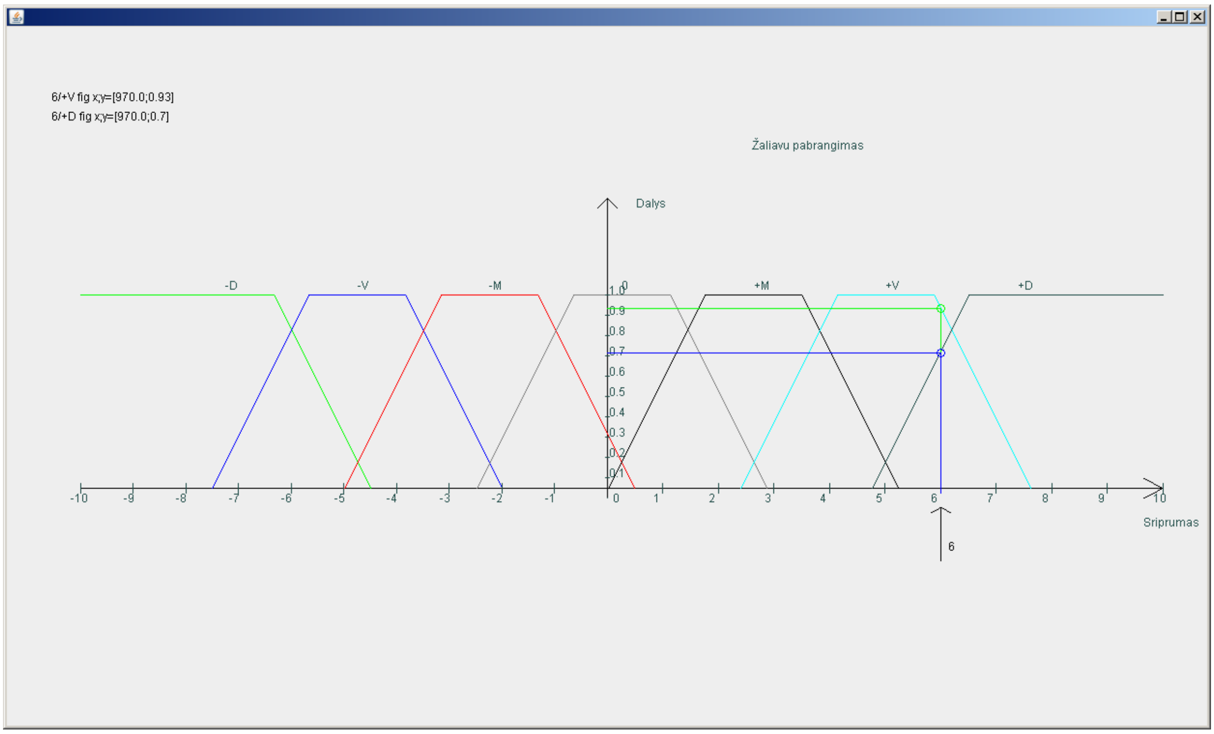
PRIEDAI

1 priedas.



35 pav. Kreivės pokyčio prognozavimo vartotojo sąsaja

Tarkim esamu laiku aktualios dvi grupės *Žaliavų pabrangimas* ir *Susisiekimo, krovinių pervežimo gerinimas*. Įrašomos jų reikšmės. Pirmos įvykių grupės reikšmė reiškia pokytį kiek pasikeitė grupės būsena. Šiuo atveju *Žaliavos pabrango* tarkim šešis kartus. Antros grupės reikšmė reiškia kad *Susisiekimas, krovinių pervežimas* pagerėjo tarkim trisdešimt procentų. Įvykių grupių pokyčių išreiškimo būdai (ar procentais ar skaičiais) pasirenkami vedant sistemos įėjimo duomenis.

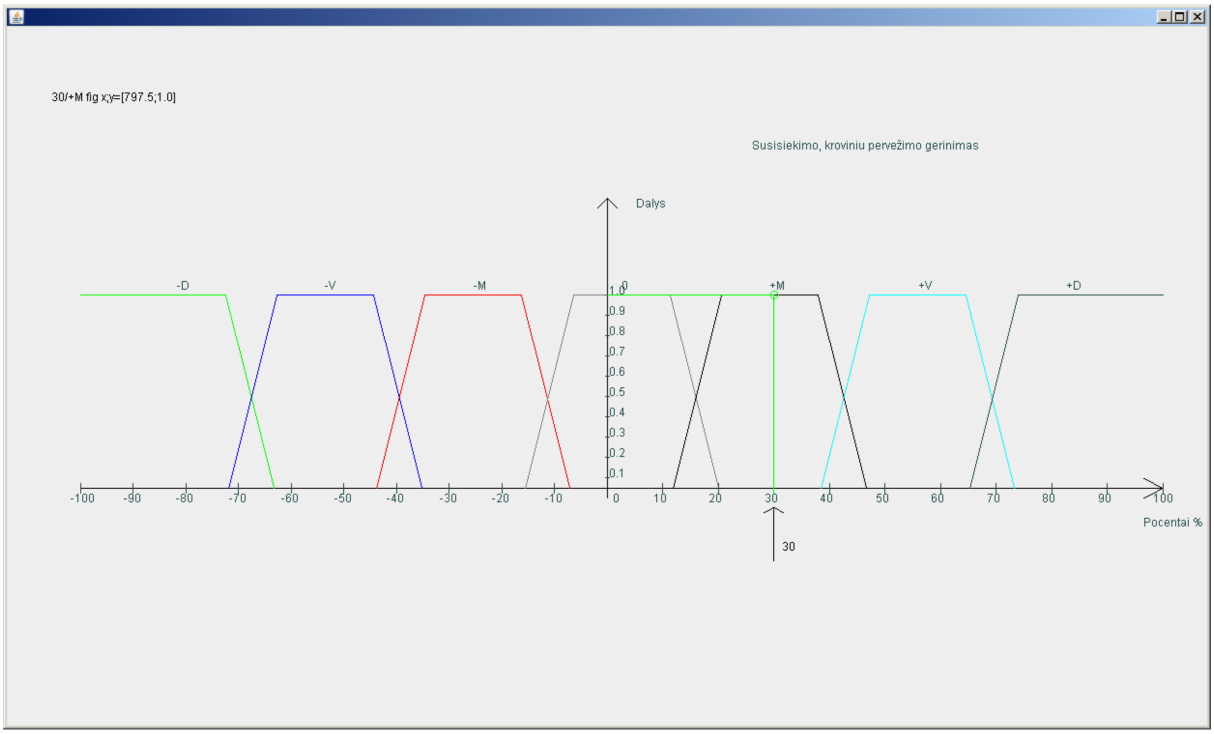


36 pav. Miglotųjų atitikmenų radimas pirmos grupės įvestai reikšmei

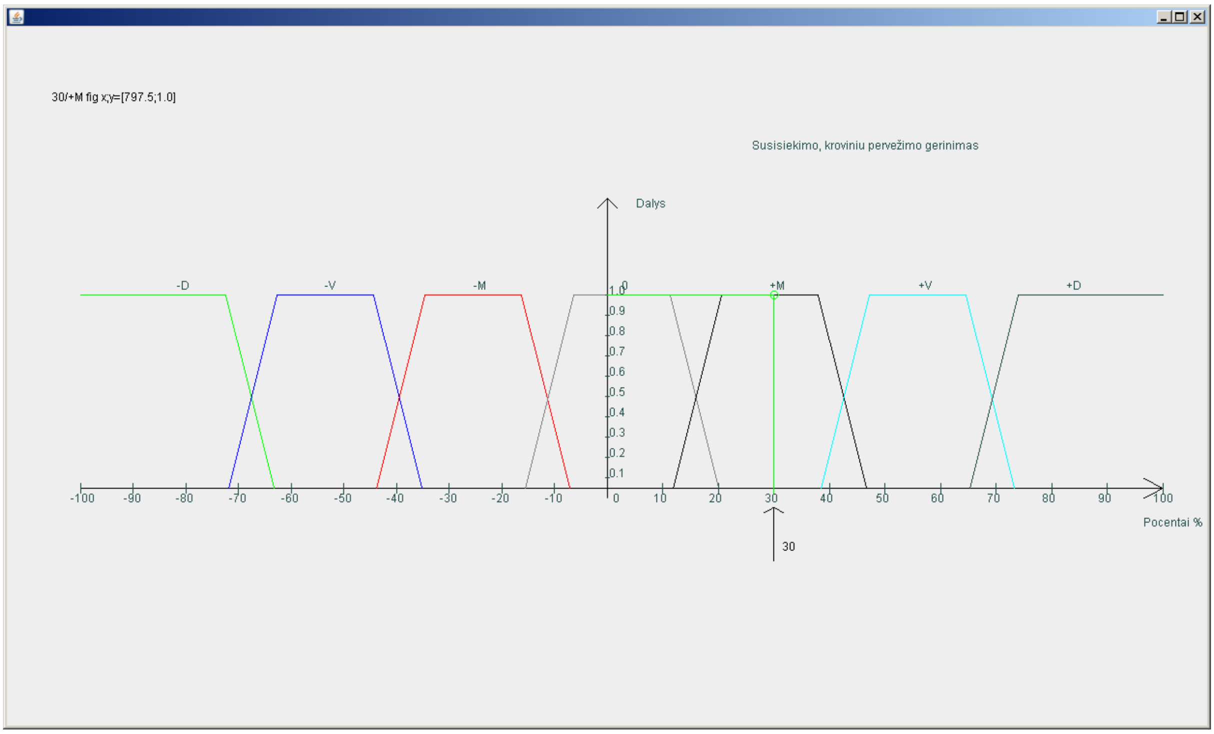
Pirmosios grupės įvestam pokyčiui randamos miglotosios reikšmės. Tai yra randamos figūros kurios yra kertamos ties pateikta reikšme.

Figūra +V kertama ties 0.93

Figūra +D kertama ties 0.7



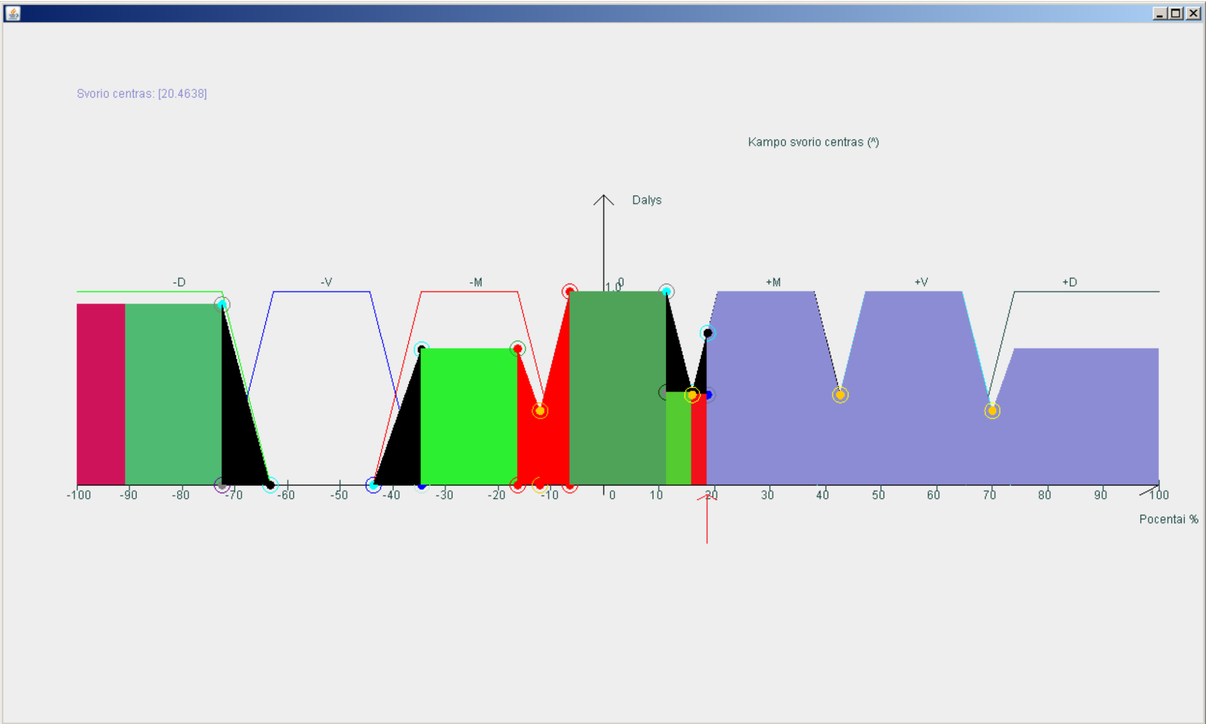
Figūra +M kertama ties 1.0



37 pav. Miglotųjų atitikmenų radimas antros grupės įvestai reikšmei

Antros grupės įvestam pokyčiui randamos miglotosios reikšmės. Tai yra randamos figūros kurios yra kertamos ties pateikta reikšme.

Galima pastebėti kad įvykių grupių miglotųjų figūrų grafikai skiriasi. Miglotųjų figūrų grafikai (trapecijų viršaus ir apačios ilgiai) įvykių grupėms priskiriami pasirenkant sistemos įėjimo duomenis.



38 pav. Zadeh OR operacija apjungti taisyklių rasti prognozuojamo kampo (^) miglotieji įverčiai

- Pirmos įvykių grupės miglotųjų reikšmių rezultatai (Figūra +V kertama ties 0.93 ir Figūra +D kertama ties 0.7) bei antros įvykių grupės miglotųjų reikšmių rezultatai (Figūra +M kertama ties 1.0) įstatomi į miglotųjų taisyklių sąlygas.
- Taisyklių rezultatuose gaunami miglotieji prognozuojamo kampo (^) ir jo vėlinimo (µ) rezultatai (r). Jeigu gaunasi kad miglotosios logikos figūrai pvz –D gautos dvi reikšmės tada atliekama Zadeh operacija OR kuri parenka didesnę reikšmę.

(^)=-D (r)=0.93

(^)=-M (r)=0.7

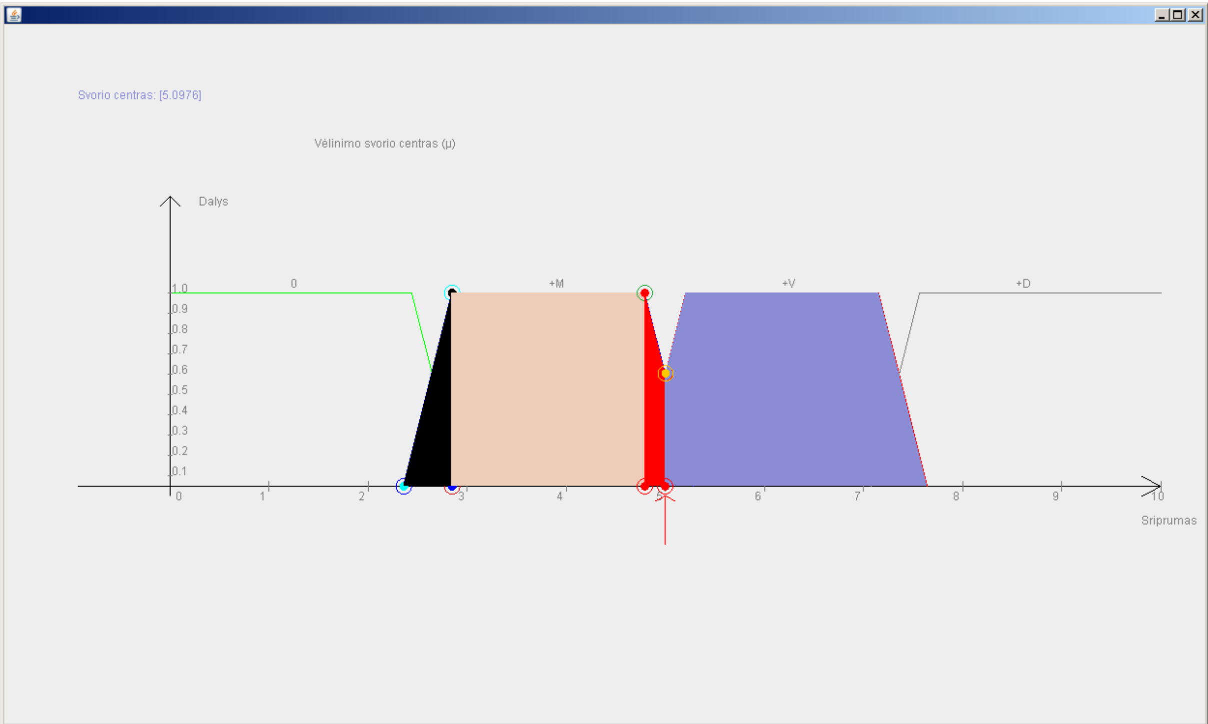
(^)=0, (r)=1.0

(^)=+M, (r)=1.0

(^)=+V, (r)=1.0

(^)=+D, (r)=0.7

Taisyklių rezultatai atvaizduojami grafike. Kurio vėliau ieškomas svorio centras. Gauta reikšmė yra miglota prognozuojamo kreivės kampo pokyčio reikšmė. Naudojant triškaitę taisyklę randamas atitikmuo maksimaliam galimam kreivės pokyčio kampo intervalui.

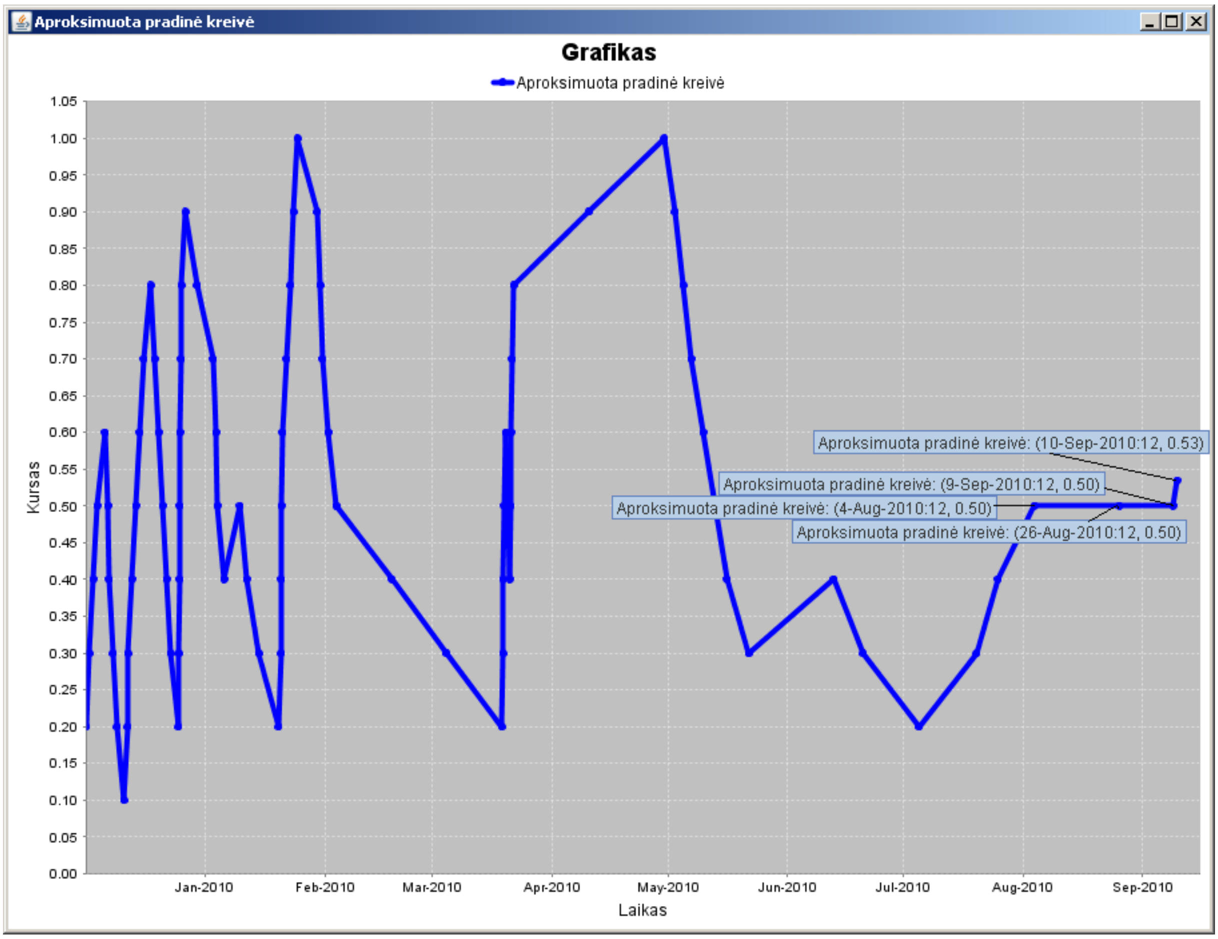


39 pav. Zadeh OR operacija apjungti taisyklių rasti prognozuojamo kreivės pokyčio vėlinimo (µ) miglotieji įverčiai

Tokiu pat būdu randama prognozuojamo kampo vėlinimo reikšmė.

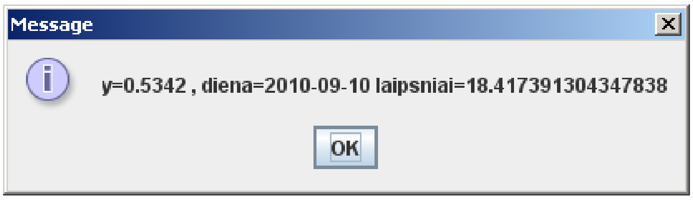
(µ)=+M, | (r)=0.93   OR   (µ)=+M, | (r)=1.0 OR   (µ)=+M, | (r)=1.0   ==   (µ)=+M, | (r)=1.0

(µ)=+V, | (r)=1.0   OR   (µ)=+V, | (r)=0.7   OR   (µ)=+V, | (r)=0.7   ==   (µ)=+V, | (r)=1.0

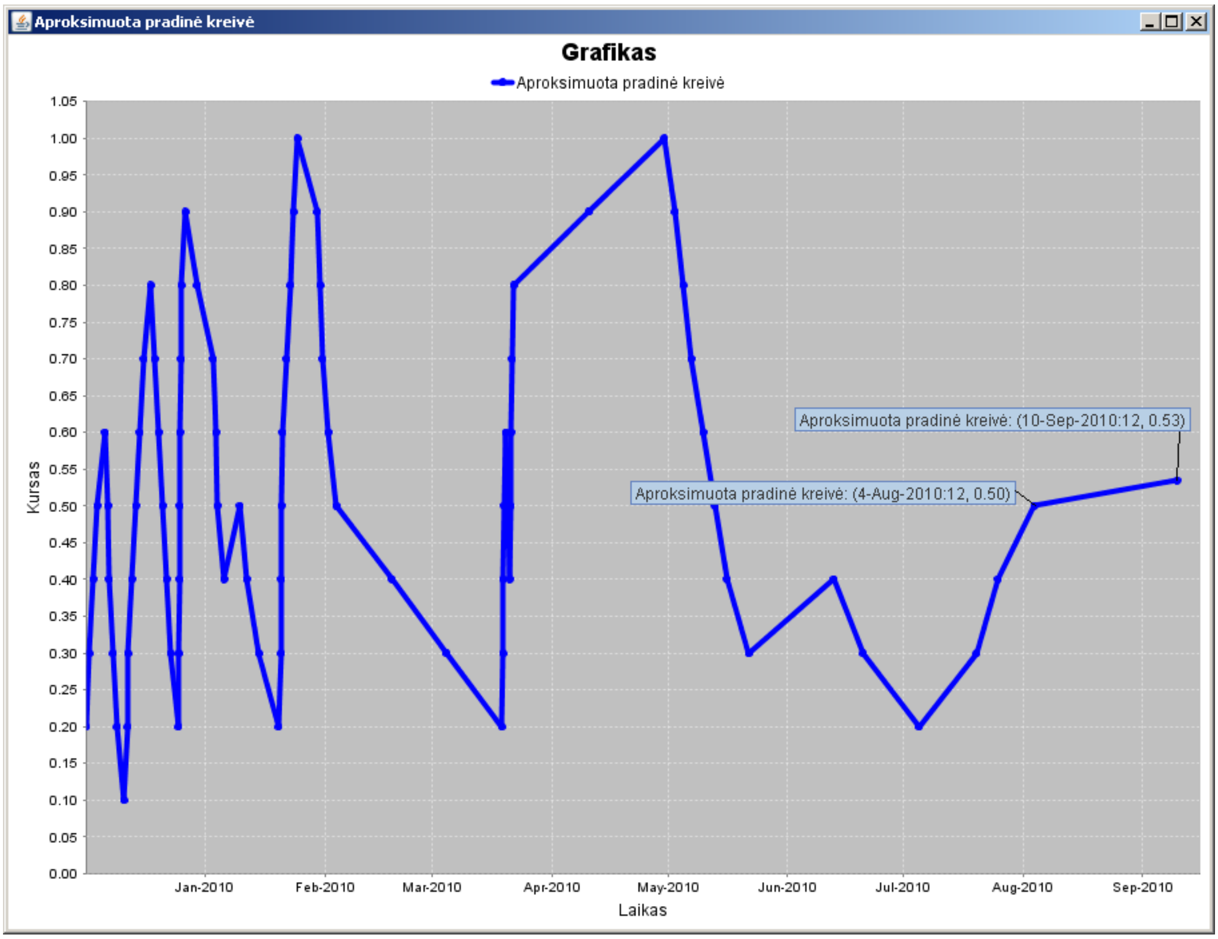


40 **pav. Dviejų naujų kreivės taškų radimas**

Radus prognozuojamą tašką jis atidedamas į kreivės grafiką. Iki jo tęsiama senoji kreivės būsena.



41 pav. Išprognozuoto kreivės taško koordinatės ir pasvirimo kampas



42 **pav. Suaproksimuoti seni ir nauji kreivės taškai**

Du naujieji taškai taip pat suaproksimuojami. Šiuo atveju aproksimavimo slenkstis 0.1. Tarp taškų išklotas atstumas 0,1, jo neradus (mažiau nei 0.1) taškai kreivės gale atmesti. Paskutinis kreivės taškas paliktas nors iki prieš tai esančio nėra atstumo 0.1.