

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Mantas Balnys

**Genetinių algoritmų pritaikymo klasifikavimo
uždaviniams spręsti tyrimas**

Magistro darbas

Darbo vadovas
dr. P. Kanapeckas

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

TVIRTINU

Katedros vedėjas
doc. dr. E. Kazanavičius
2004 05

Mantas Balnys

**Genetinių algoritmų pritaikymo klasifikavimo
uždaviniams spręsti tyrimas**

Informatikos mokslų magistro baigiamasis darbas

Kalbos konsultantė

Lietuvių kalbos katedros lektorė
dr. Jurgita Mikelionienė
2004 05

Vadovas

dr. P. Kanapeckas
2004 05

Recenzentas

dr. G. Palubeckis
2004 05

Atliko
IFM-8/1 gr. stud.
M. Balnys
2004 05

TURINYS

1	Pratarmė.....	6
2	Įvadas.....	7
3	Literatūros apžvalga ir analizė.....	8
3.1	Genetiniai algoritmai.....	9
3.2	Evoliucinės strategijos.....	10
3.3	Evoliucinis programavimas.....	11
4	Programinė įranga.....	11
4.1	Genetinio algoritmo modeliavimas.....	12
4.2	Neuroninio tinklo modeliavimas.....	14
4.3	Globalaus optimizavimo paketas „GMJ“.....	16
5	Eksperimentai.....	17
5.1	Vienasluoksnio perceptroninio tinklo apmokymas taikant S.Luke „EC“ paketą.....	17
5.1.1	Eksperimento apžvalga.....	17
5.1.2	Rezultatai.....	17
5.1.3	Eksperimento išvados.....	18
5.2	Genetinių algoritmų taikymas globalaus optimizavimo uždaviniams spręsti.....	20
5.2.1	Eksperimento apžvalga.....	21
5.2.2	Knapsako (<i>Knapsack</i>) uždavinys.....	21
5.2.3	Mokyklos tvarkaraščio optimizavimas.....	21
5.2.4	Dvikovos optimizavimo uždavinys.....	22
5.2.5	Matematinų funkcijų optimizavimas (sinuso funkcija).....	22
5.3	Genetinių algoritmų taikymas neuroninio tinklo apmokyje.....	22
5.3.1	Eksperimento apžvalga.....	22
5.3.2	Rezultatai.....	23
6	Išvados.....	25
7	Literatūra.....	26
8	Priedai.....	28
8.1	Globalaus optimizavimo paketo „GMJ“ darbo pavyzdys.....	28
8.2	Eksperimentas „Vienasluoksnio perceptroninio tinklo apmokymas“.....	30
8.2.1	Testiniai duomenys.....	30
8.3	Eksperimento „Genetinių algoritmų taikymas globalaus optimizavimo uždaviniams spręsti“ rezultatai.....	30
8.3.1	Iteracijų palyginimas.....	30
8.3.2	Metodų konvergavimas.....	32
8.4	Eksperimentas „Genetinių algoritmų taikymas neuroninio tinklo apmokyje“.....	34
8.4.1	Testiniai duomenys atpažinime.....	34
8.5	Pranešimas „Genetinių Algoritmų taikymas Globalaus optimizavimo uždaviniams spręsti“.....	36

SANTRAUKA

Vienas iš efektyviausių klasifikavimo metodų yra neuroninis tinklas. Tokių klasifikuojantį neuroninį tinklą bandoma apmokyti naudojant genetinius algoritmus. Tyrimo eigoje nustatyti efektyviausi genetinio algoritmo parametrai. Pateikiami genetinių algoritmų darbo palyginimo su kitais optimizavimo metodais rezultatai. Palyginama su tokiais žinomais metodais kaip Bajeso (*Bayes*) ir Monte Karlo paieškos (*Monte Carlo search*). Galima išskirti du šiame darbe realizuotų ir panaudotų genetinių algoritmų tipus. Pirmąjį galima vadinti paprastu, nes jis realizuotas pagal bazinę genetinių algoritmų idėją (dėl didesnio išskirtinumo darbe vadinamas parametriniu genetiniu algoritmu). Antrasis realizuotas pagal Lobo (*F. G. Lobo*) ir Goldbergo (*D. E. Goldberg*) pateiktą beparametrinio genetinio algoritmo idėją (23).

Tyrimas parodė genetinio algoritmo universalumą ir tinkamumą klasifikuojančiojo neuroninio tinklo efektyviam apmokymui. Taip pat nustatytas beparametrinio genetinio algoritmo pranašumas prieš bazinę genetinio algoritmo konfigūraciją. Sukurta programinė įranga bus panaudota ateityje mokslo vystymo ir mokymo tikslais.

SUMMARY

Neural networks are one of the most efficient classifier methods. One of such classifying neural networks we are trying to teach in this work by using genetic algorithms. In this work we test two types of genetic algorithms. One may be called parameterized genetic algorithm. It is built on the basic ideas of genetic algorithms. The other one is called parameter less genetic algorithm. It was presented by F. G. Lobo and D. E. Goldberg (23). Both genetic algorithms are tested and compared to the other well known optimization methods such as Bayes and Monte Carlo search.

Experiments show the relevance of use genetic algorithms in teaching classifying neural network. Also stated that parameter less genetic algorithm works more efficient than parametric genetic algorithm in general cases. Created programs will be used in future studies.

1 PRATARMĖ

Yra gana populiari ir madinga taikyti genetinius algoritmus įvairaus tipo optimizavimo uždaviniuose. To priežasties reiktų paieškoti genetinių algoritmų prigimtyje. Principas paimtas iš pačios gamtos yra atlaikęs milijonus ar net milijardus metų išbandymų ir ne tik išlikęs iki šių dienų, bet ir sunkiai pralenkiamas sudėtinguose uždaviniuose. Čia reiktų pastebėti, kad pakankamai paprastiems uždaviniams spręsti žmonės išrado begale puikių metodų, su kuriais genetiniams algoritmams net nebėra ką lygintis. Gal ir apskritai pasaulyje kada nors nebeliks uždavinio, kurį genetiniai algoritmai išspręstų geriausiai, bet tas laikas dar ateis negreit.

Šiuo metu pasaulyje genetiniais algoritmais domimasi vis labiau. Kasmet pasirodo vis daugiau straipsnių apie jų pritaikymo galimybes. Šie tyrimai ne tik atveria optimizavimo galimybes, bet ir veda mūsų pačių pažinimo link. Tuo metu kai biologai bando sudaryti žmogaus genomo žemėlapi, informatikai užsiima pačių genų veikimo principų tyrimu.

Vienas iš klasifikavimo uždavinių sprendimo metodų yra neuroniniai tinklai. Panaši situacija kaip ir su genetiniais algoritmais yra ir su neuroniniais tinklais. Kaip ir kur jie bus paplitę netolimoje ateityje niekas negali tiksliai pasakyti. Gali būti, kad atsiras metodai veikiantys geriau nei neuroniniai tinklai ir jų teorija numirs. Kaip ir genetiniais algoritmais, klasifikuojančiaisiais neuroniniais tinklais domimasi vis labiau ir labiau. Jų pagrindinė taikymo sritis, tai didelio sudėtingumo uždaviniai, kuriems nėra kitų efektyvesnių sprendimo metodų.

Pasiskaičius įvairios literatūros galima susidaryt išpūdį, kad neuroniniai tinklai gali beveik viską. Visą, o gal netgi ir daugiau, nei pats žmogus. Tam, kad pateisintumėme bent dalelę šios tiesos, tenka įveikti keletą kliūčių. Visų jų pagrindas milžiniškas neuroninio tinklo dydis. Bent kiek sudėtingesni tinklai susideda iš daugiau nei tūkstančio neuronų. Tarp, kurių dar sukuriama nuo dešimt iki šimto kartų daugiau ryšių. Čia ir iškyla tokio tinklo modeliavimo kompiuteriu problema. Bet ją daugiau ar mažiau išspręsus rezultatų negausime. Pats neuroninis tinklas nieko nemoka. Jį dar reikia apmokyti.

Paprastiems neuroniniams tinklams yra sukurta keletas gana efektyvių apmokymo metodų. Bet sudėtingėjant tinklams, tikslaus skaičiavimo metodus pritaikyti darosi vis sunkiau. Iš čia ir kyla idėja patyrinėti dviejų gamtos pateiktų metodų sintezę. Kaip dera gamtoje, taip jie turėtų tikti ir kompiuteriniame modelyje.

2 ĮVADAS

Šiais informacinės eros laikais klasifikavimo uždavinys tampa ypač svarbus. Neklasifikuota informacija neturi jokios vertės. Klasifikuojama viskas – nuo kojinių stalčiuje iki akcijų biržoje. Ten, kur informacijos kiekiai yra labai dideli, ją suklasifikuoti rankiniu būdu yra neįmanoma. Čia į pagalbą ateina kompiuteriai.

Šiame darbe bus dirbama su metodais, skirtais analizuoti labai dideliame kiekiui informacijos. Tokiam, su kuriuo nesusidoroja jokie tikslaus skaičiavimo metodai, o rezultatai gaunami tik euristinių metodų pagalba. Šiuo metu daugiausiai žadantis euristinis metodas yra klasifikuojantis neuroninis tinklas. Kaip ir visus neuroninius tinklus, jį reikia apmokyti, o tai padaryti nėra paprasta. Kuo daugiau informacijos klasifikuoja neuroninis tinklas, tuo jis turi būti didesnis. Taip pat atsiranda poreikis kurti sudėtingesnės architektūros neuroninius tinklus. Šitokių tinklų apmokymas tampa didžiule problema. Nėra pakankamai efektyvių vieningų metodų galinčių tai padaryti.

Genetinio algoritmo pritaikymo klasifikuojančiojo neuroninio tinklo apmokymui galimybes bus bandoma nustatyti šiame darbe.

Kaip programinės įrangos atraminis taškas, pasirinktas profesoriaus J. Mockaus vystomas globalaus optimizavimo paketas „GMJ“ (25). Šiame pakete yra sukaupia nemažai globalaus optimizavimo metodų, su kuriais galima palyginti savo rezultatus. Taip pat šis paketas yra lengvai išplečiamas. Praplėtus šį paketą genetinių algoritmų ir neuroninių tinklų modeliavimo paketais buvo atlikti eksperimentai.

Darbo metu nepavyko aptikti pakankamai kokybiškų genetinių algoritmų modeliavimo paketų tinkamų numatytiems tyrimams atlikti (4.1 Genetinio algoritmo modeliavimas). Tad buvo sukurtas universalus genetinių algoritmų modeliavimo paketas adaptuotas „GMJ“ sistemai. Panaši situacija ir neuroninių tinklų modeliavime (4.2 Neuroninio tinklo modeliavimas). Nors paketų yra daug, bet jie nėra tinkami perprogramavimui, dažniausiai jie būna skirti tik vykdymui, o veikia jie ne taip kaip norima.

Beparametrinių genetinių algoritmų pritaikymas klasifikavimo uždaviniuose buvo sekantis uždavinys. Beparametrinis genetinis algoritmas, tai tas pats genetinis algoritmas su specialiomis priemonėmis suoptimizuotais parametrais. Yra žinoma keletas būdų tam atlikti, bet šiuo atveju parametrus optimizuoti buvo pasitelktas Bajeso (*Bayes*) optimizavimo metodas. Pirmieji tokį beparametrinį algoritmą pasiūlė Lobo (*F. G. Lobo*) ir Goldbergas (*D. E. Goldberg*) straipsnyje

„Beparametrinis genetinis algoritmas praktikoje“ 2001 metais (23). Pagal jų pasiūlytą metodiką buvo sukurtas beparametrinis genetinis algoritmas pritaikytas „GMJ“ sistemai.

Toliau aprašomuose eksperimentuose patvirtinamas genetinių algoritmų tinkamumas klasifikuojančiojo neuroninio tinklo apmokymui. Beparametriniai genetiniai algoritmai pasirodė pakankamai efektyvūs ir tinkami ateityje pakeisti parametrizuotuosius.

3 LITERATŪROS APŽVALGA IR ANALIZĖ

Evoliuciniai skaičiavimo metodai aprašo kompiuterinių problemų sprendimo sistemą, kuri pagrįsta evoliucinių procesų panaudojimu skaičiavimo metodų projektavime ir realizavime.

Evoliucinių skaičiavimų pritaikymus optimizavimui pradmenis galima aptikti dar 1950 – 1960 metais (1, 2, 3, 4, 5, 6). Lygiagrečiai keletas biologijos specialistų pradėjo naudoti kompiuterius savo eksperimentams modeliuoti (2, 7, 8, 9, 10). Istoriškai susiklostė, kad iš evoliucinių skaičiavimų išsivystė trys pagrindinės srovės, tai genetiniai algoritmai (11), evoliucinės strategijos (12, 13) ir evoliucinis programavimas (14). Greta pagrindinių išsivystė ir keletas smulkesnių evoliucinių skaičiavimų sričių tokių kaip genetinis programavimas ir klasifikavimo sistemos. Tačiau evoliucinių algoritmų kombinavimas su kitais kompiuteriniais skaičiavimo metodais tokiais kaip neuroniniai tinklai, miglotos logikos sistemos, kitomis euristinėmis paieškomis, lokalia paieška, tabu paieška yra intensyviai vystomos tik pastaruosiu metu.

Evoliucinių algoritmų sukurta daug, tačiau visi jie remiasi keletu pagrindinių parametru:

- 1) Evoliucinio algoritmo kolektyvinis apsimokymas vyksta per individų populiaciją. Kiekvienas individas atitinka vieną nagrinėjamos problemos sprendinį.
- 2) Kiekvienam individui gali būti priskirta sprendinio kokybę įvertinanti reikšmė. Atsižvelgiant į šią reikšmę, kokybiškesni individai parinkimo procese dalyvaus dažniau nei prasti.
- 3) Individai generuojami atsitiktiniais principais modeliuojant mutaciją ir kryžminimą. Mutacija atitiktų klaidingą savęs nukopijavimą, o kryžminimas – informacijos pasikeitimą tarp dviejų ir daugiau individų.

Tipinio evoliucinio algoritmo veikimas pateiktas pavyzdyje (pav. 1). Individų populiacija P sukuriama, tada pakartotinai taikant kokybės įvertinimą, išrinkimą, kryžminimą ir mutaciją evoliucionuoja iš kartos t į kartą $t+1$. Paprastai evoliucinis algoritmas individus sukuria pagal atsitiktinę strategiją, tačiau yra galimi ir konkrečiam uždaviniui pritaikyti metodai. Kokybės

įvertinimas nustato individo, kaip uždavinio sprendinio vertę. Šio įvertinimo sudėtingumas svyruoja, nuo minimalios matematinės funkcijos apskaičiavimo iki sudėtingų modeliavimo procesų. Sekančiai kartai kurti, individai parenkami su tikimybe, kuo individo kokybė geresnė, tuo didesnė ir tikimybė jį parinkti. Parinkti du ar daugiau individų dalyvauja kryžminimo ir mutacijos operacijose. Galiausiai, po visų skaičiavimų dalis geriausių individų pereina į sekančią kartą.

Evoliuciniai algoritmai naudojami dideliame kiekyje mokslinių ir inžinerinių uždavinių ir modelių. Keletas iš evoliucinio algoritmo panaudojimo pavyzdžių būtų optimizavimo, automatinio programavimo, mašinų apmokymo, ekonomikos, ekologijos, evoliucijos ir apmokymo, socialinės sistemos uždaviniai (15).

```
procedure EA();
begin
  t = 0; /* Initial Generation */
  initialize population(P(t));
  evaluate(P(t));
  while (not termination condition)
  begin
    P'(t) = select parents(P(t));
    P''(t) = recombine(P'(t));
    P'''(t) = mutate(P''(t));
    evaluate(P'''(t));
    P(t+1) = select survivors(P'''(t) U Q); /* Q in {∅, P(t)} */
    t = t + 1; /* Next Generation */
  end
end
```

pav. 1 Evoliucinio algoritmo pavyzdys.

3.1 GENETINIAI ALGORITMAI

Genetinius algoritmus atrado ir išvystė J.Holandas (*J.Holland*) (11). J.Holandas studijavo natūralios adaptacijos fenomeną ir ieškojo būdų jį perkelti į kompiuterį. Genetiniai algoritmai buvo pristatyti kaip biologinės evoliucijos modeliavimo priemonė.

Tradiciškai genetinių algoritmų genomams aprašyti naudojamos dvejetainės išraiškos. Tačiau plečiantis uždavinių spektrui ilgainiui buvo pradėtos naudoti ir kitokios genomo išraiškos: sveika skaitinė, realieji skaičiai, vektoriai, masyvai, grafai (neuroniniai tinklai).

Individų parinkimas yra tikimybinė funkcija pagrįsta individo kokybe. Proporcingo individų parinkimo funkcija parenka individą su tokia tikimybe, kad jis būtų parinktas apie tiek kartų kiek gautume individo kokybe padaline iš vidutinės populiacijos kokybės. Paprastas metodas atlikti

proporcingą individų parinkimą yra ruletės rato pavyzdys (16). Sukurtų naujų individų kiekis atitinka pradinių individų kiekį. Vėliau jie pakeičia visus pradinius individus. Šis metodas neturi galimybės išsaugoti geriausių individų ir jie gali būti prarasti.

Nauji individai yra kuriami kryžminant senus individus su nustatyta tikimybe. Po to jam dar pritaikoma mutacija su labai maža tikimybe kiekvienam genomo genui. Pradinėje genetinio algoritmo idėjoje kryžminimas yra iškeliamas kaip pagrindinė paieškos operacija, o mutacija nustumiama į šalį, kaip pagalbiniis veiksnys. Susidomėjimas mutacijos operacija atsirado ne taip seniai, dalinai dėl evoliucinių strategijų ir evoliucinio programavimo įtakos.

3.2 EVOLIUCINĖS STRATEGIJOS

Evoliucines strategijas išvystė I.Rechenbergas (*I.Rechenberg*) (12), naudodamas individų parinkimą, mutacijas ir populiaciją sudarytą iš vieno individo. H.Sčvefelis (*H.Schwefel*) (13) panaudojo kryžminimą ir populiacijas sudarytas iš didesnio individų kiekio. Taip pat evoliucines strategijas sulyginu su kitom, labiau tradicinėm optimizavimo technologijom.

Evoliucinės strategijos paprastai genomui aprašyti naudoja realiųjų skaičių vektorius. Individai kryžminimui parenkami atsitiktinai iš sąrašo. Naujai gautų individų skaičius būna didesnis nei senųjų. Kad naujieji individai sudarytų naują kartą galima taikyti du deterministinius metodus. Pirmasis iš naujų individų išrenka tik reikiamą kiekį geriausių (tiek kiek buvo pirminėje populiacijoje). Antruoju metodu reikiamas kiekis individų sudaromas bendrai iš naujų ir senų atrinkus geriausius. Taikant pastarąjį metodą senieji individai taip pat turi galimybę išlikti.

Naujos kartos individai gaunami atliekant kryžminimą, paskui kurį seka mutacija. Evoliuciniuose skaičiavimuose naudojama begalė skirtingų kryžminimo strategijų, kurių operatoriai būna dviejų rūšių. Pirmu atveju kryžminimas vyksta tarp dviejų atsitiktinai parinktų individų. Kitu – vienas atsitiktinai parinktas individas yra fiksuojamas ir kryžminamas su vis kitu atsitiktinai parinktu individu. Mutacija paprastai vykdoma kiekvienam paskirstytai kiekvienam individui pagal tikimybe. Evoliuciniuose skaičiavimuose atsirado daug žadanti prisitaikanti mutacija – tikimybė įvykti mutacijai kinta priklausomai nuo aplinkybių.

Kryžminimas ir mutacija evoliuciniuose skaičiavimo metoduose yra lygiavertės svarbiausios operacijos.

3.3 EVOLIUCINIS PROGRAMAVIMAS

Evoliucinį programavimą išvystė L.Fogelis (*L.Fogel*) (14). Evoliucinis programavimas iškilo kaip troškimas sukurti dirbtinį intelektą. Naudojantis individų išrinkimu ir mutacijomis sukurti išbaigtas „protingas“ mašinas.

Evoliucinis programavimas tradiciškai panaudojo individus, kurie savo genomu yra tvirtai surišti su sprendžiama problema. Uždaviniai turintys baigtinį būsenų kiekį dažnai yra atvaizduojami grafu. Kituose uždaviniuose taikomi atitinkami atvaizdavimo būdai, pavyzdžiui vektoriai arba surūšiuoti sąrašai.

Individų parinkimas yra konkurencija pagrįsta tikimybinė funkcija. Naujų individų sukuriama tiek pat kiek ir buvo senų. Vėliau geriausi atrenkami iš bendro naujų ir senų individų sąrašo ir patenka į sekančią kartą.

Naujas individas sukuriamas iš senojo pritaikant jam mutaciją. Mutacijos operacijos tipas labai priklauso nuo uždavinio specifikos ir panašiai kaip evoliuciniuose skaičiavimuose dažnai yra prisitaikantis.

4 PROGRAMINĖ ĮRANGA

Įvairiems moksliniams tyrinėjimams atlikti visame pasaulyje naudojamos įvairios programavimo kalbos. Tačiau dažniausiai pasitaikančios ir šiuolaikiškiausios atrodančios yra tik dvi, tai C++ ir Java. C++ kalba rašyta programinė įranga yra platinama pagal GNU licenziją, o tai reiškia kad ją galima legaliai naudoti visiems mokslo darbams. Visa problema, kad šia programavimo kalba rašytos programos veikia tik Unix pagrindo operacinėse sistemose, kurių paplitimas mūsų universitete yra gana ribotas. Į kur kas geresnę situaciją pakliuna Java programavimo kalba. Jos specifika, jai leidžia veikti, bet kokioje operacinėje sistemoje. Netgi Unix pagrindo operacinėje sistemoje parašyta programa be jokių pataisymų veikia populiarioje Windows sistemoje. Šitoks perkeliamumas Java programavimo kalbą paverčia lydere visuose moksliniuose projektuose. Mokslininkams atsiranda galimybė palyginti tarpusavio rezultatus, pasinaudoti jau sukurtais skaičiavimo metodais ir toliau juos tobulinti. Vis daugiau mokslinę programinę įrangą kuriančių žmonių pasirenka Java. Tad, siekiant neatsilikti nuo mados, visiems tyrimams atliktiems šiame darbe buvo panaudota Java programavimo kalba.

Labai daug dėmesio buvo skirta J.Mockaus Globalaus optimizavimo paketui (25). Jame yra sukaupta daug optimizavimo metodų, su kuriais greitai galima atlikti palyginimą. Tik pradžioje šis paketas neturėjo nieko bendro su genetiniais algoritmais ar neuroniniais tinklais.

4.1 GENETINIO ALGORITMO MODELIAVIMAS

Džordžo Masono (*George Mason*) universiteto profesorius asistentas Šonas Lukas (*Sean Luke*) neseniai baigė daktaro disertaciją. Eksperimentams atlikti jis sukūrė didelį evoliucinių skaičiavimų paketą pavadintą „EC“ (17). Prie šio paketo buvo prijungtas mano sukurtas universalus neuroninių tinklų modeliavimo paketas (4.2 Neuroninio tinklo modeliavimas). Šia kombinacija buvo atliktas vienas pirmųjų eksperimentų (5.1 Vienasluoksniu perceptroninio tinklo apmokymas taikant S.Luke „EC“ paketą).

Eksperimento metu pastebėta, kad modeliavimams šiuo paketu reikia labai daug laiko sąnaudų. Pats paketas užėmė labai daug vietos ir netiko prijungimui prie J.Mockaus globalaus optimizavimo sistemos. Tad nors pats eksperimentas pavyko, „EC“ paketo teko atsisakyti.

Buvo imtasi kurti nuosavą genetinių algoritmų modeliavimo paketą. Sukurtame pakete uždavinys konstruojamas iš specialiai parinkto genomo, mutacijos ir kryžminimo operacijų ir vykdymo strategijos. Genomas pilnai priklauso nuo uždavinio ir baziniame variante tai yra skaičių aibė. Neuroninio tinklo apmokyme naudojama realiųjų skaičių aibė. Mutacijos ir kryžminimo operacijas galima nesunkiai kurti naujas su sumanytom strategijom. Šiuo metu pakete pateikiamos šios mutacijos operacijos:

- Vieno geno mutacija – kai mutuoja tik vienas genomo genas (pav. 2);
- Atsitiktinė mutacija – kai mutuoja atsitiktinė genų dalis (kontroliuojama parametrais) (pav. 3);
- Mutacija su paieška – nustatytą kiekį kartų vykdoma atsitiktinė mutacija, o kaip rezultatas pateikiamas tik geriausias individas;

Kryžminimo operacijos:

- Kryžminimas per vieną tašką – genomas skeliamas pusiau ir į naują individą patenka pusė genų iš vieno tėvo, o kita pusė iš kito (pav. 4);
- Kryžminimas per du taškus – genų persiskirstymas vyksta panašiai kaip ir kryžminant per vieną tašką, tik čia genomas skeliamas į tris dalis (pav. 5);

- Atsitiktinis kryžminimas – genai į naująjį individą patenka atsitiktinai iš abiejų tėvų (pav. 6);
- Strateginis kryžminimas – nustatytą kiekį kartų atliekamas kryžminimas per vieną tašką, o iš gautų individų išrenkamas tik geriausias;

senas	A	A	A	A	A	A	A	A	A	A
naujas	A	A	A	A	A	B	A	A	A	A

pav. 2 Vieno geno mutacija.

senas	A	A	A	A	A	A	A	A	A	A
naujas	A	B	B	A	A	B	A	A	A	A

pav. 3 Atsitiktinė genų mutacija.

senas A	A	A	A	A	A	A	A	A	A	A
senas B	B	B	B	B	B	B	B	B	B	B
naujas	A	A	A	A	A	A	B	B	B	B

pav. 4 Kryžminimas per vieną tašką.

senas A	A	A	A	A	A	A	A	A	A	A
senas B	B	B	B	B	B	B	B	B	B	B
naujas	B	B	A	A	A	A	B	B	B	B

pav. 5 Kryžminimas per du taškus.

senas A	A	A	A	A	A	A	A	A	A	A
senas B	B	B	B	B	B	B	B	B	B	B
naujas	A	A	B	A	B	A	B	B	A	B

pav. 6 Atsitiktinis kryžminimas.

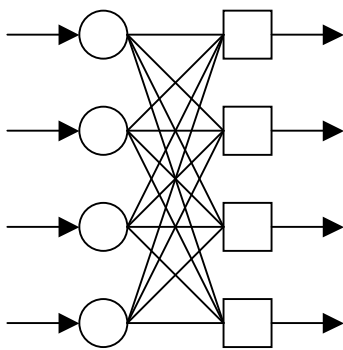
Kai buvo nuspręsta sukurti beparametrinį genetinį algoritimą, jis buvo sukurtas to paties paketo bazėje. Iš J.Mockaus paketo „GMJ“ paimtas Bajeso (*Bayes*) optimizavimo metodas, buvo pritaikytas optimizuoti genetinio algoritmo parametrus, kai tuo tarpu pats genetinis algoritmas optimizavo klasifikuojantį neuroninį tinklą.

4.2 NEURONINIO TINKLO MODELIAVIMAS

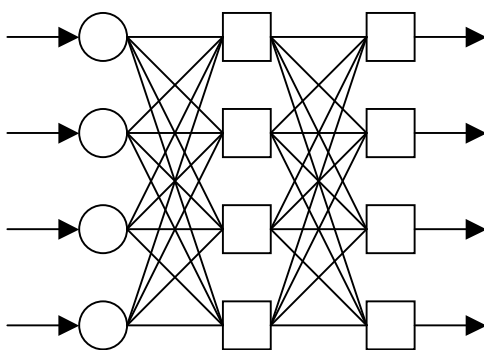
Nors neuroninius tinklus modeliuoja net ir tokios populiarios sistemos kaip MatLAB arba MathCAD, tinkamo panaudoti paketo rasti nepavyko. Absoliuti dauguma paketų modeliuoja tik labai paprastus neuroninius tinklus. Tuo tarpu sudėtingesnės sistemos pateikiamos, kaip skaičiuojantis produktas, o ne kaip modeliavimo paketas. Tokios sistemos neįmanoma papildyti savo programiniais kodais, o konkrečiai mano atveju – nėra galimybės į neuroninio tinklo apmokymo procesą įtraukti genetinius algoritmus.

Kuriant universalų neuroninį tinklą, negalima užsiduoti jo architektūros. Visas dėmesys sukonzentruojamas į žemiausią neuroninio tinklo lygmenį – neuronus. Vėliau iš neuronų galima konstruoti įvairiausias architektūras tinklus nuo standartinių kelių sluoksnių (pav. 7, pav. 8, pav. 9) iki nesuspecifikuojamų rezginių.

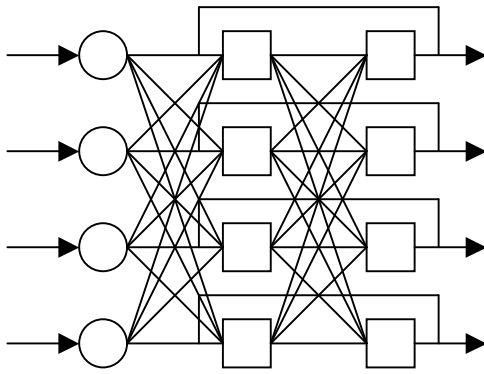
Šitoks architektūros laisvumas kuriamas tam, kad algoritmas apmokantis neuroninį tinklą galėtų ne tik keisti neurono įėjimų svorius, bet ir pačią tinklo architektūrą.



pav. 7 Vienasluoksnis neuroninis tinklas.



pav. 8 Dvisluoksnis neuroninis tinklas.



pav. 9 Dvisluoksnis neuroninis tinklas su grįžtamaisiais ryšiais.

Pagrindinis vienetas čia yra neuronas. Kiekvienas neuronas gali jungtis su bet kiek kitų neuronų, bet kokia tvarka. Net ir pats su savimi. Bet kokie architektūriniai apribojimai atmetami. Pats neuronas veikia labai paprastai, pagal sekančią formulę:

$$N = \sum_i w_i \bullet x_i, \text{ čia } w_i \text{ yra svoriniai koeficientai, } x_i \text{ įėjimai, o } i \text{ įėjimo indeksas.}$$

neuronas išduoda išėjimo signalą $O = f(N)$, kur funkcija f gali būti labai įvairi.

Šiame darbe programiškai realizuoti funkcijos f pavyzdžiai:

- 1) Šuolinė: $f(N) = \begin{cases} 0 & , N < T \\ 1 & , N \geq T \end{cases}$, čia T iš anksto nustatytas slenkstis;
- 2) Eksponentinė: $f(N) = \frac{1}{1 + e^{-N}}$;
- 3) Hiperbolinio tangento: $f(N) = \tanh N$;
- 4) Kvadratinė: $f(N) = N * N$;

Pats neuroninio tinklo darbas buvo modeliuojamas signalų trasavimo būdu. Kiekvienas signalas turi savo žingsnio skaitiklį. Pastarasis skaičiuoja per kiek neuronų signalas yra praėjęs. Tokiu būdu bandoma užkirsti kelią amžiniems ciklams ir pasenusių signalų apdorojimui. Signalas yra pasiunčiamas į pirmuosius neuronus, čia perskaičiuojamas, o jo skaitiklis padidinamas. Tada signalas siunčiamas į visus neuronus, kurie yra prijungti prie pirmojo. Šis ciklas kartojamas kol signalas nusistovi. Šuolinės funkcijos atveju, tai nustatyti gana paprasta, bet kitais atvejais signalas yra realus skaičius ir lygybė niekada nebus pasiekta. Todėl yra naudojama minimalios paklaidos reikšmė. Jei du signalai skiriasi mažiau, nei nustatyta paklaidos reikšmė, tai šie signalai yra vienodi.

4.3 GLOBALAUS OPTIMIZAVIMO PAKETAS „GMJ“

Profesorius J. Mockus globalaus optimizavimo paketą pradėjo vystyti prieš kelis metus. Kiekvienais metais jis pasipildo vis naujom priemonėm. Realizuojami nauji optimizavimo metodai. Įtraukiami nauji uždaviniai optimizavimui. Pridedama papildomų rezultatų analizės galimybių. Mano indėlis čia buvo du optimizavimo metodai ir vienas optimizavimo uždavinys:

- parametrizuotas genetinis algoritmas (pav. 12);
 - paprastas
 - su kryžminimo strategija
- beparametrinis genetinis algoritmas (pav. 13);
 - paprastas
 - su kryžminimo strategija
- neuroninio tinklo optimizavimo uždavinys (pav. 14);
 - vienasluoksnis
 - dvisluoksnis
 - su grįžtamais ryšiais

Metodų palyginimui buvo pasinaudota pakete jau buvusiais metodais. Pagrindiniai du, tai Bajeso (*Bayes*) ir Monte Karlo paieškos (*Monte Carlo search*). Šie metodai yra gerai žinomi optimizavimo mokslo srityje. Greta šių gerai žinomų metodų, dar buvo pasinaudota ir keliais labiau specializuotais ir mažiau žinomais metodais, tai:

- Exkor Bajeso metodo modifikacija su kryptinga paieška
- LBayes Bajeso metodo modifikacija su žingsnio dydžio kontrole
- Globt Klasterizacijos tipo metodas
- Unt Ekstrapoliacijos tipo metodas

Metodai tarpusavyje buvo lyginami ir skirtingų uždavinių fone. Palyginimui pasitelkti tokie skirtingi uždaviniai, kaip Knapsako (*Knapsack*) arba vagies uždavinys, mokyklos tvarkaraščio optimizavimo uždavinys, dvikovos (*Duel*) optimizavimo uždavinys, matematinių funkcijų optimizavimas.

Šio paketo pagalba buvo galima nesunkiai atlikti eilę eksperimentų (pav. 15) ir palyginti optimizavimo metodų darbą. Kadangi palyginimui buvo naudojama ta pati programinė įranga, tai rezultatai turėjo gautis labai tikslūs.

5 EKSPERIMENTAI

5.1 VIENASLUOKSNIO PARCEPTRONINIO TINKLO APMOKYMAS TAIKANT S.LUKE „EC“ PAKETĄ

Dr. Sean Luke evoliucinio skaičiavimo (*Evolutional Computation*) paketas skirtas modeliuoti genetiniams algoritmams. Prie jo buvo prijungta mano sukurta universali neuroninių tinklų modeliavimo sistema.

Eksperimentu bandoma nustatyti kaip neuroninio tinklo apmokymo greitis priklauso nuo genetinio algoritmo parametrų. Tiriama genetinio algoritmo parametrai: mutacijos tikimybė, kryžminimo tikimybė ir populiacijos dydis.

5.1.1 EKSPERIMENTO APŽVALGA

Bandoma apmokyti vieno sluoksnio parceptroninį tinklą. Neuroninį tinklą sudaro devyni parceptronai (atitinka 3x3 matricą). Slenkstinė funkcija išduoda 1, kai įėjimų verčių padaugintų iš svorių suma būna didesnė už 0,5. Tinklą bandoma išmokyti inversijos mechanizmo. Testiniai rinkiniai apmokymui pateikiami priede (8.2.1). Genetiniam algoritmui parinkti sekantys pradiniai parametrai:

Genomo dydis – 81 = 9*9 (neuroninio tinklo svorių kiekis)

Populiacijos dydis – 10

Ribinis kartų skaičius – 1000

Kryžminimo tikimybė – 100%

Mutacijos tikimybė – 20%

Naudojamas atsitiktinio pobūdžio kryžminimas ir mutacija. Kryžminama per vieną tašką.

5.1.2 REZULTATAI

Eksperimento metu sudaryta lentelė (lentelė 1). Pirmose trijose grafose pateikiamos genetinio algoritmo parametrų reikšmės, tai: mutacijos tikimybė, kryžminimo tikimybė ir populiacijos dydis. Paskutinė grafa parodo per kiek iteracijų neuroninis tinklas buvo apmokytas. Kadangi su vienodomis parametrų reikšmėmis buvo atliekami keli bandymai, tai jų rezultatai pateikiami atskirti per kablelį.

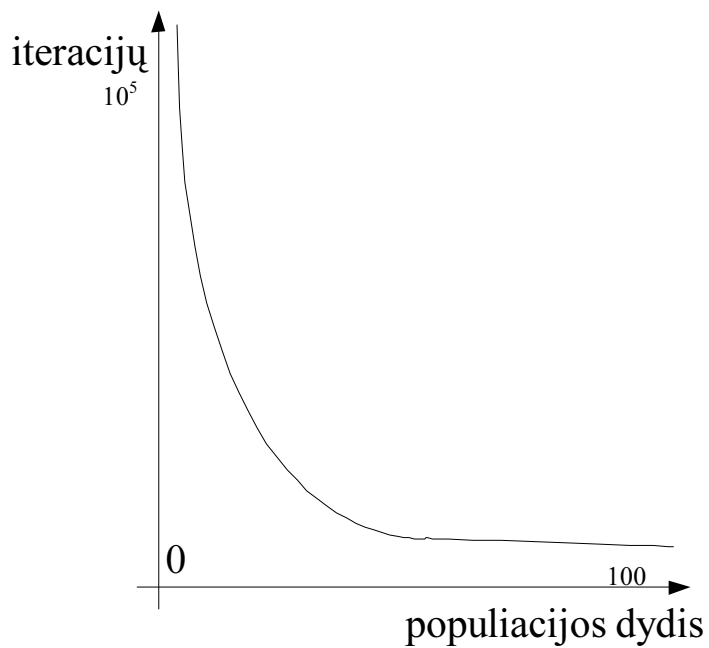
Jei neuroninio tinklo nepavykdavo apmokyti per tūkstantį iteracijų, bandymas buvo nutraukiamas ir į lentelę įrašytas apsimokymo procentas (šimtas procentų atitiktų pilną apmokymą).

lentelė 1 Vienasluoksniu perceptroninio tinklo apmokymo rezultatai, taikant genetinį algoritmą

Mutacijos tikimybė, %	Kryžminimo tikimybė, %	Populiacijos dydis	Iteracijų skaičius	Vidutinis iteracijų skaičius
20	100	10	[93%], [93%], [93%]	[93%]
1	100	10	151, 105, 82, 115, 61, 314	138
5	100	10	95, 175, 98, 282, [96%], 704	400
0	100	10	[70%], [74%], [70%]	[71%]
0.1	100	10	[96%], [93%], [96%]	[95%]
1	50	10	174, 278, 239, 64, 207	192
1	50	100	16, 18, 12, 15, 12	15
1	100	100	12, 22, 32, 11, 14	18
1	100	1000	8, 11, 9	9
1/81	80	10	118, 87, 192, 232, 21, 33, 315	142

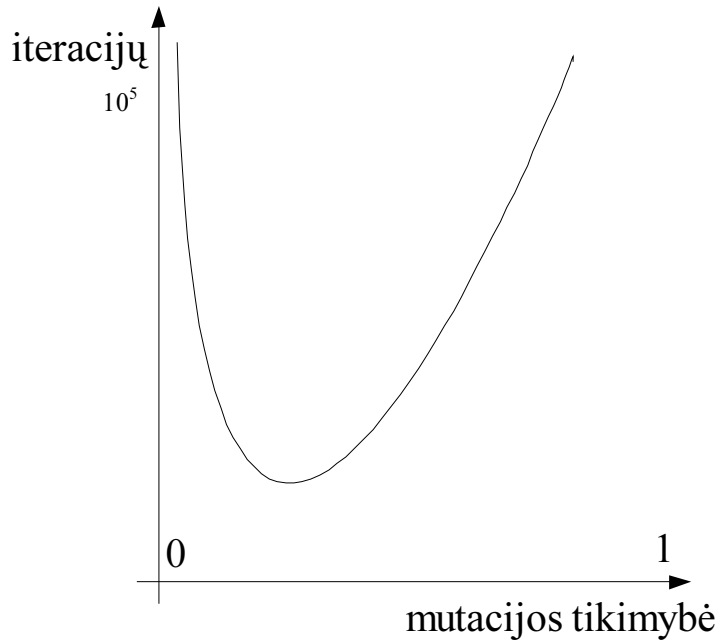
5.1.3 EKSPERIMENTO IŠVADOS

Didėjant populiacijos dydžiui, rezultatas visą laiką pasiekiamas su mažesniu iteracijų skaičiumi. Bet tai dar nenurodo kad kuo didesnė populiacija tuo bus geriau. Apsimokymo iteracijų kiekis, nuo populiacijos dydžio priklauso hiperboliškai (pav. 10). O ir kiekvienam populiacijos individui sumodeliuoti reikalingas papildomas laikas. Galima teigti, kad dvidešimt iteracijų su dešimt individų sumodeliuoti užtruks tiek pat kiek ir dešimt iteracijų su dvidešimt individų. Galiausiai peršasi išvada, kad siekiant optimalumo, populiacijos dydis turėtų būti kiek įmanoma mažesnis. Pernelyg mažas populiacijos dydis gali apskritai neduoti norimo rezultato.



pav. 10 Apsimokymo iteracijų priklausomybė nuo populiacijos dydžio

Tiek per didelė, tiek ir per maža mutacijos tikimybė blogina rezultatus. Egzistuoja aukso vidurys, ties kuriuo kitimas jau nėra toks žymus (pav. 11). Šiuo konkrečiu atveju šis vidurys labai artimas dydžiui, kuris atitiktų vieno geno mutaciją.



pav. 11 Apsimokymo iteracijų priklausomybė nuo mutacijos tikimybės

5.2 GENETINIŲ ALGORITMŲ TAIKYMAS GLOBALAUS OPTIMIZAVIMO UŽDAVINIAMS SPREŠTI

Tradicinių optimizavimo uždavinių bazėje genetiniai algoritmai yra palyginami su žinomais globalaus optimizavimo metodais. Tyrimui panaudoti uždaviniai: Knapsako (*Knapsack*) optimizavimo uždavinys; Mokyklos tvarkaraščio optimizavimas; Dvikovos (*Duel*) optimizavimo uždavinys. Palyginimui pasitelkti plačiai žinomi globalaus optimizavimo metodai: Bajeso (*Bayes*), Monte Karlo paieškos (*Monte Carlo search*), Ekstrapoliacijos tipo metodas (*Unt*), Bajeso (*Bayes*) metodo modifikacija su kryptinga paieška (*Exkor*), Klasterizacijos metodas (*Globt*) ir vienas lokalaus optimizavimo metodas – Bajeso (*Bayes*) metodo modifikacija su žingsnio dydžio kontrole.

Genetinio algoritmo parametrai yra parinkti intuityviai, atsižvelgiant į dažniausiai literatūroje pateikiamus dydžius. Viso tyrimo metu, net ir keičiant užduotis, jie išlieka pastovūs. Tiriamos dvi genetinio algoritmo modifikacijos: bazinė ir su kryptinga kryžminimo strategija.

Kaip rodo tyrimas, net ir paprasta genetinio algoritmo versija be jokių strateginių modifikacijų neatsilieka, o kartais ir pralenkia nagrinėjamus optimizavimo metodus. Dėl genetinių algoritmų vidinės struktūros atsiranda galimybė vystyti įvairias strategijas ir dar labiau pagerinti rezultatus.

Genetiniai algoritmai yra taikomi globalaus optimizavimo uždaviniams spręsti. Pagrindinis tokio taikymo trūkumas – didelis kiekis genetinio algoritmo parametrų. Šiame tyrime nuo šios problemos laikinai atsiribojama ir pasirenkami fiksuoti parametrai. Idėja yra tokia, kad įvedus genetinio algoritmo parametrų optimizavimą rezultatai būtų tik geresni. Šiuo bandymu siekiama patikrinti genetinio algoritmo universalumą sprendžiant skirtingus uždavinius ir palyginti sprendimo kokybę. Pagrindinis palyginimo metodas yra Monte Karlo paieška. Šis metodas būtų kritinė riba įvertinant optimizavimo metodo vertingumą. Optimizavimui pasirinktos skirtingo pobūdžio užduotys su skirtingo stiliaus funkcijomis ir bandoma nustatyti kokios funkcijai koks metodas būtų tinkamiausias.

Kartais nėra galimybės įvairiapusiškai įvertinti optimizavimo metodą konkrečiai užduočiai. Taip gali būti dėl užduočiai sumodeliuoti reikalaujamo per didelio kiekio kompiuterinio laiko. Tokiu atveju apsunkėja galimybės iširti genetinio algoritmo parametrų įtaką optimizavimo našumui. Jei genetiniai algoritmai pasirodytų pakankamai universalūs sprendžiant pasirinktas užduotis, tada galima būtų manyti, kad jie panašiai elgsis ir su kitomis užduotimis. Toliau reiktų atlikti genetinio algoritmų parametrų optimizavimo tyrimą, kuris nustatytų, koks geriausias būdas sukurti

beparametrinius genetinius algoritmus, kuriuos būtų galima drąsiai taikyti sunkiai modeliuojamoms užduotims optimizuoti.

5.2.1 EKSPERIMENTO APŽVALGA

Eksperimentui atlikti pasitelktas globalaus optimizavimo metodų tyrimo paketas GMJ, priklausantis profesoriui J.Mockui (Kauno technologijos universitetas, Informatikos fakultetas). Buvo tiriamas, kiek kokybiškai įvairūs metodai suoptimizuoja įvairias užduotis, atlikę tokį patį skaičių iteracijų. Tyrimams pasirinktos skirtingo pobūdžio užduotys. Tuo buvo siekiama iširti genetinių algoritmų universalumą ir įvertinti gaunamų rezultatų kokybę.

Toliau lentelėse pateikti rezultatai – suvidurkintos testinių rezultatų reikšmės. Kiekvienam vidurkiui gauti buvo atliekama iki šimto bandymų.

Genetinio algoritmo parametrai buvo fiksuoti ir nekito per visą tyrimą. Pasirinkta kryžminimo tikimybė 50 procentų, mutacijos 30 procentų. Populiacijos dydis tesiekė dešimt individų. Vienas geriausias individas buvo išsaugomas nepakitęs. Buvo tiriami du genetiniai algoritmai. Pirmasis veikė be jokios konkrečios strategijos, tiesiog taikydamas bazines genetinio algoritmo idėjas. Antrasis veikė panašiai, tik kryžminimą atlikdavo kryptingai – siekdavo, kad sukryžminti individai nebūtų blogesni už savo tėvus.

5.2.2 KNAPSAKO (KNAPSACK) UŽDAVINYS

Optimizuojant Knapsako uždavinį gaunama sprendimo kokybę nusakanti reikšmė. Kuo reikšmė didesnė, tuo geresnės kokybės yra sprendinys. Knapsako uždavinys yra mažasvoris – reikalaujantis mažai skaičiavimo sąnaudų. Dėl šios priežasties jo optimizavimui kur kas geriau tinka paprasti metodai tokie kaip Monte Karlo paieškos. Šiuo atveju Monte Karlo paieška ir išduoda akivaizdžiai geresnius rezultatus, nei Bajeso metodas (lentelė 3). Tuo tarpu genetinio algoritmo rezultatai turi tendenciją aplenkti Monte Karlo paieškos rezultatus. Palyginus konvergavimą galima pastebėti, kad tiek Bajeso, tiek ir Monte Karlo paieškos metodai konverguoja greičiau nei genetiniai algoritmai, tačiau genetiniai algoritmai pasiekia geresnę funkcijos kokybę (pav. 16).

5.2.3 MOKYKLOS TVARKARAŠČIO OPTIMIZAVIMAS

Optimizuojant mokyklos tvarkaraštį gaunama baudos funkcija. Kuo jos reikšmė mažesnė, tuo geresnis rezultatas. Šiame uždavinyje savo pranašumą parodo klasterizacijos tipo metodas „Globt“, kuris ženkliai aplenkia genetinių algoritmų rezultatus. Bet, savo ruožtu, genetiniai algoritmai lenkia tokius metodus kaip Bajeso ir Monte Karlo (lentelė 4).

5.2.4 DVIKOVOS OPTIMIZAVIMO UŽDAVINYS

Optimizuojant dvikovos uždavinį gaunama tikimybės išlikti gyvam santykinė vertė. Kuo ši vertė didesnė tuo bus didesnė tikimybė likti gyvam po dvikovos. Šiame uždavinyje išryškėja Bajeso metodo pranašumas (lentelė 5). Net ir atlikęs mažai iteracijų jis sugeba gauti neblogus rezultatus. Tuo tarpu genetinis algoritmas atlikęs mažai iteracijų sugeba tik prisivytį Monte Karlo paieškos metodą, bet pritaikius minimalią strategiją jis jau sugeba, nors ir nežymiai, bet aplenkia Monte Karlo. Atlikus daugiau iteracijų genetinį algoritmą jau galima lyginti ne su Monte Karlo paieška, bet su Bajeso metodu.

5.2.5 MATEMATINIŲ FUNKCIJŲ OPTIMIZAVIMAS (SINUSO FUNKCIJA)

Optimizuojant paprastas matematinės funkcijas, Bajeso metodas dėl savo specifikos tai atlieka labai greitai. Į pateiktas konvergavimo kreives reiktų žvelgti atsargiai, nes tai tėra vieno bandymo rezultatas, o ne vidutinė reikšmė, kuri kur kas tiksliau atspindėtų rezultatus. Vertinant bendrai galima pastebėti (pav. 16, pav. 17, pav. 18) kad genetinių algoritmų konvergavimo kreivė yra iškilesnė. Ta prasme, genetiniai algoritmai konverguoja tiesiškiau, nei kiti metodai. Esant mažam iteracijų skaičiui konvergavimo greičiu genetinius algoritmus lenkia kone visi metodai, bet genetiniai algoritmai kur kas lėčiau praranda konvergavimo greitį didėjant iteracijų kiekiui.

5.3 GENETINIŲ ALGORITMŲ TAIKYMAS NEURONINIO TINKLO APMOKYME

Eksperimentas buvo atliekamas J.Mockaus (25) globalaus optimizavimo paketo bazėje. Šis globalaus optimizavimo paketas buvo papildytas mano sukurta universalia neuroninių tinklų modeliavimo priemone (4.2 Neuroninio tinklo modeliavimas). Tokiu būdu atsirado galimybė optimizuoti neuroninius tinklus visais globalaus optimizavimo metodais, o rezultatus palyginti. Optimizavimo metodų sąrašas buvo papildytas genetiniais algoritmais – su galimybe keisti parametrus ranka ir beparametriniu genetiniu algoritmu (4.1 Genetinio algoritmo modeliavimas).

5.3.1 EKSPERIMENTO APŽVALGA

Trijų tipų neuroniniai tinklai yra apmokomi įvairiais metodais. Tai du plačiai žinomi globalaus optimizavimo metodai ir kelios genetinių algoritmų modifikacijos. Tokių metodų, kaip Monte Karlo paieškos (*Monte Carlo search*) ir Bajeso (*Bayes*) pristatinėti nereikia. Greta jų naudojamas paprastas genetinis algoritmas su fiksuotais parametrais ir beparametrinis genetinis algoritmas. Kiekvienam iš šių dviejų genetinių algoritmų pritaikoma kryžminimo strategija ir taip gaunami dar du palyginimo metodai.

Žingsnis link beparametrinio genetinio algoritmo buvo žengtas ankstesnio tyrimo metu (5.2 Genetinių algoritmų taikymas globalaus optimizavimo uždaviniams spręsti). Genetinio algoritmo parametrai: mutacijos, kryžminimo tikimybė, bei populiacijos dydis yra optimizuojami Bajeso metodu (23).

Vienasluoksnį neuroninį tinklą sudaro devyni parceptronai (atitinka 3x3 matricą). Slenkstinė funkcija išduoda 1, kai įėjimų verčių padaugintų iš svorių suma būna didesnė už 0,5. Tinklą bandoma išmokyti inversijos mechanizmo. Testiniai rinkiniai apmokymui pateikiami priede (8.2.1).

Dvisluoksnis neuroninis tinklas ir neuroninis tinklas su grįžtamais ryšiais yra apmokomi atpažinti kelias raides pateiktas matricoje 8x8. Testiniai rinkiniai apmokymui pateikiami priede (8.4.1).

Fiksuotų parametrų genetinio algoritmo parametrai:

Populiacijos dydis – 10

Kryžminimo tikimybė – 100%

Mutacijos tikimybė – 20%

Naudojamas atsitiktinio pobūdžio mutacija. Kryžminama per vieną tašką.

Kiekvienam vidurkiui gauti buvo atliekama apie dešimt bandymų.

5.3.2 REZULTATAI

Eksperimento metu buvo sudaryta lentelė (lentelė 2). Pirmame stulpelyje pateikiamas apmokančio algoritmo pavadinimas. Toliau seka apmokymo trukmė milisekundėmis su trijų tipų neuroniniais tinklais.

Į J.Mockaus globalaus optimizavimo paketą įtrauktas Bajeso metodas turi apribojimus – jis gali optimizuoti tik dvidešimt penkis parametrus. Dėl pastarosios priežasties nebuvo galimybės gauti skaičiavimo rezultatus su didesniais neuroniniais tinklais.

lentelė 2 Neuroninių tinklų apmokymo įvairiais metodais palyginimas

Algoritmas	Darbo trukmės vidurkis, ms		
	Vienasluoksnis NT	Dvisluoksnis NT	NT su grįžtamais ryšiais
Beparametrinis genetinis algoritmas	21020	3150	29301
Beparametrinis genetinis algoritmas su kryžminimo strategija	18717	3011	31070
Fiksuotų parametų genetinis algoritmas	24902	3110	33022
Fiksuotų parametų genetinis algoritmas su kryžminimo strategija	23102	3011	32011
Monte Karlo (<i>Monte Carlo</i>) metodas	29557	4812	51102
Bajeso (<i>Bayes</i>) metodas	28002	-	-

Šis eksperimentas leidžia pastebėti beparametrinio genetinio algoritmo darbo efektyvumą. Fiksuotų parametų genetinis algoritmas, gerai parinkus parametrus, visada dirbs greičiau nei beparametrinis. Tačiau tinkamų parametų atradimas taip pat atims nemažai laiko. Beparametrinis genetinis algoritmas tinkamiausius parametrus aptinka automatizuotai.

6 IŠVADOS

Atliekant eksperimentus su genetiniais algoritmais pastebėtas jų išskirtinis universalumas optimizavimo uždaviniuose. Priede (8.3.1) pateiktose lentelėse galima pastebėti kaip optimizuojant įvairiausių uždavinius genetiniai algoritmai visada yra tarp pirmaujančiųjų. Šitokią jų universalumą pastebėjo ir Holandas (*J.H.Holland*) (26). Toks universalumas eksperimento metu leido padaryti grandiozinę išvadą – jei genetiniai algoritmai gerai optimizuos įvairias lengvai modeliuojamas užduotis, tai lygiai tiek pat gerai jie optimizuos ir sunkiai sumodeliuojamus uždavinius. Vėliau ši išvada dar kartą buvo patvirtinta paskutiniojo eksperimento metu (lentelė 2). Ši genetinių algoritmų savybė leidžia atlikti bandymus su lengviau modeliuojamais uždaviniais, o gautus rezultatus tiesiogiai taikyti sudėtingesniems.

Eksperimentuojant su vienasluoksniu neuroniniu tinklu ir tyrinėjant tinkamiausią mutacijos tikimybę buvo pastebėta, kad ji artima tikimybei mutuoti vienam genui (lentelė 1). Tai sudarė prielaidą panaudoti vieno geno mutaciją. Šis mutacijos tipas yra lengviau realizuojamas ir veikia žymiai greičiau. Mutacija yra efektyvi, jei mutuoja bent vienas genas. Tačiau pernelyg didelis mutuojančių genų kiekis privestų prie atsitiktinės paieškos (pav. 11). Pasiremiant ankstesne išvada, galima būtų manyti, kad vieno geno mutacija yra tinkamiausia neuroninių tinklų apmokyje.

Beparametrinių genetinių algoritmų (23) tyrimas parodė jų tinkamumą klasifikavimo uždaviniams spręsti (lentelė 2). Nors parametrizuotas genetinis algoritmas su optimaliais parametrais niekada nenusileis beparametriniam, jo panaudojimas yra labai ribotas. Norint atrasti optimalius parametrus reikia būti šios srities specialistu, bet net ir tada paieškai bus sugaišta nemažai laiko. Beparametriniai genetiniai algoritmai užtikrina kokybišką sprendinį net ir nieko nenusimanant apie genetinius algoritmus.

7 LITERATŪRA

1. Box G. E. P. Evoliucinės operacijos: Industrijos produktyvumo didinimo metodas. Taikomoji statistika, 1957.
2. Fraser A. S. Genetinių sistemų modeliavimas automatiniais skaitmeniniais kompiuteriais 1: Įvadas. Australijos biologijos mokslo žurnalas, 1957.
3. Friedman G. J. Evoliucinių procesų skaitmeninis modeliavimas. Bendrųjų sistemų metų knyga, 1959.
4. Bledsoe W. W. Biologinių principų taikymas analitiniame sistemų studijavime. Pristatytas ORSA-TIMS nacionaliniame susitikime, San Franciskas, 1961.
5. Bremermann H. J. Optimizacija naudojant evoliuciją ir kryžminimą. Iš Yovits M. C., Jacobi G. T., Goldstein G. D. leidimo Save organizuojančios sistemos, 1962.
6. Reed J., Toombs R., Barricelli N. A. Biologinės evoliucijos ir mašinų apmokymo modeliavimas. Teorinės biologijos žurnalas, 1967.
7. Barricelli N. A. Dirbtiniais metodais realizuotas simbiogenetinis evoliucijos procesas. Metodai, 1957.
8. Barricelli N. A. Evoliucinių teorijų skaitmeninis testavimas. ACTA Bioteorija, 1962
9. Fraser A. S. Genetinių sistemų modeliavimas automatiniais skaitmeniniais kompiuteriais 2: Ryšių efektas žingsnio parinkime. Australijos biologijos mokslo žurnalas, 1957.
10. Martin F. G. Cockerham C. C. Didelio greičio parinkimo studijos. Iš Kempthorne O. leidimo Biometrinė genetika. Pergamonas, 1960.
11. Holland J. H. Prisisitaikymas natūraliose ir dirbtinėse sistemose. Mičigano universiteto leidykla, 1975.
12. Rechenberg I. Evoliucijos strategija: Optimizavimo technikos sistema biologinės evoliucijos principais. Frommann-Holzboog, 1973.
13. Schwefel H. P. Kompiuterinių modelių skaitmeninė optimizacija. Wiley, Chichester, UK, 1981.
14. Fogel L., Owens A., Walsh M. Dirbtinis intelektas evoliuciniame modeliavime. Wiley, Niu Jorkas, 1966.
15. Back T. Fogel D. B. Michalewicz Z. Evoliucinio skaičiavimo vadovas. Oksfordo universiteto fizikos instituto leidykla, Niu Jorkas ir Bristolis, 1997.
16. Goldberg D. E. Genetiniai algoritmai paieškoje, optimizavime ir mašinų mokyme. Addison-Wesley, 1989.
17. Luke S. Evoliucinių skaičiavimų ir genetinio programavimo tyrimo sistema, Java kalboje. Evoliucinių skaičiavimų laboratorija, Džordžo Masono universitetas [interaktyvus]. 2004, kovas [žiūrėta 2004-04-15]
18. Уоссермен Ф. Neurokompiuterinė technika: Teorija ir praktika = Нейрокомпьютерная техника: Теория и практика, 1992. Vertimas į rusų kalbą, Зуев Ю. А., Точенов В. А., 1992.
19. Терехов С. А. Dirbtinių neuroninių tinklų teorija ir taikymas. Paskaitos. = Лекции по теории и приложениям искусственных нейронных сетей. Dirbtinių neuroninių tinklų laboratorija, НТО-2, ВНИИТФ, Снежинск, 1994.
20. Schultz A. C. Genetinių algoritmų archyvas = The Genetic Algorithms Archive. Karinis dirbtinio intelekto taikomųjų tyrimų centras (*The Navy Center for Applied Research in Artificial Intelligence*) [interaktyvus]. 2004, gegužė [žiūrėta 2004-05-11]. Prieiga per internetą: <http://www.aic.nrl.navy.mil/galist>
21. Holland J. H. Genetiniai algoritmai. Amerikos mokslas, žurnalas, 1992.

22. Obitko M. Įvadas į genetinius algoritmus = Introduction to Genetic Algorithms. University of Applied Sciences [interaktyvus]. 2003 balandis [žiūrėta 2003-05-05]. Prieiga per internetą: <http://cs.felk.cvut.cz/~xobitko/ga/>
23. Lobo F. G., Goldberg D. E. Beparametrinis genetinis algoritmas praktikoje = The Parameterless Genetic Algorithm in Practice: IlliGAL Report No. 2001022, Iliojaus genetinių algoritmų laboratorija, 2001 birželis.
24. Pelikan M., Goldberg D. E., Cantu-Paz. E. Bajeso optimizavimo algoritmas = BOA: the bayesian optimization algorithm. GECCO-99: Genetinių ir evoliucinių skaičiavimų konferencijos medžiaga, San Franciskas, CA: Morgan Kaufmann, 1999.
25. Mockus J. Globalaus ir diskretaus optimizavimo pavydžių rinkinys = A Set of Examples of Global and Discrete Optimization. Kauno technologijos universitetas, programinės įrangos katedra [interaktyvus]. 2004, vasaris [žiūrėta 2004-04-17]. Prieiga per internetą: <http://soften.ktu.lt/~mockus>.

8 PRIEDAI

8.1 GLOBALAUS OPTIMIZAVIMO PAKETO „GMJ“ DARBO PAVYZDYS

The screenshot shows the configuration window for the MethodGA algorithm. It features a header with tabs for 'Method', 'Task', and 'Operation'. A dropdown menu on the right allows selecting the method and properties, currently set to 'MethodGA'. The main area is a table with two columns: 'Property' and 'Value'. The parameters are as follows:

Property	Value
Iterations	10000
Stop if <=	0
Individuals	10
Mutation prob.	0.3
Crossover prob.	0.5
Elites perc.	0.01
Mutation type:	Single gene mutation
Mutating part	0.05
Mutation best search level	2
Crossover type:	Simple crossover
Crossover style:	Single point crossover
Crossover best search level	2

Mantas Balnys, KTU

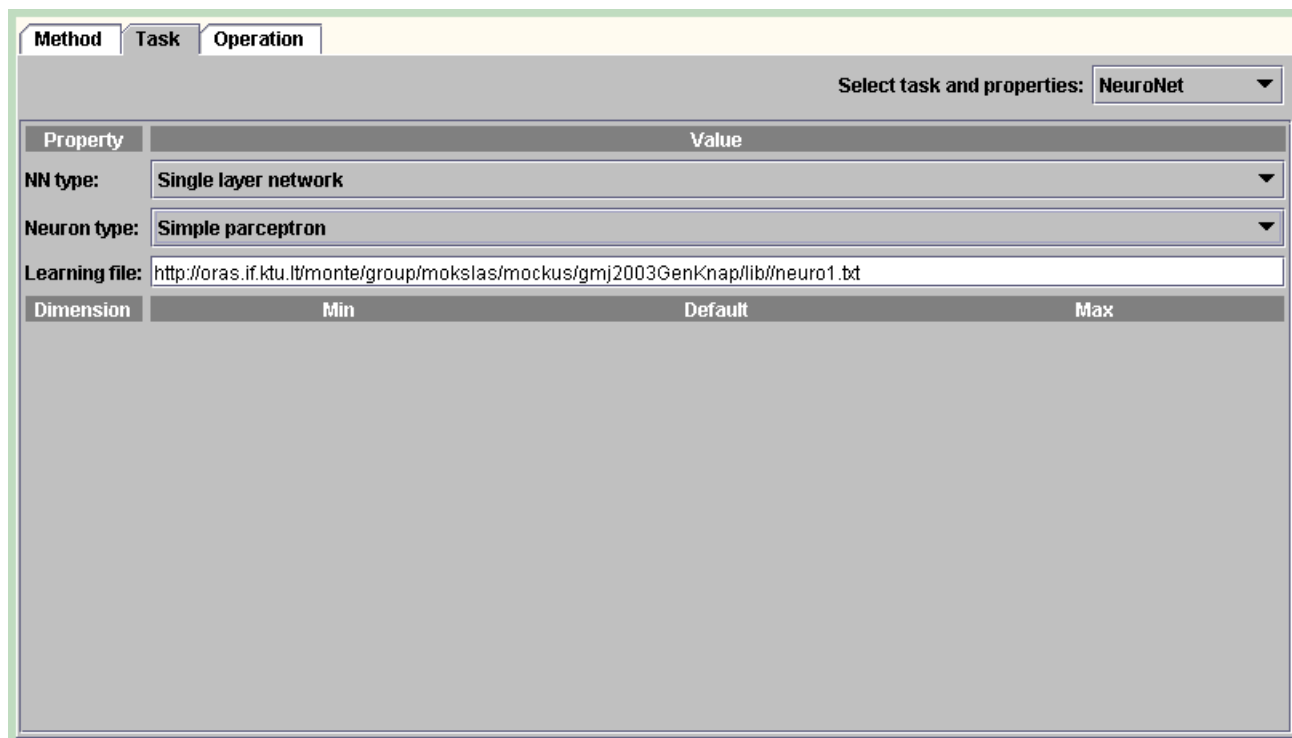
pav. 12 Parametruoto genetinio algoritmo – optimizavimo metodo pasirinkimas

The screenshot shows the configuration window for the MethodGAopt algorithm. It features a header with tabs for 'Method', 'Task', and 'Operation'. A dropdown menu on the right allows selecting the method and properties, currently set to 'MethodGAopt'. The main area is a table with two columns: 'Property' and 'Value'. The parameters are as follows:

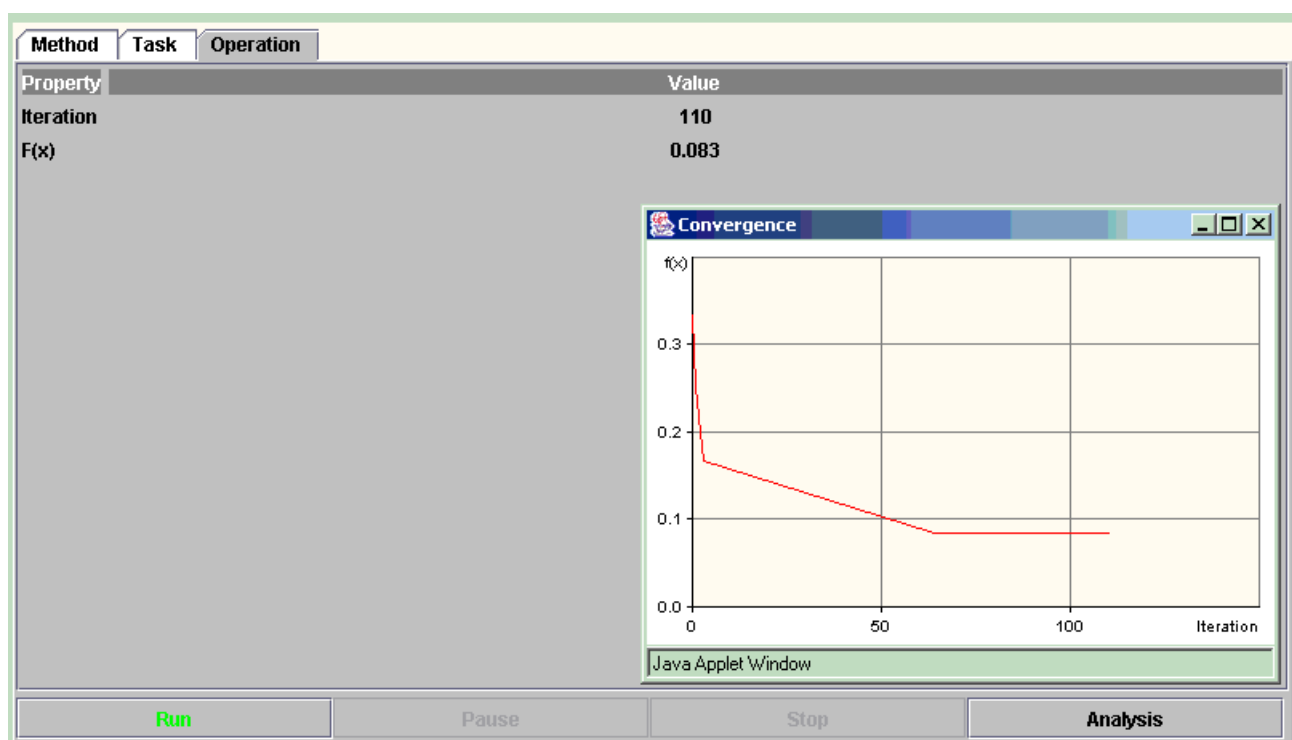
Property	Value
Bayes options	
Iterations	100
Initial points	5
GA iterations when optimized	20
advanced Genetic Algorithm options	
Iterations	100000
Stop if <=	0
Mutation type:	Simple partial mutation
Mutating part	0.05
Mutation best search level	2
Crossover type:	Simple crossover
Crossover style:	Random crossover
Crossover best search level	2

Mantas Balnys, KTU

pav. 13 Beparametrinio genetinio algoritmo – optimizavimo metodo pasirinkimas



pav. 14 Neuroninio tinklo – optimizavimo uždavinio pasirinkimas



pav. 15 „GMJ“ veikimo pavyzdys. Stebimas genetinio algoritmo konvergavimas apmokant klasifikuojantį neuroninį tinklą.

8.2 EKSPERIMENTAS „VIENASLUOKSNIO PARCEPTRONINIO TINKLO APMOKYMAS“

8.2.1 TESTINIAI DUOMENYS

INPUTS:

1 1 0

1 0 1

0 1 1

END

OUTPUTS:

0 0 1

0 1 0

1 0 0

END

INPUTS:

0 1 0

1 1 1

0 1 0

END

OUTPUTS:

1 0 1

0 0 0

1 0 1

END

INPUTS:

1 1 1

1 0 1

1 1 1

END

OUTPUTS:

0 0 0

0 1 0

0 0 0

END

8.3 EKSPERIMENTO „GENETINIŲ ALGORITMŲ TAIKYMAS GLOBALAUS OPTIMIZAVIMO UŽDAVINIAMS SPREŠTI“ REZULTATAI

8.3.1 ITERACIJŲ PALYGINIMAS

GA	genetinis algoritmas
GAopt	genetinis algoritmas su kryžminimo strategija
MC	Monte Karlo paieška
Bayes	Bajeso metodas
Exkor	Bajeso metodo modifikacija su kryptinga paieška

LBayes Bajeso metodo modifikacija su žingsnio dydžio kontrole

Globt Klasterizacijos tipo metodas

Unt Ekstrapoliacijos tipo metodas

lentelė 3 Kokybės funkcija optimizuojant Knapsako uždavinį įvairiais metodais.

iteracijų	GA	GAopt	MC	Bayes	Exkor	LBayes	Globt	Unt
100	9410	9350	9130	8770	7370	7170	8800	8770
1000	9710	9960	9700	9660	7570	7570	9650	8800

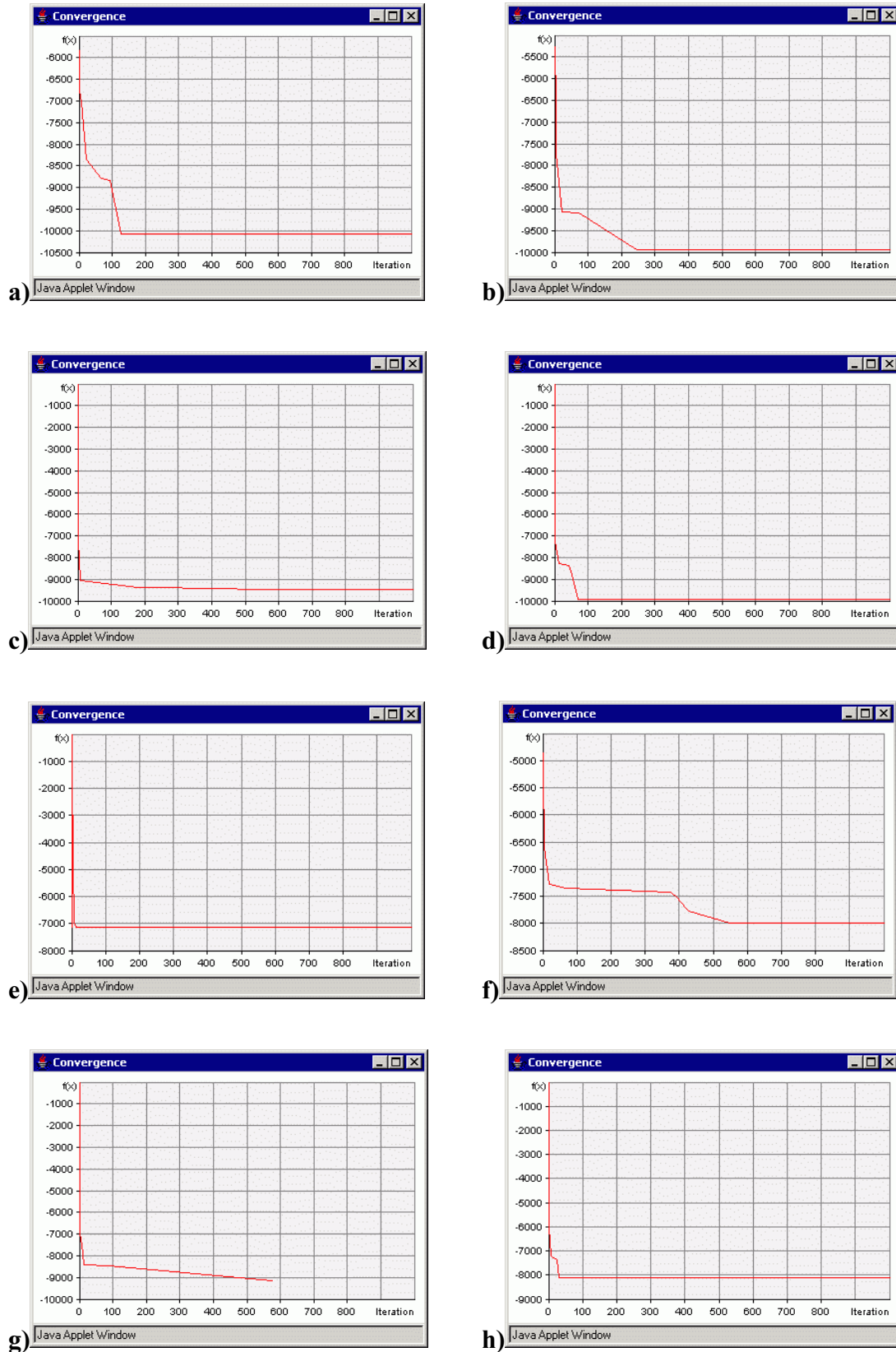
lentelė 4 Baudos funkcija optimizuojant mokyklos tvarkaraštį įvairiais metodais.

iteracijų	GA	GAopt	MC	Bayes	Exkor	LBayes	Globt	Unt
100	15,7	15,6	16,4	17,5	17,3	16,7	15	18,8

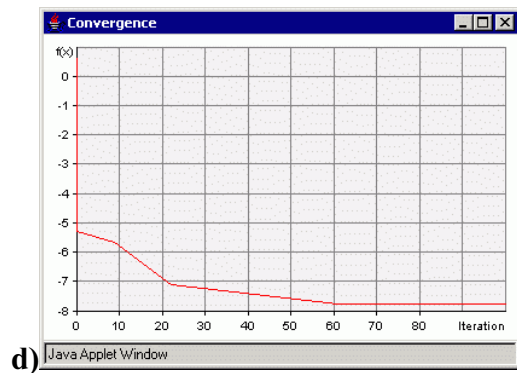
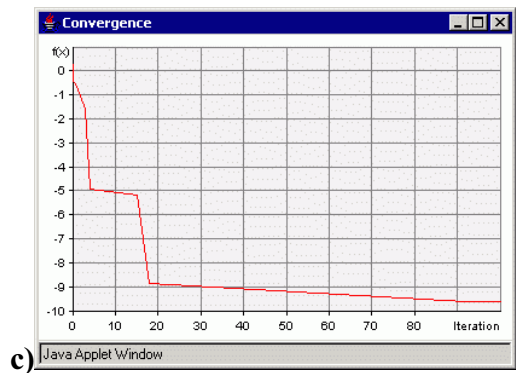
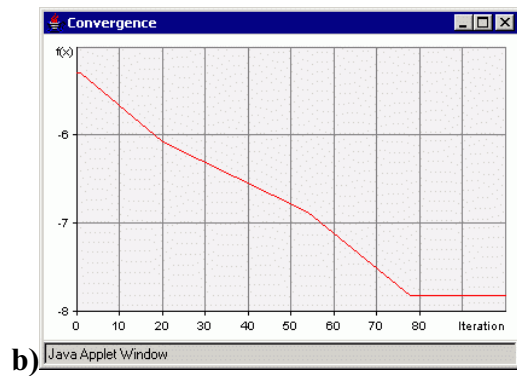
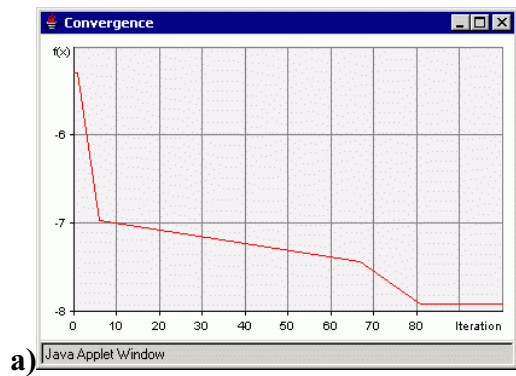
lentelė 5 Kokybės funkcija optimizuojant Dvikovos uždavinį įvairiais metodais.

iteracijų	GA	GAopt	MC	Bayes	Exkor	LBayes	Globt	Unt
100	8,5	8,8	8,6	9,3	-	-	8,0	8,3
1000	11,0	11,1	10,2	11,5	-	-	10,5	8,8

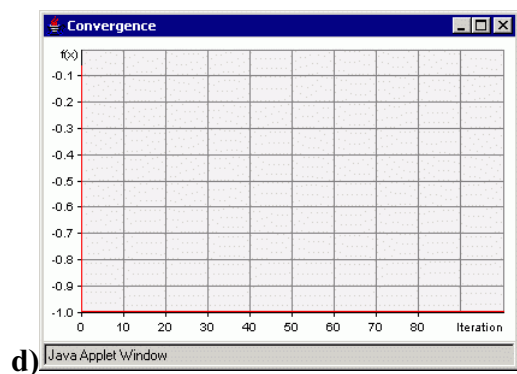
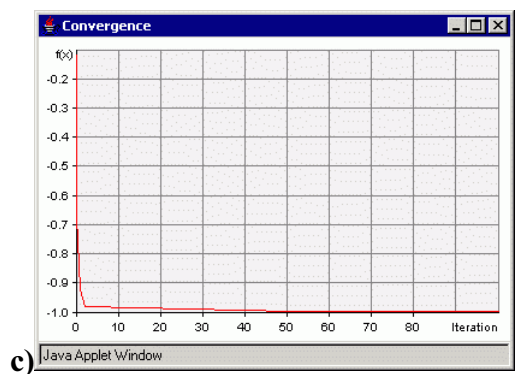
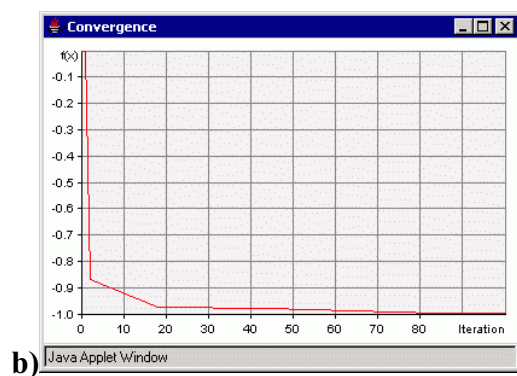
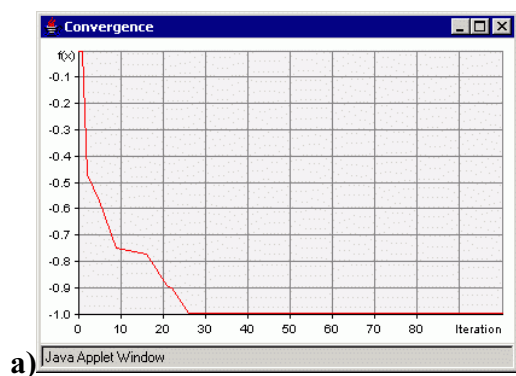
8.3.2 METODŲ KONVERGAVIMAS

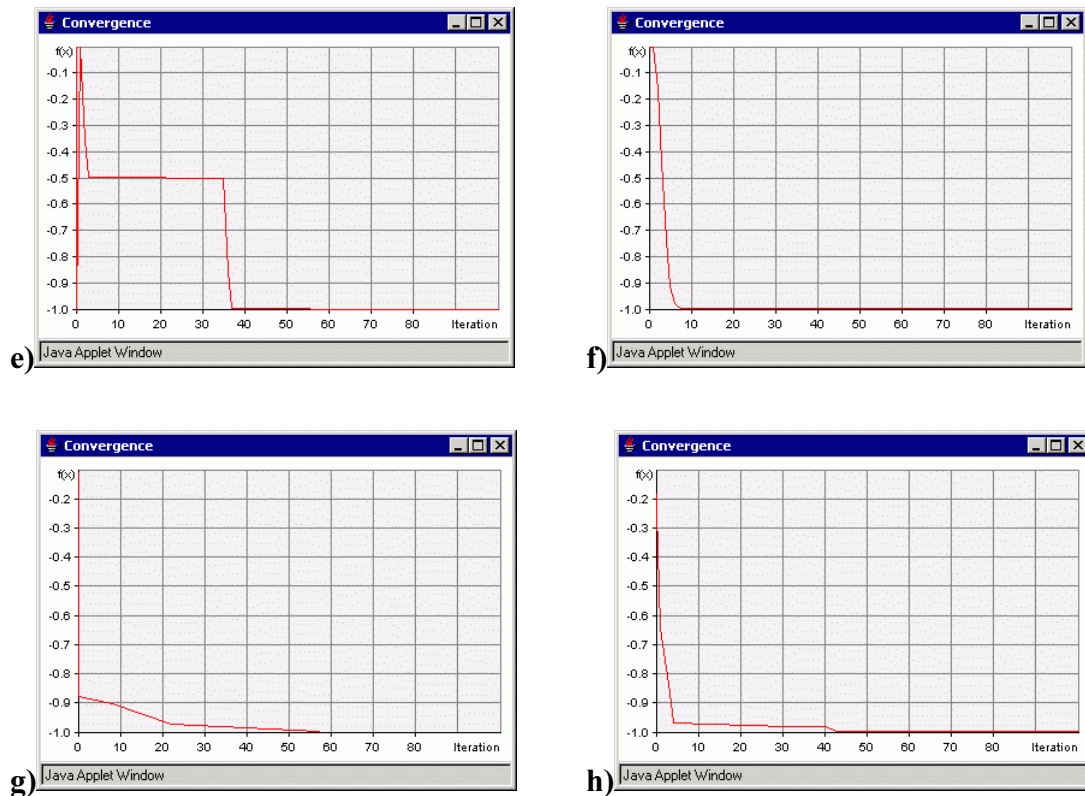


pav. 16 Optimizavimo metodų konvergavimas sprendžiant Knapsako uždavinį. a) genetinis algoritmas, b) genetinis algoritmas su kryžminimo strategija, c) Monte Karlo paieška, d) Bajeso metodas, e) "Exkor", f) "LBayes", g) "Globt", h) "Unt".



pav. 17 Optimizavimo metodų konvergavimas sprendžiant Dvikovos uždavinį. a) genetinis algoritmas, b) genetinis algoritmas su kryžminimo strategija, c) Monte Karlo paieška, d) Bajeso metodas.





pav. 18. Optimizavimo metodų konvergavimas sprendžiant paprastą matematinę funkciją (sinuso funkciją). a) genetinis algoritmas, b) genetinis algoritmas su kryžminimo strategija, c) Monte Karlo paieška, d) Bajeso metodas, e) “Exkor”, f) “LBayes”, g) “Globt”, h) “Unt”.

8.4 EKSPERIMENTAS „GENETINIŲ ALGORITMŲ TAIKYMAS NEURONINIO TINKLO APMOKYME“

8.4.1 TESTINIAI DUOMENYS ATPAŽINIME

```

INPUTS :
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
1 1 1 1 1 1 1 1
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
END
OUTPUTS :
1 0 0 0 0 0 0 0
END
INPUTS :
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 0 1 0 0 1 0 0
0 0 1 0 0 1 0 0
0 1 1 1 1 1 1 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
1 1 0 0 0 0 1 1

```

```
END
OUTPUTS:
1 0 0 0 0 0 0 0
END
INPUTS:
0 1 1 1 1 0 0 0
0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0
0 1 1 1 1 1 0 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
0 1 1 1 1 1 0 0
END
OUTPUTS:
0 1 0 0 0 0 0 0
END
INPUTS:
0 0 1 1 1 1 0 0
0 1 0 0 0 0 1 0
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
1 0 0 0 0 0 0 1
0 1 0 0 0 0 1 0
0 0 1 1 1 1 0 0
END
OUTPUTS:
0 0 1 0 0 0 0 0
END
```

8.5 PRANEŠIMAS „GENETINIŲ ALGORITMŲ TAIKYMAS GLOBALAUS OPTIMIZAVIMO UŽDAVINIAMS SPREŠTI“

INFORMACINĖS TECHNOLOGIJOS'2004

Iš konferencijų ciklo "LIETUVOS MOKSLAS IR PRAMONĖ"



Kauno technologijos universitetas
Informatikos fakultetas
2004 m. sausio 28-29 d.

Konferencijos medžiaga platinama elektroninėje formoje – kompaktinėse plokštelėse, taip pat yra ir spausdintas konferencijos medžiagos leidimas.

GENETINIŲ ALGORITMŲ TAIKYMAS GLOBALAUS OPTIMIZAVIMO UŽDAVINIAMS SPREŠTI

Mantas Balnys

Kauno Technologijos Universitetas, Informatikos Fakultetas, Kompiuterių Katedra

Tradicinių optimizavimo uždavinių bazėje genetiniai algoritmai yra palyginami su žinomais globalaus optimizavimo metodais. Tyrimui panaudoti uždaviniai: Knapsako (Knapsack) optimizavimo uždavinys; Mokyklos tvarkaraščio optimizavimas; Dvikovos (Duel) optimizavimo uždavinys. Palyginimui pasitelkti plačiai žinomi globalaus optimizavimo metodai: Bajeso (Bayes), Monte Karlo paieška (Monte Carlo search), Ekstrapoliacijos tipo metodas (Unt), Bejeso (Bayes) metodo modifikacija su kryptinga paieška (Exkor), Klasterizacijos metodas (Globt) ir vienas lokalaus optimizavimo metodas – Bajeso (Bayes) metodo modifikacija su žingsnio dydžio kontrole.

Genetinio algoritmo parametrai yra parinkti intuityviai, atsižvelgiant į dažniausiai literatūroje pateikiamus dydžius. Viso tyrimo metu, net ir keičiant užduotis, jie išlieka pastovūs. Tiriamos dvi genetinio algoritmo modifikacijos: bazinė ir su kryptinga kryžminimo strategija.

Kaip parodo tyrimas, net ir paprasta genetinio algoritmo versija be jokių strateginių modifikacijų neatsilieka, o kartais ir pralenkia nagrinėjamus optimizavimo metodus. O dėl genetinių algoritmų vidinės struktūros atsiranda galimybė vystyti įvairias strategijas ir dar labiau pagerinti rezultatus.

1 Įvadas

Genetiniai algoritmai yra taikomi globalaus optimizavimo uždaviniams spresti. Pagrindinis tokio taikymo trūkumas, tai didelis kiekis genetinio algoritmo parametrų. Šiame tyrime nuo šios problemos laikinai atsiribojama ir pasirenkami fiksuoti parametrai. Idėja yra tokia, kad įvedus genetinio algoritmo parametrų optimizavimą rezultatai būtų tik geresni. Šio bandymo metu siekiama patikrinti genetinio algoritmo universalumą sprendžiant skirtingus uždavinius ir palyginti sprendimo kokybę. Pagrindinis metodas, su kuriuo siekiama sulygtinti yra Monte Karlo paieška. Šis metodas būtų kritinė riba įvertinant optimizavimo metodo vertingumą. Optimizavimui pasirinktos skirtingo pobūdžio užduotys su skirtingo stiliaus funkcijomis ir bandoma nustatyti kokius funkcijai, koks metodas būtų tinkamiausias.

Kartais nėra galimybės įvairiapusiškai įvertinti optimizavimo metodą konkrečiai užduočiai. Taip gali būti dėl užduočiai sumodeliuoti reikalaujamo per didelio kiekio kompiuterinio laiko. Tokiu atveju ap sunkėja galimybės iširti genetinio algoritmo parametrų įtaką optimizavimo našumui. Jei genetiniai algoritmai pasirodytų pakankamai universalūs sprendžiant pasirinktas užduotis, tada galima būtų manyti, kad jie panašiai elgsis ir su kitomis užduotimis. Toliau reiktų atlikti genetinio algoritmų parametrų optimizavimo tyrimą, kuris nustatytų koks geriausias būdas sukurti beparametrius genetinius algoritmus, kuriuos būtų galima drąsiai taikyti sunkiai modeliuojamoms užduotims optimizuoti.

2 Eksperimento apžvalga

Eksperimentui atlikti pasitelktas globalaus optimizavimo metodų tyrimo paketas GMJ, priklausantis profesoriui J.Mockui, Kauno Technologijos Universitetas, Informatikos fakultetas. Buvo tiriama kiek kokybiškai įvairūs metodai suoptimizuoja įvairias užduotis atlikę tokį patį skaičių iteracijų. Tyrimams pasirinktos skirtingo pobūdžio užduotys. Tuo buvo siekiama iširti genetinių algoritmų universalumą ir įvertinti gaunamų rezultatų kokybę.

Toliau lentelėse pateikti rezultatai, tai suvidurkintos testinių rezultatų reikšmės. Kiekvienam vidurkiui gauti buvo atliekama iki šimto bandymų.

Genetinio algoritmo parametrai buvo fiksuoti ir nekito viso tyrimo metu. Pasirinkta kryžminimo tikimybė 50 procentų, mutacijos 30 procentų. Populiacijos dydis tesiekė dešimt individų. Vienas geriausias individas buvo išsaugomas nepakitęs. Buvo tiriami du genetiniai algoritmai. Pirmasis veikė be jokios konkrečios strategijos, tiesiog taikydamas bazines genetinio algoritmo idėjas. Antrasis veikė panašiai, tik kryžminimą atlikdavo kryptingai – siekdavo kad sukryžminti individai nebūtų blogesni už savo tėvus.

3 Knapsako (Knapsack) uždavinys

Optimizuojant Knapsako uždavinį gaunama sprendimo kokybę nusakanti reikšmė. Kuo reikšmė didesnė, tuo geresnės kokybės yra sprendinys. Knapsako uždavinys yra mažasvoris – reikalaujantis mažai skaičiavimo sąnaudų. Dėl šios priežasties jo optimizavimui kur kas geriau tinka paprasti metodai tokie kaip Monte Karlo paieškos. Šiuo atveju Monte Karlo paieška ir išduoda akivaizdžiai geresnius rezultatus, nei Bajeso metodas (1 lentelė). Tuo tarpu genetinio algoritmo rezultatai turi tendenciją aplenksti Monte Karlo paieškos rezultatus. Palyginus konvergavimą galima pastebėti, kad tiek Bajeso, tiek ir Monte Karlo paieškos metodai konverguoja greičiau nei genetiniai algoritmai, tačiau genetiniai algoritmai pasiekia geresnę funkcijos kokybę.

1 lentelė. Kokybės funkcija optimizuojant Knapsako uždavinį įvairiais metodais.

iteracijų	GA	GAopt	MC	Bayes	Exkor	LBayes	Globt	Unt
100	9410	9350	9130	8770	7370	7170	8800	8770
1000	9710	9960	9700	9660	7570	7570	9650	8800

GA genetinis algoritmas

GAopt genetinis algoritmas su kryžminimo strategija

MC Monte Karlo paieška

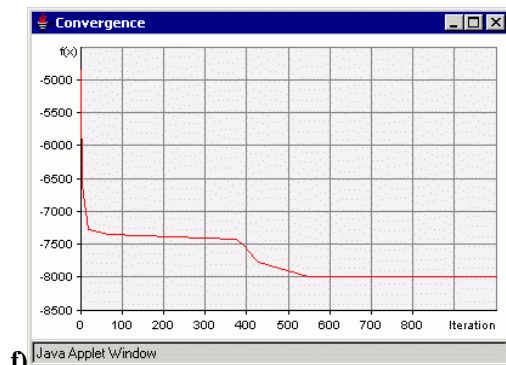
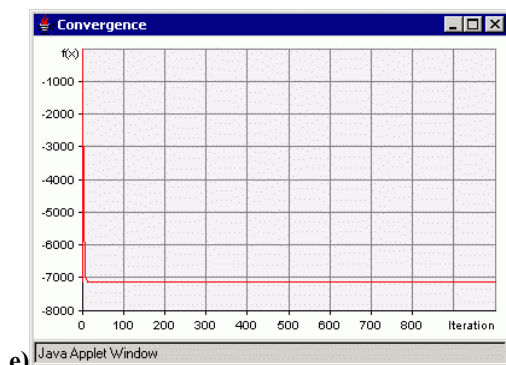
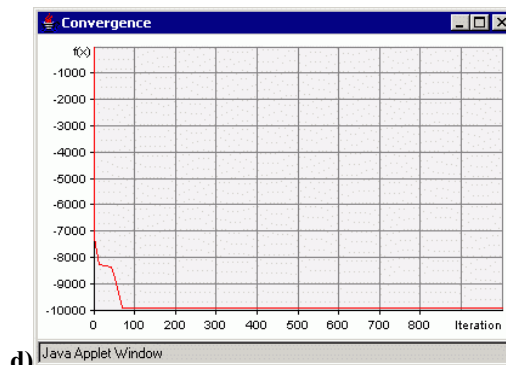
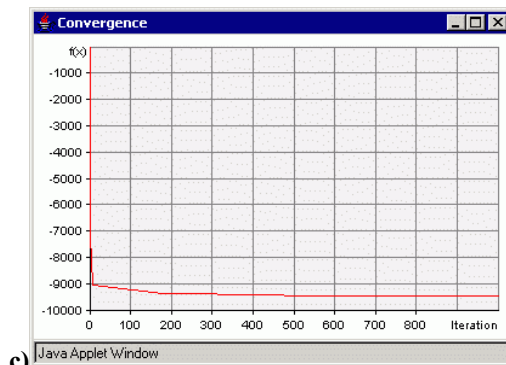
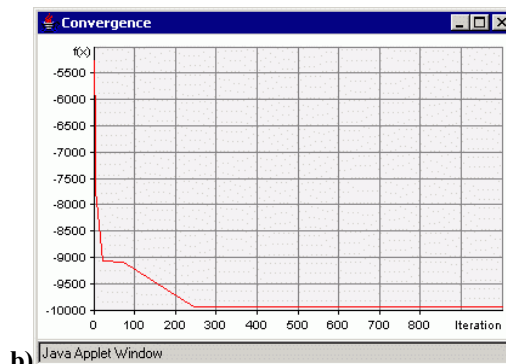
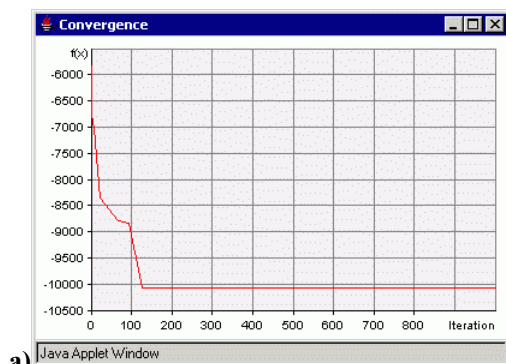
Bayes Bajeso metodas

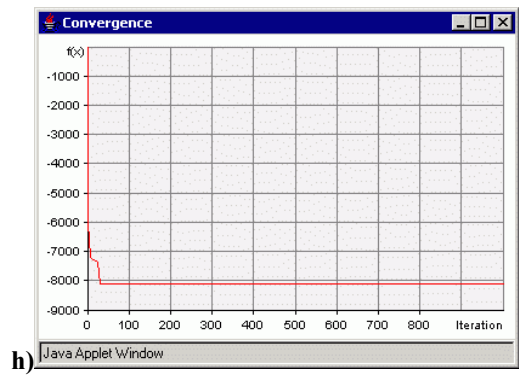
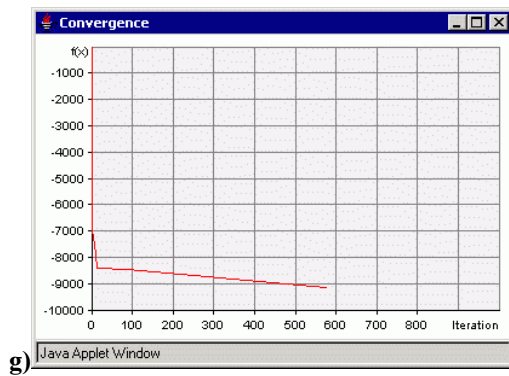
Exkor Bajeso metodo modifikacija su kryptinga paieška

LBayes Bajeso metodo modifikacija su žingsnio dydžio kontrole

Globt Klasterizacijos tipo metodas

Unt Ekstrapoliacijos tipo metodas





1 pav. Optimizavimo metodų konvergavimas sprendžiant Knapsako uždavinį. a) genetinis algoritmas, b) genetinis algoritmas su kryžminimo strategija, c) Monte Karlo paieška, d) Bajeso metodas, e) “Exkor”, f) “LBayes”, g) “Globt”, h) “Unt”.

1 Mokyklos tvarkaraščio optimizavimas

Optimizuojant mokyklos tvarkaraštį gaunama baudos funkcija. Kuo jos reikšmė mažesnė tuo geresnis rezultatas. Šiame uždavinyje savo pranašumą parodo klasterizacijos tipo metodas “Globt”, kuris ženkliai aplenkia genetinių algoritmų rezultatus. Bet savo ruožtu genetiniai algoritmai lenkia tokius metodus kaip Bajeso ir Monte Karlo (2 lentelė).

1 lentelė. Baudos funkcija optimizuojant mokyklos tvarkaraštį įvairiais metodais.

iteracijų	GA	GAopt	MC	Bayes	Exkor	LBayes	Globt	Unt
100	15,7	15,6	16,4	17,5	17,3	16,7	15	18,8

GA genetinis algoritmas

GAopt genetinis algoritmas su kryžminimo strategija

MC Monte Karlo paieška

Bayes Bajeso metodas

Exkor Bajeso metodo modifikacija su kryptinga paieška

LBayes Bajeso metodo modifikacija su žingsnio dydžio kontrole

Globt Klasterizacijos tipo metodas

Unt Ekstrapoliacijos tipo metodas

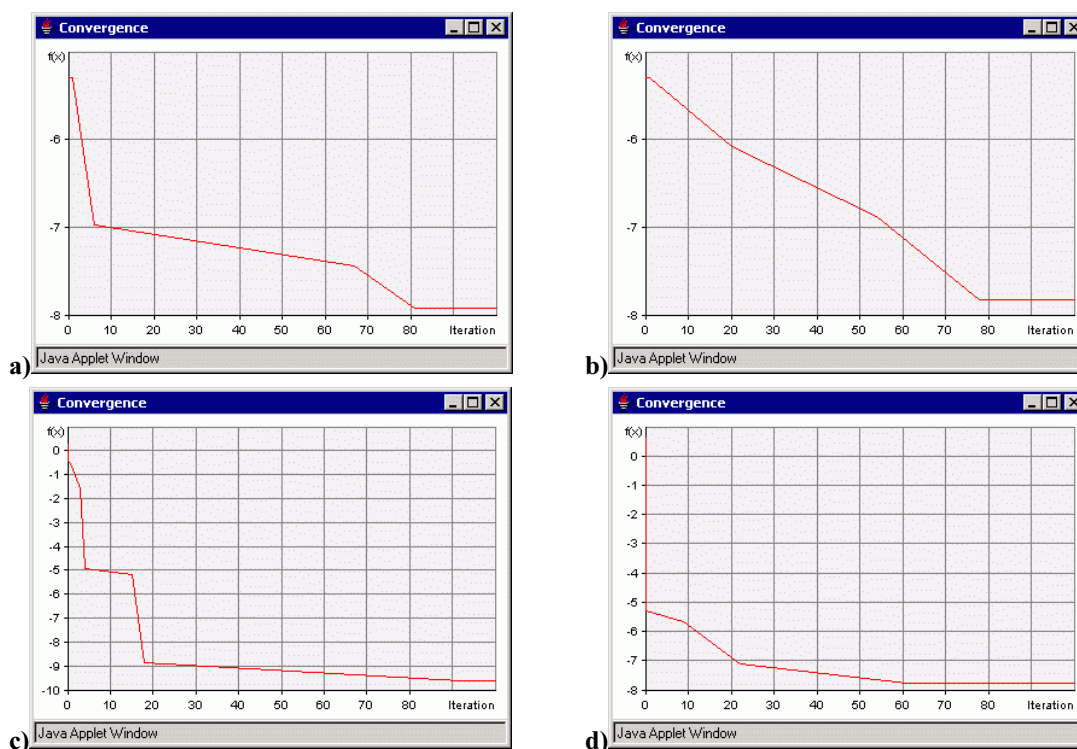
2 Dvikovos optimizavimo uždavinys

Optimizuojant dvikovos uždavinį gaunama tikimybės išlikti gyvam santykinė vertė. Kuo ši vertė didesnė tuo bus didesnė tikimybė likti gyvam po dvikovos. Šiame uždavinyje išryškėja Bajeso metodo pranašumas (3 lentelė). Net ir atlikęs mažai iteracijų jis sugeba gauti neblogus rezultatus. Tuo tarpu genetinis algoritmas atlikęs mažai iteracijų sugeba tik prisivyti Monte Karlo paieškos metodą, bet pritaikius minimalią strategiją jis jau sugeba, nors ir nežymiai, bet aplenkia Monte Karlo. Atlikus daugiau iteracijų genetinį algoritmą jau galima lyginti ne su Monte Karlo paieška, bet su Bajeso metodu.

1 lentelė. Kokybės funkcija optimizuojant Dvikovos uždavinį įvairiais metodais.

iteracijų	GA	GAopt	MC	Bayes	Exkor	LBayes	Globt	Unt
100	8,5	8,8	8,6	9,3	-	-	8,0	8,3
1000	11,0	11,1	10,2	11,5	-	-	10,5	8,8

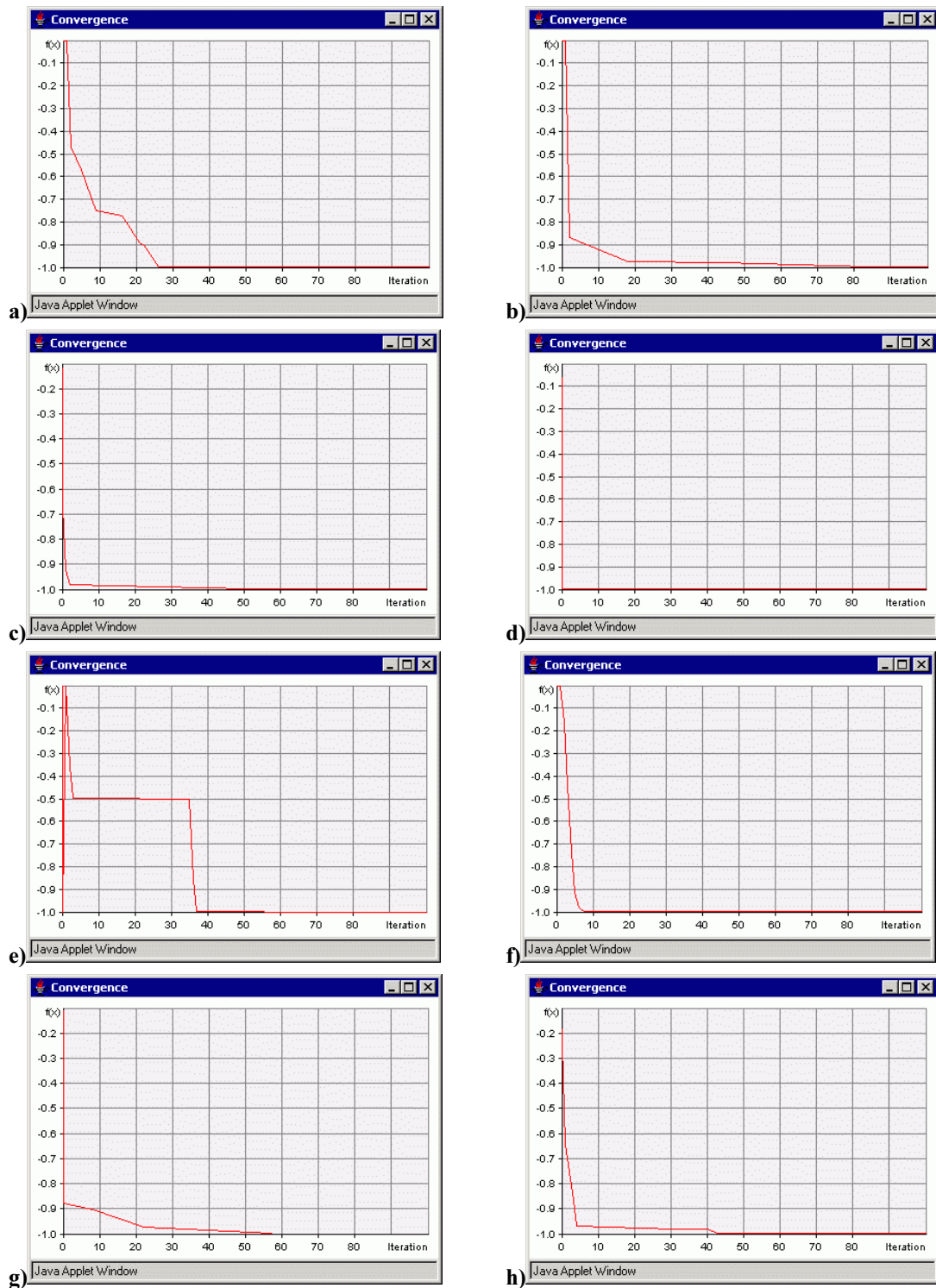
- GA genetinis algoritmas
- GAopt genetinis algoritmas su kryžminimo strategija
- MC Monte Karlo paieška
- Bayes Bajeso metodas
- Exkor Bajeso metodo modifikacija su kryptinga paieška
- LBayes Bajeso metodo modifikacija su žingsnio dydžio kontrole
- Globt Klasterizacijos tipo metodas
- Unt Ekstrapoliacijos tipo metodas



1 pav. Optimizavimo metodų konvergavimas sprendžiant Dvikovos uždavinį. a) genetinis algoritmas, b) genetinis algoritmas su kryžminimo strategija, c) Monte Karlo paieška, d) Bajeso metodas.

1 Mateminių funkcijų optimizavimas (sinuso funkcija)

Optimizuojant paprastas matemines funkcijas Bajeso metodas dėl savo specifikos, tai atlieka labai greitai. Į pateiktas konvergavimo kreives reiktų žvelgti atsargiai, nes tai tėra vieno bandymo rezultatas, o ne vidutinė reikšmė, kuri kur kas tiksliau atspindėtų rezultatus. Vertinant bendrai galima pastebėti (1 pav., 2 pav., 3 pav.) kad genetinių algoritmų konvergavimo kreivė yra iškilesnė. Ta prasme, genetiniai algoritmai konverguoja tiesiškiau, nei kiti metodai. Esant mažam iteracijų skaičiui konvergavimo greičiu genetinius algoritmus lenkia kone visi metodai, bet genetiniai algoritmai kur kas lėčiau praranda konvergavimo greitį didėjant iteracijų kiekiui.



1 pav. Optimizavimo metodų konvergavimas sprendžiant paprastą matematinę funkciją (sinuso funkciją). a) genetinis algoritmas, b) genetinis algoritmas su kryžminimo strategija, c) Monte Karlo paieška, d) Bajeso metodas, e) "Exkor", f) "LBayes", g) "Globt", h) "Unt".

1 Apibendrinimas

Tyrimas parodė genetinių algoritmų universalumą sprendžiant įvairias globalaus optimizavimo užduotis. Jo rezultatų padėtis nedaug svyruoja lyginant su Monte Karlo paieškos ir Bajeso metodais, kai tuo tarpu kiti nagrinėti metodai aiškiai išsiskyrė tik tam tikro tipo uždaviniuose. Įvertinant šiuos rezultatus galima būtų manyti, kad optimizuojant ir kitokius uždavinius rezultatai būtų panašūs. Žinoma, optimizuojant iškiliasias funkcijas su vienu minimumo tašku genetiniai algoritmai niekada nepralenks tokių metodų kaip Bajeso. Bet genetiniai algoritmai be jokių problemų optimizuoja funkcijas turinčias daug lokalių minimumų. Yra galimybė taip suoptimizuoti genetinio algoritmo parametrus, kad jis neištrigtų lokaliame minimume.

Vertinant genetinio algoritmo konvergavimą sprendžiant optimizavimo uždavinius galima pastebėti, kad jo konvergavimo kreivė yra tiesiškesnė. Ta prasme, genetinis algoritmas aplenkia kitus optimizavimo metodus prie didesnio iteracijų skaičiaus. Šios sanaudos, kurių reikia norint gauti geresnius rezultatus nei kitais optimizavimo metodais ateityje turėtų tapti priimtinomis, nes didės kompiuterių skaičiavimo sparta.

Tolimesniuose planuose numatoma iširti genetinio algoritmo parametrų optimizavimo galimybes. Optimizuotus genetinius algoritmus bus galima pritaikyti sunkiai modeliuojamiems uždaviniams spręsti, tokiems kurių apskaičiavimui reikia daug kompiuterinio laiko sąnaudų.

Literatūros sąrašas

- [1.] **Fernando G. Lobo, David E. Goldberg.** The Parameter-less Genetic Algorithm in Practice. *IlliGAL Report No. 2001022, Illinois Genetic Algorithms Laboratory*, 2001 June.
- [2.] **M.Pelikan, D.E.Goldberg & E.Cantu-Paz.** BOA: the bayesian optimization algorithm. *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, CA: Morgan Kaufmann, 1999, pp. 525-532.
- [3.] **J.Mockus.** A Set of Examples of Global and Discrete Optimization. Available at: <http://soften.ktu.lt/~mockus>.
- [4.] **Alan C.Schultz.** The Genetic Algorithms Archive. *The Navy Center for Applied Research in Artificial Intelligence*. Available at: <http://www.aic.nrl.navy.mil/galist>.
- [5.] **Web page of the International Society for Artificial Life (ISAL).** Available at: <http://www.alife.org>
- [6.] **Marek Obitko (1998).** Introduction to Genetic Algorithms. *University of Applied Sciences*. Available at: <http://cs.felk.cvut.cz/~xobitko/ga>.

Applying Genetic Algorithms in Global Optimization Tasks solving

Genetic Algorithms are compared to the other global optimization methods in the base of well-known optimization tasks. The tasks are such as Knapsack, School schedule optimization and Duel. Genetic Algorithms are compared to such base methods as Monte Carlo and Bayes, also few Bayes method modifications, like Exkor – Bayesian coordinate line search, LBayes – the stochastic approximation method with the Bayesian step size control, and few other methods: Unt – the extrapolation type method, Globt – the clustering method.

Parameters of genetic algorithms were selected intuitively considering suggestions proposed in literature. And through all testing process parameters stay unaltered. There were two modifications of genetic algorithms taken into testing. First is simplest, with no strategy. And the second has purposeful crossover strategy.

The experiment shows that even the simplest genetic algorithm with no special strategic modification keep in step with the pending global optimization algorithms and in some cases even outrun them.