VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEEDRA

Vaidas Juzonis

SPECIALIZUOTO MODELIAVIMO ĮRANKIO, PAREMTO GENETINIAIS ALGORITMAIS, KŪRIMAS

DEVELOPMENT OF SPECIALIZED SIMULATION TOOL BASED ON GENETIC ALGORITHMS

Baigiamasis magistro darbas

Inžinerinės informatikos studijų programa, valstybinis kodas 62409P111

Informacinių sistemų specializacija

Informatikos studijų kryptis

Vilnius, 2011

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIU MOKSLU FAKULTETAS
INFORMACINIU SISTEMU KATEDRA

TVIRTINU
Katedros vedėjas

_____
(Parašas)

Dalius Mažeika
(Vardas, pavardė)

_____
(Data)

Vaidas Juzonis

SPECIALIZUOTO MODELIAVIMO ĮRANKIO, PAREMTO GENETINIAIS ALGORITMAIS, KŪRIMAS

DEVELOPMENT OF SPECIALIZED SIMULATION TOOL BASED ON GENETIC ALGORITHMS

Baigiamasis magistro darbas

Inžinerinės informatikos studijų programa, valstybinis kodas 62409P111

Informacinių sistemų specializacija

Informatikos studijų kryptis

Vadovas prof. habil. dr. Antanas Čenys _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Konsultantas dr. Nikolaj Goranin _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Konsultantas doc. dr.Angelė Kaulakienė _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Vilnius, 2011

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

Informatikos studijų kryptis
Inžinerinės informatikos studijų programa, valstybinis kodas 62409P111
Informacinių sistemų specializacija

_____
(Parašas)

Dalius Mažeika
(Vardas, pavardė)

_____
(Data)

BAIGIAMOJO MAGISTRO DARBO

UŽDUOTIS

…….....Nr. .....
Vilnius

Studentui (ei) Vaidui Juzoniui
(Vardas, pavardė)

Baigiamojo darbo tema: Specializuoto modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas

patvirtinta 201…m. ……..…… d. dekano potvarkiu Nr. …….

Baigiamojo darbo užbaigimo terminas 201…m. ……..…… d.

BAIGIAMOJO DARBO UŽDUOTIS:
Sukurti įrankį, leidžiantį realizuoti esybių iš informacijos saugos srities evoliucijos modeliavimą naudojant genetinius algoritmus.

Baigiamojo darbo rengimo konsultantai:
dr. Nikolaj Goranin
(Moksl. laipsnis, vardas, pavard.)

Vadovas ..... prof. habil. dr. Antanas Čenys
(Parašas) (Moksl. laipsnis, vardas, pavardė)

Užduotį gavau

……..
(Parašas)
Vaidas Juzonis
(Vardas, pavardė)

……..……...
(Data)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Informatikos studijų kryptis
Inžinerinės informatikos studijų programa, valstybinis kodas 62409P111
Informacinių sistemų specializacija

PAŽYMA
APIE BAIGIAMĄJĮ MAGISTRO DARBĄ
…….....Nr. .....
Vilnius

Studentas (ė) Vaidas Juzonis
(Vardas, pavardė)

Studento (ės) studijų svertinis įvertinimų vidurkis.....balo.

Baigiamojo darbo tema: Specializuoto modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas

Baigiamasis darbas peržiūrėtas ir studentui (ei) Vaidui Juzoniui
leidžiama ginti šį baigiamąjį darbą magistro laipsnio suteikimo komisijoje.

Katedros vedėjas .....…..... doc. dr. Dalius Mažeika
(Parašas) (Moksl. laipsnis/pedag. vardas, vardas, pavardė)

VADOVO ATSILIEPIMAS
APIE BAIGIAMĄJĮ MAGISTRO DARBĄ

.....

.....

.....

.....

.....

.....

.....

.....

Vadovas .....……..... prof. habil. dr. Antanas Čenys
(Parašas) (Moksl. laipsnis/pedag. vardas, vardas, pavardė)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Vaidas Juzonis 20054500

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Inžinerinės informatikos studijų programa, INSFm-09

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO

SĄŽININGUMO DEKLARACIJA

2011 m. d.

(Data)

Patvirtinu, kad mano baigiamasis darbas (projektas) tema

Specializuoto modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas

patvirtintas 20 m. d. dekano potvarkiu Nr. , yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Parenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: dr. N. Goranin, doc. dr. A. Kaulakienė

Mano darbo (projekto) vadovas prof. habl. dr. Antanas Čenys

Kitų asmenų indėlio į parengtą baigiamąjį darbą (projektą) nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs(-usi).

Vaidas Juzonis

(Parašas) (Vardas ir pavardė)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

RECENZIJA
APIE BAIGIAMĄJĮ MAGISTRO DARBĄ

…….....Nr. .....
Vilnius

Studentas(ė) …… Vaidas Juzonis ……..….. parengė baigiamąjį magistro darbą tema:

Specializuoto modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas .....……

……

RECENZIJA

……...……

……

……

……

……

……

……

……...…..

……...

……..….

……..….

……..…….

……..…….

……..…….

Magistro laipsnio suteikimo komisijos narys

…… ……..
(Parašas) (Moksl. laipsnis/pedag. vardas, vardas, pavardė)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

RECENZIJA
APIE BAIGIAMĄJĮ MAGISTRO DARBĄ

…….....Nr. .....
Vilnius

Studentas(ė) …… Vaidas Juzonis ..…….….. parengė baigiamąjį magistro darbą tema:

Specializuoto modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas .....……

……

RECENZIJA

……...……

……

……

……

……

……

……

……..…..

……...

……..….

……..….

……..…….

……..…….

……..…….

Magistro laipsnio suteikimo komisijos narys

…… ……..
(Parašas) (Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz.sk. .....
Data .....-.....-.....

Inžinerinės informatikos studijų programos baigiamasis magistro darbas

Pavadinimas: **Specializuoto modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas**

Autorius: **Vaidas Juzonis** Vadovas: **prof. habil.dr. Antanas Čenys**

Kalba
× lietuvių
užsienio

Anotacija

Šiame darbe išanalizuoti genetinių algoritmų (GA) veikimo principai. Analizuojamos egzistuojančios modeliavimo aplinkos ir genetiniais algoritmais pagrįsti modeliavimo įrankiai. Kuriant modeliavimo įrankį, nustatyti funkciniai ir nefunkciniai reikalavimai. Realizuotas sukurtas įrankis ir atliktas pasirinktos esybės evoliucijos modeliavimas.

"16th International Conference on Information and Software Technologies" konferencijoje buvo pristatytas pranešimas “Genetic Algorithm Modeling Approach for Mobile Malware Evolution Forecasting”. Panaudojus jame pateiktus parametrų duomenis, atlikti bandymai su sukurtu modeliavimo įrankiu. Taip pat XIV jaunųjų mokslininkų konferencijoje „Mokslas - Lietuvos ateitis“ 2011, pristatytas straipsnis „Informacijos saugos dalykinės srities esybių evoliucijos modeliavimo įrankio, paremto genetiniais algoritmais, kūrimas “, šis straipsnis buvo parašytas remiantis šiuo darbu.

Darbą sudaro: 7 skyriai, 26 paveikslai, 7 lentelės, 2 priedai. Literatūros sąraše 52 šaltiniai.

Reikšminiai žodžiai:
evoliucija, genetiniai algoritmai, kenksmingas programinis kodas, modeliavimas, MATLAB, Jgap, virusai.

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN ISSN
No. of copies : 2
Date .....-.....-.....

Master‘s Thesis of **Engineering informatics** study program

Title: **Development of specialized simulation tool based on genetic algorithms**

Author: **Vaidas Juzonis** Academic supervisor prof. habil. dr. **Antanas Čenys**

Language
[×] Lithuanian
[ ] English

Annotation

This study analyzes operating principles of the genetic algorithms (GA), also submit proposals for the calculation of GA. Discuss the existing simulation environment and tools to implement GA. Towards a modeling tool to determine the functional and non-functional requirements. Marketed developed tool and to carry out tests for selected test of evolutionary analysis.

‘16th International Conference on Information and Software Technologies’ was presented the article ‘Genetic Algorithm Modeling Approach for Mobile Malware Evolution forecasting’ using the parameters details of this article perform the tests with simulation tool. Also XIV Conference of Young Scientists ‘Science - The future of Lithuania‘ 2011, was introduced the article ‘Development of the subject area of the information security beings evolutionary modeling tool based on genetic algorithms’. This article was written on the basis of this work.

Thesis consist of: 7 chapters, 26 pictures, 7 tables, 2 appendixes, 52 bibliographical entries.

Keywords:
evolution, genetic algorithm, Jgap, Malware, MATLAB, tools, virus,

Turinys

Turinys 10
LENTELIŲ SĄRAŠAS 11
PAVEIKSLŲ SĄRAŠAS 12
Santrumpų sąrašas 13
Įvadas 14
1. UŽDAVINIO FORMULAVIMAS 15
1.1. Tikslas ir uždaviniai 15
1.2. Darbo svarba 15
1.3. Temos naujumas 16
2. GENETINIAI ALGORITMAI 17
2.1. Kryžminimasis 20
2.2. Mutacija 21
2.3. Genų priklausomybė 22
3. GENETINIAIS ALGORITMAIS PAGRĮSTŲ MODELIAVIMO ĮRANKIŲ ANALIZĖ 23
3.1. MATLAB genetinių algoritmų ir tiesioginės paieškos paketas 23
3.2. Globalaus optimizavimo paketas „GMJ“ 25
3.3. Demonstracinis modeliavimo įrankis pagrįstas genetiniais algoritmais 27
3.4. JGAP paketas 29
4. INFORMACIJOS SAUGOS ESYBIŲ PERSPEKTYVIOS MODELIAVIMO KRYPTIES NUSTATYMAS 33
5. ĮRANKIO FUNKCINIAI IR NEFUNKCINAI REIKALAVIMAI 35
5.1. Funkciniai reikalavimai 35
5.2. Nefunkciniai reikalavimai 36
6. MODELIAVIMO ĮRANKIO, PAGRĮSTO GENETINIAIS ALGORIMAIS, REALIZAVIMAS 39
6.1. Įrankio testavimas 39
6.2. Įrankio aprašymas 41
6.2.1. Pagrindinis programos langas 41
6.2.2. Meniu juosta 41
6.2.3. Pagrindinių duomenų įvedimas 42
6.2.4. Duomenų failai 43
6.2.5. Iteracijų vykdymas 44
6.3. Informacijos saugos esybės evoliucijos modeliavimas 45
7. IŠVADOS IR PASIŪLYMAI 53
Literatūra 54
PRIEDAI 57
A priedas 57
B priedas 64

LENTELIŲ SĄRAŠAS

1 lentelė. Lango sąsajos testavimas 39
2 lentelė. Lango laukų testavimas 40
3 lentelė. Lango veikimo testavimas 40
4 lentelė. Įrankio funkcionalumo testavimas 40
5 lentelė. Chromosomos struktūra 45
6 lentelė. Genų parametrų reikšmės 46
7 lentelė. Atnaujintos genų parametrų reikšmės 50

PAVEIKSLŲ SĄRAŠAS

1 pav. GA pseudokodas [5] 18
2 pav. Klasikinio GA schema 19
3 pav. Kryžminimosi pavyzdys 20
4 pav. Kryžminimas trim būdais 20
5 pav. Mutacijos pavyzdys 21
6 pav. Plitimo pavyzdys 22
8 pav. Genetinių algoritmų paketo grafinė vartotojo sąsaja [11] 24
9 pav. Parametrizuoto genetinio algoritmo – optimizavimo metodo pasirinkimas 26
10 pav. Pradinis EGALT įrankio langas 28
11 pav. Funkcijų sąrašas 29
12 pav. Programos su genetiniais algoritmais veikimas 30
13 pav. JGAP svarbiausių klasių diagrama(V3.1) [10] 32
14 pav. Nefunkcinių reikalavimų tipai [46] 37
15 pav. Pagrindinis programos langas 41
16 pav. Įvedimo laukai 42
17 pav. Duomenų failas 43
18 pav. Geno reikšmių pavadinimų failas 43
19 pav. Duomenų failų įvedimo laukai 44
20 pav. Užpildyta genų parametrų lentelė 44
21 pav. Skaičiavimų paleidimo langas 45
22 pav. Geriausios plitimo strategijos kaita [45] 48
23 pav. Geriausios plitimo strategijos kaita, gauta paiūlytu įrankiu 48
24 pav. Išmaniųjų telefonų OS rinkos statistika 49
25 pav. Išmaniųjų telefonų rinkos statistika 49
26 pav. KPK plitimo strategijos kaita gauta su naujom parametrų reikšmėm 51

Santrumpų sąrašas

KPK – kenksmingas programinis kodas

DNS - (angl. Domain Name System) srities vardų struktūra

MMS - (angl. Multimedia Messaging Service) daugialypių pranešimų paslauga

GA – genetiniai algoritmai

IP – (angl. Internet Protocol) interneto protokolas

JGAP - Java genetinių algoritmų paketas

CSV - (angl. comma separated values) taškais atskirtos reikšmės

WPA – (angl. Wi-Fi Protected Access) apsaugotas priėjimas prie Wi-Fi

XML – išplėsta žymėjimo kalba (Extensible Markup Language)

PĮ – programinė įranga

Įvadas

Genetiniai algoritmai (toliau - GA) – tai kompiuterinių algoritmų šeima, kurių pagrindas evoliuciniai procesai [1]. Šiuo metu GA apima ir gimininges sąvokas, tokias kaip evoliucinės strategijos (toliau - ES), evoliucinis programavimas (toliau - EP), dirbtinė gyvybė (angl. artificial life, toliau - AL), klasifikatorių sistemos (angl. classifier systems, toliau - CS) ir evoliucionuojančių įrenginių koncepcija [2]. GA naudojami įvairioms užduotims spręsti – nuo keliaujančio pirklio problemos iki „Stels“ paviršių modeliavimo [3]. Vykdomi GA pritaikomumo tyrimai informacijos saugos sistemose: biometrinėse, kriptografinėse, skverbties detektavimo, atakų modeliavimo, kodo analizės ir t. t.

Kiekvienais metais didėja grėsmė prarasti svarbius duomenis ar gyvybiškai svarbių įrenginių valdymą. Didelę įtaką turi ir vis didesnis kompiuterizuotų sistemų paplitimas, aukštesnis kompiuterizacijos lygis, bendros tendencijos jungti visą kompiuterizuotą įrangą prie interneto. Vis labiau plinta specializuoti kompiuteriai, integruoti į buities prietaisus, mobiliuosius telefonus ir kitą įrangą, kur seniau buvo naudojamos paprastesnės elektroninės sistemos, kuriose kenksmingo programinio kodo veikimas buvo faktiškai neįmanomas. Taip pat didėjančią grėsmę įtakoja paties kenksmingo programinio kodo plėtra, panaudojimas vis naujesnių ir efektyvesnių metodų kenksmingo programinio kodo veikimui.

Todėl kuriant apsaugos infrastruktūrą, būtina nuspėti naujų kenkėjų evoliuciją, reikia turėti gerą kvalifikavimo techniką ir aptikti kenkėją, kol jis nepadarė žalos. Norint numatyti informacijos saugos srities esybių egzempliorių evoliucijos tikslesnį kelią, reikia turėti tam pritaikytą modeliavimo įrankį, kuris būtų tinkamas įvairesniems informacinės saugos užtikrinimo modeliams, tiek kenksmingo programinio kodo plitimo, tiek evoliucijos ar aptikimo modeliams.

Modeliuojant bandoma iš daugelio galimų faktorių išskirti pagrindinius, išryškinti esminius jų tarpusavio ryšius ir atmesti kitus faktorius ir ryšius, kurie konkrečiame kontekste yra neesminiai. GA pasirinkti, kaip modeliavimo metodas atsižvelgiant į jų gebėjimą atvaizduoti natūralios evoliucijos procesus, patogų užduoties parametrų nustatymą, efektyvų taikymą optimizavimo ir modeliavimo uždaviniams bei sėkmingus taikymus įvairiose informacinėse saugos sistemose.

1. UŽDAVINIO FORMULAVIMAS

1.1. Tikslas ir uždaviniai

- Darbo tikslas – sukurti įrankį, leidžiantį realizuoti esybių iš informacijos saugos srities evoliucijos modeliavimą naudojant genetinius algoritmus.

Atsižvelgiant į iškeltą tikslą, suformuluoti tokie uždaviniai:

- Išanalizuoti genetinių algoritmų veikimo principą ir jį aprašyti.
- Aptarti informacijos saugos srities esybių specifiką ir nustatyti perspektyvias modeliavimo kryptis.
- Išanalizuoti egzistuojančias modeliavimo aplinkas ir įrankius, realizuojančius genetinių algoritmų funkcionalumą.
- Suformuluoti įrankiui funkcinius ir nefunkcinius reikalavimus.
- Realizuoti įrankio prototipą ir atlikti bandymus pritaikant įrankį pasirinktos esybės evoliucijos modeliavimui.

1.2. Darbo svarba

Informacijos sauga pastaruoju metu tampa vis aktualesnė problema, galinti paliesti ne tik šalies elektroninių ryšių tinklus ar informacines sistemas, turinčias įtakos nacionaliniam saugumui, bet ir kiekvieną pilietį (pvz.: internetinės bankininkystės triktis, neprieinami gyventojų registro duomenys). Po vis dažnesnių kibernetinių atakų, kuomet įsilaužiama į valstybinių institucijų interneto svetaines, sutrikdomas jų darbas. Žiniasklaidoje nuolat diskutuojama apie tai, koks yra optimalus būdas užtikrinti valstybei svarbios elektroninės informacijos saugą. Dažniausiai nebūna sukurta bendra kritinės informacinės infrastruktūros saugos užtikrinimo sistema, todėl nėra aišku, kaip reikėtų valdyti kritines situacijas, kylančias dėl įsilaužėlių atakų ir techninių nesklandumų elektroninėje erdvėje. Nesuvaldžius kritinės situacijos, gali nukentėti ne tik valstybinės reikšmės informacija ir kiekvienas pilietis, bet ir šalies prestižas, pasitikėjimas naujomis technologijomis.

Informacijos saugos esybė – tai objektų klasės kurios susijusios su informacijos saugumu. Skiriamos informacijos saugą užtikrinančios ir skirtos piktaviliškom atakom prieš informacijos

saugą. Pastarosios esybės, tai kenksmingi programiniai kodai- internetiai kirminai, DoS atakos, trojos arkliai. Informacijos saugą užtikrinančios esybės – apsaugos programos, ugnesienės ir t.t.

Norint apsisaugoti nuo galimų naujų atakų, apsaugos sistemų kūrimo procesas tampa labai sudėtingas. Todėl kuriant apsaugos infrastruktūrą, būtina nuspėti informacijos saugos esybių evoliuciją, reikia turėti gerą kvalifikavimo techniką ir aptikti kenkėją, kol jis nepadarė žalos. Norint numatyti informacijos saugos srities esybių egzempliorių evoliucijos tikslesnį kelią, reikia turėti tam pritaikytą modeliavimo įrankį, kuris būtų tinkamas įvairesniems informacinės saugos užtikrinimo modeliams, tiek kenksmingo programinio kodo plitimo, tiek evoliucijos ar aptikimo modeliams.

1.3. Temos naujumas

Nespecializuotuose modeliavimo įrankiuose, kurie realizuoja genetinių algoritmų funkcionalumą, parametrai išreiškiami dvejetaine sistema, kas nėra intuityvu, taip pat nepatogu ir vartotojui. Egzistuojantys, genetiniais algoritmais paremti modeliai, realizuoti kaip atskiros programos, todėl negalima lengvai pritaikyti kitų esybių modeliavimui. Dauguma programų neleidžia įvesti norimos tinkamumo funkcijos, pasirinkti parametro kiekio ar nustatyti genų priklausomybės. Norint supaprastinti informacijos saugos srities esybių evoliucijos modeliavimą, įrankis turi būti patogus vartotojui, lankstus, galintis atvaizduoti skirtingų esybių specifiką. Modeliavimo įrankis būtų pritaikomas įvairesniems informacinės saugos užtikrinimo modeliams, tiek kenksmingo programinio kodo plitimo, tiek evoliucijos ar aptikimo modeliams.

2. GENETINIAI ALGORITMAI

GA – tai kompiuterinių algoritmų šeima, kurių pagrindas – evoliuciniai procesai [1]. Šiuo metu GA apima ir gimininges sąvokas, tokias kaip evoliucinės strategijos (ES), evoliucinis programavimas (EP), dirbtinė gyvybė (AL) (angl. artificial life), klasifikatorių sistemos (CS) (angl. classifier systems) ir evoliucionuojančių įrenginių koncepcija [2]. GA naudojami įvairioms užduotims spręsti – nuo keliaujančio pirklio problemos iki „*Stels*“ paviršių modeliavimo. Atliekami GA pritaikomumo tyrimai tokiose informacijos saugos sistemose:

- ✓ biometrinėse;
- ✓ kriptografinėse;
- ✓ skverbies detektavimo;
- ✓ atakų modeliavimo;
- ✓ kodo analizės ir t. t.

Genetinius algoritmus galima imituoti gamtos procesais, nes apima tokius tikslus kaip išlikimas, populiacijos didinimas, dauginimasis, mutacija. Paprastai specifikacijai atstovauja žinomas konfigūracijos kiekis, kur kiekvienas iš jų gamtoje veikia kaip vienas gyvas organizmas (chromosoma ar viena rūšis gamtoje). Jie vienijasi, kad sudarytų populiaciją. Gyventojai paprastai turi didelį kiekį, dažniausiai kintamą dydį struktūrų, kad palaikytų atranką. Ši struktūra susideda iš fundamentalių vienetų, vadinamųjų bitų, pozicija arba simboliškai vadinama genais, kurie struktūras daro skirtingas viena nuo kitos. Vienam problemos parametrui gali atstovauti vienas ar daugiau vienetų.

Algoritmas prasideda nuo pradinės sprendimų (individų) populiacijos sudarymo. Kiekvienas sprendimas yra išreiškiamas chromosomomis. Tada iš vienos populiacijos yra imami individai ir sukuriama nauja populiacija. Šio proceso motyvacija yra paremta tikėjimu, kad naujoji karta bus geresnė už senąją. Imant individus naujai populiacijai sudaryti, yra atsižvelgiama į sprendimo teisingumą, t.y. kuo tikslesnis sprendimas, tuo didesnė tikimybė, kad jis bus pasirinktas. Toks procesas yra kartojamas iki užbrėžto tikslo: kartų skaičius arba sprendimo tikslumas.

Genetinis algoritmas leidžia eksperimento parametrus konvertuoti į genus, kurie gali būti sistemingai pripažinti kompiuterio. Paprasti parametrai, kurių reikšmės gali būti tik „Taip“ arba „Ne“, genuose gali būti išreikšti 1 arba 0. Sudėtingesni parametrai, kurių reikšmės gali būti su raidėmis, išreiškiami dvejetaine sistema, arba gali turėti skirtingus išreiškimo metodus. Vienos kombinacijos parametrai formuoja genų rinkinį, kuris vadinamas chromosoma. Kiti skirtingi

parametrai formuoja kitas chromosomas, kurios bendrai sudaro populiaciją. Rezultatų kokybę sąlygoja chromosomos sugebėjimai, tai vadinama tinkamumu [6].

Galima pateikti genetinių algoritmų ir inžinerinių sąvokų palyginimus:

GA terminas	Inžinerijos terminas
✓ Aleninis genas	Parametro reikšmė
✓ Chromosoma	Parametro vektorius
✓ Evoliucija	Konvergencija
✓ Genas	Parametras
✓ Mutacija	Triukšmas
✓ Populiacija	Bandymų masyvas
✓ Kryžminimasis	Atsinaujinimas

Kryžminimas ir mutacija yra pagrindiniai genetinio algoritmo operatoriai, lemiantys problemų sprendimų raidą. Jų tipų pasirinkimas ir realizavimas priklauso nuo problemos bei naudojamų chromosomų kodavimo.

```
sugeneruoti pradinę populiaciją;
      while (nepildoma ciklo nutraukimo sąlyga) do
            pasirinkti tėvus iš populiacijos;
            atlikti crossover operaciją;
            įvesti mutaciją;
            suformuoti naują populiaciją;
      end while;
end
```

1 pav. GA pseudokodas [5]

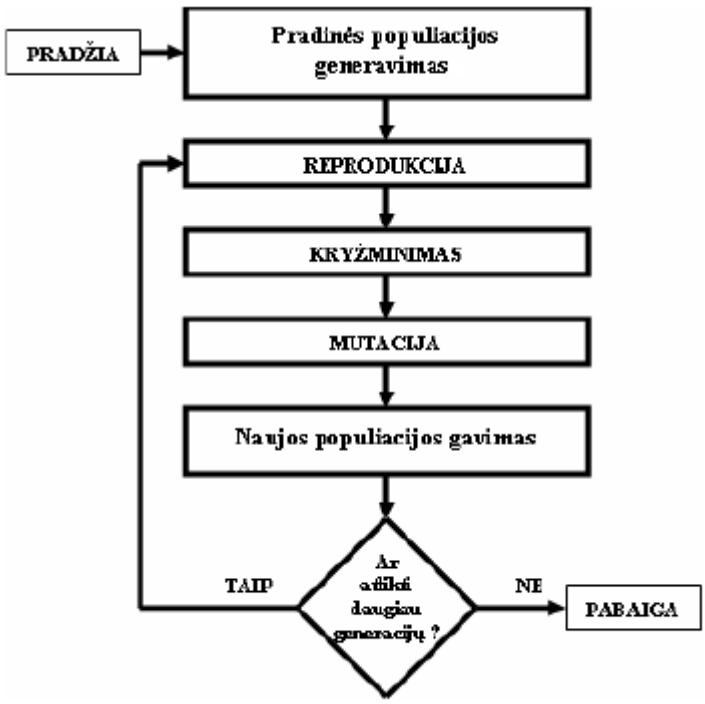
Algoritmo eskizas:

1. Sugeneruoti n chromosomų, kurios reprezentuoja problemų sprendimus ir sudarys pradinę populiaciją.
2. Apskaičiuoti kiekvienos chromosomos (sprendimo) x tinkamumą f(x). T.y. randame taško x paieškos erdvėje reikšmę.
3. Sukurti naują populiaciją kartojant šiuos etapus:
    - Pasirinkti dvi chromosomas x, y iš populiacijos atsižvelgiant į jų tinkamumą.

- Su tikimybe P_c sukryžminti x ir y chromosomas, kad galima būtų gauti naujus individus. Jei kryžminimas neįvyko, tada jie yra tikslios tėvų kopijos.
- Su tikimybe P_m mutuoti kiekvieną chromosoma kiekviename gene (pozicija chromosomoje).
- Patalpinti naujas chromosomas į populiaciją.

4. Kartoti algoritmą nuo 2 punkto tol, kol yra patenkintos tam tikros sąlygos.
5. Grąžinti geriausią sprendimą.

Šį GA eskizą galima pavaizduoti taip (2 pav.)



2 pav. Klasikinio GA schema

Šis algoritmas atrodo paprastai, tačiau yra daugybė parametrų ir būdų jam realizuoti. Pirmiausia iškyla klausimas, kaip reprezentuoti sprendimus chromosomomis, ir kokiu būdu jas užkoduoti. Tada reikia apibrėžti pagrindinius genetinio algoritmo operatorius: kryžminimą ir mutaciją.

2.1. Kryžminimasis

Kryžminimasis, arba kitaip rekombinacija, yra būdas genetiniuose algoritmuose perteikti vienos chromosomos ar kelių chromosomų sandarą kitai kartai. Ji yra biologinės rekombinacijos analogas, kadangi būtent biologiniu procesu ir paremta [42].

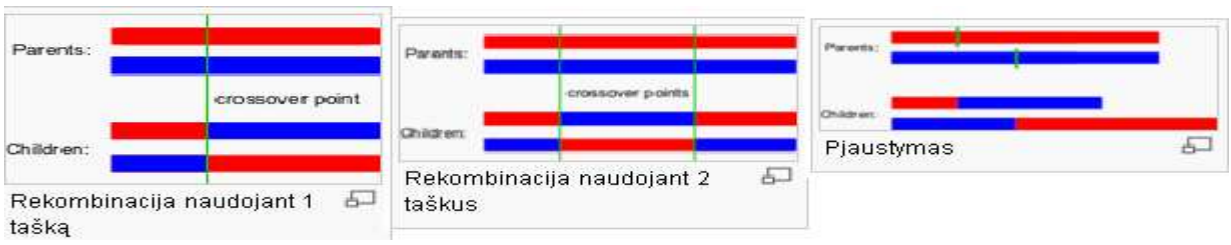
Yra nemažai kryžminimosi būdų, kurie skirtingai perteikia biologinės rekombinacijos esmę:



3 pav. Kryžminimosi pavyzdys

Kryžminimo metu du atrinkti individai keičiasi bitais. Sakykime, turime tokius individus:
11111111
00000000,
ir žinome, kad kryžminimosi pozicija yra 3 nuo dešinės:
11111^111
00000^000,
tuomet turėsime du naujus individus:
11111000
00000111.

Vieno taško kryžminime yra pasirenkamas vienas taškas, tada naujai chromosomai sudaryti dalis iki to taško yra paimama iš vieno tėvo, o likusi dalis iš kito (3 pav.). Lygiai taip pat veikia ir dviejų taškų kryžminimas, tik čia figūruoja trys dalys – pirma ir paskutinė iš vieno tėvo, o likusi vidurinė iš kito. Pastovus kryžminimas primena vieno ir dviejų taškų, tik čia šių taškų yra daug, t.y. kiekvienas genas yra atsitiktinai imamas (dažniausiai su tikimybe 0,5) arba iš vieno tėvo, arba iš kito. Be šių operatorių chromosomoms, koduojamoms bitų metodu (ir ne tik), gali būti taikomi ir kitokie būdai, pvz., standartinė AND operacija [6].



4 pav. Kryžminimas trim būdais

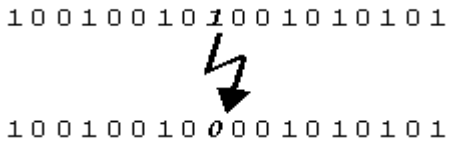
Koduojant perstatas reikia atsižvelgti į tai, kad dažnai skaičius chromosomoje gali kartotis tik viena kartą. Tokiais atvejais vieno taško kryžminime yra kopijuojama pirmoji dalis iš vieno tėvo, o kita dalis papildoma trūkstamais elementais nuskaitant antrąjį. Tačiau egzistuoja ir kitokių būdų antrai daliai užpildyti.

Taip pat yra ir kitokie kryžminimosi būdai (4 pav.):

- ✓ Su dviem rekombinacijos taškais. Pasirenkami du rekombinacijos taškai iš tėvų duomenų. Vėliau visi tėvų duomenys, esantys tarp šių taškų sukeičiami vietomis. Rezultatas –vaikai, kurie turi dalį vieno ir kito tėvo duomenų.
- ✓ Pjaustymas – rekombinavimo technika, kai keičiamas vaikų duomenų ilgis. Skirtumas nuo ankščiau minėtų technikų yra tas, kad čia iš tėvų duomenų pasirenkami taškai skirtingose vietose.
- ✓ Vienoda ir pusiau vienoda rekombinacija. Abiejų technikų atvejais tėvų duomenų pagrindu sukuriami du nauji vaikai. Pirmuoju atveju tėvų duomenys sulyginami tarpusavyje ir sukeičiami vietomis su 50 proc. tikimybe. Pusiau vienodos rekombinacijos atveju sukeičiama vietomis pusė nesutampančių tėvų duomenų.

2.2. Mutacija

Mutacijos reikšmė yra apsaugoti sprendimų populiaciją nuo lokalaus konvergavimo. Mutacijos yra atsitiktinės arba su labai maža tikimybe kiekvienam individui.



5 pav. Mutacijos pavyzdys

Mutacijos operatorius yra gana paprastai realizuojamas. Bitų kodavime kiekvienas bitas su pasirinkta tikimybe yra keičiamas į priešingą, t. y. iš „0“ į „1“ arba iš „1“ į „0“. Jeigu turime individą 11111111 ir žinome, kad ketvirtas bitas iš dešinės mutuoja 11111111, tai turėsime individą 11110111. Tiesioginiame metode jis priklauso nuo naudojamų reikšmių, pvz., naudojant realius skaičius, mutacija gali būti tiesiog atsitiktinio skaičiaus pridėjimas ar atėmimas. Perstatų metode gali būti tiesiog sukeičiami du genai vietomis, o medžio kodavime yra keičiami mazgai.

Nors mutacijos operatoriai atlieka tik paprastas funkcijas, tačiau jie gali būti pakeisti truputį sudėtingesniais. Pavyzdžiui, keliaujančio pirklio uždavinyje naudojant perstatų metodą būtų netikslinga atlikti mutaciją, jei galima lengvai patikrinti ar sprendimas turės geresnį tinkamumą [6].

2.3. Genų priklausomybė

Genetinius algoritmus galima papildyti nauju funkcionalumu, tokiu kaip genų priklausomybė. Genu priklausomybė naudinga, kai yra tam tikros genų grupės, kuriose genai vienas nuo kito priklauso, dažniausiai vienas iš genų būna aktyvavimo genas, kuris sužadina kitų genų veikimą individe. Tačiau, jei evoliucijos metu individas netenka geno iš priklausančiųjų grupės, kiti genai, priklausantys tai pačiai grupei, tampa neaktyvūs, nes vienas be kito funkcionuoti negali.

Šis naujas genetinių algoritmų funkcionalumas atsispindi ir gamtoje, iš kurios ir yra pasisemta genetinių algoritmų idėja. Kenksmingo programinio kodo evoliucijos atveju galima pateikt tokį pavyzdį: kenksmingas programinis kodas gali plisti per kelias operacines sistemas, Windows OS ir Linux OS. Plintant per šias operacines sistemas naudojamos skirtingos įsilaužimo programos, ir jei evoliucijos metu individas nepaveldi jokios įsilaužimo programos, tai jo plitimas per tam tikra operacinę sistemą neįmanomas.

Geno pavadinimas	Linux	Exp	NOKIA
Reikšmė	1	0	1

6 pav. Plitimo pavyzdys

Linux- operacinė sistema, kurioje galimas kenkėjiško kodo plitimas, "Exp"- naudojama įsilaužimo programa, NOKIA- telefono modelis, kuriam gali būti pritaikyta įsilaužimo programa.

Matome, kad šio individo egzistavimas yra negalimas, kadangi evoliucijos metu "Exp" genas neteko reikšmės.

3. GENETINIAIS ALGORITMAIS PAGRĮSTŲ MODELIAVIMO ĮRANKIŲ ANALIZĖ

Šiame skyriuje analizuojami GA pagrįsti modeliavimo įrankiai. Aprašomos modeliavimo įrankių funkcinės galimybės, parametrų įvedimo ar pasirinkimo variantai. Taip pat išanalizuojama Java programavimo kalbos biblioteka JGAP, jos atsiradimo idėjos principai, išanalizuojamos pagrindinės JGAP klasės, jų konfigūravimo objektai.

3.1. MATLAB genetinių algoritmų ir tiesioginės paieškos paketas

MATLAB yra galingas matematinis programinės įrangos paketas. MATLAB paketas pasižymi puikiu grafikos ir matricų apdorojimu. Esminės MATLAB ypatybės yra jo grafinės funkcijos ir vartotojui skirti įrankiai. Taip pat labai svarbi MATLAB ypatybė yra gebėjimas sujungti MATLAB taikomąsias programas su programomis, parašytomis C, C++, Fortan ar Java programavimo kalbomis. MATLAB palaiko duomenų importavimo galimybę iš išorinių įrenginių bei failų. Todėl į modeliavimui skirtą aplinką galima lengvai įsikelti reikiamus garsinius failus. Dauguma funkcijų MATLAB yra orientuotos į matricas, todėl jos gali veikti su įvairiais tinkamo dydžio duomenų masyvais.

Genetinio algoritmo ir tiesioginės paieškos paketas išplečia MATLAB įrankių, naudojančių genetinius ir tiesioginės paieškos algoritmus, optimizavimo galimybęs. Genetinio algoritmo ir tiesioginės paieškos paketas gali papildyti kitus optimizavimo metodus, padėdamas surasti gerus pradžios taškus. Tuomet toliau galima naudoti tradicinius algoritmus, norint nustatyti tikslų sprendinį.

Paketo funkcijos yra pasiekiamos per MATLAB komandinę eilutę. Visos funkcijos yra parašytos atvira MATLAB kalba. Tai reiškia, kad kiekvienas gali patikrinti naudojamus algoritmus, juos modifikuoti ir sukurti sau reikalingas funkcijas.

MATLAB genetinio algoritmo ir tiesioginės paieškos paketas pasižymi [49]**:**

- Grafinės sąsajos ir komandinės eilutės funkcijomis, greitam problemų nurodymui, algoritmų parametrų nustatymui, skaičiavimų stebėjimui.
- Genetinio algoritmo įrankiais su daugeliu parametrų kūrimu, selekcija, kryžminimu, mutacija.
- Funkcijos integracija į MATLAB ir optimizavimo paketą. Automatinio MATLAB kodo generavimo palaikymas.

Paketas pateikia paruoštus šablonus, kurie yra skirti optimizavimo uždaviniams spręsti panaudojant genetinius algoritmus ir (arba) tiesioginės paieškos metodus. Visos šio paketo

funkcijos yra surašytos į Matlab failus, kurie realizuoja optimizavimo metodus. Šios funkcijos laisvai peržiūrimos ir redaguojamos.

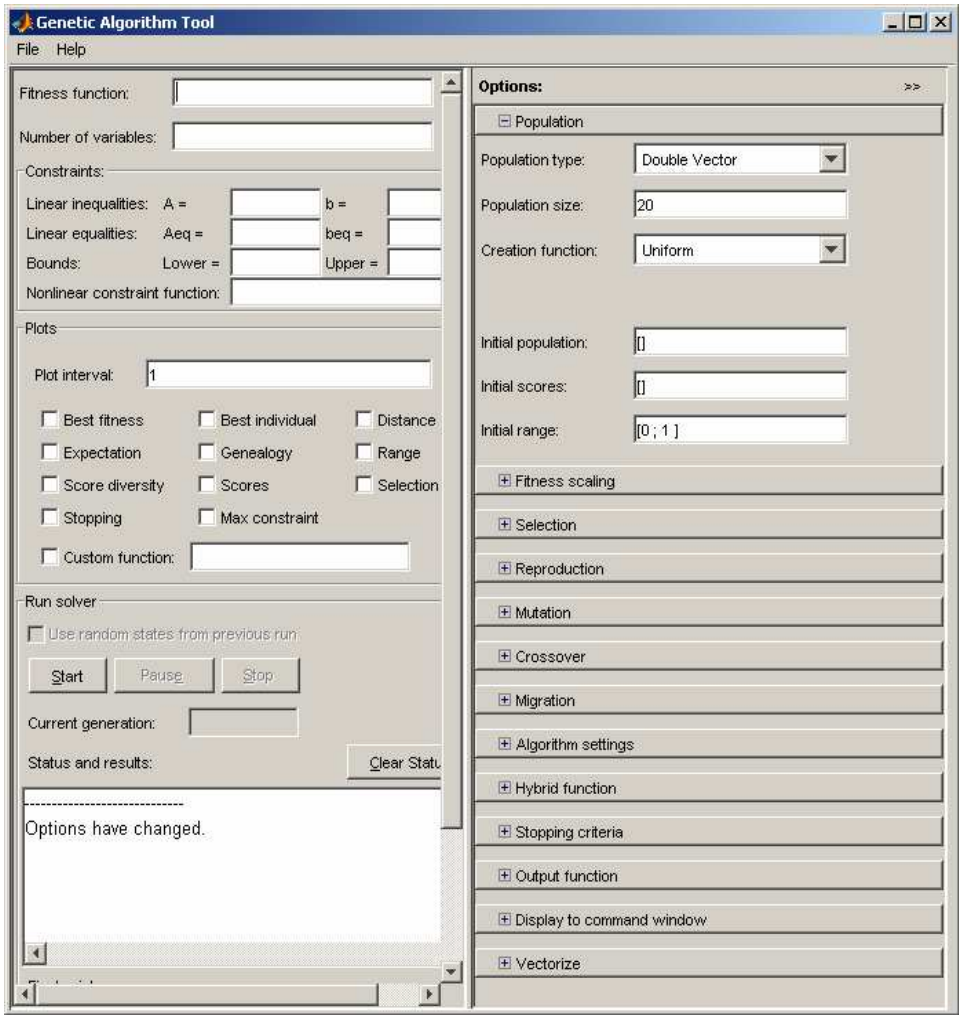
Genetinio algoritmo funkcija `ga` yra iškviečiama užrašant komandinėje eilutėje komandą: [x fval] = ga(@fitnessfun, nvars, options), kur [33]

- @fitnessfun – fitneso funkcijos nuoroda;
- nvars – nepriklausomų kintamųjų skaičius;
- options – struktūra, turinti informaciją apie genetinį algoritmą. Jei kviečiant funkciją nenurodomas šis argumentas, `ga` panaudos pirminius numatytus nuostačius.

Šios funkcijos rezultatai:

- x – taškas, kuriame buvo pasiekta paskutinė reikšmė;
- fval – tinkamumo funkcijos paskutinė vertė.

Genetiniai algoritmai gali buti taikomi optimizavimo uždaviniams spręsti panaudojus grafinę sąsają, kuri yra iškviečiama panaudojus `gatool` funkciją.



7 pav. Genetinių algoritmų paketo grafinė vartotojo sąsaja [11]

Šiai grafinei aplinkai reikia nurodyti tinkamumo funkciją, kurią norima minimizuoti. Tinkamumo funkcija yra užrašoma į MATLAB failą, į kurį yra kreipiamasi per simboli @. Nurodomas įėjimo kintamųjų skaičius.

Apibendrinant MATLAB genetinių algoritmų ir tiesioginės paieškos paketą nustatomi šie privalumai ir trūkumai:

Privalumai:

- Daugybės parametrų pasirinkimas:
  - Mutacijos
  - Kryžminimo.
  - Duomenų pateikimo.
  - Skaičiavimų užbaigimo.
- Rezultatų grafinis atvaizdavimas

Trūkumai:

- Nėra genų priklausomybės.
- Genų reikšmės nustatomos pasirinkus intervalą.
- Genas gali turėti tik po vieną parametrą.
- MATLAB paketo kaina.

3.2. Globalaus optimizavimo paketas „GMJ“

Profesorius J.Mockus globalaus optimizavimo paketą pradėjo plėtoti maždaug 2002 m. Kiekvienais metais jis papildomas vis naujomis priemonėmis. Realizuojami nauji optimizavimo metodai.

Įtraukiami nauji uždaviniai optimizavimui, pridedama papildomų rezultatų analizės galimybių.

GMJ paketą sudaro pagrindiniai optimizavimo metodai [36]:

- Parametrizuotas genetinis algoritmas:
  - paprastas;
  - su kryžminimo strategija.

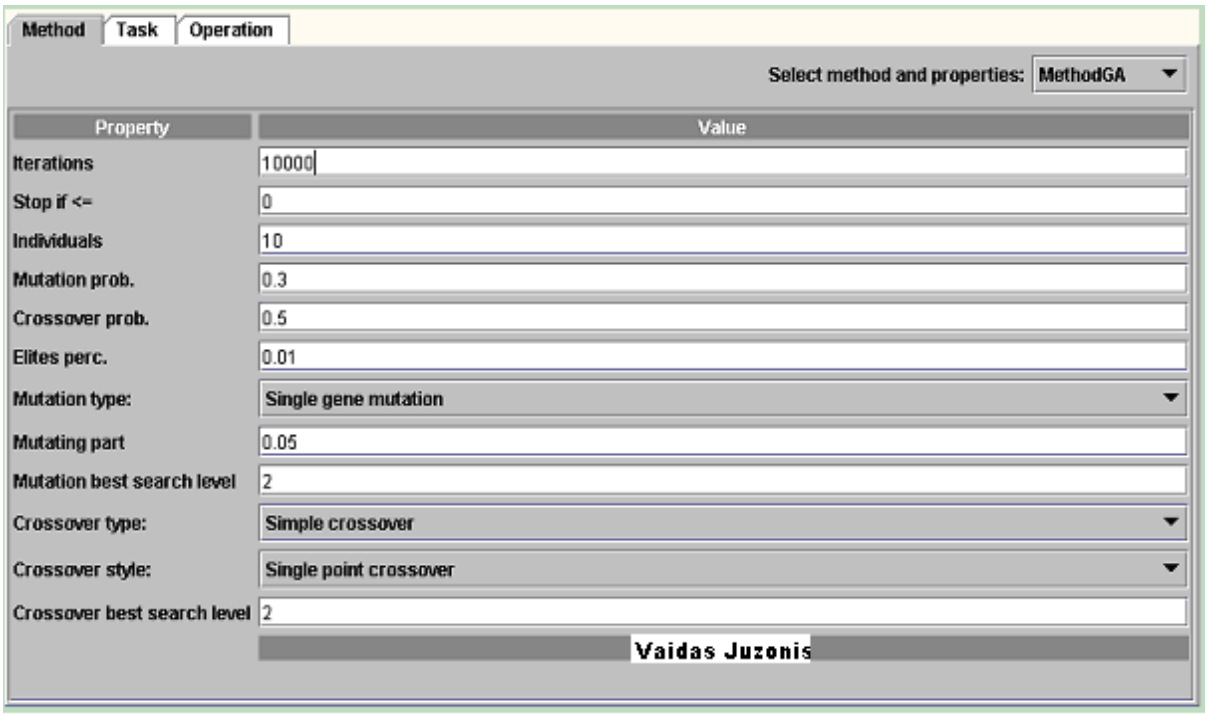
- Beparametris genetinis algoritmas :
  - paprastas;
  - su kryžminimo strategija.

- Neuroninio tinklo optimizavimo uždavinys:
  - vienasluoksnis;

- dvisluoksnis;

o su grįžtamaisiais ryšiais.

- Bajeso.
- Monte Karlo paieškos.
- Exkor Bajeso metodo modifikacija su kryptinga paieška.
- LBayes Bajeso metodo modifikacija su žingsnio dydžio kontrole.
- Globt klasterizacijos tipo metodas.
- Unt ekstrapoliacijos tipo metodas.



8 pav. Parametrizuoto genetinio algoritmo – optimizavimo metodo pasirinkimas

ga.jar-genetinio algoritmo(GA) modeliavimo paketas.
methodGA.jar - GA kaip optimizavimo metodas GMJ pakete. "Method" dalyje pasirinkti "MethodGA" metodą.
KnapGen2.jar - GA paketas sprendžiantis Knapsack uždavinį. Į GMJ paketą jis įtrauktas kaip užduotis "Task" dalyje pavadinimu "KnapGen2".

GA kaip metodas ir kaip užduotis turi tuos pačius parametrus:

- Iteracijų skaičius.
- Nurodoma kiek GA iteracijų atlikti.
- Individų skaičius.
- Individų skaičius vienoje iteracijoje.

- Mutacijos tikimybė.
- Su tokia tikimybe bus atliekama mutacija.
- Kryžminimo tikimybė.
- Su tokia tikimybe bus atliekamas kryžminimas.
- Geriausi individai.

Nurodytas procentas (0-1) geriausių individų bus perkeliamą į naują iteraciją be jokių pokyčių. Skaičiuojant perkeliamų individų skaičių apvalinama į viršų, t.y. net ir 0,0...01 iššauks vieno individo perkėlimą.

Mutacijos tipas

Vieno geno mutacija, mutuoja tik vienas individo genas. Visi kiti mutacijos parametrai neturi įtakos. Paprasta mutacija. Mutuoja nurodyta individo genų dalis. Kiek individų genų turi mutuoti nusako esantis parametras *Mutating part* (*Mutuojanti dalis).* Mutacija ieškant geriausio varianto. Atliekama paprasta mutacija, bet į rezultatus patenka tik geriausias individas.

Kryžminimo tipas

Paprastas kryžminimas. Atliekamas paprastas individų kryžminimas.

Kryžminimas ieškant geriausio varianto. Atliekama paprastas kryžminimas, bet į rezultatus patenka tik geriausi individai. Parametru *Kryžminimo geriausio paieškos lygis (*angl. *Crossover best search level)* galima nurodyti kiek kartų bandyti atrasti geriausią.

Kryžminimo stilius

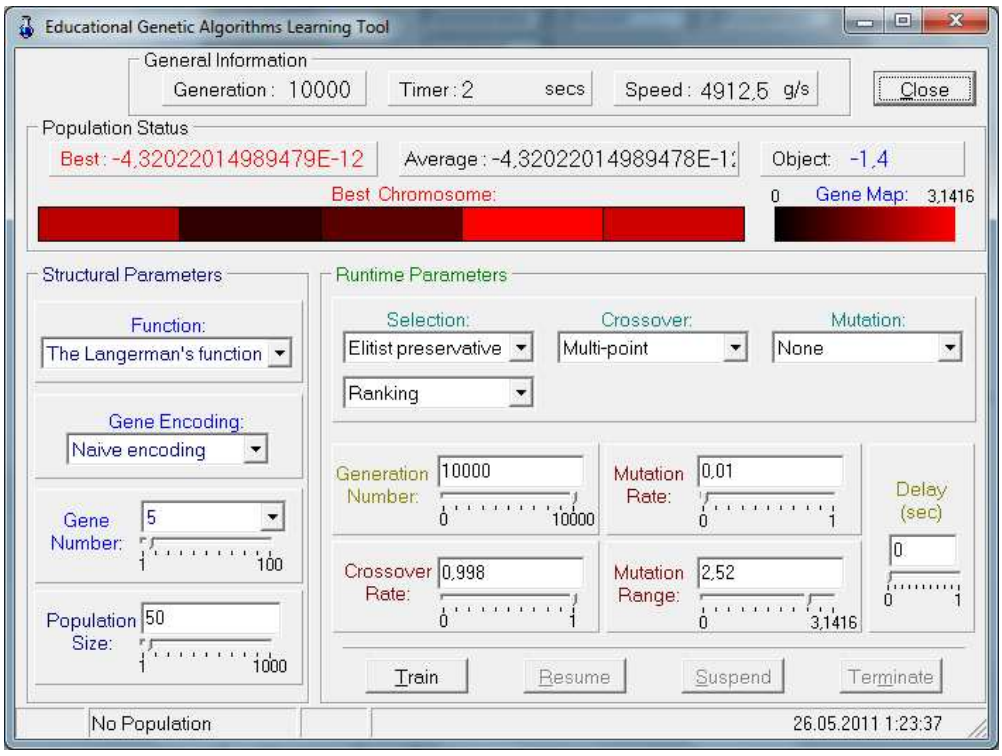
Kryžminimas per vieną tašką.

Kryžminimas per du taškus.

Kryžminimas atsitiktinai. Kryžminant, genus išrenkant atsitiktinai.

3.3. Demonstracinis modeliavimo įrankis pagrįstas genetiniais algoritmais

Per pastarąjį dešimtmetį ėmė sparčiai didėti susidomėjimas genetiniais algoritmais, tai pastebima iš padidėjusio kiekio mokslinių straipsnių, seminarų, konferencijų skaičiaus, kuriuose pristatomi darbai susiję su genetiniais algoritmais. Kai kuriuose mokymų moduliuose pradėta dėstyti apie genetinius algoritmus. Tam tikslui buvo sukurtas mokymo įrankis su genetiniais algoritmais, kuris pavadintas `EGALT` (angl. educational genetic algorithm learning tool). `EGALT` įrankio autoriai Ying-Hong Liao ir Chuen-Tsai Sun. Naudojant `EGALT` įrankį, studentai gali lengviau ir greičiau susipažinti su genetinių algoritmų veikimu. Patogi vartotojo sąsaja buvo sukurta siekiant padėti studentams keisti genetinių algoritmų struktūrą ir parametrus.



9 pav. Pradinis EGALT įrankio langas

EGALT įrankis parašytas naudojant C++ programavimo kalbą ir skirtas Windows operacinei sistemai. Įrankį galima parsisiųsti iš www.ewh.ieee.org/soc/es/May2001/14/Begin.htm. 9 pav. pavaizduotas pagrindinis įrankio langas. Viršutinėje lango dalyje yra šeši komponentai:

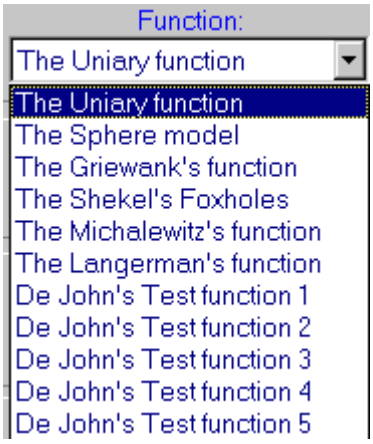
- *Generation (generacija)*: generacijų skaičius, kuris keičias po kiekvienos atliktos generacijos.
- *Timer (laikas):* vykstančio skaičiavimo laikas.
- *Speed (sparta):* parodo kiek generacijų įvykdoma per vieną sekundę.
- *Best (geriausias):* didžiausia funkcijos reikšmė.
- *Average (vidutinis):* vidutinė funkcijos reikšmė per praėjusias generacijas.
- *Best chromosome (geriausia chromosoma):* chromosomos geno reikšmė susieta su spalva. Vartotojas paspaudęs ties geno gali pažiūrėti jo tikrą reikšmę.
- *Gene map (geno priskirimas):* Funkcijos reikšmę susieja su spalva. Kuo spalva raudonesnė, tuo reikšmė didesnė.

Įrankyje galima pasirinkti testuojamą funkciją, genų skaičių, populiacijos dydį, iteracijų dydį, taip pat mutacijos, kryžminimo tikimybes. Įrankyje pateikta dešimt tinkamumo funkcijų.

Funkcijos, kurios pavadinimas "Griewank's" formulė yra [31]:

$$f_2(\bar{x}) = \frac{1}{4000}\sum_{i=1}^{N}(x_i - 100)^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (1)$$

su $x_i \in$ [-600,600].



10 pav. Funkcijų sąrašas

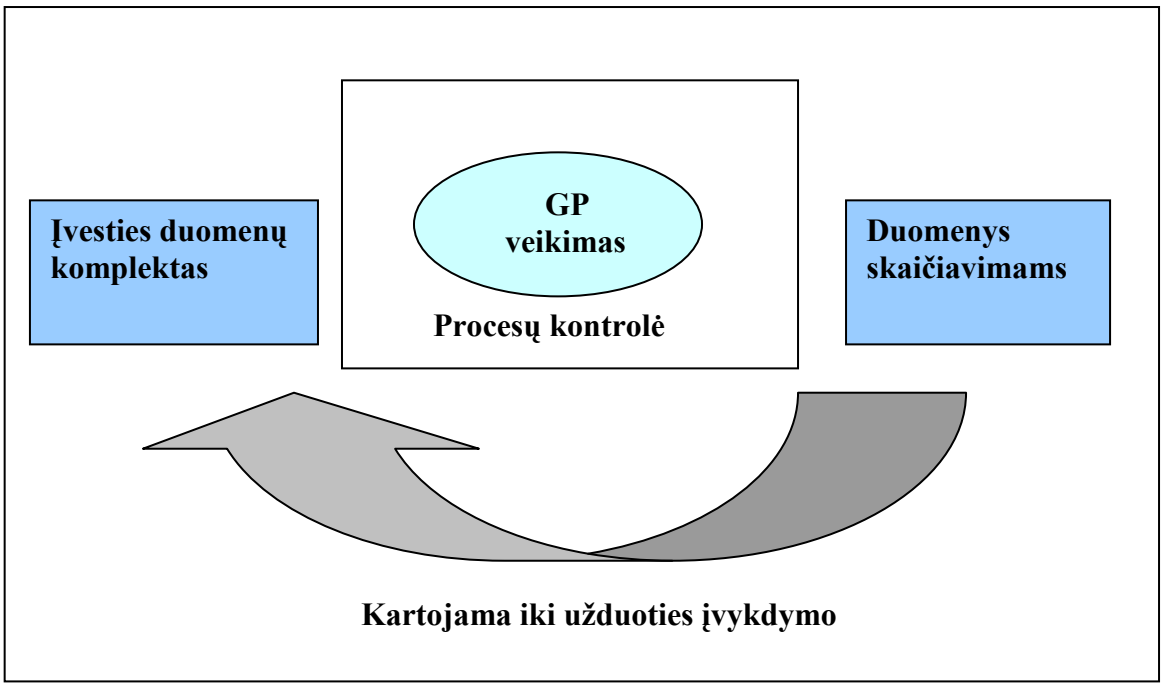
Taip pat galima pasirinkti kryžminimo būdą: vieno taško kryžminimas, dviejų taškų kryžminimas, su daug kryžminimo taškų, tolygus kryžminimas. Vieno taško kryžminimas yra nustatytas, kaip pradinis parametras.

10 pav. pateiktas vių funkcijų sąrašas, kurių formules galima surasti www.ewh.ieee.org/soc/es/Begin.htm puslapyje.

Šis mokymo įrankis vartotojui tinkamas susipažinti su genetiniais algoritmais. Įrankio paleidimas yra labai paprastas, nereikia jo įdiegti į savo kompiuterį, o tik paleisti EGALT.EXE failą. Tačiau vykdyti skaičiavimams, kurie turėtų mokslinę reikšmę, įrankis yra nepritaikytas. Šio įrankio didžiausias trūkumas, kad vartotojas pats negali įvesti tinkamumo funkcijos. Galima pasirinkti funkciją, kuri numatyta įrankyje. Taip pat vartotojas negali suvesti genų parametrų reikšmių, genų reikšmės generuojamos atsitiktiniu būdu. Nerealizuotas rezultatų atvaizdavimas grafike, todėl gautus iteracijų duomenys yra sudėtingiau palyginti. Šio įrankio, pagrįsto genetiniais algoritmais, veikimo sparta yra pakankamai aukšta - 1000 iteracijų suskaičiuoja vidutiniškai per 3 sekundes.

3.4. JGAP paketas

1990 m. John Koza pasiūlė genetinio programavimo paradigmą, kuri yra naudojama, kaip pagrindinė idėja daugumose genetinių algoritmų sistemų. Sudaromas tam tikras duomenų komplektas, kurį apdoroja programa, turinti genetinius algoritmus, ir gautas rezultatas vėl naudojamas skaičiavimams. Ciklas vykdomas tol, kol patenkinama nurodyta sąlyga.



11 pav. Programos su genetiniais algoritmais veikimas

Java genetinių algoritmų paketo (toliau - JGAP) idėjos principas buvo - sudaryti bendrą genetinių programų funkcijų sistemą, į kurią būtų įtrauktos pagrindinės genetinės operacijos. Naudojant tokias sistemas galima greičiau ir patikimiau kurti genetines programas, kuriose galima įgyvendinti savo logiką.

Java genetinių algoritmų paketas – tai genetinių algoritmų ir genetinių programų komponentas, pateiktas kaip Javos struktūra. JGAP turi pagrindinius genetinius mechanizmus, kurie gali būti panaudoti pritaikant evoliucinius principus problemoms spręsti. Šis paketas yra nemokamas, naudojama viešoji GNU licencija, o tai reiškia, kad ją galima legaliai naudoti visiems mokslo darbams.

JGAP pakete yra daug pavyzdinių modelių (>1400), kurie padeda problemos sprendimui, taip pat detali dokumentacija apie JGAP paketą.

JGAP yra aukšto lygio modulinė sistema, kuri gali būti lengvai pritaikoma ir kituose Java taikomosiose programose. Taip pat JGAP leidžia taikyti pasirinktas GA operacijas, tokias kaip rekombinacija su trimis pasirinktais taškais, nors JGAP turi įgyvendintas pagrindines genetinių algoritmų funkcijas.

JGAP yra plėtojamas pagal tam tikrą iš anksto numatytą misiją:

- Pateikti stabilią ir patogią sistemą genetiniam programavimui, vykdant daugybę testų kiekvienai klasei.
- Pasiūlyti paprasto naudojimo sistemą, kad ją būtų lengva įdiegti ir pradedantiesiems programuotojams.

- Pateikti visapusišką ir detalią dokumentaciją, taip pat paruoštų programų pavyzdžių.

JGAP yra Java bibliotekos paketas, kurio pavadinimas jgap.jar. Šis failas turi būti įtrauktas į programos klasės kelią, kurį naudoją JGAP. Sistemos kodą galima rasti oficialiame JGAP tinklapyje: http://jgap.sourceforge.net.

JGAP suteikia galimybę išsaugoti duomenis XML formatu. Tai yra naudinga, jei norime išsaugoti esamą populiaciją faile ir kitą kartą pratęsti likusį iteracijų ciklą. Taip pat suteikta galimybė išsaugoti atskirus chromosomos genotipus ir net atskirus genus. Šis procesas vadinamas rūšiavimu. Žinoma JGAP suteikia galimybę importuot xml failą į programą. Už šias funkcijas yra atsakinga *org.jgap.xml.XMLManager* klasė.

Genetinės aplinkos konfigūravimas

Bet kurioje programoje, kurioje yra naudojama JGAP, prieš pradedant genetinius procesus, tokius kaip kryžminimas, reikia nustatyti genetinę aplinką. Į genetinės aplinkos nustatymą įeina genomo galimų sprendimų struktūros apibrėžimas. Tūri būti taikomos genetinės operacijos, tinkamumo funkcijos, kurios įvertina rezultatus, įvertinimas, taip pat skirtingų GA parametrų genetinių procesų kontrolė. Šiam tikslui JGAP turi klasę pavadinimu *org.jgap.Configuration,* kuri tūri būti sukurta ir tinkamai sukonfigūruota mūsų programoje.

```
Configuration conf = new DefaultConfiguration();
```

DefaultConfiguration yra konfigūracinis poklasis, kuris priskiria parametrams numatytas reikšmes. Į šią konfiguraciją taip pat įeina kryžminimo ir mutacijos operacijos objektai. Tačiau, nors ir naudojamas *DefaultConfiguration* poklasis, vis tiek lieka galimybė keisti parametrų ir konfigūracijos objektų reikšmes [29].

JGAP chromosomos struktūra

Genetinėse programose iš anksto turėtų būti apibrėžti struktūros sprendimai, t.y. chromosomos. Programoje reikia sukurti chromosomos tipą, *org.jgap.Ichromosome* ir įvesti konfigūravimo objektą [48].

```
IChromosome sampleChromosome = new Chromosome(conf, sampleGenes);
conf.setSampleChromosome(sampleChromosome);
```

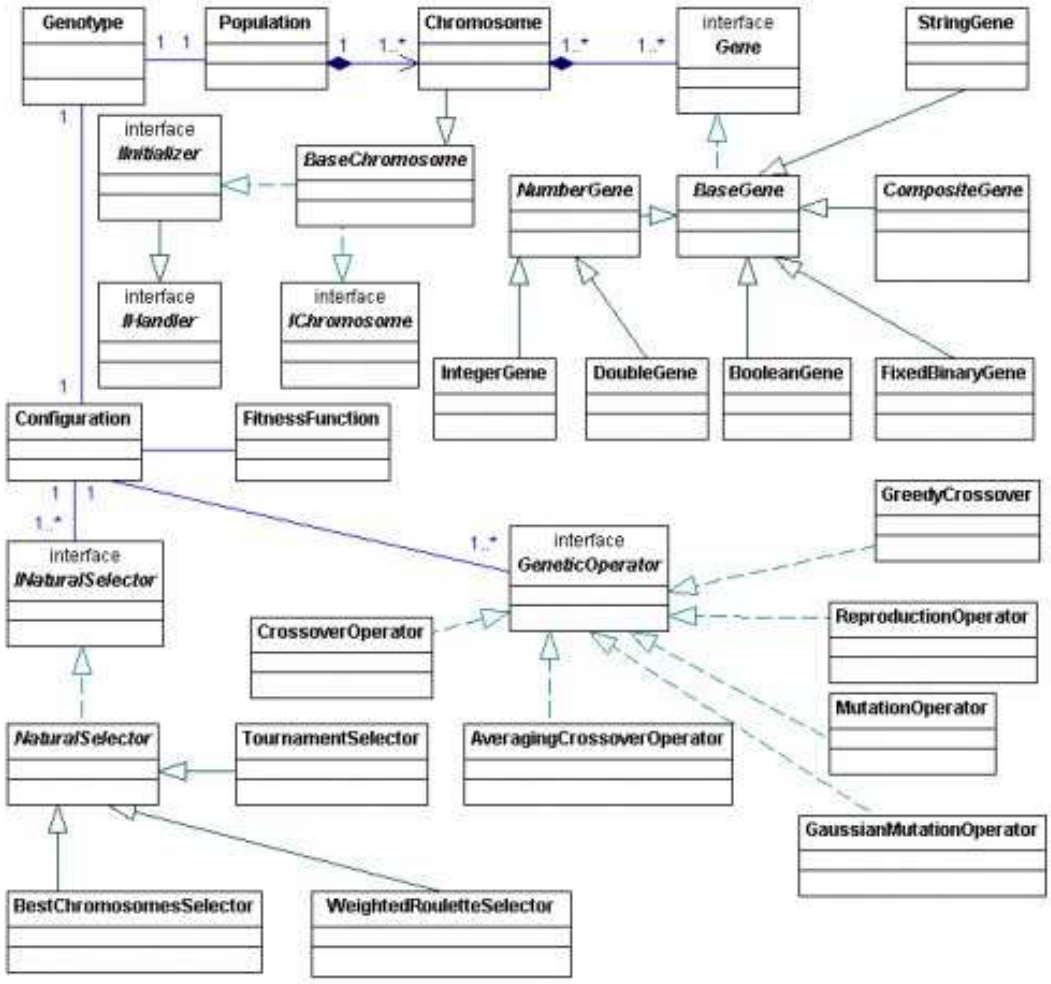
Pateiktame pavyzdyje *sampleGenes* yra vektorius iš *org.jgap.Gene* tipo.

Genetinių algoritmų vykdymas

Atsižvelgdami į genetinių chromosomų apibrėžtą struktūrą, galima pradėti sudaryti genetinius individus. Kaip jau buvo minėta, genetiniams algoritmams reikia tinkamumo funkcijos,

norint įvertinti sudaromos chromosomos sprendinius. JGAP tinkamumo funkcija yra apibrėžta kaip poklasis *org.jgap.FitnessFunction,* kur vyksta objektų konfigūravimas ir vertinama evoliucija [28].

```
FitnessFunction myFunc = new GAFitnessFunction(graph, vertices);
try {
conf.setFitnessFunction(myFunc);
}
catch(Exception e){
System.err.println("Invalid Function");
System.exit(-1);
}
```



12 pav. JGAP svarbiausių klasių diagrama(V3.1) [10]

4. INFORMACIJOS SAUGOS ESYBIŲ PERSPEKTYVIOS MODELIAVIMO KRYPTIES NUSTATYMAS

Kompanijų ir organizacijų verslo procesai šiuo metu yra labai priklausomi nuo informacinių technologijų infrastruktūros, kuri tampa konkurencingumo ir efektyvumo pagrindu [26]. Jei sutriktų informacinių sistemų veikla, neleistinai pakeista arba konfidenciali informacija patektų į svetimas rankas, gali iškilti pavojus, kad organizacija negalės tinkamai atlikti savo funkcijų ir įgyvendinti pavestų uždavinių [53]. Po vis dažnesnių kibernetinių atakų, kuomet įsilaužiama į valstybinių institucijų interneto svetaines, sutrikdomas jų darbas, žiniasklaidoje nuolat diskutuojama apie tai, koks yra optimalus būdas užtikrinti valstybei svarbios elektroninės informacijos saugą.

Kritinė informacinė infrastruktūra – elektroninių ryšių tinklas, informacinė sistema ar informacinių sistemų grupė, prie kurios neteisėtas prisijungimas ir sąlygų neteisėtai prisijungti sudarymas, kurios neteisėtas sutrikimas ar pakeitimas, kurioje saugomų, tvarkomų, iš jos išrenkamų arba ja perduodamų elektroninių duomenų sunaikinimas, sugadinimas, ištrynimas ar pakeitimas, galimybės naudotis tokiais elektroniniais duomenimis panaikinimas arba apribojimas turi ar gali turėti ženklios neigiamos įtakos nacionaliniam saugumui, šalies ūkiui ar visuomenės gerovei.

[22] Atliktoje literatūros analizėje teigiama, kad pagrindiniai GA taikymo tyrimai informacinės saugos srityje yra orientuoti į taikymus kriptologijoje, IDS ir biometrinėse autentifikacijos sistemose. Yra publikuoti tik keli straipsniai, kuriuose nagrinėjami GA taikymai informacinės saugos procesų modeliavimo uždaviniui spręsti. Kriptologiniai GA taikymai gali būti vertinami kaip įdomūs, tačiau neturintys didelės praktinės reikšmės, kadangi siūlomi tik su standartais nesuderinti patobulinimai egzistuojantiems šiuolaikiniams šiframs, o GA paremtos atakos neaktualios, kadangi taikomos prieš klasikinius (perstatymo, sumaišymo), jau pripažintus nesaugiais („lengvoji“ ir Merkle-Hellman kuprinės) arba nepopuliarius (pvz.,Chor-Rivest) šifrus. GA aktyviai taikomi IDS sistemose, bet siūlomi sprendimai dar neperžengę mokslinių tyrimų ir prototipų bandymų ribos ir dažnai būna siaurai specializuoti. Tyrimai vertinami kaip perspektyvūs, kadangi sukurti prototipai demonstruoja aukštą efektyvumo lygį ir gali būti optimizuoti, parenkant GA parametrus.

Iš informacijos saugos esybių, kenkėjiškas programinis kodas laikomas vienas iš didžiausių grėsmių informacinių technologijų infrastruktūros srityje. Sistemų, nukreiptų prieš kenksmingus programinius kodus, plėtra turi didelį ekonominį poveikį ir yra pastoviai besikeičiantis procesas. Kenkėjiškų programų rizikos lygių vertinimas ir raidos tendencijų prognozavimas yra svarbus mokslinio pagrindimo ekspertų įvertinimas, KPK vystymosi tendencijų supratimas, kenksmingų programų raidos prognozavimas naudojant atskirus parametrus arba jų kompleksus. Taip pat

prevencija įgyvendinant greito reagavimo mechanizmus kenksmingo programinio kodo epideminiams protrūkiams [21].

Iš 3 skyriaus "ĮRANKIAI PAGRĮSTI GENETINIAIS ALGORITMAIS" analizės ir ištestuotų įrankių, naudojančių genetinius algoritmus, nustatyta, kad šiuo metu nėra tinkamo įrankio, kurį būtų galima taikyti informacijos saugos dalykinės srities esybių rizikos vertinimui ir evoliucijos prognozavimui.

5. ĮRANKIO FUNKCINIAI IR NEFUNKCINAI REIKALAVIMAI

Kuriant įrankį, kuris tūri būti patogus vartotojui, lankstus savo įvairiapusiškumu, būtina nusistatyti funkcinius ir nefunkcinius reikalavimus. Programinės įrangos reikalavimai, tai paslaugų nustatymų procesas, kurio vartotojas reikalauja iš sistemos ir apribojimai, pagal kuriuos sistema dirba ir baigia darbą. Reikalavimai skirstomi į keletą tipu [47]:

- Vartotojo reikalavimai:

paprasta (ne technine) kalba surašyti vartotojo pageidavimai būsimai sistemai, gali būti pridedamos laisvos formos diagramos, parašyta taip, kad suprastų bet kuris sistemos vartotojas.

- Sistemos reikalavimai:

struktūrizuotas dokumentas, išdėstantis detalius sistemos funkcijų, paslaugų ir operacijų aprašymus.

5.1. Funkciniai reikalavimai

Funkciniai reikalavimai specifikuoja kiekvienos sąsajos atlikimo savybes, apibūdinančias įeigos ir išeigos procesus (sąsajos funkcionalumą).

Idealiu atveju funkciniai reikalavimai specifikuoja tokius aspektus [39]:

a) įeigos duomenis, išeigos duomenis, galimas duomenų sekas (duomenų transformavimą);

b) sistemos būseną – duomenis, kuriuos sistema turi saugoti. Saugomi duomenys apibrėžia sistemos būseną. Sistemos išeiga priklauso ne tik nuo įeigos, bet ir nuo sistemos būsenos;

c) procesą (transformavimą) – tai, kaip sistema įeigą transformuoja į išeigą.

Reikalavimai sąsajoms aprašomi atskirai. Aprašomos programinės įrangos sąsajos su kita programine įranga ar su vartotoju (duomenų įvedimas ir išvedimas).

Sistemos servisų aprašymai turėtų paaiškinti, kaip sistema turėtų reaguoti į ypatingus duomenų įvedimus ir kaip sistema elgsis ypatingose situacijose. Aprašo funkcionalumą arba sistemos servisus. Priklauso nuo programinės įrangos tipo, laukiamo vartotojų ir sistemos tipo, kur sistema yra naudojama. Funkciniai vartotojų reikalavimai gali būti aukšto lygio teiginiai, apie tai, ką sistema turi daryti, bet funkciniai sistemos reikalavimai turi detaliai aprašyti sistemos servisus.

Kuriamam įrankiui suformuoti šie funkciniai reikalavimai:

Kuriama programinė įranga turi palengvinti bei paspartinti informacijos saugos esybių evoliucijos modeliavimą. Todėl modeliavimo įrankis turi realizuoti tokias funkcijas:

- suvesti ir koreguoti pirminius dokumentus;
- suvesti ir koreguoti normatyvinę informaciją;
- pateikti evoliucijos eigos ataskaitą su evoliucijos metu gautu geriausiu individu.

Funkcinių reikalavimų detalesnis apibūdinimas:

1.Tinkamumo funkcijos įvedimas.

1.1. Tinkamumo funkciją galima įvesti panaudojus pagrindinius aritmetikos simbolius, trigonometriją, šaknies traukimą, kėlimą laipsniu, sumavimo ženklą (Σ), daugybos ženklą (Π)

1.2.Vartotojui įvedus tinkamumo funkciją, ji išsaugoma, kad prireikus būtų galima panaudoti tą pačią funkciją.

1.3.Po genų įvedimo, tinkamumo funkcija sugeneruojama pagal genų priklausomybę.

2. Genų įvedimas.

2.1.Vartotojas prieš įvedant genus gali pasirinkti jų skaičių ir geno parametrų kiekį.

2.2. Genų parametrai išsaugomi, kad prireikus būtų galima panaudoti tuos pačius duomenis.

2.3.Atskirame faile išsaugomi genų parametrų vardai.

3. Kitų parametrų pasirinkimas.

3.1.Vartotojas taip pat gali savarankiškai pasirinkti:

3.1.1. Individų skaičių - kiek chromosomų rinkinių bus sugeneruota.

3.1.2. Iteracijų skaičius- kiek ciklų vyks skaičiavimas.

3.1.3. Mutacijos tikimybė – su kokia tikimybe individo genas mutuos.

4. Rezultatų pateikimas.

4.1.Kiekvienos iteracijos metu generuojamas grafikas, kuriame žymima reikšmė gauta paskaičiavus geriausio individo tinkamumą. Šiame grafike galima matyti kaip kinta evoliucija ir kyla/krenta individų tinkamumas.

4.2.Baigus vykdyt visas iteracijas, ar pasiekus kitą numatytą tikslą, pateikiamas gautas geriausias individas ir individo genų reikšmės. Rezultatas išsaugomas.

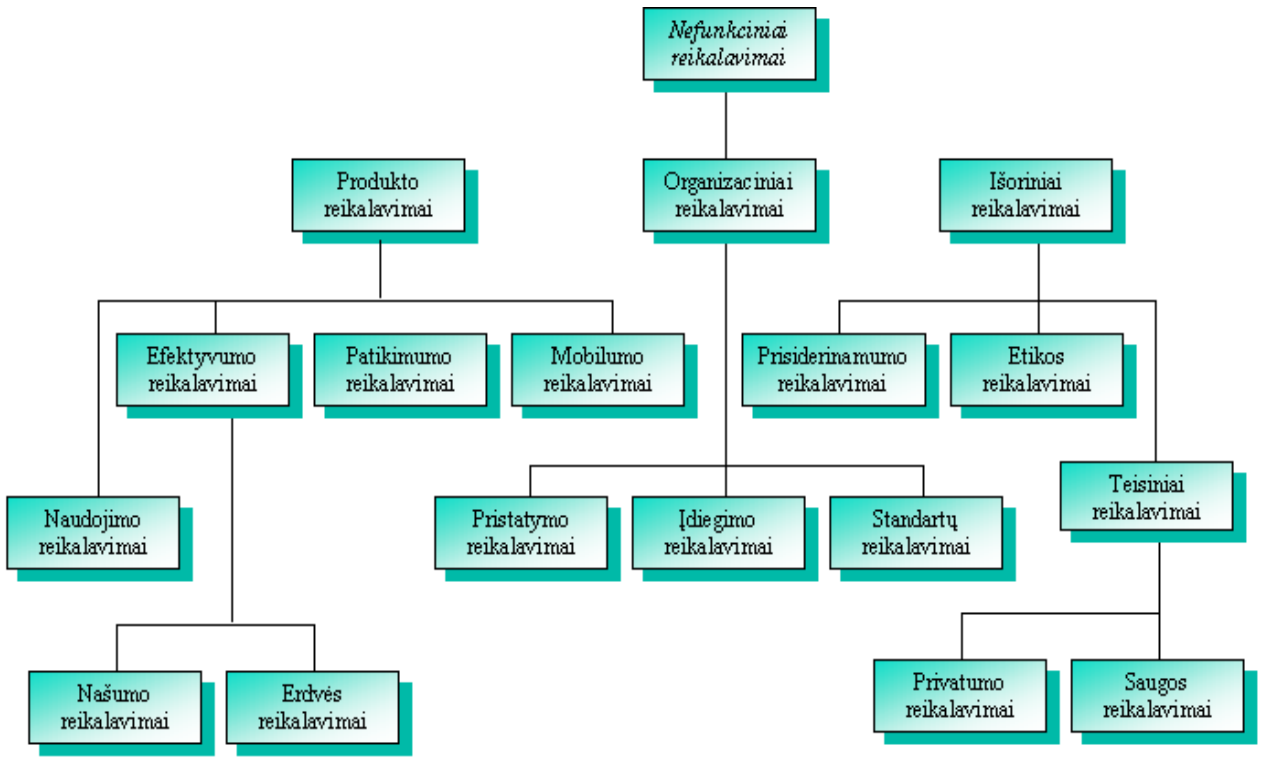
5.2. Nefunkciniai reikalavimai

Nefunkciniai reikalavimai gali būti labiau svarbūs nei funkciniai reikalavimai. Jei jie nėra išpildomi, sistema yra nenaudinga. Nefunkciniai reikalavimai specifikuoja numatomų (projektuojamų) sistemos funkcijų savybes – apibrėžia kokybines funkcijų charakteristikas. Todėl nefunkciniai reikalavimai dar vadinami kokybės reikalavimais.

Kokybės reikalavimai apima tris aspektus:

- Našumą – sistemos efektyvumą, realizavus konkrečią techninę įrangą: koks turėtų būti sistemos reagavimo laikas, naudojamų kompiuterinių resursų kiekis, gaunamų rezultatų tikslumas, saugomų duomenų apimtis (reikalavimai atminčiai).
- Naudojimo savybes – teikiamas vartotojams sistemos savybes: sistemos įsisavinimo paprastumas, kasdieninio darbo našumas, kitos;
- Eksploatavimo savybes – klaidų taisymo paprastumas, papildymo naujomis funkcijomis sudėtingumas, kitos.

Kokybės faktorių yra ir daugiau, čia paminėti svarbesnieji.



14 pav. Nefunkcinių reikalavimų tipai [46]

Nefunkcinius reikalavimus gali būti labai sudėtinga tiksliai nustatyti, o netikslius reikalavimus gali būti labai sunku patikrinti.

Nefunkciniai reikalavimai:

Reikalavimai sistemos išvaizdai

- Lengvai suvokiama naudotojo sąsaja.
- Sistemos naudotojo sąsajos valdymo elementai yra tradiciniai - pelytė ir klaviatūra.
- Pagrindiniame įrankio lange nėra papildomų įvedimo laukų, kurie nesusija su pagrindiniais duomenimis ir skaičiavimo paleidimui.
- Galimybė pildyti įvedimo laukus ne iš eilės.

Reikalavimai panaudojimui

- Duomenų įvedimas turi būti aiškus ir paprastas.
- Įrankis turi padėti vartotojui nedaryti klaidų, arba išsamiai apie jas informuoti jį.
- Vartotojas gali naudotis modeliavimo įrankiu be išankstinio apmokymo, pakanka perskaityti įrankio naudojimo vadovą.
- Dėl patogesnio prieinamumo, įrankis turi būti patalpintas žiniatinklyje. [38]

Reikalavimai vykdymo charakteristikoms

- Įrankis turi atlikti greitus skaičiavimus, naudojant genetinius algoritmus.
- Greitas duomenų įkelimas į failą .

Reikalavimai veikimo sąlygoms

Programinė įranga:

- Operacinė sistema: Microsoft Windows,Unix.
- MS Office: MS Excel.
- Internetinė naršyklė: Internet explorer, Mozila, Google Chrome, Opera, Safari (Versijai reikalavimai nekeliami, pageidautina naudoti kuo naujesnę versiją).
- Java platforma.

Kiti reikalavimai:

- Planuojamas programos dydis (be duomenų) iki 10Mb.
- Galima naudoti iš daugelio darbo vietų.

Reikalavimai sistemos priežiūrai

- Laiku vykdomas dokumentacijos atnaujinimas.
- Sistemos priežiūros instrukcijos, galimos klaidos ir jų sprendimo būdai turi būti aprašyti atskirame dokumente arba pateikti kaip vartotojo vadovo dokumentacijos dalis.

Kultūriniai-politiniai reikalavimai

- Sistemoje naudojama taisyklinga ir aiški lietuvių kalba
- Sistema nenaudoja įžeidžiančių terminų ir simbolių.

6. MODELIAVIMO ĮRANKIO, PAGRĮSTO GENETINIAIS ALGORIMAIS, REALIZAVIMAS

Modeliavimas yra esminė ir neatsiejama mokslinės veiklos dalis. Profesionalus modeliuotojas pasitelkia specialius gebėjimus ir metodus tam, kad galėtų gauti įžvalgius, patikimus ir naudingus rezultatus. Metodai susidedantys iš modernių statistinių metodų, kompiuterių simuliacijos, sistemos atpažinimo, ir jautrumo analizės yra vertingi įrankiai. Tačiau jie nėra tokie svarbūs, kaip galimybė suprasti kompleksinės sistemos dinamikos esmę. Šis supratimas yra reikalingas tam, kad galima būtu nustatyti ar modelio prielaidos yra teisingos ir pilnos. Modeliuotojas turėtų sugebėti atpažinti ar modelis atspindi tikrovę ir turi mokėti atskirti teoriją ir duomenis [23].

6.1. Įrankio testavimas

Testavimo objektas – patikrinti programiniam įrankiui darbo uždaviniuose keltus reikalavimus.

- Testuoto įrankio versija 0. 8.
- Įrankis bus testuojamas Windows 7 operacinėje sistemoje.

Siekiant nustatyti sukurto įrankio trūkumus, koreguotinus aspektus, buvo parengtas tyriamo įrankio testavimo planas.

1. Lango interfeiso testavimas.
2. Pasirinkimo laukų testavimas.
3. Lango laukų testavimas.
4. Lango veikimo testavimas.
5. Įrankio funkcionalumo testavimas.

1 lentelė. Lango sąsajos testavimas

Lango interfeiso testavimas	**Rezultatas**
1. Ar lange naudojamas vienas šriftas;	Taip
2. Ar laukai ir jų pavadinimai sulygiuoti;	Taip
3. Ar prie įvedimo laukų yra jų pavadinimai;	Taip
4. Ar pavadinimų pirmas žodis iš didžiosios raidės;	Taip

5. Ar meniu punktai turi greito priėjimo klavišus ir jie veikia teisingai;	Taip
6. Ar, atidarius įrankio langą, jis yra aktyvus;	Taip
7. Ar lange yra mygtukas: išsaugoti gautus rezultatus;	Taip
8. Ar nenaudotini mygtukai yra uždrausti	Tokiu nėra

2 lentelė. Lango laukų testavimas

Lango laukų testavimas	**Rezultatas**
1. Patikrinti ar programa reikalauja užpildyti visus laukus.	Ne
2. Jei neužpildomas laukas ar gaunamas klaidos pranešimas.	Taip
3. Patikrinti ar ribojamas simbolių įvedimo kiekis	Ne
4. Patikrinti ar grafikas teisingai atvaizduoja iteracijų rezultatus	Taip
5. Ar gaunamas klaidos pranešimas blogai nurodžius kelią iki duomenų failo.	Taip

3 lentelė. Lango veikimo testavimas

Lango veikimo testavimas	**Rezultatas**
1. Vaikščiojimą per redaguojamus laukus su TAB klavišu.	Taip
2. Kaip veikia klavišas ESCAPE.	Funkcijos nėra
3. Kaip veikia klavišų kombinacija CTRL+F4.	Funkcijos nėra
4. Ar veikia klavišų kombinacija ALT+F4.	Taip

4 lentelė. Įrankio funkcionalumo testavimas

Įrankio funkcionalumo testavimas	**Rezultatas**
1. Patikrinti ar įrankis leidžia išsaugoti gautus interacijos rezultatus.	Taip
2. Patikrinti ar galima įvesti tinkamumo funkcija su sin, cos, ^, +,-,*,/,	Taip
3. Patikrinti ar galima įvesti tinkamumo funkcija naudojant sigma ir pi.	Taip
4. Patikrinti ar įrankis leidžia nurodyti genų pavadinimų failą	Taip
5. Patikrinti ar įrankis leidžia naudotis pagalbos langu.	Taip
6. Ar neatitikus duomenų failo vardui ir kintamojo vardui gaunamas pranešimas	Taip

Testavimo išvados

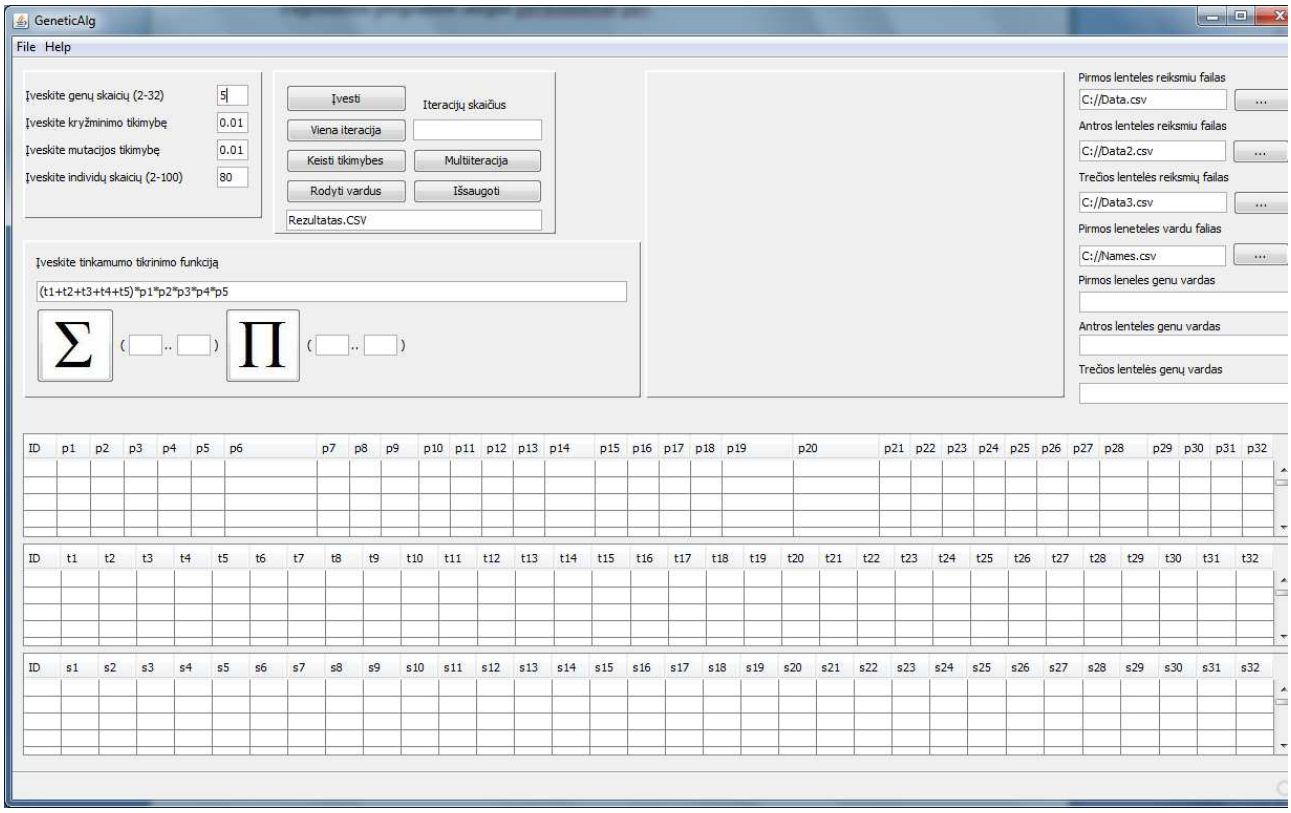
Informacijos saugos dalykinės srities esybių evoliucijos modeliavimo įrankis, paremtas genetiniais algoritmais atitiko testavimo keliamus funkcinius reikalavimus. Apibendrinus testavimo išvadas galima teigti, kad programa yra pakankamai ištestuota, kad galėtų būti pateikta vartotojui.

6.2. Įrankio aprašymas

6.2.1. Pagrindinis programos langas

Pagrindinis programos langas susideda iš meniu juostos bei laukų pradiniams duomenims įvesti.

Pagrindinis programos langas pavaizduotas 15pav.



15 pav. Pagrindinis programos langas

6.2.2. Meniu juosta

- **File (failas)**

Exit (išeiti) - iškvietus šią komandą programa automatiškai užsidaro.

- **Help (pagalba)**

About (apie) – čia galima pamatyti naudojamo įrankio versiją.

6.2.3. Pagrindinių duomenų įvedimas

Duomenys susiję su algoritmo skaičiavimais nustatomi pagrindiniame įrankio lange, atitinkamuose laukeliuose:

Įveskite genų skaičių- šiame laukelyje nurodomas genų skaičius, kuris nurodo kiek genų paimti iš duomenų failo, skaičiuojant nuo B stulpelio.

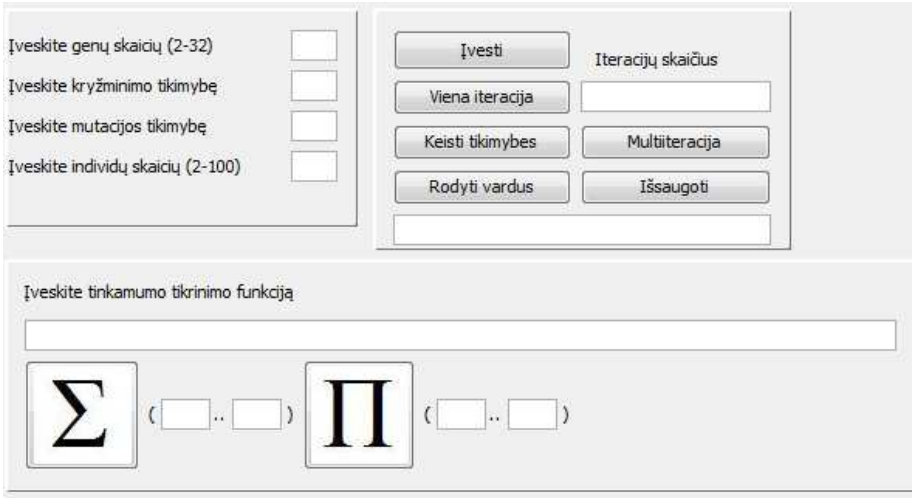
Įveskite kryžminimo tikimybę- tikimybė, kuri nurodoma kryžminimo tikimybės lauke, apatinio sąrašo individas pakeičiamas dviejų viršutinio sąrašo nariu palikuoniu. Palikuonis sudaromas pasirenkant atsitiktinai tėvų genu reikšmes.

Įveskite mutacijos tikimybę- tikimybė su kuria mutuos genas kiekviename individe. Dažniausiai ši tikimybė būna labai maža.

Įveskite (2-100) individų skaičių - individų kiekis, su kuriais bus vykdomas algoritminis skaičiavimas.

Iteracijų skaičius – kiek iteracijų bus vykdoma skaičiavimuose.

Įveskite tinkamumo funkciją – funkcija, pagal kurią bus vykdomi geriausio individo skaičiavimai. Sigma ir Pi įvedimas - į laukelius esančius skliaustuose įvedam kintamąjį su jo eilės numeriu ir spaudžiam norimą ženklą - $\sum$ arba $\prod$. Programa automatiškai sugeneruos ir įdės į tinkamumo funkciją norimą matematinį veiksmą. Tarkim, reikia, kad tinkamumo funkcijoje būtų $\sum_{j=1}^{22} t_j$, tada į pirmą laukelį rašome t1, į antrą laukelį įrašom t22 ir spaudžiam sigmos ženklą.



16 pav. Įvedimo laukai

6.2.4. Duomenų failai

Į duomenų failą įvedamos genų parametrų reikšmės, jų pavadinimai. Taip pat į atskirą failą išsaugomi geriausio individo rezultatai. Duomenų failo plėtinys CSV.

	A	B	C	D	E	F	G
1	Nothing	1	2	3	4	5	
2		6	66	6	6	6	
3		46	6	66	3	1	
4		3	2	16	7	66	
5		4	3	4		4	
6		40.545				3	
7						84	
8							

17 pav. Duomenų failas

Geno parametrų reikšmės pradedamos vesti nuo B stulpelio ir pirmos eilutės. Kadangi tai csv - taškais atskirtos reikšmės formatas, vietoj kablelio dedamas taškas.

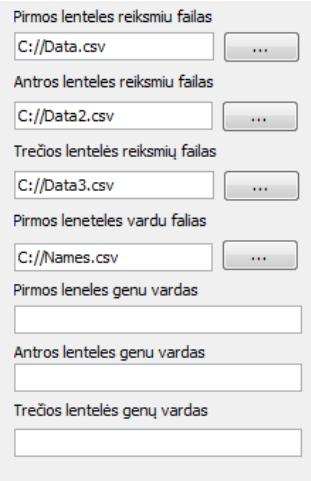
Kiekvienas parametras turi savo vardą, kurį turi atitikti ir jo duomenų failo pavadinimas. Tarkim jei parametras yra pavadinimu r, tai ir duomenų failas turi būti r.csv

Kiekvienai geno reikšmei galima suteikti pavadinimą, jie taip pat pildomi į failą, kurio pavadinimas Name.csv. Failas pradedamas pildyt nuo B stulpelio. Pavadinimų koordinatės turi sutapti su jom priskirtom reikšmių koordinatėm.

	A	B	C	D	E	F	G	H
1	0	MMS	Linux	nokia	Adresų knygos numeriai	10:00val.-20:00val	SMS	Bluetooh
2			mac	rim	Priimtų skambučių	Visada		
3			windows	palm	Atsitiktinis generavimas	20:00-10:00		
4			symbian	aplle				
5			RIM					
6								
7								
8								
9								

18 pav. Geno reikšmių pavadinimų failas

Pradiniame įrankio lange nurodomas kelias iki duomenų failų ir kintamųjų vardai. Duomenų failo pavadinimas ir kintamojo vardas turi sutapti. Įrankis gali būti realizuotas su trim kintamaisiais.



19 pav. Duomenų failų įvedimo laukai

6.2.5. Iteracijų vykdymas

Suvedus visus duomenys reikalingus skaičiavimams ir nurodžius kintamųjų reikšmių failus, spaudžiama "Įvesti". Įrankio lango apačioje užsipildo lentelė individų genų parametrų reikšmėmis.

ID	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20
1	0.7	0.084	0.53	0.2	0.4	0.7	0.2	0.05	0.4	0.02	0.0	0.2	0.0	0.04	0.0	0.0	0.0	0.01	0.0	0.01
2	0.7	0.107	0.13	0.1	0.6	0.7	0.2	0.05	0.4	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.01
3	0.7	0.084	0.53	0.2	0.4	0.7	0.2	0.05	0.4	0.02	0.0	0.2	0.0	0.04	0.0	0.0	0.0	0.01	0.0	0.01
4	0.7	0.107	0.13	0.1	0.6	0.7	0.2	0.05	0.4	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.01

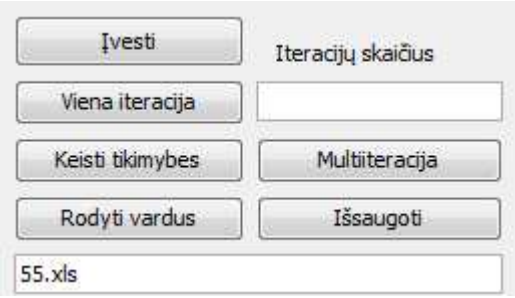
ID	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20
1	3.0	0.0...	0.0...	0.01	0.01	3.0	1.0	0.01	0.0...	0.01	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.01
2	3.0	0.0...	0.0...	0.01	0.01	3.0	1.0	0.01	0.0...	0.01	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.01
3	3.0	0.0...	0.0...	0.01	0.01	3.0	1.0	0.01	0.0...	0.01	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.01
4	3.0	0.0...	0.0...	0.01	0.01	3.0	1.0	0.01	0.0...	0.01	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.0...	0.0	0.01

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

20 pav. Užpildyta genų parametrų lentelė

Skaičiavimus galima vykdyti po viena iteracija, šiuo atveju reikia spausti "Viena iteracija". Jie norima skaičiuot vykdant daugiau iteracijų į langelį "Iteracijų skaičius" įvedam norima iteracijų skaičių ir spaudžiam "Multiiteracija"

Pasibaigus genetinio algoritmo vykdymui, galima pažiūrėt ne tik genų reikšmes, bet ir jų pavadinimus. Reikia spausti "Rodyti vardus". Taip pat galima visą rezultatą išsaugoti į atskirą failą. Reikia laukelyje įrašyti failo pavadinimą su plėtiniu csv arba xls ir spausti "Išsaugoti". Failas yra išsaugomas toj direktorijoj, kurioj yra įrankio paleidimo failas.



21 pav. Skaičiavimų paleidimo langas

6.3. Informacijos saugos esybės evoliucijos modeliavimas

Išanalizavus įrankio funkcionalumu ir naudojimo instrukciją, galima atlikti modeliavimą, kurio rezultatas jau yra žinomas. Šis testas yra skirtas įsitikinti, ar modeliavimo įrankis veikia tinkamai- gaunami teisingi rezultatai, ar panaudoti tinkami optimizavimo raktai, kad skaičiavimai užtruktų kuo trumpiau.

Kaune, konferencijoj "16th International Conference on Information and Software Technologies" konferencijoje buvo pristatytas straipsnis “Genetic Algorithm Modeling Approach for Mobile Malware Evolution Forecasting”[45]. Šiame straipsnyje nagrinėjama mobiliųjų telefonų kenksmingų programinių kodų plitimo problema. Pasiūloma mobiliųjų telefonų kenksmingo programinio kodo plitimo strategijos chromosoma. Panaudojant genetinius algoritmus, apskaičiuojama geriausia kenksmingo programinio kodo plitimo strategija. Remiantis gautais skaičiavimais sudaroma mobiliųjų telefonų KPK raidos evoliucija.

5 lentelėje [45] pateikta mobiliųjų telefonų kenksmingo programinio kodo plitimo chromosoma su išnagrinėtais genais. Ši pasiūlyta plitimo chromosoma yra bendra visiems mobiliųjų telefonų kenksmingiems programiniams kodams. Skaičiavimo metu, genetinių algoritmų pagalba yra sudarinėjami individai, naudojant reikšmes iš genų parametrų reikšmių lentelės (6 lentelė).

5 lentelė. Chromosomos struktūra

Geno numeris / Pavadinimas / Tipas / Aprašymas / Komentaras	**Galimos geno reikšmės**	**Geno numeris / Pavadinimas / Tipas / Aprašymas / Komentaras**	**Galimos geno reikšmės**
1/TRANSF1/AA/ Apsprendžia plitimo tipą / genas apibrėžią, kad KPK galės plisti MMS žinutėmis	MMS	10/OS_PLATF/AA/ nurodo OS, kurioje gali veikti KPK	Linux; WIN MOBILE; SYMBIAN;Android…
2/TRANSF2/AA/ Apsprendžia plitimo tipą / genas apibrėžią, kad KPK galės plisti SMS žinutėmis	SMS	11/TEL/AA/ nurodo mob. telefono modelį kuriame gali veikti KPK	NOKIA, SAMSUNG, Apple, RIM;…
3/TRANSF3/AA/ Apsprendžia plitimo tipą / genas apibrėžią, kad KPK galės plisti BT ryšiu	Bluetooth	12,13,14/EN_EXPL_N/AA/EXPL_N (N=1-3) aktivavimo genai	ON=ExploitRef / OFF

4/TRANSF4/AA/ Apsprendžia plitimo tipą / genas apibrėžią, kad KPK galės plisti el. paštu	e-mail	15,16,17/EXPL_N(N=1-3)/AE/Įsiskverbimo programos, kurias gali naudoti KPK	Pasirenkama atsitiktinai, pasirenkama tik tinkamą.
5/TRANSF5/AA/ Apsprendžia plitimo tipą / genas apibrėžią, kad KPK galės plisti Wi-Fi ryšiu	Wi-Fi	18/NR_TIME/AA/NR geno nustatymai- aktyvavimo valandos	Visada; 10:00-20:00; 20:00-10:00
6/NR/AE/ Ieško arba generuoja numerius / Reikalingas SMS arba MMS plitimo tipui	Priimtų skambučių, Adresų knygos numeriai; Atsitiktinis generavimas; Generavimas numeriu, kurio yra fiksuotas šalies kodas: …	19/BT_TIME/AA/BT geno nustatymai- aktyvavimo valandos	Visada; 10:00-20:00; 20:00-10:00
7/BT/AE/ Paieškos modulis, ieško mobiliųjų telefonų su įjungtu bluetooth ryšiu	Skenuoti/Neskenuoti	20/WIFI_TIME/AA/WIFI geno nustatymai- aktyvavimo valandos	Visada; 10:00-20:00; 20:00-10:00
8/EMAIL/AE/E-mail siuntimo modulis	Adresu knyga; e-mail adresu duomenų bazė.	21/EXEC/AA/EXEC_CHAN aktyvavimo genas	Standartinės funkcijos; Atnaujinimo funkcijos; Standartinės + atnaujinimo funkcijos;
9/WIFI/AE/Paieškos modulis, ieško mobiliųjų telefonų su įjungtu Wi-Fi ryšiu	Skenuoti/Neskenuoti	22/EXEC_CHAN/AE/ nustato KPK atnaujinimo aplinką	e-mail; WI-FI; web-update

Čia AA- visada aktyvus genai, AE- genai, kurių veikimą aktyvuoja kiti genai.

6 lentelė. Genų parametrų reikšmės

Geno Nr.	1	2	3	4	5	6		
Reikšmė	MMS	SMS	Bluetoothe	e-mail	Wi-Fi	Adresų knygos numeriai	Priimtų skambučių	Atsitiktinis generavimas
p	0,7	0.7	0.2	0.05	0.4	0.2	0.1	0.4
t	3	3	1	0.01	0.005	0.01	0.01	0.03
Geno Nr.	7		8	9		10		
Reikšmė	Skenuot	Neskenuot	E-mail adresu knyga	Skenuot	Neskenuot	MAC OS	MS WINDOWS MOBILE	SYMBIAN
p	0	0	0.2	0	0	0.107	0.124	0.471
t	0	0	0.01	0	0	0.005	0.005	0.005
Geno Nr.		11				12/13/14	15	16
Reikšmė	RIM	NOKIA	RIM	PALM	Aplle	EXP_ON/OFF	Expl1	Expl2
p	0.195	0.53	0.13	0.08	0.06	0	0.1	0.05
t	0.005	0.005	0.005	0.005	0.005	0	0.005	0.005
Geno Nr.	17	18			19		20	
Reikšmė	Expl3	10:00val.-20:00val	Visada	20:00-10:00	10:00val.-20:00val	Visada	Visada	20:00-10:00

p	0.2	0.4	0.6	0.1	0.4	0.6	0.6	0.1
t	0.005	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Geno Nr.		21	22		
Reikšmė	10:00val. -20:00val	Stand.+ at. Funkc.	e-mail	Wi-Fi	web-update
p	0.4	0	0.01	0.01	0.1
t	0.01	0.005	1	0.5	0.5

Genetinių algoritmų skaičiavimuose naudojama tinkamumo funkcija, kuria nustatoma gauto individo tinkamumo vertė esančioje populiacijoje. Tinkamumo funkcija sudaroma pagal pasirinktos chromosomos struktūrą. Taip pat yra svarbu kokį modelį norime sudaryti. Kadangi šiame darbe nagrinėjamas informacijos saugos esybių evoliucijos modeliavimo klausimas, kaip modelių pavyzdį galima pateikti kompiuterinių virusų- kirminų, BotNet ar kitų klasių virusų evoliucinį modeliavimą ar mobiliųjų telefonų kenksmingų programinių kodų evoliucijos modeliavimą.

Vykdant pasirinktą iteracijų skaičių kiekviename cikle gaunama populiacija, kurioje gyvuoja individai su evoliucijos metu gauta plitimo strategija. Gautos strategijos efektyvumas nustatomas sudaryta tinkamumo funkcija.

Pateiktai chromosomos struktūrai (5 lentelė), sudaroma tinkamumo funkcija, 2 formulė.

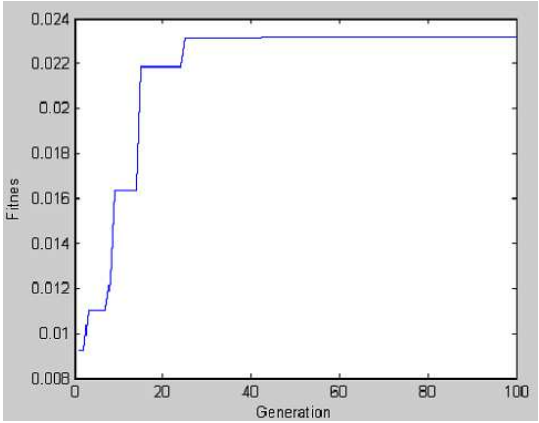
$$F(S) = k \cdot \left(\begin{array}{l} \left(1-\left(1-p_6\left(NR_TIME\right)\right)\cdot\left(1-p_7\left(BT_TIME\right)\right)\cdot\left(1-p_8\right)\cdot\left(1-p_9\left(WIFI_TIME\right)\right)\right)\cdot p_{10}\cdot p_{11}\cdot \\ \cdot\left(1-\prod_{i=15}^{17}\left(1-p_i\right)\right) \end{array} \right), \qquad (2)$$

čia S- evoliucijos strategija; p_6-p_9 – tikimybė, kad įsilaužimo programa sėkmingai pasieks numatytą įrenginį. 6, 7, 9 genai turi priklausomybę nuo laiko genų, kurių numeriai chromosomos struktūros lentelėje 18, 19 ir 20. Parametras k- ciklų skaičius, kuriuos KPK atlieka per vieną sekundę.

$$k = \frac{1}{\sum_{j=1}^{22} t_j}, \qquad (3)$$

čia t_j – būtinos laiko sąnaudos reikalingos *j* genui funkcionuoti [45].

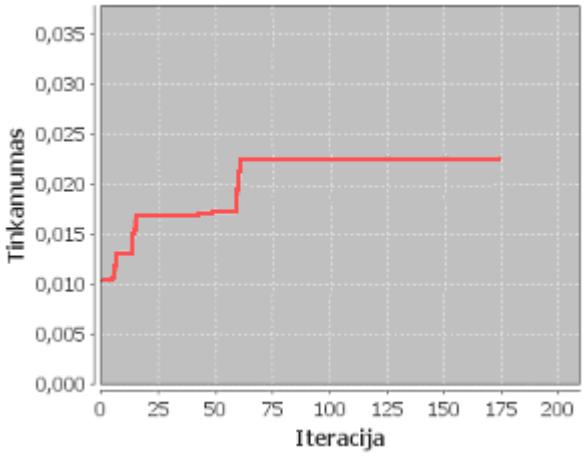
Iš straipsnio [45] žinome, kad skaičiuojant pagal 6 lentelėje pateiktas genų reikšmes ir tinkamumo funkciją, geriausios strategijos tinkamumas yra $F(S_d)= 0.023,$ o geriausias rezultatas gautas 42 iteracijos metu.



22 pav. Geriausios plitimo strategijos kaita [45]

Panaudojus tas pačias genų reikšmes ir tą pačią tinkamumo funkciją, apskaičiuojam geriausią plitimo strategiją pasiūlyto specializuoto modeliavimo įrankio pagalba.

Modeliavimo metu gauti rezultatai panašūs su straipsnyje pateiktais rezultatais (paklaida mažesnė nei 0,01 buvo įvertinta kaip neesminė.). 22 pav. pateikta geriausių plitimo strategijų evoliucija. Geriausios strategijos tinkamumas yra $F(S_d)= 0.0229$.

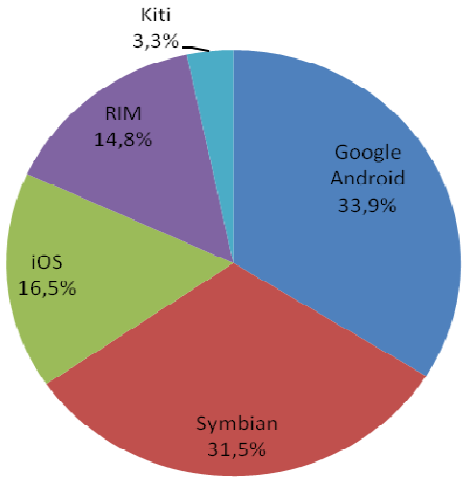


23 pav. Geriausios plitimo strategijos kaita, gauta paiūlytu įrankiu

Kadangi išmaniųjų telefonų rinka per pastarąjį pusmetį pasikeitė, dėl į rinką įžengusio naujo OS gamintojo. Būtų tikslinga atlikti naują išmaniųjų telefonų kenksmingo programinio kodo strategijos evoliucijos modeliavimą. Atlikti naują modeliavimą, reikia išanalizuoti dabartinę išmaniųjų telefonų ir jų operacinių sistemų rinką ir jų paplitimą pasaulyje.

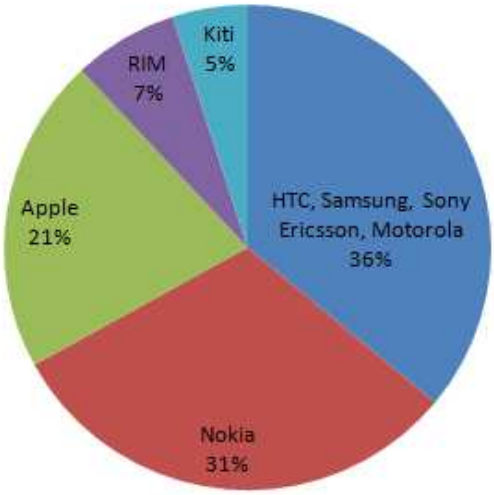
„Canalys“ duomenimis, Google Android operacinė sistema šiuo metu iš viso užima 32,9% globalios išmaniesiems telefonams skirtų operacinių sistemų rinkos. Antroje vietoje rikiuojasi lyderio pozicijas praradusi Symbian su 30,6% dalimi, po jos seka Apple įrenginiuose veikianti iOS platforma (16%). Populiariausių operacinių sistemų penketuką užbaigia RIM išmaniuosiuose veikianti BlackBerryOS (14,4%) ir dar palyginus neseniai pasirodžiusi Microsoft Windows Phone 7

platforma. 2,9% likusios rinkos užima kitos OS, į kurių sąrašą patenka tokie vardai kaip Bada, Maemo, Brew, Windows Mobile, webOS ir kiti. Nepriklausomo tinklaraščio „Communities-dominate“ savininkas Tomis Ahonenas mano, jog Android pardavimai apskaičiuoti netinkamai ir gerokai padidinti. Kaip svarbiausią savo argumentą jis pasitelkia ne tik skaičiavimus bei įvairios statistikos palyginimus, bet ir faktą, kad šių „Canalys“ analitikų duomenų nepatvirtino pati Google [24] [43].



24 pav. Išmaniųjų telefonų OS rinkos statistika

„Canalys“ taip pat pateikia populiariausių išmaniųjų telefonų sarašą. Dižiausią rinkos dalį užima telefonai kurie naudoja Android operacinę sistemą, tai HTC, Samsung, Sony Ericsson, Motorola (36%). Po jų seka Nokia (31%). Į juos dažniausiai įdiegta Symbian operacinė sistema. Trečiąją ir ketvirtąją vietą dalinasi Apple (21%) su iOS operacine sistema ir RIM (7%) mobilieji telefonai. Likusią rinkos dalį užima kiti mobilieji telefonai (5%) [25].



25 pav. Išmaniųjų telefonų rinkos statistika

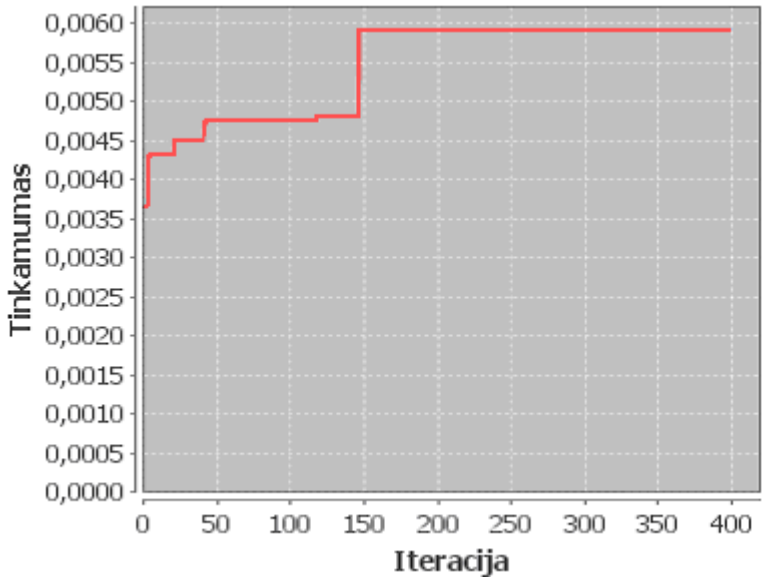
Pasitelkus šiuos statistinius duomenis, genų reikšmių lentelėje pakeičiamos atitinkamų genų parametrų reikšmės. Taip pat pakoreguojamos NR geno parametrų reikšmės, kadangi daugumoje išmaniųjų telefonų yra įdiegtos apsaugos programos, kurios turi tokias funkcijas kaip „SMS/MMS Antispam“ – funkcija, atskirianti patikimus kontaktus pagal nustatytus sąrašus arba blokuojanti nežinomus telefono numerius. Todėl kenksmingas programinis kodas plitimui naudodamas adresų knygos kontaktų numerius turi didesnę tikimybę patekti į išmanųjį telefoną, kuriame yra minėta apsaugos funkcija.

7 lentelė. Atnaujintos genų parametrų reikšmės

Geno Nr.	1	2	3	4	5	6		
Reikšmė	MMS	SMS	Bluetoothe	e-mail	Wi-Fi	Adresų knygos numeriai	Priimtų skambučių	Atsitiktinis generavimas
p	0,7	0.7	0.2	0.05	0.4	0.4	0.1	0.3
t	3	3	1	0.01	0.005	0.01	0.01	0.03
Geno Nr.	7		8	9		10		
Reikšmė	Skenuot BT	Neskenuot BT	E-mail adresu knyga	Skenuot	Neskenuot	Google Android	Symbian	iOS
p	0.2	0	0.2	0.4	0	0.329	0.306	0.16
t	0.5	0	0.01	0.005	0	0.005	0.005	0.005
Geno Nr.		11				12/13/14	15	16
Reikšmė	BlackBerryOS	NOKIA	RIM	HTC, Samsung, Sony Ericsson, Motorola	Aplle	EXP_ON/OFF	Expl1	Expl2
p	0.144	0.31	0.07	0.36	0.21	0	0.1	0.05
t	0.005	0.005	0.005	0.005	0.005	0	0.005	0.005
Geno Nr.	17	18			19			20
Reikšmė	Expl3	10:00val.-20:00val	Visada	20:00-10:00	10:00val.-20:00val	Visada	20:00-10:00	20:00-10:00
p	0.2	0.4	0.6	0.1	0.4	0.6	0.1	0.1
t	0.005	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Geno Nr.	20		21	22				
Reikšmė	10:00val.-20:00val	Visada	Stand.+ at. Funkc.	web-update	e-mail	Wi-Fi		
p	0.4	0.6	0	0.1	0.01	0.01		
t	0.01	0.01	0.005	0.5	1	0.5		

Atlikus skaičiavimus su naujom genų parametrų reikšmėm, gauta nauja KPK plitimo strategija, kurios tinkamumas yra mažesnis, nei buvo gauta su ankstesnėm parametrų reikšmėm. Buvo atlikti 5 modeliavimo bandimai, su įvairiais kryžminimo ir mutacijos parametrais. Skaičiavimuose mažiausiai iteracijų reikėjo, kai kryžminimo tikimybė buvo lygi 0.3, o mutacijos

tikimybė 0.01. Taip pat pasirinkti mažą tikimybės dydį rekomenduojama ir literatūroje [6] apie genetinius algoritmus. Populiacijos dydis buvo pasirenkamas nuo 60 iki 80 individų, tačiau tai skaičiavimams didesnės įtakos neturėjo.



26 pav. KPK plitimo strategijos kaita, gauta su naujomis parametrų reikšmėmis

(26 pav.) pateikta geriausių plitimo strategijų evoliucija, kuri yra gauta naudojant naujas parametrų reikšmes. KPK plitimo strategijos tinkamumas $F(S_p)$=0,0059. Panaudojus duomenis iš rezultatų failo galima sudaryti gautos geriausios KPK plitimo strategijos chromosomą.

S_p=(TRANSF1=" SMS", TRANSF2='" Bluetooth", TRANSF3=" e-mail", TRANSF4=" Wi-Fi", TRANSF5="MMS", NR=" Adresų knygos numeriai", BT=" Skenuot BT", EMAIL=" E-mail adresu knyga", WIFI=" Skenuot", OS_PLATF=" Google Android", TEL=" HTC, Samsung, Sony Ericsson, Motorola" EN_EXPL_N="EXP1_ON_ALL", EXPL_1=" Expl1", EXPL_2=" Expl2", EXPL_3=" Expl3", NR_TIME=" Visada", BT_TIME=" Visada", WIFI_TIME=" Visada", EXEC=" Stand.+ at. Funkc.", EXEC_CHAN=" web-update",)

Plitimo strategijoje pakito kelios genų reikšmės - OS_PLATF=" Google Android", TEL=" HTC, Samsung, Sony Ericsson, Motorola". OS platformos ir išmaniųjų telefonų gamintojų pasikeitimas įtakoja, kad rinkoje įsitvirtino Google Android operacinė sistema, o taip pat ir mobilieji telefonai į kuriuos diegiama ši OS. Prieš tai geriausia plitimo strategija naudojo Symbian operacinę sistemą ir NOKIA mobiliuosius telefonus.

Geriausia KPK plitimo strategija yra 3,7 karto mažesnė nei prieš tai buvusi. Strategijos efektyvumo sumažėjimą įtakoja nelabai stiprus operacinės sistemos ir mobiliųjų telefonų įsitvirtinimas rinkoje, nes panašią rinkos dalį užima Android ir Symbian OS. Taip pat plitimo strategijos efektyvumo sumažėjimą lėmė atsiradusios efektyvesnės apsaugos programos nuo

kenksmingų programinių kodų. Dėl to sumažėjo tikimybė užkrėsti kitą išmanųjį telefoną plintant KPK per trumpąsias žinutes, nes atsitiktinai generuojami vartotojų numeriai tapo ne tokia efektyvi priemonė. Apsaugos programos filtruoja atėjusias trumpąsias žinutes iš numerių nesančių telefonų knygoje. Ateityje tokia plitimo strategija tūrėtų būti neefektyvi dėl tobulėjančių apsaugos programų, kurios vis bando pasivyti KPK plitimo strategijos evoliuciją. Tačiau KPK evoliucija irgi nesustos ir ateityje atsiras naujų genų, kurie pakels plitimo strategijos efektyvumą. Norint išvengti masiškų epidemijos protrūkių reikia iš anksto nuspėti KPK plitimo strategiją.

7. IŠVADOS IR PASIŪLYMAI

Šiame darbe analizuojami genetinių algoritmų veikimo principai, GA panaudojimas įvairiose mokslo srityse. Taip pat pasiūloma taikyti GA funkcionalumą - genų priklausomybė nuo kitų genų.

Išanalizavus apžvelgus modeliavimo įrankius, kuriuose realizuoti genetiniai algoritmai ir atlikus informacijos saugos esybių analizę, pasirenkama kurti modeliavimo įrankį, kurį būtų galima taikyti informacijos saugos rizikos vertinimui ir evoliucijos prognozavimui.

Suformuoti funkciniai ir nefunkciniai reikalavimai, į kuriuos atsižvelgiama realizuojant specializuotą modeliavimo įrankį. Modeliavimo įrankio skaičiavimų tikslumui nustatyti yra naudojami duomenys, kurių rezulatas jau yra žinomas. Atlikus įrankio testavimus, nustatyta, kad įrankis veikia korektiškai.

Analizės metu nustatyta, kad iš informacijos saugos esybių tinkamiausia modeliuoti kenksmingo programinio kodo plitimo strategijos evoliuciją. Šios esybės evoliucijos modeliavimas pasirinktas dėl nuolatinės KPK kaitos, atsiradus naujoms plitimo strategijoms.

Modelaivimo metu buvo atlikti 5 bandymai po 400 iteracijų. Skaičiavimuose mažiausiai iteracijų reikėjo, kai kryžminimo tikimybė buvo lygi 0.3, o mutacijos tikimybė 0.01. Didžiausia KPK plitimo strategijos tinkamumo reikšmė buvo lygi 0,0059, kuri yra 3,7 karto mažesnė, nei prieš tai buvusi geriausia mobiliojo KPK plitimo strategija. Strategijos tinkamumo kritimą leme į rinką atėjusios naujos operacinės sistemos, nes sumažėja tikimybė, kad išmaniajame telefone bus būtent kenksmingam programiniam kodui tinkama operacinė sistema. Taip pat dėl atsiradusių apsaugos programų išmaniesiems telefonams. KPK plitimas per trumpąsias žinutes nėra toks efektyvus kaip seniau, nes žinutės yra filtruojamos, tikrinamas siuntėjo numeris su adresų knygoje esančiais numeriais.

Literatūra

1. WHITLEY, D. A Genetic Algorithm Tutorial. Colorado State University,USA. 37 p.

2. REEVES, R. C.; ROWE, E. J. Genetic Algorithms: Principles and Perspectives. Kluwer A.P., 2002. 327 p.

3. HAUPT, L. R.; HAUPT, S. E. Practical Genetic Algorithms. John Wiley & Sons, 2004. 272 p.

4. Weaver, N.; Paxson, V.; Staniford, S.; Cunningham, R.. A taxonomy of computer worms. Proceedings of the 2003 ACM workshop on Rapid malcode, October 27-27, 2003, Washington, DC, USA.

5. Choy, K. L.; Yfung, W.S. Application of Genetic Algorithms in Optimizing Parameters in a Decision Support System for Resource Allocation.// Annual Journal of HE (HK) 1996-97.

6. WHITLEY, D. A. Genetic Algorithm Tutorial. Colorado State University. USA. 37 p.

7. REEVES, R. C.; ROWE, E. J. Genetic Algorithms: Principles and Perspectives. Kluwer A.P., 2002. 327 p.

8. Goranin, N; Čenys, A. Genetinių algoritmų taikymas kriptografijoje ir kriptoanalizėje.// 10-osios Lietuvos jaunųjų mokslininkų konferencija „Mokslas – Lietuvos ateitis“. Vilnius 2007 m. balandžio 16 d., medžiaga Vilniaus Gedimino technikos universitetas.

9. Getting Started with MATLAB. [žiūrėta 2009-10-18].Prieiga per internetą:www.mathworks.com.

10. SourceForge.[žiūrėta 2009-10-18]. Prieiga per internetą: sourceforge.net/dbimage.php?id=7995.

11. R2010 Documentation. [žiūrėta 2009-10-25]. Prieiga per internetą: http://www.mathworks.com/access/helpdesk/help/toolbox/f9120.html.

12. An Educational Genetic Algorithms Learning Tool. [žiūrėta 2009-10-25]. Prieiga per internetą: http://ewh.ieee.org/soc/es/May2001/14/Begin.htm.

13. [žiūrėta 2010-02-18]. Prieiga per internetą: http://www.karnig.co.uk/ga/ggat.html.

14. GAtree Intro. [žiūrėta 2010-02-18]. Prieiga per internetą: http://www.gatree.com/tutorial.php.

15. Staniford, S; Paxson, V; Weaver, N. How to 0wn the Internet in Your Spare Time. Proceedings of the 11th USENIX Security Symposium, 2002, 149-167.

16. Jarmo, T. Alander 2006. Genetic Algorithms: An Introduction: Introductory lecture for the Ninth Scandinavian Conference on Artificial Intelligence Genetic Algorithms Session

17. Murray, J.D. 2002. Mathematical Biology II: Spatial Models. 3rd Edition, Springer, New York.

19. Serazzi, G; Zanero, S. 2004. Computer Virus Propagation Models. Lecture Notes in Computer Science. – Springer-Verlag,

20. Turner, M.D. 2008. The Logistic Function. [žiūrėta 2010-04-01]. Prieiga per internetą: http://academic.cuesta.edu/mturner/m242/proj2.pdf.

21. Goranin, N. 2010. Genetinių algoritmų taikymas informacijos saugos sistemose. Daktaro disertacija. technologijos mokslai, informatikos inžinerija (07T) ; Vilniaus Gedimino Technikos Universitetas. Vilnius Technika 2010.

22. Silvert, W. 2001. Modelling as a Discipline, in: Int. J. General Systems Vol. 30(3), pp. 261.

23. Android tampa populiariausia išmaniųjų telefonų OS. [žiūrėta 2011-05-02]. Prieiga per internetą: http://www.mobinews.lt/?p=7749.

24. Prieiga per internetą: http://mobilesystem.info/index.php?tp=81350e0ebb536599 [žiūrėta 2011-05-02].

25. S, Mehdi; A, Kumar; M.Farooq. IMAD: In-Execution Malware Analysis and Detection**.** Next Generation Intelligent Networks Research Center. National University of Computer & Emerging Sciences.

26. Goranin, N. Genetinių algoritmų taikymas informacijos saugos sistemose. Daktaro disertacija. technologijos mokslai, informatikos inžinerija (07T) ; Vilniaus Gedimino Technikos Universitetas. Vilnius Technika 2010.

27. Helmer, G; Wong, J; Honavar, V; Miller, L. Feature Selection Using a Genetic Algorithm for Intrusion Detection", In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO, pp. 13-17, Orlando, 1999.

28. Java Package class. Prieiga per internetą: http://jgap.sourceforge.net/javadoc/3.5/ [žiūrėta 2010-10-24].

29. Prieiga per internetą: http://expression-tree.sourceforge.net/ [žiūrėta 2010-10-26]

30. Java Package class. [žiūrėta 2010-09-02]. Prieiga per internetą: http://www.jfree.org/jfreechart/api/javadoc /index.html

31. An Educational Genetic Algorithms Learning Tool. [žiūrėta 2011-05-12]. Prieiga per internetą: www.ewh.ieee.org/soc/es/May2001/14/Begin.htm.

32. Koza, J. R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992.

33. Dokumentation. [žiūrėta 2009-12-10]. Prieiga per internetą: http://www.mathworks.com/help/techdoc/ informacija.

34. Simulink Model-Based and System-Based Design. Using Simulink Version 5, 2003.

35. Matlab The Language of Technical Computing. Getting Started with Matlab Version 7, 2006.

36. Global and Discrete Optimization. Kauno technologijos universitetas, programinės įrangos katedra. [žiūrėta 2010-04-17]. Prieiga per internetą: http://soften.ktu.lt/~mockus.

37. Back, T; Fogel, D; Michalewicz, Z. Evoliucinio skaičiavimo vadovas. Oksfordo universiteto fizikos instituto leidykla, Bristolis, 1997.

38. Dagienė, V; Grigas, G; Jevsikova, T. Enciklopedinis kompiuterijos ţodynas. Anglų–lietuvių kalbų kompiuterijos žodynėlis. [žiūrėta 2011-04-08]. Prieiga per internetą: http://www.likit.lt/en-lt/angl.html.

39. Wilson, K. Data-Driven Design. [žiūrėta 2011-04-05]. Prieiga per internetą: http://www.gamearchitect.net/Articles/DataDrivenDesign.html

40. Gudas, S. Architektūra grindžiamas IS projektavimas. [žiūrėta 2011-04-02]. Prieiga per internetą: https://isd.ktu.lt/ISD/Gudas/DBIS/

41. Vasilecas, O.; Cenys, A.; Sosunovas, S.; Goranin, N. 2008. Informacinių sistemų sauga [Information System Security]. Vilnius. 274 p.

42. Aickelin, U. 1999. Genetic Algorithms for Multiple-Choice Optimisation Problems. [žiūrėta 2009-11-27]. Prieiga per internetą: http://www.eprints.nottingham.ac.uk/306/1/99thesis.pdf.

43. Smartphone statistics and market share. [žiūrėta 2011-05-20]. Prieiga per internetą: http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm.

44. Smartphone statistics. [žiūrėta 2011-05-20]. Prieiga per internetą: http://mobilefanboy.com/category/smartphone-statistics.

45. Juzonis, V.; Goranin, N.; Cenys, A. 2010. Genetic Algorithm Modelling Approach for Mobile Malware Evolution Forecasting, in Proc. of − The 16th International Conference on Information and Software Technologies, Kaunas, 259–264. ISSN 2029-0063.

46. Reikalavimai programinei įrangai. [žiūrėta 2011-03-26]. Prieiga per internetą: http://kopustas.elen.ktu.lt/~rsei/SE4/07.Reikalavimai.ppt

47. Programų sistemų inžinerija. [žiūrėta 2011-03-26]. Prieiga per internetą: http://www.techmat.vgtu.lt/konspekta/PSI/psi07_reikalavimai.pdf

48. Java Class Libraries. [žiūrėta 2010-04-26]. Prieiga per internetą: http://home.cogeco.ca/~ve3ll/jatutor6.htm.

49. Global Optimization Toolbox. [žiūrėta 2010-10-26]. Prieiga per internetą: http://www.mathworks.com/help/toolbox/ gads/ ga.html.

50. *Sundgot, J. First Symbian OS virus to replicate over MMS appears.2005.* [žiūrėta 2010-11-18]. Prieiga per internetą. http://www.infosyncworld.com/ 2005.

51. A Java-based Evolutionary Computation Research System. [žiūrėta 2011-05-20]. Prieiga per internetą: http://cs.gmu.edu/~eclab/projects/ecj/

52. Informacijos sauga. [žiūrėta 2011-05-23]. Prieiga per internetą: http://www.vipt.lt/infosauga/index2.html.

PRIEDAI

A priedas

Straipsnis

Genetic Algorithm Modeling Approach for Mobile Malware Evolution Forecasting

Vaidas Juzonis[1], Nikolaj Goranin[2], Antanas Cenys[3]

[1]VilniusGediminas Technical University, Department of Information System, Sauletekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania, vjuzonis@gmail.com
[2]VilniusGediminas Technical University, Information Technology Security Laboratory, Department of Information System, Sauletekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania, ngrnn@fmf.vgtu.lt
[2]VilniusGediminas Technical University, Information Technology Security Laboratory, Department of Information System, Sauletekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania, ac@fm.vgtu.lt

Abstract. Mobile malware is a relatively new but constantly increasing threat to information security and modern means of communication. Mobile malware evolution speedup is highly expected due to the increase of the SmartPhone and other mobile device market and malware development shift from vandalism to economic aspect. Forecasting evolution tendencies is important for development of countermeasure techniques and prevention of malware epidemic outbreaks. Existing malware propagation models mainly concentrate on malware epidemic consequences modeling, i.e. forecasting the number of infected computers, simulating malware behavior or economic propagation aspects and are based only on current malware propagation strategies or oriented to other malware types. In this article we propose using the genetic algorithm modeling approach for mobile malware evolution forecasting. Genetic algorithm is selected as a modeling tool taking into consideration the efficiency of this method while solving optimization, modeling problems with large solution space and successful application for other malware type evolution forecasting. The model includes the Genetic algorithm description, operating conditions, chromosome that describes mobile malware characteristic and the fitness function for propagation strategy evolution evaluation. Model was implemented and tested on the MATLAB platform.

Keywords: Mobile, malware, genetic, algorithm, model, evolution, forecast.

Introduction

Nowadays malware, i.e. software created with malicious purposes in order to harm the computer software or to be installed on computer without allowance of the legal user [21], is considered to be one of the major threats to information security, information systems and modern communication methods. The number of malware in the wild and rate of malware usage by e-criminals has the tendency to increase making protection against it a crucial task [33]. Significant shift in motivation for malicious activity has taken place over the past several years: from vandalism and recognition in the hacker community, to attacks and intrusions for financial gain. This shift has been marked by a growing sophistication in the tools and methods used to conduct attacks, thereby escalating the network security arms race [2]. Mobile malware is defined as viruses, worms, Trojans or other types that spread on the SmartPhones or other mobile devices running mobile OS. Although it is a relatively new malware type and not very common in the wild yet its portion is highly expected to increase with increase of the smart mobile device market. IDC [29] predicts that 1 billion mobile devices will go online by 2013. Protection against malware on mobile platforms is not very common, compared to traditional computer systems, making them especially attractive for e-criminals. Mobile devices can also provide a variety of services to e-criminals, the traditional systems cannot do: SMS-spam, MMS-spam, call-proxy, etc.

Model is a physical, mathematical or logical representation of system entities, phenomena or processes [6]. Modeling allows forecasting the malware propagation consequences damage [34] and evolution trends [11], understand the behavior of malware, including spreading characteristics [10], understand the factors affecting the malware spread, determine the required effectiveness of countermeasures in order to control the spread and facilitate network designs that are resilient to malware attacks [26], predict the failures of the global network infrastructure [28] and many other tasks that cannot be investigated without harm to production systems in the wild. Existing malware propagation models mainly concentrate on malware epidemic consequences modeling,

simulating malware behavior or economic propagation aspects and are oriented to traditional malware. In this article we propose using the genetic algorithm (GA) modeling approach for mobile malware propagation strategy evolution forecasting which may be also used as a framework for other characteristics evolution forecasting. Genetic algorithm [15] was selected as a modeling tool since it simulates natural selection by means of repeatedly evolving population of solutions and therefore may be used for predicting and modeling possible future propagation strategies. Genetic algorithm modeling has been proved to be effective in many areas such as business decision making, bioinformatics [3], [14], [31], information security [7], [11], [12], [13] and other.

Mobile Malware Evolution and Technical Analysis

According to [17] the first mobile virus to appear was the "*Cabir*" virus which appeared on the 15th of June 2004, infected mobile phones running the "Symbian" OS and used Bluetooth wireless network as a propagation channel. After the successful infection the virus appended the telephone software with its code, activated the Bluetooth and started searching for another Bluetooth device to forward the infected file. Since Bluetooth network coverage is limited to 10 meters the propagation rate of the first mobile virus was rather limited. The first Trojan malware („*Skulls*") also appeared in 2004, November [22],[24]. It infected NOKIA mobile phones, running „Symbian" operating system. „*Skulls*" propagated by pretending to be a software update, usually as „Macromedia Flash" update file with *.sis* extension. When the phone user activated the Trojan it changed the phone configuration settings and depicted the skulls on the screen. It also blocked many functions, such as SMS, MMS, calendar, camera, etc. The phone user could only perform telephone calls. The mobile Trojan evolution continued in 2005. A new Trojan „*Locknut.A*" was detected [16]. Also created for the Symbian platform it was particular in size. The „*patch.sis*" file that contained the infection was only 2KB size, making it the smallest known Trojan for mobile platform.

The first mobile malware that started using propagation methods, other than Bluetooth, was the „*Commwarrior.A*" virus, also running on the „Symbian" platform [8],[32]. It was using much quicker propagation by MMS, since this method does not have limitations by distance, although Bluetooth was also supported. MMS message included text in English, which proposed the phone user a new game, update for antivirus software or similar. The message was sent to all contacts, found in the phone address book. In this case virus authors have relied on the social engineering since when the recipient receives the message from his friend or familiar person the probability of opening it is higher than when it comes from the unknown number. An interesting thing is that Bluetooth was activated during the working hours and MMS were sent in the evening and at night. After each successful infection the virus makes a one minute delay and after that starts searching for a new victim. In 2009 the Kaspersky Labs has discovered a new mobile malware named *„sms.python.flocker",* written in Python language and designed to manipulate the mobile phone accounts. The main malware functionality is dedicated to financial gain. Virus sends SMS messages to the specific number, which allows transferring money from the account of the infected phone to the account of the malware author [17].

Prior and related work

Although it is widely accepted that malware evolution forecasting is an important information security task, the first model-based research paper on this topic appeared only in 2008 [11], which discussed the Internet worm evolution trends. In this article we proposed the general framework for Internet worm evolution (propagation strategy) forecasting. Propagation strategy was selected since it is one of the most descriptive malware characteristics. The Internet worm characteristics representation structure, fitness function and the experiment results were provided. It was shown, that GA may be used for malware characteristic evolution forecasting. The proposed model was tested on existing worms' propagation strategies with known infection probabilities. The tests have proved the effectiveness of the model in evaluating propagation rates and have shown the tendencies of worm evolution. Rather similar concept was proposed almost one year later in [25]. Authors validate the notion of evolution in viruses on a well-known *Bagle* virus family. The results of the proof-of-concept study showed that new viruses–previously unknown of *Bagle* family have successfully evolved starting from a random population. This paper is more malware specific compared to our previous article [11] since its characteristic representation is created for the specific malware type (*Bagle* virus), code-dependent, mainly demonstrates evolution concept and is not specialized for evolution forecasting.

Non-GA models mainly cover malware epidemic consequences modeling, i.e. forecasting the number of infected computers, simulating malware behavior or economic propagation aspects and are based only on current malware propagation strategies or oriented to other malware types. The first epidemiological model to computer virus propagation was proposed by [18]. Epidemiological models abstract from the individuals, and consider them units of a population. Each unit can only belong to a limited number of states. A SIR model assumes the Susceptible-Infected-Recovered state chain and SIS model – the Susceptible-Infected-Susceptible chain. In a technical report [37] described a model of e-mail worm propagation. The authors model the Internet e-mail service as an undirected graph of relationship between people. In order to build a simulation of this graph, they assume that each node degree is distributed on a power-law probability function.

Malware propagation in Gnutella type P2P networks was described in [26] by Ramachandran et al. An analytical model that emulates the mechanics of a decentralized Gnutella type of peer network was formulated and the study of malware spread on such networks was performed. The Random Constant Spread (RCS) model [30] was developed by Staniford et al. using empirical data derived from the outbreak of the CodeRed worm. The model assumes that a machine cannot be compromised multiple times and operates the constant average compromise rate K, which is dependant on worm processor speed, network bandwidth and location of the infected host, etc. The model can predict the number of infected hosts at time t if K is known. As [23] states, that although more complicated models can be derived, most network worms will follow this trend. Other authors [5] propose the AAWP discrete time model, in the hope to better capture the discrete time behavior of a worm. However, according to [28] continuous model is appropriate for large scale models. On the other hand Zanero et al in [28] propose a sophisticated compartment based model, which treats the Internet as the interconnection of autonomous systems, i.e. subnetworks. Interconnections are so-called "bottlenecks". The model assumes, that inside a single autonomous system the worm propagates unhindered, following the RCS model. The authors motivate the necessity of their model via the fact that the network "bottlenecks" may be flooded by malware. Zou et al in [34] propose a two-factor propagation model, which is more precise in modeling the satiation phase taking into consideration the human countermeasures and the decreased scan and infection rate due to the large amount of scan-traffic. The same authors have also published an article on modeling worm propagation under dynamic quarantine defense [36] and evaluated the effectiveness of several existing and perspective worm propagation strategies [35].

Lelarge in [19] introduces an economic approach to malware epidemic modeling (including botnets). Li et al. [20] model botnet-related cybercrimes as a result of profit-maximizing decision-making from the perspectives of both botnet masters and renters/attackers. From this economic model, they derive the effective rental size and the optimal botnet size. Fultz in [9] describes DDoS attacks organized with the help of botnets as economic security games.

The increase of mobile device popularity has called out the appearance of models dedicated to the mobile malware modeling. Ruitenbeek et al. in [27] simulates virus propagation using parameterized stochastic models of a network of mobile phones, created with the help of Mobius tool and provides insight into the relative effectiveness of each response mechanism. Two models of the propagation of mobile phone viruses were designed to study the impact of viruses on the dependability and security of mobile phones: the first model quantifies the propagation of MMS viruses and the second - of Bluetooth viruses. Bulygin in [4] analysis two viruses using different propagation methods (MMS and Bluetooth) in SI (Susceptible->Infected) model.

Evolution forecasting model

General assumptions

Simulation environments serve many purposes, but they are only as good as their content [1]. While designing the model it is necessary to select main factors out of many and reject those that are not important or may cause result distortion. In case of GA modeling the main task consists of three parts: appropriate selection of chromosome structure, which represents the solution, definition of the fitness function and GA operating conditions, such as population size, mutation rates, parent selection, etc.

The model proposed in this article is based on the model previously proposed in [11] with some modifications, adapting it for mobile malware evolution forecasting. Although the proposed model is adapted to propagation strategy evolution forecasting with some modifications (fitness function change) it can be used for other characteristic evolution forecasting. Here we define the propagation strategy as a combination of methods and techniques, used by malware to insure malware population increase. In the current study, we have chosen to model strategies for a theoretical mobile virus, which aims infecting the largest amount of mobile devices during a fixed relatively short period of time.

Experiment conditions

GA consists of initialization, selection and evolution stages. During the initialization stage initial population of strategies is generated. Each strategy is represented as a chromosome. At selection stage strategies are selected through a fitness-based process and in case termination condition is not met evolutionary mechanisms are started. If termination condition is reached, algorithm execution is ended. If not – evolutionary mechanisms are activated.

Initial population is generated on a random basis, i.e. each individual, representing separate strategy is combined of random genes' values. Population size N is equal to 50. Population size remains constant after each new generation. The algorithm would stop producing new generations in case the number of generations have reached 100. Parent selection is random. Mutation operator is activated to each newly generated individual with a 0.05 probability. MATLAB platform was used for model implementation.

Strategy representation

Each strategy is represented as a chromosome (Table 1), which is combined of genes, i.e. combination of techniques and methods. Genes are divided into AA (always active compulsory or activating gene) and AE (active if enabled by AA gene). Such division insures representation flexibility and fixed chromosome length.

Table 1. Chromosome structure.

Gene number / Name / Type / Description / Comments	Value range or sample values	Gene number / Name / Type / Description / Comments	Value range or sample values
1/TRANSF1/AA*/Defines the 1st supported propagation type/Enables NR	MMS	10/OS_PLATF/AA/OS platform affected by malware	Linux; WIN MOBILE; SYMBIAN;…
2/TRANSF2/AA/Defines the 2nd supported propagation type/Enables NR	SMS	11/TEL/AA/Telephone models, affected by malware	NOKIA, SAMSUNG, Apple, RIM;…
3/TRANSF3/AA/Defines the 3rd supported propagation type/Enables BT	Bluetooth	12,13,14/EN_EXPL_N/AA/EXPL_N (N=1-3) activation gene	ON=ExploitRef / OFF
4/TRANSF4/AA/Defines the 4th supported propagation type/Enables EMAIL	e-mail	15,16,17/EXPL_N(N=1-3)/AE/Defines the exploit used for propagation	Random exploit out of suitable exploit array
5/TRANSF5/AA/Defines the 5th supported propagation type/Enables WIFI	Wi-Fi	18/NR_TIME/AA/Defines the NR gene's activity hours	Always; 10:00-20:00; 20:00-10:00
6/NR/AE/Telephone number search or generation module/ Effective if SMS or MMS transfer methods.	Address book; Accepted/ Dialed numbers; Random; …	19/BT_TIME/AA/Defines BT gene's activity	Always; 10:00-20:00; 20:00-10:00
7/BT/AE/Scanner module, that searches for mobile devices with Bluetooth support.	Scan	20/WIFI_TIME/AA/Defines WIFI gene's activity	Always; 10:00-20:00; 20:00-10:00
8/EMAIL/AE/E-mail sending module	Address book; e-mail address DB.	21/EXEC/AA/Defines additional malware functionality/Activates EXEC_CHAN	None; Manage; Update; Manage+Update
9/WIFI/AE/Scanner module, searches for mobile devices with WIFI support.	Scan	22/EXEC_CHAN/AE/Defines malware update channel	e-mail; WI-FI; web-update

Fitness function

From [30] we can say that the propagation strategy efficiency can be evaluated by value K – the number of computers the first malware individual in the wild can infect in a fixed time period. That means that the higher is K, the higher is the fitness of a propagations strategy. Our K calculations by fitness function (Eq.1.) are based on combined statistical and empirical evaluation of time expenditures of strategy's functionality and probabilistic evaluation of strategy's functionality efficiency. Probabilities and time consumption values for activation genes and genes that are not enabled are equal to 0 and may be excluded from calculations.

$$F(S) = k \cdot \left(\begin{array}{l} \left(1-\left(1-p_6\left(NR_TIME\right)\right)\right)\cdot\left(1-p_7\left(BT_TIME\right)\right)\cdot\left(1-p_8\right)\cdot\left(1-p_9\left(WIFI_TIME\right)\right)\cdot p_{10}\cdot p_{11}\cdot \\ \cdot\left(1-\prod_{i=15}^{17}\left(1-p_i\right)\right) \end{array} \right) \quad (1)$$

where: S – evaluated strategy; p_6-p_9 – probability, that exploits will be successfully transferred to the target device (p_6, p_7 and p_9 are time dependant); p_{10} – probability, that the target device will run the supported OS; p_{11} – probability, that device hardware is compatible; p_{15}-p_{17} – probabilities, that exploit will result in infection; k – the number of cycles the virus, using the evaluated strategy, can perform in one second time interval (Eq.2).

$$k = \frac{1}{\sum_{j=1}^{22} t_j} \tag{2}$$

where t_j are time expenditures needed for j^{th} gene functionality. The fitness function can be read as: "The evaluated strategy S can perform k cycles per second. During each cycle the virus, using this strategy, will infect a target host in case at least one of the transfer methods successfully transfers the exploits to the target, the target runs the supported OS on the supported platform and at least one of exploits result in target infection. Compared to our previous model for Internet worms described in [11] limitations for probabilities' size were removed.

Experiment results

The best fitness result achieved during algorithm test was equal to $F(S_d)= 0.023$. Compared to fitness of a sample strategy $F(S_p)=0,017$ of the current mobile malware (Transfer method – MMS only; OS platform - Symbian; Telephone platform - NOKIA; activity hours – Always; Numbers used – Address book; one exploit) fitness of the predicted mobile virus has increased almost 1.687 times. The fitness change during evolution of the best individual is shown on Fig.1., average population fitness change - Fig.2. It should be noticed, that general population fitness also increases in time and that the number of individuals with "better" strategies increase even though the best individual evolution stops after the 42 generation.

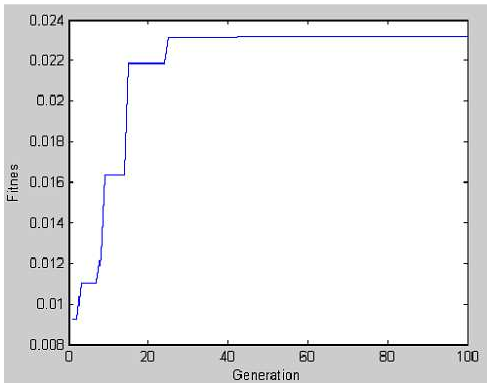


Figure 1. Best strategy fitness change graph

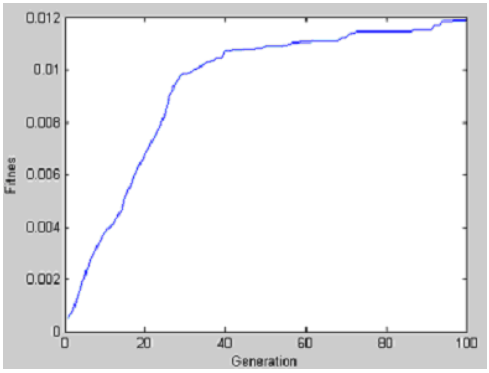


Figure 2. Average population fitness change graph

Compared to the sample strategy the following functionality (genes) was enabled in the best strategy during evolution: Windows mobile support, Wi-Fi transfer method support. We can make an assumption that these methods were included since they provide rather high infection efficiency (additional popular OS and W-Fi with relatively high network coverage). Other potentially efficient methods were not included since their added-value to propagation efficiency was neglected by time consumption, other methods do not result in infection at all (additional functionality) or even minimize the propagation rate (e.g. limitation by hours).

Conclusions

In this article the genetic algorithm modeling approach for mobile malware evolution forecasting was proposed. This is an absolutely new modeling approach for this malware type since it forecasts mobile malware evolution trends compared to traditional models that concentrate on epidemic consequences modeling. Model tests were performed for the mobile malware propagation strategy forecasting.

The proposed model included the Genetic algorithm description, operating conditions, chromosome that describes mobile malware characteristics and the fitness function for propagation strategy evolution evaluation. Model was implemented and tested on the MATLAB platform. The model test results have shown that in case malware creators will intend to optimize the propagation strategy mobile malware evolution will tend to inclusion of additional OS platform and propagation by Wi-Fi networks. The forecasted propagation strategy tends not to be function overloaded due to time consumption increase.

The main model application area is countermeasures planning, since the model predicts the propagation strategy trends. The current study shows that special attention should be paid to wireless security on mobile devices. The model can be also used as a framework (fitness function modification would be needed) for evolution modeling of other mobile malware parameters, such as stealth, functionality or their complexes.

References

[1] **Banks S.B., Stytz M.R.** Challenges Of Modeling BotNets For Military And Security. *Proceeding of SimTecT 2008.* 2008.

[2] **Barford P., Yegneswaran V.** An Inside Look at Botnets. *Advances in Information Security, Springer US.* 2007, volume 27, 171-191.

[3] **Birchenhall C., Kastrinos N., Metcalfe S.** Genetic algorithms in evolutionary modeling. *Journal of Evolutionary Economics.* 1997, volume 7, 375-393.

[4] **Bulygin Y.** Epidemics of Mobile Worms. *Performance, Computing, and Communications Conference, 2007. IPCCC 2007, IEEE International.* 2007, 475-478.

[5] **Chen Z., Gao L., Kwiat K.** Modeling the Spread of Active Worms. *Proceedings of NFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies.*2003, volume 3, 1890-1900.

[6] **Defense Acquisition University.** Systems Engineering Fundamentals: January 2001. *Defense Acquisition University Press.* 2001.

[7] **Faraoun K.M., Boukelif A.** Genetic Programming Approach for Multi-Category Pattern Classification Applied to Network Intrusions Detection. *International Journal of Computational Intelligence.* 2007, volume 3(1), 79-90.

[8] **F-Secure.** Worm:SymbOS/Commwarrior. *F-Secure Corporation, Interactive: http://www.f-secure.com/* 2006.

[9] **Fultz N.** Distributed attacks as security games. *Master thesis, US Berkley School of Information.* 2008.

[10] **Garetto M.W., Towsley G. D.** Modeling Malware Spreading Dynamics. *Proceedings of INFOCOM.* 2003.

[11] **Goranin N., Cenys A.** Genetic Algorithm Based Internet Worm Propagation Strategy Modeling. *Information Technology And Control.* 2008, volume 37, 133-140.

[12] **Goranin N., Cenys A.** Genetic algorithm based Internet worm propagation strategy modeling under pressure of countermeasures. *Journal of Engineering Science and Technology Review.*2009, volume 2, 43-47.

[13] **Goranin N., Cenys A.** Malware Propagation Modeling by the Means of Genetic Algorithms. *Electronics and Electrical Engineering.* 2008, volume 86, 23-26.

[14] **Hill R.R., McIntyre G.A., Narayanan S.** Genetic Algorithms for Model Optimization. *Proceedings of Simulation Technology and Training Conference* (*SimTechT*). 2001.

[15] **Holland J.** Adoption in natural and artificial systems. *The MIT press.* 1975.

[16] **Jarno U**. Disinfection tool for SymbOS/Locknut.A (Gavno.A and Gavno.B). *F-Secure Corporation, Interactive: http://www.f-secure.com/* 2005.

[17] **Kaspersky Lab.** Kaspersky Lab reports. *Interactive: http://www/.kaspersky.com* 2009.

[18] **Kephart J.O., White S.R.** Directed-graph epidemiological models of computer viruses. *Proceedings of IEEE Computer Society Symposium.* 1991, 343-359.

[19] **Lelarge M.** Economics of Malware: Epidemic Risks Model, Network Externalities and Incentives. *Proceedings of Fifth biannual Conference on The Economics of the Software and Internet Industries.* 2009.

[20] **Li Z., Liao Q., Striegel A.** BotnetEconomics: Uncertainty Matters. Managing Information Risk and the Economics of Security, Springer US. 2009, 1-23.

[21] **Monga R.** MASFMMS: Multi Agent Systems Framework for Malware Modeling and Simulation. *Lecture Notes in Computer Science, Springer Berlin / Heidelberg.* 2009, volume 5269/2009, 97-109.

[22] **Naraine R.** Cell Phone Security: New Skulls Mutant Comes with Virus Extras, *Interactive: http://www.eweek.com/* 2004.

[23] **Nazario J.** Defense and Detection Strategies against Internet Worms. *Artech House Publishers.* 2003.

[24] **Niemela J.** F-Secure Virus Descriptions : Skulls.D. *F-Secure Corporation, Interactive: http://www.f-secure.com* 2005.

[25] **Noreen S., Murtaza S., Shafiq M.Z., Farooq M.** Evolvable malware. *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM.* 2009, 1569-1576.

[26] **Ramachandran K., Sikdar B.** Modeling malware propagation in Gnutella type peer-to-peer networks. *Proceedings of the Parallel and Distributed Processing Symposium, IPDPS.* 2006, volume 20, 8 pp.

[27] **Ruitenbeek E.V., Courtney T., Sanders W.H., Stevens F.** Quantifying the Effectiveness of Mobile Phone Virus Response Mechanisms. *IEEE/IFIP International Conference on Dependable Systems and Networks.*2007, 790-800.

[28] **Serazzi G., Zanero S.** Computer Virus Propagation Models. *Lecture Notes in Computer Science, Springer-Verlag.* 2004, 26–50.

[29] **Shah A.** IDC: 1 Billion Mobile Devices Will Go Online by 2013. *IDG News Service, Interactive:http://www.pcworld.com/* 2009.

[30] **Staniford S., Paxson V., Weaver N.** How to 0wn the Internet in Your Spare Time. *Proceedings of the 11th USENIX Security Symposium, USENIX Association.* 2002, 149-167.

[31] **Stender J., Hillebrand E., Kingdon J.** Genetic Algorithms in Optimization, Simulation and modeling. *IOS Press.* 1994.

[32] **Sundgot J.** First Symbian OS virus to replicate over MMS appears.2005. *Interactive:* http://www.infosyncworld.com/ 2005.

[33] **Turner D.** Symantec Global Internet Security Threat Report. *Symantec Corporation*. 2008.

[34] **Zou C.C., Gong W., Towsley D.** Code Red Worm Propagation Modeling and Analysis. *CCS '02: Proceedings of the 9th ACM Conference on Computer and communications security, ACM*. 2002, 138-147.

[35] **Zou C.C., Gong W., Towsley D.** On the performance of Internet worm scanning strategies // *Performance Evaluation, Elsevier Science Publishers B. V.* 2005, volume 63, 700–723.

[36] **Zou C.C., Gong W., Towsley D.** Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense. *WORM '03: Proceedings of the 2003 ACM workshop on Rapid malcode, ACM.* 2003, 51-60.

[37] **Zou C.C., Towsley D., Gong W.** Email Virus Propagation Modeling and Analysis. *Technical report TRCSE-03-04, University of Massachusetts.* 2004.

B priedas

```
/*
 * GeneticAlgView.java
 */
package geneticalg;
//-------- failu nuskaitymui – irasymui----
import java.io.*;
//---------Jgap-----
import org.jgap.*;
//----------------------interfeisui--------
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import
org.jdesktop.application.SingleFrameApplication
;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import javax.swing.*;
//-----------matematikai + tinkamumo funkcijai
skaiciuoti (koreguoti)-------
import geneticalg.com.graphbuilder.math.*;
import java.lang.Math.*;
import java.util.*;
import java.awt.image.*;
//---------grafiko piesmui-------
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
//---------------------------------
/**
 * The application's main frame.
 */
public class GeneticAlgView extends
FrameView {

    public
GeneticAlgView(SingleFrameApplication app) {
        super(app);

        initComponents();
        Numeracija();
        // status bar initialization - message timeout,
idle icon and busy animation, etc
        ResourceMap resourceMap =
getResourceMap();
        int messageTimeout =
resourceMap.getInteger("StatusBar.messageTim
eout");
        messageTimer = new
Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent
e) {
                statusMessageLabel.setText("");


            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate =
resourceMap.getInteger("StatusBar.busyAnimati
onRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] =
resourceMap.getIcon("StatusBar.busyIcons[" + i
+ "]");
        }
        busyIconTimer = new
Timer(busyAnimationRate, new
ActionListener() {
            public void actionPerformed(ActionEvent
e) {
                busyIconIndex = (busyIconIndex + 1)
% busyIcons.length;
```

```
statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);

        }
    });
    idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
    statusAnimationLabel.setIcon(idleIcon);
    progressBar.setVisible(false);

    // connecting action tasks to status bar via TaskMonitor
    TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());

taskMonitor.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
        public void propertyChange(java.beans.PropertyChangeEvent evt) {
            String propertyName = evt.getPropertyName();

            if ("started".equals(propertyName)) {
                if (!busyIconTimer.isRunning()) {

statusAnimationLabel.setIcon(busyIcons[0]);
                    busyIconIndex = 0;
                    busyIconTimer.start();
                }
                progressBar.setVisible(true);
                progressBar.setIndeterminate(true);
            } else if ("done".equals(propertyName)) {
                busyIconTimer.stop();

statusAnimationLabel.setIcon(idleIcon);
                progressBar.setVisible(false);
                progressBar.setValue(0);
            } else if ("message".equals(propertyName)) {
                String text = (String)(evt.getNewValue());
                statusMessageLabel.setText((text == null) ? "" : text);
                messageTimer.restart();
            } else if ("progress".equals(propertyName)) {
                int value = (Integer)(evt.getNewValue());
                progressBar.setVisible(true);
                progressBar.setIndeterminate(false);
                progressBar.setValue(value);
            }
        }
    });
  }
  @Action
  public void showAboutBox() {
    if (aboutBox == null) {
      JFrame mainFrame = GeneticAlgApp.getApplication().getMainFrame();
      aboutBox = new GeneticAlgAboutBox(mainFrame);

outBox.setLocationRelativeTo(mainFrame);
    }

eneticAlgApp.getApplication().show(aboutBox);
  }
  public void Numeracija() //numeracija, uzpildo ID lauka chromosomu numeriais
  {
  int i = 0;
  while (i < IndividuSkaicius)
    {
    this.lentele.setValueAt(i+1, i, 0);
    this.lentele2.setValueAt(i+1, i, 0);
    this.lentele3.setValueAt(i+1, i, 0);
```

```
      i++;
      }
   }

   public void Mutacija(double a[], double b[], double c[], int destination) // gauna dvi chromosomas (p ir t), mutuoja jas ir grazina mutanta
   {
      Random generatorius = new Random();
      int i = 1;
      int j = 0;
      int k = 0;
      double x = 0;
      double y = 0;
      double z = 0;
      double q = 0;
      x = generatorius.nextFloat(); //sukuriame randomini skaiciu nuo 0 iki 1
      i = 1;
      if (x < MutacijosTikimybe) //jei sukurtas skaicius mazesnis uz tikimybe ivyksta mutacija
      {
         i = generatorius.nextInt(GenuSkaicius+1); //atsitiktinai randame mutuojanti gena
         y = 0;
         while ((y == 0.0)&(k < 20)) //randame netuscia geno verte is pradiniu ivestu duomenu & apsauga nuo uzsiciklinimo
         {
            j = generatorius.nextInt(10); //randame eilutes nr
            y = eData[j][i];
            z = eData2[j][i];
            q = eData3[j][i];
            k++;
         }
         if (k > 20)// jei ivyko uzsiciklinimas imama pirmoji mutojancio geno variantu reiksme is duomenu failo//sansas kad bus panaudota sekanti kodo dalis 1/(200*(kiek_yra_mutojancio_geno_variantu))
         {
            j = 0;
            y = eData[j][i];
            z = eData2[j][i];
            q = eData3[j][i];
         }
         a[i] = y;  //irasome reikiamus rezultatus
         b[i] = z;
         eVardai[destination][i] = eNames[j][i];
      }
   }

   public void MutuotiVisus() // atlieka masini lenteles mutavima, galima nustatyti kiek geriausiu individu nemutuos (Naudoja metodus Mutacija(), Isimti Chromosoma(), Ideti Chromosoma())
   {
      int i = 2; //nemutuojame 2 geriausiu
      double[] masyvas;
      double[] masyvas2;
      double[] masyvas3;
      masyvas = new double[GenuSkaicius + 1];
      masyvas2 = new double[GenuSkaicius + 1];
      masyvas3 = new double[GenuSkaicius + 1];
      while (i < IndividuSkaicius)
      {
         IsimtiChromosoma (masyvas,masyvas2,masyvas3, i);
         Mutacija (masyvas, masyvas2, masyvas3,i);
         IdetiChromosoma (masyvas,masyvas2,masyvas3, i);
         i++;
      }
   }
   //atlieka randomini dvieju pn chromosomu kryzminima, rezultatas 3. Sinchroniskai kryzmina tn chromosoma. Nr1 - 1 tevo adresas.
```

```
Nr2 - 2 tevo adresas. Destination - kryzminamo geno adresas.
   public void Kryzminimas(double a[], double b[], double c[], int Nr1, int Nr2, int destination, double a2[], double b2[], double c2[], double a3[], double b3[], double c3[])
   {
      Random generatorius = new Random();
      int i = 1;
      double x = 0;
      while (i < GenuSkaicius + 1)  //praeiname paeiliui per visus individo genus
      {
         x = generatorius.nextFloat(); //generuojame skaiciu nuo 0 iki 1
         if (x < 0.5)  //jei gautas skaicius maziau nei puse genas igauna pirmojo tevo reiksmes
         {
            c[i] = a[i];
            c2[i] = a2[i];
            c3[i] = a3[i];

Vardai[destination][i]=eVardai[Nr1][i];
         }
         else  //jei gautas skaicius daugiau nei puse genas igauna antrojo tevo reiksmes
         {
            c[i] = b[i];
            c2[i] = b2[i];
            c3[i] = b3[i];

eVardai[destination][i]=eVardai[Nr2][i];
         }
         i++;
      }
   }

   public void KryzmintiVaikus() // funkcija pakeicia antros puses chromosomas atsitiktiniais pirmos puses chromosomu kryzminimais (Naudoja metodus Kryzminimas(), Isimti Chromosoma(), Ideti Chromosoma());
   {
      Random generatorius = new Random();
      double x = 0;
      int i = 2;//nekryzminame 2 geriausiu chromosomu
      int i1 = 0;
      int i2 = 0;
      double a[];
      double a2[];
      double a3[];
      double b[];
      double b2[];
      double b3[];
      double c[];
      double c2[];
      double c3[];
      a = new double[GenuSkaicius + 1];
      a2 = new double[GenuSkaicius + 1];
      a3 = new double[GenuSkaicius + 1];
      b = new double[GenuSkaicius + 1];
      b2 = new double[GenuSkaicius + 1];
      b3 = new double[GenuSkaicius + 1];
      c = new double[GenuSkaicius + 1];
      c2 = new double[GenuSkaicius + 1];
      c3 = new double[GenuSkaicius + 1];
      while (i < IndividuSkaicius)
      {
         x = generatorius.nextFloat();
         if (x < KryzminimoTikimybe)
         {
            i1 = generatorius.nextInt(IndividuSkaicius/2);
            i2 = generatorius.nextInt(IndividuSkaicius/2);
            IsimtiChromosoma(a,a2,a3,i1); //isimame pi ir ti is vietos i1
            IsimtiChromosoma(b,b2,b3,i2); //isimame antra pi ir ti is vietos i2
```

```
Kryzminimas(a,b,c,i1,i2,i,a2,b2,c2,a3,b3,c3);//su kryzminame pi pirmo su pi antro  ir t1 su ti
            IdetiChromosoma(c,c2,c3,i);
//idedame kryziuka i vieta
          }
          i++;
       }
    }
    public void TinkamumoFunkcija(double a[], double b[], double c[]) //atlieka tinkamumo skaiciavimus visai //lentelei. Palaiko genu neaktyvuma.
    {
       int i;
       boolean f = false;
       double z = 99.99;
       Neaktyvumas = 0;
       for (i = GenuSkaicius; i < 33; i++)
       {
          a[i]=0;
          b[i]=0;
          c[i]=0;
       }
       String s = Lygtis.getText();
       funk = s.toCharArray();
       for (i = 0; i < GenuSkaicius; i++)
       {
          if (a[i] == z)
          {
             if (f == false) NebutinasGenas1(s, i+1);
             //System.out.print(funk);
             //System.out.println(Neaktyvumas);
             char funk2[] = new char[s.length()-Neaktyvumas];
             for (i = 0; i < (s.length()-Neaktyvumas); i++)
             {
                funk2[i] = funk[i];
             }
             s = new String(funk2);
             f = true;
          }
       }
       Expression x = ExpressionTree.parse(s);
       VarMap vm = new VarMap(true);
……..
       FuncMap fm = new FuncMap();
       fm.loadDefaultFunctions();
       a[GenuSkaicius] = x.eval(vm, fm);
    }

    public void ConvertToJgap (double a[], double b[], double c[], Chromosome chrom1, Chromosome chrom2, Chromosome chrom3)
    {
       int i;
       //apnuliname 3 chromosomas ir gena
       Gene gene = null;
       chrom1 = null;
       chrom2 = null;
       chrom3 = null;
       for (i = 1; i < GenuSkaicius-1; i++) //praeiname per visus individo laukus ir perkeliame ju reiksmes i jgap duomenu struktura
       {
          gene.setEnergy(a[i]);
          chrom1.setGene(i, gene);
          gene.setEnergy(b[i]);
          chrom2.setGene(i, gene);
          gene.setEnergy(c[i]);
          chrom3.setGene(i, gene);
       }
    }

    public void TinkamumoTikrinimas() //ima is abieju lenteliu paeiliui chromosomas ir tikrina jas su tinkamumo funkcija. Iraso tinkamumo verte i lentele. (Naudojasi metodu TinkamumaFunkcija())
    {
```

```
    {
        int i = 0;
        int j = 1;
        double[] masyvas;
        double[] masyvas2;
        double[] masyvas3;
        masyvas = new double[33];
        masyvas2 = new double[33];
        masyvas3 = new double[33];

        while (i < IndividuSkaicius)
        {
            j = 1;
            while (j < GenuSkaicius + 1)
            {
                masyvas[j-1] =
Double.parseDouble(String.valueOf(this.lentele.
getValueAt(i,j)));
                masyvas2[j-1] =
Double.parseDouble(String.valueOf(this.lentele2
.getValueAt(i,j)));
                masyvas3[j-1] =
Double.parseDouble(String.valueOf(this.lentele3
.getValueAt(i,j)));
                j++;
            }
            TinkamumoFunkcija(masyvas,
masyvas2, masyvas3);

this.lentele.setValueAt(masyvas[GenuSkaicius],i
,GenuSkaicius + 1);
            i++;
        }
    }
  }

  public void Rusiavimas() //atlieka rusiavima
visoje lenteleje pagal tinkamuma mazejimo
tvarka
  {
    double[][] masyvas;
    double[][] masyvas2;
    double[][] masyvas3;
    String masyvas4[][];
    masyvas4 = new String[100][100];
    int i = 0;
    int j = 0;
    int N = 0;
    double max = 0;
    masyvas = new
double[IndividuSkaicius][GenuSkaicius + 2];
    masyvas2 = new
double[IndividuSkaicius][GenuSkaicius + 2];
    masyvas3 = new
double[IndividuSkaicius][GenuSkaicius + 2];

    while (i < IndividuSkaicius)
    {
      j = 0;
      while (j < GenuSkaicius + 2)
      {
        masyvas[i][j] =
Double.parseDouble(String.valueOf(this.lentele.
getValueAt(i,j)));
        if (j < GenuSkaicius + 1)
        {
          masyvas2[i][j] =
Double.parseDouble(String.valueOf(this.lentele2
.getValueAt(i,j)));
          masyvas3[i][j] =
Double.parseDouble(String.valueOf(this.lentele3
.getValueAt(i,j)));
          masyvas4[i][j] = eVardai[i][j];
        }
        j++;
      }
      i++;
    }
    i = 0;
    while (i < IndividuSkaicius)
    {
      j = 0;
```

```
        max = -999999999;
        while (j < IndividuSkaicius)
        {
          if (max < masyvas[j][GenuSkaicius +
1])
          {
            max = masyvas[j][GenuSkaicius +
1];
            N = j;
          }
        j++;
        }
        masyvas[N][GenuSkaicius + 1] = -
999999999;
        j = 0;
        while (j < GenuSkaicius + 1)
        {

this.lentele.setValueAt(masyvas[N][j],i,j);

this.lentele2.setValueAt(masyvas2[N][j],i,j);

this.lentele3.setValueAt(masyvas3[N][j],i,j);
            eVardai[i][j] = masyvas4[N][j];
          j++;
        }

        max = -999999999;
        j = 0;
      i++;
      }
      TinkamumoTikrinimas();
   }

   public void IdetiChromosoma (double a[],
double b[], double d[], int c) //funkcija iveda
masyve esancia chromosoma i lentele nurodytoje
pozicijoje
   {
   int j = 1;
   //while (j < GenuSkaicius + 1)
   while (j < GenuSkaicius + 1)
   {
     this.lentele.setValueAt(a[j],c,j);
     this.lentele2.setValueAt(b[j],c,j);
     this.lentele3.setValueAt(d[j],c,j);
     j++;
   }
   }
   public void IsimtiChromosoma (double a[],
double b[], double d[], int c) //funkcija iveda i
masyva chromosoma is lenteles
   {
   int j = 1;
   while (j < GenuSkaicius + 1)
   {

     a[j]
=
(Double.parseDouble(String.valueOf(this.lentele.
getValueAt(c,j))));
     b[j]
=
(Double.parseDouble(String.valueOf(this.lentele
2.getValueAt(c,j))));
     d[j]
=
(Double.parseDouble(String.valueOf(this.lentele
3.getValueAt(c,j))));
     j++;
   }
   }

   public void PilnaIteracija()//atliekame pilna
iteracija (Kryzminimas -> Mutacija ->
TinkamumoTikrinimas -> Rusiavimas ->
Numeracija -> Grafiko Atnaujinimas)
   {
     KryzmintiVaikus();
     MutuotiVisus();
     TinkamumoTikrinimas();
     Rusiavimas();
     Numeracija();
     TinkamumoTikrinimas();
```

```
        double b = (Double.parseDouble(String.valueOf(this.lentele.getValueAt(0,GenuSkaicius+1))));
        Piesimas(b);
    }

    public void NepilnaIteracija()//atliekame pilna iteracija (Kryzminimas -> Mutacija -> TinkamumoTikrinimas -> Rusiavimas -> Numeracija -> Grafiko Atnaujinimas)
    {
        KryzmintiVaikus();
        MutuotiVisus();
        TinkamumoTikrinimas();
        Rusiavimas();
        Numeracija();
        TinkamumoTikrinimas();
        double b = (Double.parseDouble(String.valueOf(this.lentele.getValueAt(0,GenuSkaicius+1))));
        Piesimas2(b);
    }

    public void GeneruotiIndividus() //Generuoja IndividuSkaicius chromosomu is duomenu failo
    {
        double a[][];
        a = new double [100][100];
        double b[][];
        b = new double [100][100];
        double c[][];
        c = new double [100][100];
        int i;
        int j;
        int k;
        int N = 0;
        for (j =1;j < GenuSkaicius+1; j++) //stulpelio nr
        {
            N = 0;
            for (i = 0; i < 100; i++)
            {
                if (this.lentele.getValueAt(i,j) != null) //eilutes nr;
                {
                    a[i][j] = Double.parseDouble(String.valueOf(this.lentele.getValueAt(i,j)));
                    N++;
                }
                else i = 100;
            }
            i = 0;
            k = 0;
            while (k < IndividuSkaicius)
            {
                if (i > N-1)
                {
                    i = 0;
                }
                this.lentele.setValueAt(a[i][j],k,j);
                eVardai[k][j] = eVardai[i][j];
                i++;
                k++;
            }
        }
        //antra lentele
        N = 0;
        for (j =1;j < GenuSkaicius+1; j++) //stulpelio nr
        {
            N = 0;
            for (i = 0; i < 100; i++)
            {
                if (this.lentele2.getValueAt(i,j) != null) //eilutes nr;
                {
                    b[i][j] = Double.parseDouble(String.valueOf(this.lentele2.getValueAt(i,j)));
                    N++;
                }
```

```
            else i = 100;
        }
        i = 0;
        k = 0;
        while (k < IndividuSkaicius)
        {
            if (i > N-1)
            {
                i = 0;
            }
            this.lentele2.setValueAt(b[i][j],k,j);
            i++;
            k++;
        }
    }
    //end of antra lentele
    //trečia lentele
    N = 0;
    for (j =1;j < GenuSkaicius+1; j++)
//stulpelio nr
    {
        N = 0;
        for (i = 0; i < 100; i++)
        {
            if (this.lentele3.getValueAt(i,j) != null)
//eilutes nr;
            {
                c[i][j] =
Double.parseDouble(String.valueOf(this.lentele3
.getValueAt(i,j)));
                N++;
            }
            else i = 100;
        }
        i = 0;
        k = 0;
        while (k < IndividuSkaicius)
        {
            if (i > N-1)
            {
                i = 0;
            }
            this.lentele3.setValueAt(c[i][j],k,j);
            i++;
            k++;
        }
    }
    //end of trečia lentele
}

public void Piesimas(double i) //Prideda i
grafika nauja reiksme ir atnaujina grafika
{
    xy.add(iteracija, i);
    iteracija++;
    XYSeriesCollection dataset = new
XYSeriesCollection();
    dataset.addSeries(xy);
    JFreeChart chart =
ChartFactory.createXYLineChart(null,
"Iteracija", "Tinkamumas", dataset,
PlotOrientation.VERTICAL, false, false, false);
    BufferedImage image =
chart.createBufferedImage(400,300);
    grafikoLabel.setIcon(new
ImageIcon(image));
    grafikoLabel.show();
    grafikoLabel.setVisible(true);
}
public void Piesimas2(double i) //Prideda i
grafika nauja reiksme, neatnaujina grafiko
{
    xy.add(iteracija, i);
    iteracija++;
    XYSeriesCollection dataset = new
XYSeriesCollection();
    dataset.addSeries(xy);
    //JFreeChart chart =
ChartFactory.createXYLineChart(null,
"Iteracija", "Tinkamumas", dataset,
PlotOrientation.VERTICAL, false, false, false);
```

```
      //BufferedImage image =
chart.createBufferedImage(400,300);
      //grafikoLabel.setIcon(new
ImageIcon(image));
      //grafikoLabel.show();
      //grafikoLabel.setVisible(true);
   }
   public void ParodytiVardus() // Parodo
lenteleje genu pavadinimus
   {
      int i = 0;
      int j = 0;
      String zodis;
      while (i < IndividuSkaicius)
      {
                while (j < GenuSkaicius)
         {
            j++;
            zodis =
String.valueOf(this.lentele.getValueAt(i,j));
            if (zodis!= null && zodis.length() != 0)
            {
               eBuferis[i][j]=zodis;

this.lentele.setValueAt(eVardai[i][j],i,j);
               eVardai[i][j]=eBuferis[i][j];
            }

         }
         i++;
         j = 0;
      }

      if (Iteracija.isVisible() == true)
      {
         Iteracija.setVisible(false);
      }
      else Iteracija.setVisible(true);

      if (Multiiteracija.isVisible() == true)
      {
         Multiiteracija.setVisible(false);
      }
      else Multiiteracija.setVisible(true);
      if (Save.isVisible() == true)
      {
         Save.setVisible(false);
      }
      else Save.setVisible(true);
      zodis = showNames.getText();
      if ("Rodyti vardus".equals(zodis))
      {
         showNames.setText("Slepti vardus");
      }
      else showNames.setText("Rodyti vardus");
   }
   public void LoadData () //Apdoroja duomenu
failus: Isima duomenys is failu ir uzpikldo jais
duomenu strukturas
   {

PervardintiLenteles(stulpeliuVardas1.getText(),
stulpeliuVardas2.getText(),
stulpeliuVardas3.getText());
      String aLine ;
      String[] words;
      words = new String[100];
      int i, j;
      try
      {
         FileInputStream File = new
FileInputStream(Failas1.getText());
         BufferedReader br = new
BufferedReader(new InputStreamReader(File));
         j = 0;
         while ((aLine = br.readLine()) != null)
         {
            words = aLine.split("\\;");
            for (i = 1; i < GenuSkaicius+1; i++)
            {
               if (words[i]!= null &&
words[i].length() != 0)
```

```
            {

this.lentele.setValueAt(Double.parseDouble(words[i]), j, i);
                eData[j][i] = Double.parseDouble(words[i]);
            }
          }
          j++;
        }
        br.close();
      }
      catch (FileNotFoundException e1){jLabel18.setText("Pirmas duomenų failas nerastas"); jLabel18.setVisible(true);}
      catch (Exception e){}


      try
      {
        //FileInputStream File2 = new FileInputStream("C://Data2.csv");
        FileInputStream File2 = new FileInputStream(Failas2.getText());
        BufferedReader br2 = new BufferedReader(new InputStreamReader(File2));

        j = 0;
        while ((aLine = br2.readLine()) != null)
        {
          words = aLine.split("\\;");
          for (i = 1; i < GenuSkaicius+1; i++)
          {
            if (words[i]!= null && words[i].length() != 0)
            {

this.lentele2.setValueAt(Double.parseDouble(words[i]), j, i);
                eData2[j][i] = Double.parseDouble(words[i]);
            }
          }
          j++; //eilutes nr
        }
        br2.close();
      }
      catch (FileNotFoundException e1){jLabel18.setText("Antras duomenų failas nerastas"); jLabel18.setVisible(true);}
      catch (Exception e){ }
      try
      {
        //FileInputStream File2 = new FileInputStream("C://Data2.csv");
        FileInputStream File4 = new FileInputStream(Failas4.getText());
        BufferedReader br4 = new BufferedReader(new InputStreamReader(File4));
        j = 0;
        while ((aLine = br4.readLine()) != null)
        {
          words = aLine.split("\\;");
          for (i = 1; i < GenuSkaicius+1; i++)
          {
            if (words[i]!= null && words[i].length() != 0)
            {

this.lentele3.setValueAt(Double.parseDouble(words[i]), j, i);
                eData3[j][i] = Double.parseDouble(words[i]);
            }
          }
          j++; //eilutes nr
        }
        br4.close();
      }
      catch (FileNotFoundException e1){jLabel18.setText("");
jLabel18.setVisible(true);}
```

```
      catch (Exception e){ }
      try
      {
        //FileInputStream File3 = new
FileInputStream("C://Names.csv");

        FileInputStream File3 = new
FileInputStream(Failas3.getText());
        BufferedReader br3 = new
BufferedReader(new InputStreamReader(File3));
        j = 0;
        while ((aLine = br3.readLine()) != null)
        {
          words = aLine.split("\\;");
          for (i = 1; i < GenuSkaicius+1; i++)
          {
            if (words[i]!= null &&
words[i].length() != 0)
              {
                eVardai[j][i] = words[i];
                eNames[j][i] = words[i];
              }
          }
          j++; //eilutes nr
        }
        br3.close();
      }
      catch (FileNotFoundException
e1){jLabel18.setText("Vardų failas nerastas");
jLabel18.setVisible(true);}
      catch (Exception e){}
    }

    public void PervardintiLenteles (String a,
String b, String c)//Funkcija generuoja stulpeliu
pavadinimus is vartotojo ivestu reiksmiu
    {
      if (a.length() == 0) {a = "p";}
      if (b.length() == 0) {b = "t";}
      if (c.length() == 0) {c = "s";}
      int i;
      for (i=1; i < GenuSkaicius+1; i++)
      {

lentele.getColumnModel().getColumn(i).setHead
erValue(a+i);

lentele2.getColumnModel().getColumn(i).setHea
derValue(b+i);

lentele3.getColumnModel().getColumn(i).setHea
derValue(c+i);
      }
      for (i=GenuSkaicius+1; i < 33; i++)
      {

lentele.getColumnModel().getColumn(i).setHead
erValue(" ");

lentele2.getColumnModel().getColumn(i).setHea
derValue(" ");

lentele3.getColumnModel().getColumn(i).setHea
derValue(" ");
      }
      lentele2.updateUI();
      lentele3.updateUI();
    }
    public void NebutinasGenas1 (String lygtis,
int genas) //Pasalina is tinkamumo funkcijos
neaktyvuji gena. lygtis - tinkamumo funkcija,
genas - geno eiles numeris.
    {
      int ilgis = lygtis.length();
      char funk2[] = new char[ilgis];
      char c = Character.forDigit(genas, 10);
      int i = 0;
      int k = 0;
      int j = 0;
      for (i = 0; i < ilgis; i++)
      {
        if (funk[i] == 'p')
```

```
        {
            if (funk[i+1] == c)
            {
                k = i;
                if ((i+2 == ilgis)||(funk[i+2] == ')'))
                {
                    j = 3;
                }
                else
                {
                    j = 2;
                }
            }
        }
    }
    if (j > 0)
    {
        for (i = 0; i < (k+2-j); i++)
        {
            funk2[i] = funk[i];
        }
        for (i = (k+2-j); (i < (ilgis-3)); i++)
        {
            funk2[i] = funk[i+3];
        }
        Neaktyvumas = Neaktyvumas +3;
    }
    else
    {
        funk2 = funk;
    }
    funk = funk2;
    NebutinasGenas2(genas, ilgis);
  }

  public void NebutinasGenas2 (int genas, int ilgis)  //Antra neaktyvaus geno salinimo funkcijos dalis.
  {
    char funk2[] = new char[ilgis];
    char c = Character.forDigit(genas, 10);
    int i = 0;
    int k = 0;
    int j = 0;
    for (i = 0; i < (ilgis-3); i++)
    {
      if (funk[i] == 't')
      {
        if (funk[i+1] == c)
        {
          k = i;
          if ((i+2 == (ilgis-3))||(funk[i+2] == ')'))
          {
            j = 3;
          }
          else
          {
            j = 2;
          }
        }
      }
    }
    if (j > 0)
    {
      for (i = 0; i < (k+2-j); i++)
      {
        funk2[i] = funk[i];
      }
      for (i = (k+2-j); (i < (ilgis-6)); i++)
      {
        funk2[i] = funk[i+3];
      }
      Neaktyvumas = Neaktyvumas +3;
    }
    else
    {
      funk2 = funk;
    }
    funk = funk2;
  }
```

```
    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
    private void initComponents() {
        bindingGroup = new org.jdesktop.beansbinding.BindingGroup();

        mainPanel = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        lentele = new javax.swing.JTable();
        grafikoLabel = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        lentele2 = new javax.swing.JTable();
        jPanel1 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        p3 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        p2 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        p1 = new javax.swing.JTextField();
        j1 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        KeistiTikimybe = new javax.swing.JButton();
        Multiiteracija = new javax.swing.JButton();
        Iteracija = new javax.swing.JButton();
        Ivedimas = new javax.swing.JButton();
        iteracijuSkaicius = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        showNames = new javax.swing.JButton();
        Save = new javax.swing.JButton();
        Rezultatas = new javax.swing.JTextField();
        jPanel3 = new javax.swing.JPanel();
        Lygtis = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        .....
        stulpeliuVardas1 = new javax.swing.JTextField();
        jLabel17 = new javax.swing.JLabel();
        stulpeliuVardas2 = new javax.swing.JTextField();
        Failas4 = new javax.swing.JTextField();
        jLabel19 = new javax.swing.JLabel();
        stulpeliuVardas3 = new javax.swing.JTextField();
        jLabel20 = new javax.swing.JLabel();
        Find1 = new javax.swing.JButton();
        Find2 = new javax.swing.JButton();
        Find3 = new javax.swing.JButton();
        Find4 = new javax.swing.JButton();
        jScrollPane3 = new javax.swing.JScrollPane();
        lentele3 = new javax.swing.JTable();
        menuBar = new javax.swing.JMenuBar();
        javax.swing.JMenu fileMenu = new javax.swing.JMenu();
        javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
        javax.swing.JMenu helpMenu = new javax.swing.JMenu();
        javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
        statusPanel = new javax.swing.JPanel();
        javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
        statusMessageLabel = new javax.swing.JLabel();
        statusAnimationLabel = new javax.swing.JLabel();
        progressBar = new javax.swing.JProgressBar();

        mainPanel.setName("mainPanel"); // NOI18N
```

```
        jScrollPane1.setName("jScrollPane1");      // NOI18N

        lentele.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                ……….
            },
            new String [] {
                "ID", "p1", "p2", "p3", "p4", "p5", "p6", "p7", "p8", "p9", "p10", "p11", "p12", "p13", "p14", "p15", "p16", "p17", "p18", "p19", "p20", "p21", "p22", "p23", "p24", "p25", "p26", "p27", "p28", "p29", "p30", "p31", "p32"
            }
        ) {
            Class[] types = new Class [] {
……..
        grafikoLabel.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        grafikoLabel.setName("grafikoLabel");      // NOI18N

        jScrollPane2.setName("jScrollPane2");      // NOI18N

        lentele2.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
……..
            },
            new String [] {
                "ID", "t1", "t2", "t3", "t4", "t5", "t6", "t7", "t8", "t9", "t10", "t11", "t12", "t13", "t14", "t15", "t16", "t17", "t18", "t19", "t20", "t21", "t22", "t23", "t24", "t25", "t26", "t27", "t28", "t29", "t30", "t31", "t32"
            }
        ) {
            Class[] types = new Class [] {
……..
        );

        Failas1.setText("C://Data.csv");
        Failas2.setText("C://Data2.csv");
        Failas3.setText("C://Names.csv");
        jLabel13.setText("Pirmos lenteles reiksmiu failas");
        jLabel14.setText("Antros lenteles reiksmiu failas");
        jLabel15.setText("Pirmos leneteles vardu falias");
        jLabel16.setText("Pirmos leneles genu vardas");
        jLabel17.setText("Antros lenteles genu vardas");
        Failas2.setText("C://Data2.csv");
        jLabel14.setText("Antros lenteles reiksmiu failas");
        jLabel17.setText("Antros lenteles genu vardas");

        jScrollPane3.setName("jScrollPane3");      // NOI18N

        lentele3.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
……
            },
            new String [] {
                "ID", "t1", "t2", "t3", "t4", "t5", "t6", "t7", "t8", "t9", "t10", "t11", "t12", "t13", "t14", "t15", "t16", "t17", "t18", "t19", "t20", "t21", "t22", "t23", "t24", "t25", "t26", "t27", "t28", "t29", "t30", "t31", "t32"
            }
        ) {
            Class[] types = new Class [] {
                java.lang.Float.class,
…….……..
```

```
        };

        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }
    });
    lentele3.setName("lentele3"); // NOI18N

lentele3.getTableHeader().setReorderingAllowed(false);
    jScrollPane3.setViewportView(lentele3);
    int i;
    ……..
    );

    setComponent(mainPanel);
    setMenuBar(menuBar);
    setStatusBar(statusPanel);

    bindingGroup.bind();
  }// </editor-fold>//GEN-END:initComponents

  //Veiksmas, atliekamas paspaudus mygtuka Ivedimas
  private void IvedimasMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_IvedimasMouseClicked
    if (Lygtis.getText().equals(""))
    {
      String a = JOptionPane.showInputDialog("Įveskite tinkamumo funkciją ir bandykite dar kartą");
      Lygtis.setText(a);
    }
    else
    {
    int j;
    int i;
    for (i = 0; i < 100; i++)
    {
      for (j = 0; j < 100; j++)
      {
        eData[i][j] = 0.0;
        eData2[i][j] = 0.0;
        eData3[i][j] = 0.0;
      }
    }
    GenuSkaicius = Integer.parseInt(this.j1.getText());
    MutacijosTikimybe = Double.parseDouble(this.p2.getText());
    KryzminimoTikimybe = Double.parseDouble(this.p1.getText());
    IndividuSkaicius = Integer.parseInt(this.p3.getText());
    LoadData();
    GeneruotiIndividus();
    Numeracija();
    TinkamumoTikrinimas();

lentele.getColumnModel().getColumn(GenuSkaicius+1).setHeaderValue("Tinkamumas");

lentele.getColumnModel().getColumn(GenuSkaicius+1).setMinWidth(100);

lentele.getColumnModel().getColumn(GenuSkaicius+1).setMaxWidth(750);
    i = GenuSkaicius+2;
    while (i < 33)
    {

lentele.getColumnModel().getColumn(i).setMinWidth(1);

lentele.getColumnModel().getColumn(i).setWidth(1);
      i++;
    }
```

```
lentele.getTableHeader().resizeAndRepaint();
        Ivedimas.setVisible(false);
        showNames.setText("Rodyti vardus");
        }
    }//GEN-LAST:event_IvedimasMouseClicked
    //Veiksmas atliekamas paspaudus mygtuka Iteracija
    private void IteracijaMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_IteracijaMouseClicked
        PilnaIteracija();
    }//GEN-LAST:event_IteracijaMouseClicked
    //Veiksmas atliekamas paspaudus mygtuka KeistiTikimtbes
    private void KeistiTikimybeMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_KeistiTikimybeMouseClicked
        MutacijosTikimybe = Double.parseDouble(this.p2.getText());
        KryzminimoTikimybe = Double.parseDouble(this.p1.getText());
    }//GEN-LAST:event_KeistiTikimybeMouseClicked
    //Veiksmas atliekamas paspaudus mygtuka Multiiteracija
    private void MultiiteracijaMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_MultiiteracijaMouseClicked
        int a = Integer.parseInt(iteracijuSkaicius.getText());
        iteracijuSkaicius.setText("");
        for (int i = 0; i < a-1; i++)
        {
           NepilnaIteracija();
        }
        PilnaIteracija();
    }//GEN-LAST:event_MultiiteracijaMouseClicked
    //Veiksmas atliekamas paspaudus mygtuka Sigma
    private void SigmaButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_SigmaButtonMouseClicked
        String sigma01 = this.sigma1.getText();
        String sigma02 = this.sigma2.getText();
        String lygtele = this.Lygtis.getText();
        String priedas = null;
        String torp; //t or p
        torp = String.valueOf(sigma01.charAt(0));
        int i=0;
        int j=0;
        int k=0;
        if (sigma01.length() == 2)
        i = sigma01.charAt(1)-48;
        else i = (sigma01.charAt(1)-48)*10 + (sigma01.charAt(2)-48);
        if (sigma02.length() == 2)
        j = (sigma02.charAt(1)-48);
        else j = (sigma02.charAt(1)-48)*10 + (sigma02.charAt(2)-48);
        priedas = "(" + torp + String.valueOf(i);
        for (k=i+1; k < j+1;k++)
        {
           priedas = priedas + "+" + torp + String.valueOf(k);
        }
        priedas = priedas + ")";
        lygtele = lygtele + priedas;
        this.Lygtis.setText(lygtele);
……
```