

VILNAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Baigiamasis bakalaurinis darbas

Genetiniai algoritmai eismui optimizuoti
(Genetic algorithms for traffic simulation)

Atliko: 4 kurso 1 grupės studentas

Linas Jurgelėnas (parašas)

Darbo vadovas:

lekt. Irmantas Radavičius (parašas)

Recenzentas:

Linas Laibinis (parašas)

Vilnius - 2020

TURINYS

SAVOKŲ APIBRĖŽIMAI	2
ĮVADAS	3
1. EISMO MODELIAVIMAS	5
1.1. APIBRĖŽIMAS	5
1.2. EISMO MODELIŲ TIPAI.....	6
1.3. EISMO MODELIAVIMO ELEMENTAI	7
2. ĮRANKIŲ PALYGINIMAS PAGAL REALAUS PASAULIO MODELIOUS	8
2.1. APIBRĖŽIMAS	10
2.2. PAGRINDINIAI KOMPONENTAI	11
2.3. GENETINIAI OPERATORIAI.....	11
2.3.1. Atrinkimas	11
2.3.2. Kryžminimas	12
2.3.3. Mutacija	13
2.4. GENETINIO ALGORITMO FAZĖS	14
2.5. PSEUDOKODAS	15
3. TYRIMAS	16
3.1. EISMO SIMULIAVIMO OPTIMIZAVIMO REIKŠMĖ	16
3.2. GALIMI OPTIMIZAVIMO VARIANTAI	16
3.3. GENETINIO ALGORITMO KONFIGŪRAVIMAS	17
3.3.1. Optimizavimo srities pasirinkimas	17
3.3.2. Duomenų aibė.....	18
3.3.3. Duomenų genetinė reprezentacija	19
3.3.4. Populiacijos dydžio pasirinkimas	20
3.3.5. Tinkamumo funkcijos sudarymas.....	21
3.3.6. Genetinių operatorių naudojimas	21
3.4. TYRIMO DUOMENŲ AIBĖ	22
3.5. TYRIMO EIGA	23
REZULTATAI IR IŠVADOS	29
CONCLUSIONS	30
LITERATŪROS SĄRAŠAS	31
PRIEDAS NR. 1	34

Sąvokų apibrėžimai

Sutrumpinimai:

GA – Genetiniai algoritmai

GO – Genetiniai operatoriai

TP – Transporto priemonės

Mikro – mikroskopinis

Mezo – mezoskopinis

Makro - makroskopinis

Mikro simuliacija – mažo masto eismo simuliacija

Mezo simuliacija – vidutinio masto eismo simuliacija

Makro simuliacija – didelio masto eismo simuliacija

Const. – konstanta

Šv. S. – šviesoforo signalas

IVADAS

Kasmet daugėjant vairuotojų, plečiantis miestams bet negerinant miestų infrastruktūros vis labiau aktuali spūsčių problema. Vien Vokietijoje, kuri pasižymi vienu geriausiu pasaulyje greitkelių tinklu, 2018 metais buvo užfiksuota 745,000 automobilių spūsčių – tai 3%, daugiau nei 2017 metais. Taip pat spūsčių eilės pailgėjo net 5% palyginus su 2017 metais. [ADAC19]

Vidutiniškai, Amerikoje kamščiuose praleistas laikas nuo 2000 metų, kai per metus juose buvo praleidžiama 37 valandos, iki 2015 metų padidėjo beveik 15% - iki 42 valandų. Šis skaičius vis didėja, ir vis daugiau laiko yra leidžiama veltui. [TTI15]

Ne visos jau esamos sankryžos yra pritaikytos tokiems srautams kurie ir toliau didėja. Kadangi automobiliai yra ne prabangos, o būtinumo priemonė, jų skaičius tikrai nemažės, tad spūsčių problemą galima spręsti pertvarkant jau esamas sankryžas ar, jeigu jos yra reguliuojamos, šviesoforų persijungimus, arba pritaikyti naujas planuojamas ir modeliuojamas sankryžas optimaliam pralaidumui.

Eismo modeliavimo įrankiai nuo jų sukūrimo pradžios (maždaug nuo 1960 metų) buvo išvystyti nuo įrankių skirtų tirti tik ekspertams, iki plačiai viešai visiems prieinamų ir paprastai naudojamų įrankių tirti, planuoti, demonstruoti ir vystyti eismo sistemas. [P99]

Yra sukurta daug programinės įrangos skirtos eismo modeliavimui bei simuliacijai. Daugumos programų pagrindinės galimybės sutampa, bet papildomos jų opcijos skiriasi. Pagal atskirus tikslus ir reikiamą funkcionalumą galima rinktis iš plataus spektro komercinių, ir ne tik, programų, pavyzdžiui: „Anylogic PLE“, „Autodesk Infracore“, „Aimsun Next“, „PTV VISSIM 11“, „Eclipse SUMO“ (toliau – „SUMO“) ir kitų. Dauguma jų yra uždaro kodo, o „SUMO“ – viena iš nedaugelio atviro kodo programinė įranga.

Optimizuoti procesus galima daugeliu optimizavimo būdų, kaip pavyzdžiui pritaikant neuroninius tinklus, optimizavimo algoritmus ir kitus variantus. Šiame darbe eismo simuliacijos optimizavimui autorius bandys pritaikyti genetinį optimizavimo algoritmą ir parodyti kokią įtaką gali daryti skirtingi genetinio algoritmo atrinkimo, kryžminimo bei mutacijos operatoriai.

Eismo optimizavimas yra daug pastangų ir labiausiai resursų reikalaujantis procesas. Tikrinant kiekvieną norimą atlikti pakeitimą keisti simulatoriuje rankiniu būdu gali užimti labai daug laiko ir prireiks daug žmogiškųjų resursų. Norint ištirti kuo daugiau galimų modelių ir optimizuoti variantų pasirinkimą per kuo trumpesnę laiką, šį procesą būtų praktiška ir naudinga automatizuoti.

Šio **darbo tikslas** – panaudoti genetinius algoritmus eismo optimizavimo automatizavimui.

Darbas susideda iš dviejų pagrindinių dalių: pirmoji dalis skirta įrankių palyginimui realaus pasaulio modeliams – kiek kiekvienas įrankis leidžia nusakyti įvairius modelius. Antroji dalis

skirta genetinio algoritmo pritaikymui eismo simuliacijai ir palyginti, kokią įtaką gali daryti skirtingi atrinkimo, kryžminimo ar mutacijos operatoriai.

Darbo **uždaviniai**:

- Apibrėžti, kas yra eismo modeliavimas
- Palyginti eismo simuliacijos įrankius
- Apibrėžti, kas yra genetiniai algoritmai ir kaip jie naudojami
- Išanalizuoti galimus optimizavimo variantus
- Apibrėžti į kokius aspektus bus atsižvelgta taikant genetinį algoritmą eismo optimizavimo uždavinyje
- Pritaikyti genetinį algoritmą eismo optimizavimui
- Pateikti išvadas

1. Eismo modeliavimas

1.1. Apibrėžimas

Eismo modeliavimas – simuliacija yra matematinis transporto sistemų modeliavimas pasitelkiant kompiuterinę programinę įrangą padedant planuoti, projektuoti bei valdyti transporto sistemas. [L08]

Eismo modeliavimas turi labai platų panaudojimo spektrą. Tipiniai eismo modeliavimo panaudojimai atspindi:

- Transportavimo planavime: įrankiais galima įvertinti skirtingus scenarijus miestų vystymosi infrastruktūrai kuo tiksliau atkuriant esamą situaciją bei būsimus pasikeitimus per tam tikrą laiko tarpą. Kadangi, atkurti situacijas pagal visus aspektus būtų per daug sudėtinga, reikia tikslingai atrinkti duomenis pagal kuriuos bus simuliuojama.[NER99]

Simuliuojant yra įvertinamas regiono miestų vystymosi modelio įtaka transportavimo infrastruktūrai. Dažniausiai modeliai priima schematiškus populiacijos, darbingumo toje apskrityje, žemės panaudojimo duomenis. Iš gautų duomenų galima nustatyti reikiamą norimos kelionės trukmę, poveikį oro kokybei ir kitus dalykus.

Viena iš naujausių modeliavimo metodų – TRANSIMS (TRansportation ANalysis SIMulation System) – transportavimo analizinė simuliacijos sistema – sujungia paklausos modeliavimą ir eismo tėkmę keliuose vienu metu, taip siekiant apibūdinti visą eismo sistemos elgesį tam tikromis sąlygomis. [L08]

- Transportavimo projektavime: sankryžų sudėtingumas, kelių geometrija ir kt.
- Transportavimo valdyme: tokie modeliai kaip CORSIM (CORridor SIMulation) – jame integruoti du mikroskopiniai simuliacijos modeliai atstovaujantys visą eismo aplinką.

Šiuolaikiniuose eismo simulatoriuose realiam eismui sudėtingėjant, vis dažniau atkreipiama dėmesį į įrankių trūkumus, kuriuos sudėtinga išspręsti [N08]:

- Detalumo trūkumas – įrankiai dažnai turi tam tikrų duomenų apribojimus, ar iš vis tam tikrų duomenų nėra galimybės pateikti. Dėl tokių logikos ribotumo sudėtinga atkurti realų vairuotojų elgesį bei TP atsakus į aplinkos veiksnus.
- Įrankių lankstumas – norint atlikti mezo simuliacijas, dažnai tenka atsisakyti kai kurių duomenų, kuriuos galima būtų naudoti atliekant mikro simuliacijas, ir atvirkščiai.
- Brangus modelių konstravimas – kuriant modelius ir ruošiant jiems tinklo duomenis, dažnai kaina gali išaugti eksponentiškai priklausomai nuo tinklo dydžio.

Dėl šių trūkumų, naujos kartos eismo modeliavimo įrankiai nurodoma, kad turėtų būti kaip kelių mastų dydžio (*angl. multi-scale resolution*), paraleliai galėtų atlikti skaičiavimus ir būtų vedami objektų. [N08]

1.2. Eismo modelių tipai

Eismo modeliai gali būti trijų bendrinių tipų arba hibridiniai:

- Mikro modeliai – transportavimo simuliacijos srityje yra bendras susitarimas, kad yra tam tikri kompiuterinio skaičiavimo nuostatai, kai simuliacijoje yra atsižvelgiama net į kiekvieną individualų simuliuojamą eismo dalyvį.
Tokie modeliai bendrai naudoja tokius fizinių komponentų modelius, kaip kelių tinklus, eismo valdymo sistemas, vairuotojo-TP vienetų, ir kt.; ir tokius elgsenos modelių komponentus, kaip vairuotojo elgesio kelyje ir kelio pasirinkimo modeliai. [CLOR03]
- Mezo modeliai – šie modeliai yra paprastesni už mikro modelius atsisakant kai kurių rezultatų tikslumo norint sutrumpinti skaičiavimų laiką.
Mezo modelyje esmė yra bandyti užfiksuoti automobilių judėjimą vieno paskui kitą, tarpus tarp automobilių ir juostų keitimo fenomenus, bet nebūtina bandyti aptikti kiekvieną individualią TP modelio tinkle. Taip yra sumažinamas tikslumas, bet simuliacija atliekama greičiau ir joje vis tiek galima aptikti laikinų spūsčių fenomenus. [FMT01]
- Makro modeliai – lyginant su mikro modeliais, makro modeliais yra apibrėžiamas bendrasis greitis ir bendras TP tankis. Šie dydžiai yra lokalūs vidurkiai mikro modelių TP greičių ir TP atstumų tarp jų centrų. Visos kitos sąlygos kurios tiktų mikro ir mezo simuliacijoms yra pašalinamos. Dėl to, makro simuliacijos yra deterministinės, ir visos savarankiškai organizuotos struktūros (tokios kaip spūstys, sustojimo ir pajudėjimo bangos) yra išlyginamos. Žinoma, atsižvelgiant į tai, kad ši informacija buvo apdorota mikro simuliacijose, į jas yra kažkiek atsižvelgiama ir priimama, kad makro simuliacijose šie efektai yra galimi. [TH99]
- Mezo – mikro hibridinis modelis – tai modelis, kuriame yra apjungiami mikro modelio ir mezo modelio ypatumai. Jame naudojamas mikro modelis tose simuliacijos vietose, kur reikia daugiau detalumo, ir mezo modelis tose vietose, kur nereikia tokio didelio detalumo.

Svarbiausias šio modelio aspektas – identifikuoti, kurioje vietoje bus mikro modelis, o kurioje - makro modelis ir jų tarpusavio sąsaja tiek tarp duomenų perdavimo, tiek tarp kitų komponentų. [BKA05]

Priklausomai nuo pasirinkto modeliavimo tipo skirsis ir reikalingų simuliacijai elementų aibė.

1.3. Eismo modeliavimo elementai

Ne visi eismo modeliavimo įrankiai naudoja tuos pačius modelius dėl skirtingo jų pritaikomumo, panaudojamumo, keičiamų elementų kiekio ir norimų rezultatų. Dažniausiai eismo simuliacijai modeliai yra kuriami kaip vienalytės sistemos puikiai pritaikytos realiams atvaizdavimams. Kitu atveju modeliai gali būti moduliniai, sudaryti iš keleto ar keliolikos mažesnių segmentų. Abu modelių tipai gali turėti tiek statinius tiek dinامينius elementus.

Statiniai modelio elementai nesikeičia simuliacijos vykdymo metu, dėl to galima juos laikyti konstantomis. Jie gali būti įrašyti programos kode ar įvesti programos inicializacijos metu. Vienalytės sistemos su statiniais modelio atributais yra laikomos klasikinio stiliaus modeliais. Dažniausiai tokiomis galima laikyti mikroskopinius eismo simuliacijai modelius kuriuose yra reikalingas didelis duomenų tikslumas. [KSS98] Pagrindiniai mikro eismo modeliavimo elementai yra kelių tinklai, eismo poreikis, TP maršruto nustatymai, TP judėjimas ir išvesties modulis. [YKB00] Tokiu atveju dažnai statiniai elementai yra eismo juostos, sankryžos, šviesoforai ir kiti eismo infrastruktūros elementai, ir eismo charakteristikos – eismo intensyvumas, šviesoforų dažniai, o dinaminiai – tik simuliuojami objektai.

Dinaminiai modelio elementai gali leisti keisti savo parametrus ar atributus modelio vykdymo metu. [KSS98]

2. Įrankių palyginimas pagal realaus pasaulio modelius

Eismo simuliacijos modeliai ir įrankiai vis tobulėjo, kad simuliacijos, modeliavimai, planavimai ir analizės atspindėtų kuo realistiškesnę eismo eigą. 2016 metais buvo atliktas septyniolikos įrankių skirtų eismui simuliuoti palyginimas remiantis moksliniais straipsniais bei techninėmis specifikacijomis [PMO17]. Įrankių palyginimas rėmėsi simuliacijų realistiškumu simuliuojant 6000 kilometrų kelių tinklą kurį sudarė keliai miesto ir atokesnėse teritorijose, smailūs posūkiai, kliūtys, įkalnės, pėsčiųjų perėjės, dviejų juostų bei vienos juostos keliai ir eismu panašiu kaip greitkeliuose.

Šiame modeliuojamame eisme buvo simuliuojamas nevienalytis transporto eismas, kurių greičiai ir judėjimo ypatybės skyrėsi: automobiliai, sunkusis transportas, viešasis transportas, žemės ūkio TP, motociklai, dviračiai ir pėstieji.

Lentelėje galima matyti įrankių palyginimą ir pranašumus, kurie yra suskirstyti grupėmis, įskaitant ir įrankių funkcionalumus, kurie, atliekant tyrimui buvo pastebėta, kad yra svarbūs realaus pasaulio eismo simuliacijos modeliams (žr. lentelė nr.3 prieduose). [PMO17]

Apibendrintai iš tyrimo lentelės matoma, kad:

- visi eismo modeliavimo įrankiai gali įvertinti esamas eismo situacijas ir nuspėti eismo sąlygas.
- Nei viena programa neturi visiškai pilno funkcionalumo.
- Nei viena programa nėra susikoncentravusi tik į vieną taikymo sritį.
- Kai kurie iš įrankių naudoja hibridinius modelius (mikro + mezo, mikro + makro, mikro + mezo + makro)
- Kai kurie iš įrankių turi apribojimų. Genetiniai algoritmai

Nagrinėjamo tyrimo metu buvo analizuojama 17 eismo simuliacijos programų, kurios buvo vertinamos pagal nustatytus kriterijus. Yra išskiriamos 6 pagrindinės kriterijų grupės, kuriomis remiantis buvo nagrinėjamos eismo simuliacijų programos:

1. Eismo modelių tipų palaikomumas (*Angl. Flow model category*)
2. Eismo modelių dydžių apribojimai (*Angl. Model size restrictions*)
3. Sumanaus eismo funkcijos (*Angl. ITS (Intelligent Transportation System) functionalities*)
4. Gatvių apribojimai (*Angl. Street restrictions*)
5. Modeliuojamos TP ir su tuo susiję reiškiniai (*Angl. Modelled objects and phenomena*)
6. Duomenų integravimas realiuoju laiku (*Angl. Real-time data integration and analysis*)

Visos šios grupės apėmė iš viso 64 programų vertinimo kriterijus. Jiems vertinti buvo naudojami žymėjimai X, O, ! ir tuščia vieta. X simbolizuoja „taip“, O - „ne“, ! - „reikalauja tobulinimo“, tuščia vieta - „nepateikta“.

Analizuojant tyrimo lentelę galima išvesti statistiką, kiek kriterijų kiekviena programa atitiko. Statistika buvo išvesta skaičiuojant atvejus, kai programa pilnai patenkina kriterijus, t.y. neskaičiuojant atvejų, kuomet programa turėjo veikiančią funkciją, bet reikalauja tobulinimo. (žr. pav. 1)

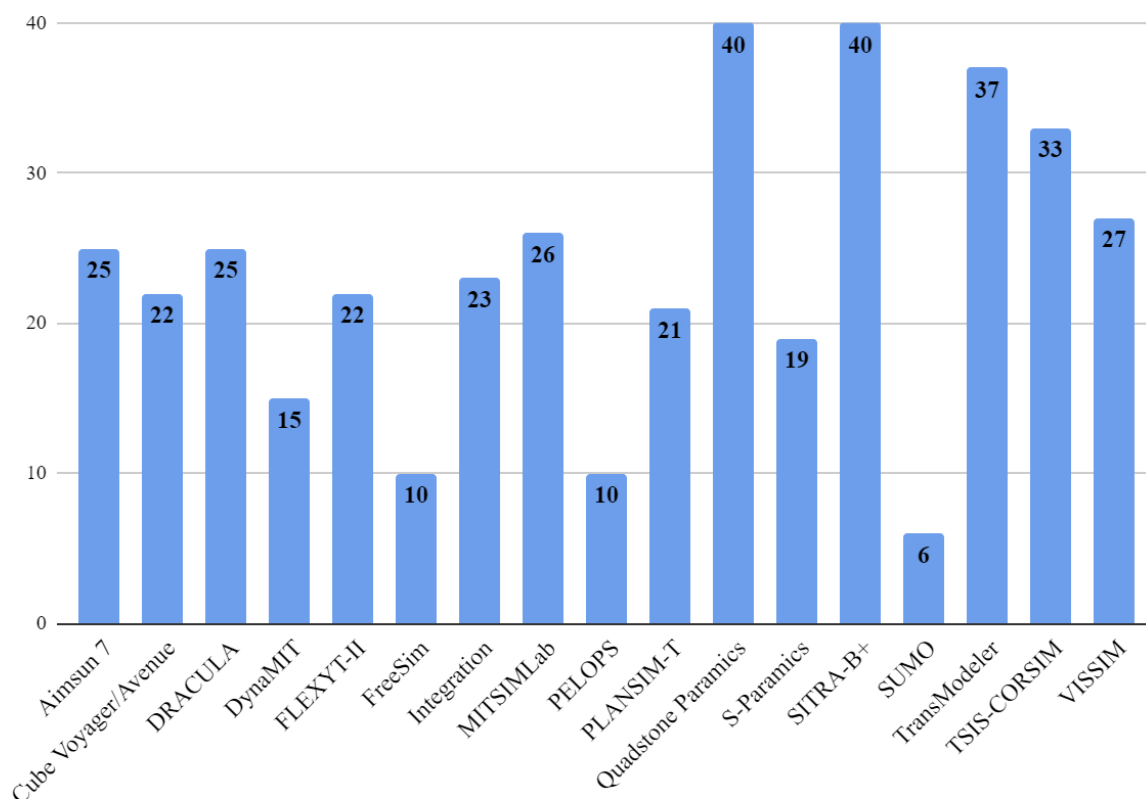


Diagrama 1. Programų kriterijų atitikimo statistika

Iš visų 17 nagrinėjamų programų matomos dvi lyderės: „Quadstone Paramics“ bei „SITRA-B+“. Šios programos atitiko daugiausiai numatytų kriterijų, surinkusios po 40 X simbolių, tai reiškia, kad jos atitiko 62.5 proc. visų 64 numatytų kriterijų. Antroje bei trečioje vietoje, pagal surinktą taškų skaičių – „TransModeler“ ir „TSIS-CORSIM“, surinkusios 37 bei 33 taškus. Mažiausiai taškų surinkusios programos: „SUMO“, „PELOPS“ bei „FreeSim“. „PELOPS“ ir „FreeSim“ atitiko 10 numatytų kriterijų, t.y. tik 15.6 proc. visų kriterijų.

„SUMO“ – mažiausiai kriterijų atitikusi programa, surinkusi tik 6 taškus, t.y. 9.4 proc. visų numatytų simulatoriams kriterijų. Dėl šios priežasties, tolimesniame šio bakalauro darbe bus naudojama ši programa, siekiant įsitikinti, ar turint daug apribojimų, bet tuo pačiu tik būtiną,

paprastą ir reikalingą funkcionalumą, galima efektyviai optimizuoti eismo simuliaciją genetiniu algoritmu.

2.1. Apibrėžimas

GA priklauso evoliucinių skaičiavimo modelių grupei. Šie algoritmai gali rasti potencialiai teisingą atsakymą specifinei problemai išspręsti remdamiesi panašia į chromosomose esančias duomenų struktūras pritaikant atranką ir pertvarkymus taip išsaugant kritinę informaciją ir sugeneruojant naujus atskaitos duomenis.

Dažnai GA yra traktuojami kaip funkcijų optimizatoriai, nors jų panaudojimas ir pritaikymas yra ganėtinai platus.

Sprendimo „gerumas“ yra apibrėžiamas lyginant gautus atsakymus su prieš tai esamais duomenų įverčiais.[W94]

Norint su GA išspręsti realaus gyvenimo situaciją, turi būti patenkinti du pagrindiniai reikalavimai:

- Yra eilė duomenų, kurie gali reprezentuoti sprendinius iš sprendinių aibės;
- Objektyvi tinkamumo funkcija, kuri nustato kokio tinkamumo sprendinys gali būti sugeneruotas.

Primityvus GA veikimo principas toks, kad yra atsitiktinai sugeneruojama pradinė duomenų eilės populiacija, ir yra pritaikomi trys pagrindiniai operatoriai sukurti naują, ir tikėtina, geresnę duomenų eilės aibę nei prieš tai buvusi. Visų pirma yra pritaikoma reprodukcijos (atrinkimo) operatorius, kur duomenų eilutės yra nukopijuojamos į kitą duomenų kartą su tam tikromis tikimybėmis pagrįstomis pagal tinkamumo funkciją. Antras operatorius yra kryžminimo, kur atsitiktinai atrenkamos duomenų eilių poros yra sukryžminamos, taip sukuriant naujas eiles. Trečias operatorius yra mutacijos, kuris atsitiktinai pakeičia kurią nors eilės reikšmę. Kryžminimo ir reprodukcijos operatoriai yra svarbiausi atliekant paiešką su GA. Mutacijos operatorius pajvairina paieškos erdvę ir apsaugo nuo svarbių praradimų, kurie gali atsirasti atliekant reprodukciją ir kryžminimą.[A10]

Genetinis algoritmas yra puikiai pritaikomas optimizavimo problemoms spręsti dėl keleto esminių privalumų:

- Pritaikomumas: GA neturi daug matematikai specifinių reikalavimų. Tik dėl evoliucinės prigimties GA ieškos tinkamiausio sprendimo neatsižvelgiant į vidinius pačios problemos veikimo principus.

- Tvirtumas/Atsparumas: daugybės tyrimų įrodyta, kad GA yra žymiai efektyvesnis ir atsparesnis ieškant optimalaus sprendimo ir reikalaujantis mažiau skaičiavimo resursų nei kiti tradiciniai euristicos metodai.
- Lankstumas: GA leidžia juos prijungti kartu prie kitų euristicinių metodų, taip leidžiant dar efektyviau ieškoti problemų sprendimų. [GL08]

2.2. Pagrindiniai komponentai

Bendrai, GA sudaro penki pagrindiniai komponentai [GL08]:

1. Potencialaus problemos sprendimo genetinė reprezentacija.
2. Būdas kaip sukurti populiaciją, arba populiacijos imtis.
3. Įvertinimo (atitikties) funkcijos vertinimo sprendimas pagal jų tinkamumo funkciją (*angl. fitness function*).
4. Genetiniai operatoriai (GO), kurie padeda modifikuoti genetinę kompoziciją – pagrindiniai yra atrinkimas, kryžminimas, mutacija.
5. Parametrų reikšmės kurias naudoja GA – populiacijos dydis, tikimybės priskirti genetinius operatorius ir kt.

2.3. Genetiniai operatoriai

Kiekvienas GO atlieka atskirą savo funkciją. Genetinis operatorius imituoja paveldimumo procesą sukuriant naujus palikuonis kiekvienoje kartoje. Jie atsitiktinai keičia esamus duomenis be garantijos gauti pagerintus duomenis naujoje kartoje.

2.3.1. Atrinkimas

Atrinkimo, arba kitaip – selekcijos, operatorius skirtas padidinti vidutinę populiacijos kokybę priskiriant didesnę galimybę naudingoms chromosomoms patekti į kuriamą kartą. Yra keturios dažniausiai naudojamos atrinkimo schemas [GD91]:

1. Proporcingas atrinkimas (*Angl. proportionate selection*) – šis atrinkimo variantas greičiausiai konverguoja nei kiti, bet labai didelė tikimybė, kad bus atmesti geri variantai. Jo veikimo principas: sudedama visų elementų atitikties funkcijų suma, sudaromos visų populiacijos narių atitikties funkcijų proporcijos – gaunama, kad didžiausią proporciją turintis genas turi didžiausią tikimybę būti atrinktas kryžminimui. Toliau yra generuojamas atsitiktinis skaičius, ir atrenkamas gautas genas kryžminimui. Taip

atrenkami du nariai, kurie naudojami kryžminimui. Tai kartojama tiek, kiek reikalauja populiacijos apimtis.[J87]

2. Reitingavimo atrinkimas (*Angl. ranking selection*) – šio atrinkimo metu populiacijos nariai yra išrūšiuojami, nuo blogiausio iki geriausio ir kiekvienam nariui suteikiama vertė lygi jo pozicijai eilėje – t.y blogiausiam genui bus priskirta 1, antram pagal blogumą – 2, ir taip kol geriausiam bus priskirta populiacijos dydžio, arba jo eilės numeris. Sudedama visų jų suma, sugeneruojamas atsitiktinis skaičius riboje iki sumos, ir iteruojama per populiacijos narius sumuojant. Atrenkamas pirmas narys kurio iteracijų suma didesnė už atsitiktinį skaičių. [GD91]
3. Turnyrinis atrinkimas (*Angl. tournament selection*) – šio atrinkimo metu yra pasirenkama atsitiktinis populiacijos narių skaičius, iš šių narių yra pasirenkamas geriausias narys, ir taip kartojama tiek kartų kiek reikia sudaryti naujai populiacijai. [GD91]
4. GENITOR (pastovios būsenos) atrinkimas (*Angl. Genitor or „steady state“ selection*). Šiame tyrime jis nebus naudojamas

2.3.2. Kryžminimas

Kryžminimo esmė yra sukombinuoti du atrinkimo etape atrinktus tėvinės populiacijos narius, gaunant naują populiacijos portą tikintis, kad ji bus geresnė už ankstesnius. Pagal užduoties populiacijos sudėtį pasirenkamas vienas iš kryžminimo metodų [LKMY96]:

1. PMX (*Angl. Partially-mapped Crossover*) – dalinai sužymėtas kryžminimo principas – atsitiktinai pasirenkamas vienodo ilgio ir tos pačios vietos informacijos dalys iš abiejų tėvų, jos sukeičiamos tarpusavy, ir iš jų sudaroma santykių matrica. Pagal ją likusi informacija yra perkeliama į vieną ir kitą vaiką pagal informacijos atitikimų sekas.
2. CX (*Angl. Cycle Crossover*) – ciklinis kryžminimas – pirma, sudaromas ciklas – pradedama iš pirmo tėvo (P1) pirmu elementu, pažiūrima į antro tėvo (P2) pirmo elemento reikšmę; peržiūrima P1 poziciją su gautu P2 elementu, ir pridedame šį elementą į ciklą. Taip kartojama tol, kol grįžtama į pirmą P1 elementą. Šie elementai keliauja į pirmą vaiką į tas pačias vietas, o likusios vietos užpildomos su elementais iš antro tėvo tų pačių vietų. Analogiškai kartojama su P2 ir antru vaiku, jeigu to reikia.
3. OX1 (*Angl. Order-based Crossover*) – tvarka paremtas kryžminimas – atsitiktinai pasirenkamas segmentas iš pirmo tėvo, kuris nukopijuojamas į pirmąjį vaiką. Toliau, iš eilės į po įkelto segmento sekančias vietas cikliškaiki iki įkelto segmento pradžios, iš eilės įrašomos antrojo tėvo nesikartojančios reikšmės. Analogiškai kartojama su antru vaiku ir tėvų tomis pačiomis vietomis, jeigu to reikia.

4. OX2 (*Angl. Order-based Crossover #2*) – OX1 kryžminimo variacija – šis kryžminimas iš antrojo tėvo atsitiktinai pasirenka atsitiktinį skaičių narių. Jų reikšmės pašalina iš pirmojo tėvo, ir į šias vietas įrašomos trūkstamos reikšmės iš antrojo tėvo pagal elementų seką eilės tvarka. Taip pat kartojama ir antram vaikui su tomis pačiomis vietomis, jeigu to reikia.
5. POS (*Angl. Position-Based Crossover*) – vieta paremtas kryžminimas – atsitiktinai pasirenkamos identiškos narių vietos tiek pirmoje tiek antroje tėvinėje duomenų imtyje. Pirmajame vaike įrašomi nariai iš antrojo tėvo į tas pačias vietas kaip antrajame tėve. Tuomet įrašomi nariai iš pirmosios imties į laisvas vietas tikrinant iš eilės. Taip pat kartojama su antru vaiku tomis pačiomis vietomis, jeigu to reikia.
6. VR (*Angl. Voting recombination crossover*) – balsavimo rekombinacijos kryžminimas – tinkamas N-nariai populiacijai, kur N – daugiau nei du. Pirmiausia nurodomas slenkstinis dydis x ne didesnis nei N. Tuomet, tikrinama visų tėvinių imčių i -tosios vietos, kur $i = [1;N]$, kiek kartų kiekvienoje pozicijoje kartojasi kiekviena reikšmė. Jeigu yra reikšmių ne mažesnių nei x , jos yra perduodamos vaikui, o likusios vietos yra užpildomos likusių reikšmių mutacijomis.
7. AP (*Angl. Alternating-position crossover*) – kintančios pozicijos kryžminimas – iš tėvų yra imama iš eilės po vieną elementą ir jie dedami vaikui, atmetant jau esamus elementus vaike. Pirmam vaikui pradedama dėti iš pirmojo tėvo, antram – iš antrojo.

2.3.3. Mutacija

Tai operatorius, keičiantis atsitiktines sukurto naujo vaiko chromosomas. Pagrindinė mutacijų esmė GA – atkurti atrinkimo metu prarastus kai kuriuos populiacijos narius, ar narių elementus ir patikrinti juos naujame kontekste; teikti narius ar narių elementus kurių dar nebuvo populiacijoje iki mutacijos įsikišimo. Tuo pačiu, tai neturėtų sugriauti pačios Genetinio algoritmo esmės ir padaryti per daug įtakos kuriamiems naujiems vaikams.

Mutacijos naudojimo pagrindas – apsaugoti GA nuo per anksti visam laikui pašalinamų elementų, kurie GA veikimo eigoje gali būti esminiai. Ypač mažoms populiacijoms nenaudojant mutacijos operatoriaus yra galimybė pirmalaikiam konvergavimui vien dėl to, kad nebuvo galimybės vėl įvesti prarastų elementų. [W94]

Yra daug mutacijų tinkamų darbui su bitų eilutėmis, keletas iš kurių puikiai tinka ir skaičiams ir eilutėms – inversijos mutacija, įterpimo mutacija, poslinkio mutacija, keitimo mutacija. Dažniausiai, kiekvienos ląstelės mutacijos tikimybė yra $1/L$, kur L yra duomenų eilutės ilgis, bet

mutacijos tikimybę elementams galima nustatyti atskirai pagal uždavinį ir mutacijos reikalingumą. Galimas iš tokių mutacijų operatorių pasirinkimas [LKMY96]:

1. DM (*Angl. Displacement mutation*) – perkėlimo mutacija – atsitiktinai parinkta vaiko atsitiktinio ilgio elementų grandinė perkeliama į kitą vietą tarp kitų elementų vaike.
2. EM (*Angl. Exchange mutation*) – sukeitimo mutacija – atsitiktinai parenkami du elementai ir jie sukeičiami vietomis.
3. ISM (*Angl. Insertion mutation*) – įterpimo mutacija – atsitiktinai parenkamas ir išimamas vienas elementas, atsitiktinai parenkama vieta kur įterpiamas parinktas elementas.
4. SIM (*Angl. Simple-inversion mutation*) – paprasto perstatymo (inversijos) mutacija – pasirenkama atsitiktinė elementų grandinė ir įterpiama į tą pačią vietą šios grandinės atvirkštinė tvarka.
5. IVM (*Angl. Inversion mutation*) – inversijos (perstatymo) mutacija – pasirenkama atsitiktinė elementų grandinė ir šios grandinės atvirkštinė tvarka įterpiama į atsitiktinai pasirinktą naują vietą.
6. SM (*Angl. Scramble mutation*) – sumaišymo mutacija – atsitiktinai pasirenkama elementų grandinė, jos nariai sumaišomi, ir įterpiama atgal į tą pačią vietą.

2.4. Genetinio algoritmo fazės

Pats genetinis algoritmas turi savo struktūrą, yra pagrindinės genetinio algoritmo fazės:

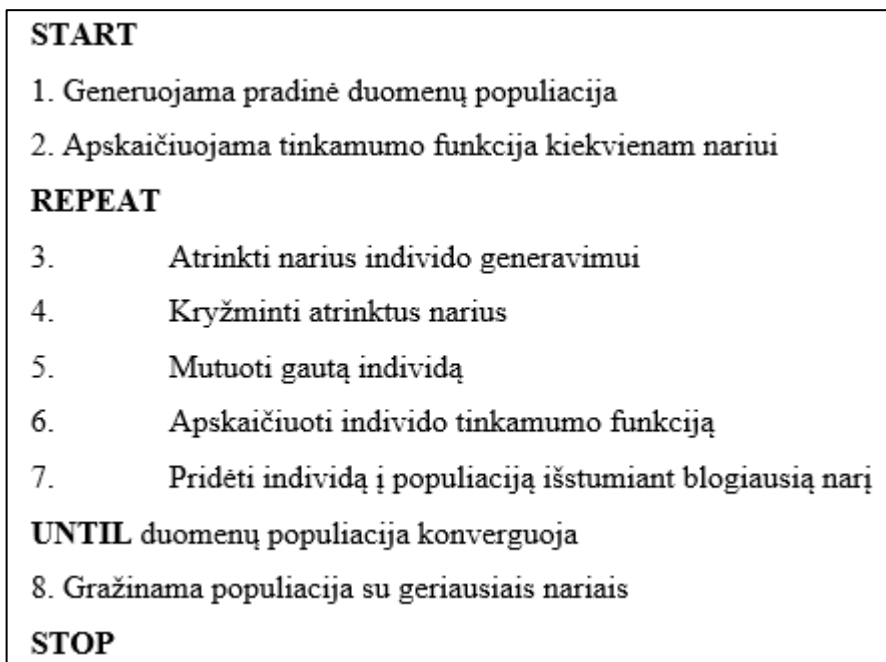
1. Populiacijos reprezentacija – klasikiniame GA chromosomos yra atvaizduojamos kaip bitų eilutės, sudarytos iš nulių ir vienetų. Tačiau, specifiniame problemos sprendime žinių aibė turi atsispindėti reprezentacijoje – ją turėtų sudaryti visa reikalinga informacija skirta pasiekti geriausio rezultato.
2. Tinkamumo funkcijos sukūrimas – bendrai, tai yra funkcija, kurią pritaikius gausime rezultatą, nusakantį, kaip arti geriausio tikslo yra gautas sprendinys. To pasekoje, ja galima remtis ar sprendimas vyksta teisinga kryptimi, ar genetinis algoritmas pasiekė maksimalią teisingiausią reikšmę. Jeigu ši funkcija bus per daug sudėtinga ir komplikuotai implementuota, ji gali sudaryti butelio kaklelio efektą ir stabdyti visą genetinį algoritmą.
3. Populiacijos inicializacija – inicializacijos metu yra sukuriamas uždavinio sprendimui reikalingo dydžio atsitiktinė populiacija galimų sprendinių – pirmoji sprendinių generacija. Jos metu turėtų būti sukurti validūs duomenys tam, kad algoritmas turėtų realią teigiamą pradžią. Kitu atveju, jeigu algoritmas yra labai apribotas, yra rizika, kad didžiąją laiko dalį vertins neteisingus populiacijos variantus ir efektyvios paieškos galimam geriausiam sprendiniui tikimybė būtų labai menka arba tai užtruktų žymiai ilgiau.

4. Tinkamumo funkcijos panaudojimas – esamos populiacijos sprendinių atitikimo paskaičiavimas.
5. Atrinkimas – inicializuotos, ar jau sukurtos naujos populiacijos sprendinių atrinkimas pagal iš anksto pasirinktą atrinkimo schemą.
6. Kryžminimas – sprendinių, atrinktų porų ar imties kryžminimas pagal pasirinktą kryžminimo operatorių.
7. Mutacijos pritaikymas – pagal mutacijos tikimybę mutuojami sprendiniai pagal mutacijos operatorių.
8. Geriausio sprendinio identifikavimas

Šios 4-7 fazės yra kartojamos tol, kol yra gaunamas artimas idealiam rezultatas, arba sprendiniai konverguoja ir neįmanoma pasiekti geresnio rezultato.

2.5. Pseudokodas

Pirmame paveikslėlyje yra pavaizduotas pseudokodas. Visų pirma yra sugeneruojama pradinė duomenų populiacija, tuomet apskaičiuojama tinkamumo funkcija kiekvienam populiacijos individui – šio darbo atveju, su individualiais yra leidžiamos simuliacijos ir iš simuliacijos išvesties duomenų yra suskaičiuojamas tinkamumas. Toliau, cikle kartojamas narių atrinkimas, jų kryžminimas ir mutavimas. Naujai gautam individui yra apskaičiuojama jo atitiktis, ir įdedamas į populiaciją išstumiant blogiausią populiacijos narį, lyginant atitikties vertes. Tai kartojama tol kol populiacija konverguoja ir gaunama populiacija su geriausiais individualiais.



pav. 1. Pseudokodas

3. Tyrimas

Autoriaus atliekamas GA pritaikymo eismo optimizavimui tyrimas yra atliekamas smulkiu mastu, norint parodyti, kad iš esmės GA įmanoma pritaikyti eismo optimizavimui. Eismo optimizavimo galimybės realiu mastu yra žymiai didesnės nei primityvaus tyrimo metu, bet sukūrus universalų, tinkamai veikiančią algoritmą jį galima išvystyti iki reikiamo dydžio. Tuo atveju, žymiai sumažėtų apribojimų skaičius nei atliekamame tyrime.

3.1. Eismo simuliacijos optimizavimo reikšmė

Kadangi rankiniu būdu vesti kiekvieną atskirą simuliacijos variantą ir tikrinti galimybes tol, kol bus pasiektas potencialiai optimalus simuliacijos variantas užimtų per daug laiko ir žmogiškų pastangų, būtų naudinga turėti automatizuotą ir stabilų įrankį, kuris leistų per trumpą laiką patikrinti žymiai daugiau galimų sprendimo variantų ir rastų galimai optimaliausią variantą.

Tuo pačiu, turint automatizuotą įrankį, būtų didesnė laisvė vartotojui ir galimybė rinkti skirtingus genetinių operatorių variantus, taip padidinant galimybę rasti dar teisingesnę ar greičiau konverguojantį sprendinį.

GA naudojimas eismo simuliacijos optimizavimui naudingas tuo, kad juo galima spręsti parametrines problemas, ir yra vienas iš paprasčiausių ir primityviausių algoritmų, leidžiančių lengvai optimizuoti uždavinius. Taip pat šiuo sprendimo būdu galima rasti daugiau nei vieną panašius, taip pat gerus ir tinkamus naudoti sprendinius.

3.2. Galimi optimizavimo variantai

Eismo simuliacijos uždaviniuose yra galimybė optimizuoti ne vieną naudingą sritį. Pavyzdžiui, kritines situacijas, tokias kaip:

- eismo pralaidumas – kiek TP pravažiuoja tam tikrą atkarpą per nurodytą laiko matą;
- vietos apribojimas – padidinti pralaidumą kelio struktūros atžvilgiu, atsižvelgiant į vietos apribojimus;
- laiko prie sankryžos praleidimas – kiek laiko TP praleidžia prie šviesoforo esant tam tikram eismui ar šviesoforų suregulavimui;
- sustojimų skaičius – kiek mažiausiai kartų TP privalo sustoti prieš pravažiuojant tam tikrą kelio dalį, ar kiek mažiausiai kartų TP sustoja kelyje judant nuo maršruto pradžios iki maršruto pabaigos;

- kelionės laikas – kiek laiko praleidžiama TP važiuojant nuo maršruto pradžios iki pabaigos;
- kelionės greitis – kiek užtrunka TP nuvykti iš maršruto pradžios iki pabaigos;
- kaina – jeigu yra apribota suma, kaip galima ją kuo naudingiau išnaudoti sudarant kuo geresnes sąlygas TP eismui;
- sankryžos geometrijos sudarymas – rasti kuo naudingiausią sankryžos geometriją atsižvelgiant į kitus aktualius parametrus, be kurių optimizuoti nebūtų įmanoma;
- tarša- TP taršos mažinimas priklausantis nuo TP greitėjimų, lėtėjimų ir kitų parametru;
- TP greičiai – vidutinio greičio didinimas ar mažinimas;
- ir kitos.

Žinoma, galimi ir hibridiniai optimizavimo uždaviniai jungiantys kelis optimizavimo variantus. Taip būtų apimama daugiau kriterijų ir simuliacijos dar labiau atitiktų realybę, bet tuo pačiu skaičiavimo sudėtingumas gali žymiai padidėti ir darbo laikas ilgėti.

3.3. Genetinio algoritmo konfigūravimas

Norint teisingai pritaikyti GA svarbiausia jį tinkamai sukonfigūruoti, sudaryti reikiamas funkcijas, eigą, nuo ko ir priklausys visas eismo simuliacijos optimizavimo rezultatas.

Visų pirma reikia apsibrėžti, vieną iš krypčių, kuria norima optimizuoti eismo simuliaciją; įvardinti apribojamus ir naudojamus parametrus, kurių bus laikomasi; pagal naudojamus parametrus sudaryti genetinę duomenų reprezentaciją; sudaryti teisingą tinkamumo funkciją su naudojamais parametrais.

Norint pritaikyti genetinio algoritmo panaudojimui reikia realizuoti anksčiau darbe minėtą pseudokodą norima programavimo kalba ir integruoti kartu su pasirinktu simuliacijos įranku.

3.3.1. Optimizavimo srities pasirinkimas

Norint įrodyti, kad GA galima naudoti eismui simuliuoti autorius bandys pritaikyti GA apribotam primityviam modeliui su mažai kintamų sąlygų. Įgyvendinant GA pritaikymą didesniai modeliui su daugiau galimybių, gali atsirasti papildomų sąlygų, bet pagrindinis principas nesikeis.

Tarkime, kad eismo tinklą reikia optimizuoti taip, kad TP keliaudamos maršrutuose per visus sustojimus kuo mažiau laiko būtų sustoję.

3.3.2. Duomenų aibė

Kad būtų galima įgyvendinti tokią simuliaciją reikia apibrėžti pagrindines sąvokas – žinių bazę, nuo kurios bus pradedamas tyrimas ir į ką bus atsižvelgta – konstantas bei kintamuosius parametrus su jų leistinomis ribomis (jeigu tokios nustatytos):

- Pagrindinės genetinio algoritmo konstantos bus nurodomos prieš paleidžiant programą. Tokios, kaip populiacijos dydis ir atvaizdavimas, bus apibrėžta vėliau. Tarkime, kad pradinė mutacijos tikimybė – 5% kiekvienam nariui, kryžminimo tikimybė – 100%, o iteracijų skaičius apribotas iki 20. Šie parametrai turi būti identiški kiekvienai simuliacijai kai bus keičiami atrinkimo, kryžminimo ar mutacijos operatoriai.
- Tinkamumo funkcija priklauso nuo pasirinktos optimizavimo srities ir galimų kintamųjų. Ji gali kisti tik norint pakeisti optimizavimo sritį. Ji bus apibrėžta vėliau tyrimo eigoje.
- Atrinkimo, kryžminimo bei mutacijos operatoriai bus nurodomi kiekvienos simuliacijos paleidimo pradžioje. Šie parametrai bus keičiami, analizuojant, kaip su skirtingais operatoriais greičiau ar lėčiau konverguoja sprendiniai.
- Simuliacijos konstantos bus nurodomos algoritmo pirmosios iteracijos pradžioje, pasirinktam eismo tinklui automatiškai sugeneravus atsitiktinius TP maršrutus – kelius, o kintamieji bus keičiami kiekvienoje GA iteracijoje.
- Bus simuliuojami tik lengvieji automobiliai remiantis tuo, kad būtent šių TP yra daugiausiai, jos nuolat yra naudojamos susisiekimui, o papildomas išplėtimas kitomis TP galimas. Toliau, tyrimo kontekste TP sąvoka bus naudojama kaip lengvųjų automobilių sinonimas;
- TP greičiai – konstantos nurodytos SUMO simuliacijos programoje;

Visų simuliacijų pradinė duomenų aibė – konstanta – yra iš atviro gatvių žemėlapis (*angl. OpenStreetMap.com*) atviro kodo programos išeksportuotas realių kelių žemėlapis „*.osm.xml*“ formatu. SUMO programos komanda *NETCONVERT* šis žemėlapis konvertuojamas į simuliacijoms priimtina formatą „*.net.xml*“, kuris ir bus naudojamas simuliacijų metu. Tinklas nebus keičiamas simuliacijų metu.

Pirmosios iteracijos metu bus sugeneruojami pirmieji visi tinklo failai skirti redagavimui genetiniu algoritmu optimizacijos metu. Jų redagavimai bus parinkti vėliau. Pirmąjį kartą bus sugeneruojami keturi skirtingi failai:

- Kelių tipų failas („*typ.xml*“) – jame nurodomi atitinkamai pažymėtų kelio atkarpų: prioritetai prieš kitus kelius; numatytas eismo krypčių kiekis viena kryptimi; leidžiamų ir draudžiamų transporto priemonių klasės; nurodymas ar tai vienpusės ar abiejų krypčių kelias; numatytas kelio tipo greitis ir šaligatvio plotis (skirtas tik grafiniam simuliacijos atvaizdavimui).
- Sankryžų / kelių sujungimų failas („*nod.xml*“) – privalomai nurodomi: sankryžų pavadinimas; jų vieta plokštumoje; tipas (pvz.: pirmumo, reguliuojama, pagrindinis iš kairės, aklakelis ir pan.). kiti atributai yra neprivalomi ir gali būti reikalingi atitinkant tik tam tikras sąlygas.
- Kelių failas („*net.xml*“) – privalomai nurodyti: unikalus identifikacinis numeris; identifikacinis sankryžos numeris arba pavadinimas iš kurios ir į kurią turi eiti kelias; kelio tipo identifikacinis numeris – nuoroda į kelių tipų failą. Papildomai galima nurodyti šviesoforų logikos ID, juostų skaičių; maksimalų greitį kelyje; kelio formą; leidžiamų ir draudžiamų TP tipus; ir kita.
- Šviesoforų logikos failas („*ttl.xml*“) – šiame faile nurodomi: identifikacinis pavadinimas egzistuojantis kelių faile; šviesoforų logikos tipas programos ID, programos laiko kompensavimas bei šviesoforo logikos fazių atributai, tokie kaip: trukmė; signalo būsenų sąrašas ir kiti neprivalomi laukai.

Visi šie failai gali būti redaguojami pagal atitinkamas nurodytas reikšmes ar reikšmių ribose, ir iš jų iš naujo bus sugeneruotas tinklas simuliacijai.

Kadangi autoriaus pasirinkta optimizavimo sritis yra TP sustojimų trukmių mažinimas, bus redaguojami tik šviesoforų logikos failai.

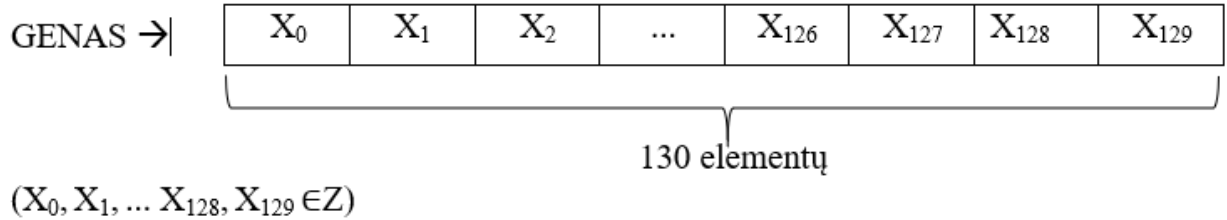
Visų iteracijų metu bus naudojamas tas pats pirmos iteracijos metu atsitiktinai sugeneruotas maršrutų failas, kad nebūtų keičiami pradiniai ir galutiniai TP maršrutų taškai ir būtų galima stebėti ir analizuoti šių maršrutų – bendrai ar atskirai – pakitimus.

3.3.3. Duomenų genetinė reprezentacija

Teisinga duomenų aibės reprezentacija genetiniais duomenimis yra pagrindinis bet kokio GA uždavinys. Kadangi, šiame darbe norima palyginti kokią įtaką turės skirtingi atrinkimo, kryžminimo bei mutacijos operatoriai, ir su kuriais bus greičiausiai pasiektas geriausias rezultatas, tyrime autorius pasirinko genetinei reprezentacijai naudoti šviesoforų logikos fazių trukmes.

Kiekvienas genas – N-tosios populiacijos narys – yra galimas geriausias sprendinys.

Genai bus X elementų ilgio – tiek, kiek fazių turi visos šviesoforų logikos kartu. Šio tyrimo atveju pagal sugeneruoto tinklo duomenis, genai bus 130 elementų ilgio. Elementai – sveikieji skaičiai – sekundės (2 pav.).



pav. 2. Geno reprezentacija

Tarkime, kad negali būti variantų, kad nei viena fazė nebūtų išjungta, t.y. $X_0, X_1, \dots, X_{129} \neq 0$. Reiškia, kad visos fazės būtų įjungtos bent trumpam, jų veikimo laikas turi būti ne mažesnis nei 1.

Tarkime, kad maksimalus fazės veikimo laikas daugiausiai galimas iki 100 sekundžių tam, kad fazių veikimas neužsitęstų be galo ilgai. Tai daroma tam kad nebūtų galimybės susikurti nepravažiuojamoms sankryžoms kitomis kryptimis. Iš to išplaukia fazių – elementų apribojimas: $X_0, X_1, \dots, X_{129} \in [1; 100]$.

3.3.4. Populiacijos dydžio pasirinkimas

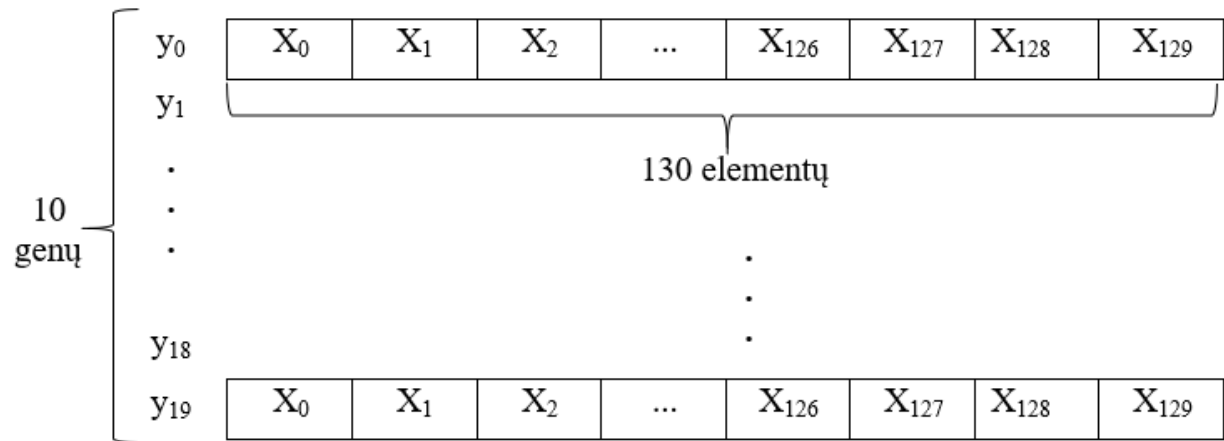
Populiacija yra atsitiktinai sugeneruotų genų masyvas. Apskritai, nėra tikslų nuorodų pasirinkti populiacijos dydžiui nors nuo jos dydžio priklauso ir GA sėkmė. Per mažas jos dydis – GA gali pasiekti per daug netikslų rezultatą, per didelis – GA turės skaičiuoti daug papildomų variantų, kurie gali būti nereikalingi. Vienintelė aiški koreliacija – kuo didesnė ir sunkesnė problema, tuo didesnė turėtų būti ir populiacija. [LL05]

Bendrai yra galimi du populiacijos dydžio pasirinkimo metodai:

- Bandymo metodu rankiniu būdu nustatyti norimą dydį;
- Implementuoti nuo proceso eigos kintantį populiacijos pasirinkimo dydį.

Kadangi šio tyrimo sudėtingumas yra palyginus mažas, bet genai sudaryti net iš 130 elementų šiam tyrimui bus naudojama 10 genų populiacija (3 pav.).

Tinkamiausio populiacijos dydžio pasirinkimas taip pat gali būti labai reikšminga dedamoji dalis gautam rezultatui.



pav. 3. Populiacijos atvaizdavimas

3.3.5. Tinkamumo funkcijos sudarymas

Tinkamumo funkcija kiekvienam GA gali būti unikali nuo užduoties parametų. Ji priklauso nuo pasirinktos optimizavimo srities ir gali kisti tik norint pakeisti optimizavimo sritį. Šio tyrimo atveju, kadangi siekiama sumažinti TP sustojimų laiką, tinkamumo funkcija bus ganėtinai primityvi – tarkime, kad simuliacijoje yra y TP, o kiekvienos TP sustojimo laikas žymimas S_{TPy} – kuo simuliacijos gautų sustojimo laikų suma mažesnė, tuo genas bus geresnis. Potencialiai – idealiausiu atveju galimas nulinis visų TP sustojimų skaičius:

$$\text{Tinkamumo funkcija} = \frac{1}{\sum_{x=0}^y S_x}$$

Kuo didesnė tinkamumo funkcijos reikšmė, tuo geresnis. Kadangi dalyba iš nulio negalima, jeigu gausime simuliacijos sustojimų sumą lygią nuliui, tinkamumo funkciją vertinsime teigiama begalybe.

3.3.6. Genetinių operatorių naudojimas

Kiekvienos simuliacijos metu bus siekiama kuo ankstesne GA iteracija rasti kuo geresnę šviesoforų logikos fazių kombinaciją. Tam, kad kiekvieno GO naudingumą galima būtų vertinti objektyviai, keičiant atrinkimo, kryžminimo bei mutacijos operatorius, kurie aprašyti 3.3. poskyryje, simuliacijų metu pradinė genų populiacija bus naudojama ta pati. Iš viso galimų genetinių operatorių kombinacijų: $3 * 7 * 6 = 126$. Šio tyrimo metu bus ištirtos ne visos galimos kombinacijos.

3.4. Tyrimo duomenų aibė

Papildomai, pradinei SUMO programos konfigūracijai kurti konfigūracijos failą reikia nurodyti simuliacijos pradžios ir pabaigos laiką realaus eismo atžvilgiu – trukmę sekundėmis. Tarkime, kad pradžia bus 0, o pabaiga 1000. Šis laiko tarpas turėtų atspindėti pakankamai realią eismo situaciją susidaryti įvairioms galimybėms.

Apibrėžus genetinį algoritmą, SUMO konfigūracijai reikalingas konstantas, galima nurodyti pradinę konstantų (žr. lentelė 1) bei kintamųjų (žr. lentelė 2) aibes, su kuriais bus tiriama.

lentelė 1 Tyrimo konstantos

Konstanta	Reikšmė
Pradinis tinklo failas	„savanoriuZiedas.net.xml“ (pridėta programos kode aplanke „...\\SimulatedNetworkFile“)
SUMO simuliacijos pradžia	0
SUMO simuliacijos pabaiga	1000
Mutacijos tikimybė, %	20
Kryžminimo tikimybė, %	100
Iteracijų skaičius, vnt.	20
Geno ilgis, el. sk.	130
Populiacijos dydis, genų sk.	10
Elementų apribojimai	$X_0, X_1, \dots, X_{129} \in [1; 100]$
Tinkamumo funkcija	$\sum_{x=0}^y S_x$ kur y – TP kiekis, S_x – sustojimo laikas

lentelė 2 Tyrimo kintamieji.

Kintamasis	Reikšmė
Genetinė reprezentacija	šviesoforų logikos fazių trukmės

Kintamasis	Reikšmė
Atrinkimo operatoriai	Proporcingas atrinkimas; reitingavimo atrinkimas, turnyrinis atrinkimas
Kryžminimo operatoriai	PMX, CX, OX1, OX2, POS, VR, AP
Mutacijos operatoriai	DM, EM, ISM, SIM, IVM, SM

Taip pat tyrimo eigoje bus varijuojamas iteracijų skaičius, populiacijos dydis norint ištirti kaip kis GA skaičiavimas ir galutiniai rezultatai.

3.5. Tyrimo eiga

Tyrimui atlikti autorius pasirinko naudoti anksčiau minėtą SUMO¹ simuliacijos programą dėl keleto priežasčių:

- Tai yra atviro kodo simuliacijos įranga;
- Ji yra nemokama;
- Ja galima simuliuoti eismą be grafinio atvaizdavimo naudojantis tik komandine eilute.
- Mažai sudėtingų funkcionalumų;
- Yra visos paprastam eismo simuliacijai reikalingos funkcijos.

Tyrimui integruoti SUMO simuliacijos programą ir implementuoti genetinį algoritmą buvo parašyta programa² naudojant:

1. JAVA programavimo kalbą (*Java versija – 11.0.1*) ;
2. „Lombok“³ biblioteką, supaprastinant implementavimą.

Kompiuteryje, kuriuo bus simuliuojama turi būti ne mažiau 8GB operatyviosios atminties, procesoriaus sparta – ne mažiau 2.40Ghz.

Programoje visų pirma sugeneruojama atsitiktinių maršrutų schema importuotam pradiniam tinklo failui.

Iš tinklo failo sugeneruojami keturi failai – kelių tipų, sankryžų, kelių ir šviesoforų logikos failai – skirti modifikavimui. Iš šių failų sukuriamas redaguojamo tinklo subjektas (*Angl. entity*), kurio šviesoforų logikos subjektas šiame tyrime bus redaguojamas.

¹ Atsisiųsti iš <https://sumo.dlr.de/docs/index.html>

² Atsisiųsti galima iš <https://github.com/LinasJu/GeneticAlgorithmForTrafficSim>

³ Atsisiųsti iš <https://projectlombok.org/>

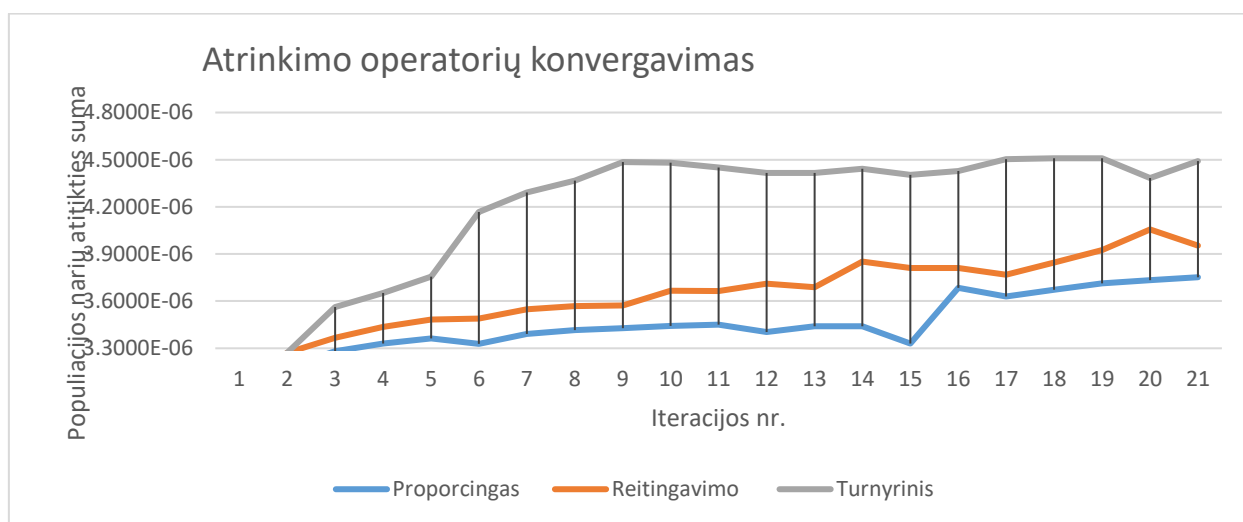
Toliau, šviesoforų logikos subjektas, iš tinklo subjekto, konvertuojamas į reikalingo dydžio pradinę genų populiaciją.

Iš visų genetinių operatorių visų pirma implementuoti visi trys atrinkimo operatoriai – proporcingas, reitingavimo ir turnyrinis; vienas kryžminimo – dalinai sužymėto kryžminimo; ir vienas mutacijos – perkėlimo. Šios implementacijos leis visų pirma palyginti kokią įtaką konvergavimui bei tikslumui turi skirtingi atrinkimo operatoriai su nurodytais pradiniais duomenimis.

Programoje, kiekvienos iteracijos metu atrinkus tėvinius individus yra sukuriamas naujas individas, su juo yra simuliuojamas eismas SUMO programa, ir iš gauto simuliacijos išvesties failo duomenų skaičiuojamas individo atitiktis pagal atitikties funkciją. Iteracijos paskutinis veiksmas – iš esamos populiacijos pašalinamas individas su mažiausia atitikties verte, ir įterpiamas naujai gautas individas.

Iteracijų kartojimas kartojamas su visais trimis atrinkimo operatoriais.

Šių iteracijų su skirtingais GO rezultatais galima palyginti linijiniais grafikais – kiekvieną atrinkimo operatoriaus panaudojimą atitinka skirtingos spalvos linija, vertikalioje ašyje – individų atitikties verčių suma, horizontalioje ašyje – iteracijos skaičius. Pirmajai programai buvo nustatyta: populiacijos dydis – 10 individų, iteracijų skaičius – 20, mutacijos tikimybė – 10%. Kryžminimo tipas – dalinai sužymėtas kryžminimas.



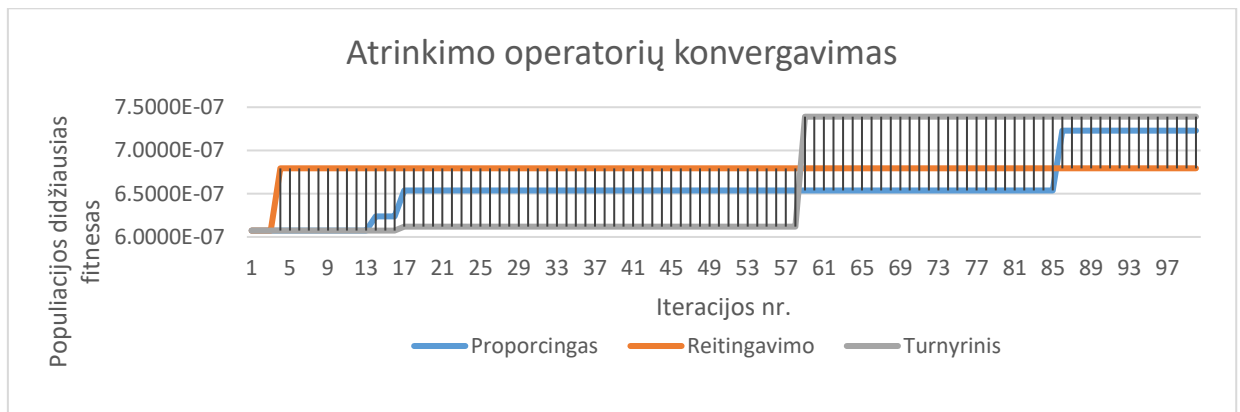
pav. 4 Atrinkimo operatorių atitikties funkcijų sumų konvergavimas su atsitiktine populiacija. Eksperimentas nr. 1

Iš ketvirto pav. grafiko matoma, kad naudojant proporcingą atrinkimo operatorių greičiausiai pasiekta didžiausia populiacijos atitikties suma, tik matoma, kad tęsiant skaičiavimus ši vertė sumažėjo.

Reitingavimo operatorius, lyginant su kitais atrinkimo operatoriais, iš vis labai lėtai konvergavo, bet ir tai nepasiekė aukščiausios kitų operatorių atitikties sumos.

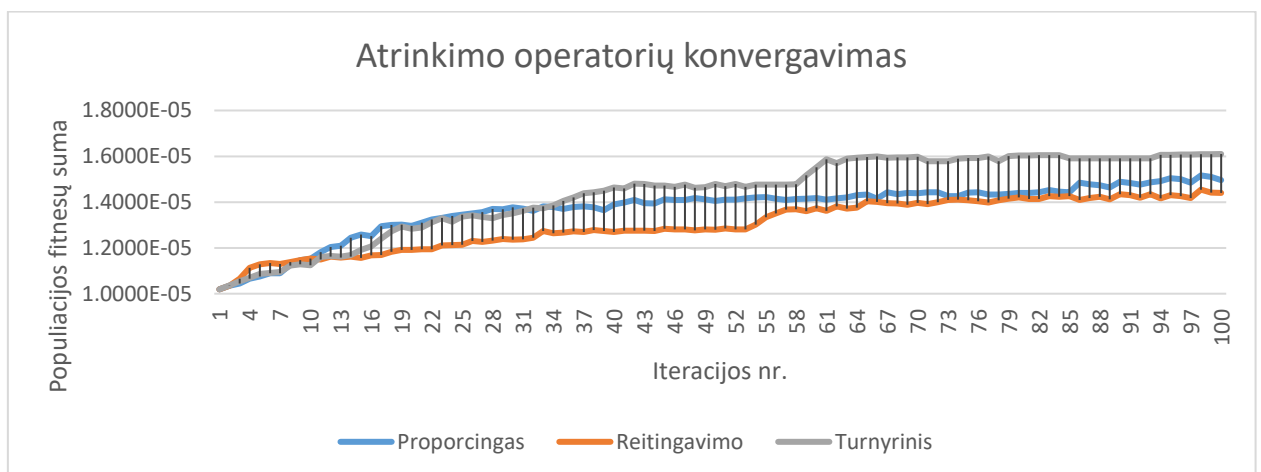
Naudojant turnyrinį operatorių atitikties suma tolygiai konvergavo ir galiausiai pasiekė didžiausią atitikties sumą.

Paleidus antrą programą su kitais parametrais – padidinus iteracijų skaičių iki 100, mutacijos tikimybę iki 20% ir populiacijos dydį iki 30 individų gauta proporcingo atrinkimo operatoriaus didžiausias atitikimo funkcijos rezultatas bei visų atitikties rezultatų sumų grafikai. (žr. pav. 5, 6).



pav. 5 Atrinkimo operatorių didžiausia atitiktis iteracijose. Eksperimentas nr. 2

Pagal pav. 5 taip pat iš antrosios programos matoma, kad per visas 100 iteracijų pagrinde naudojant atrinkimo operatorių, dalinai sužymėtą kryžminimą ir perkėlimo mutaciją su 20% tikimybe, didžiausia atitiktis padidėjo su visais atrinkimo operatoriais. Greičiausiai savo maksimumą pasiekė reitingavimo operatorius, bet per visą laiką didžiausią pasiekė turnyrinis atrinkimas.



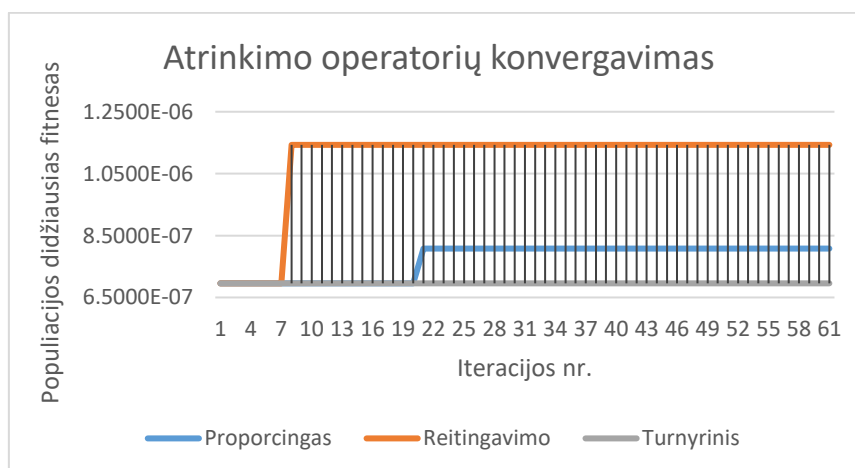
pav. 6. Operatoriaus populiacijų atitikties sumos kiekvienoje iteracijoje. Eksperimentas nr.2

Pagal šesštą pav matoma, kad per 100 iteracijų bendra genų atitikties funkcijų sumos didėjo ganėtinai apylygiai. Iš to galima spręsti, kad nors per tokį iteracijų skaičių dėl didelės populiacijos didžiausia atitiktis padidėjo, bet bendrai iš populiacijos buvo pašalinami prastesni ir mažesnę atitikties funkcijos reikšmę turintys individai.

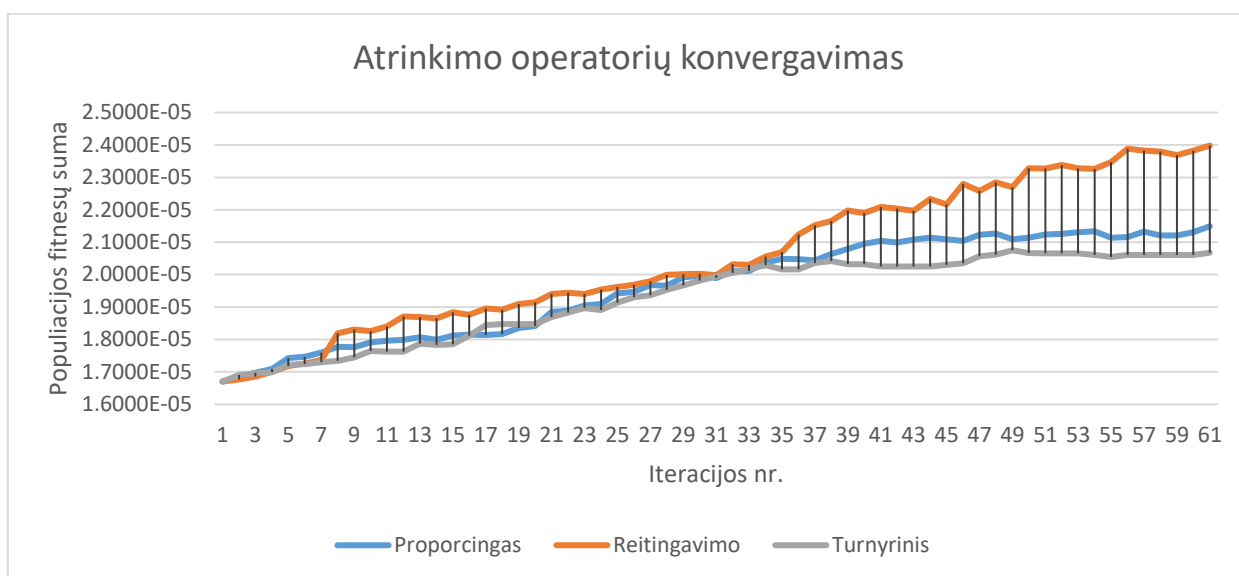
Iš šių dviejų grafikų galima spręsti, kad jeigu populiacija yra apie 30 individų, o iteracijų skaičius – ne mažiau nei 89 – tiek kiek užtruko paskutiniam operatoriui pasiekti savo maksimumą – yra galimybė rasti geresnę atitiktį nei buvo pirminėse populiacijose su visais atrinkimo operatoriais.

Papildomai, palyginus atrinkimo, reitingavimo ir turnyrinio operatorių atitikties sumas, matoma, kad atrinkinėjant su turnyriniu operatoriumi dažniausiai gaunamos didesnės reikšmės sumos nei su kitais operatoriais. Tokie analogiški rezultatai gauti dar tris kartus perskaičius su naujomis pradinių duomenų aibėmis – turnyrinis operatorius visada pasiekia didesnes atitikties funkcijos reikšmes.

Papildomai buvo atliktas dar vienas, trečiasis eksperimentas – simuliacija – su skirtingais atrinkimų operatoriais. Tik šįkart tyrime populiacijos dydis pasirinktas dar didesnis – 40 individų, o iteracijų skaičius beveik per pus mažesnis – 60, o kiti duomenys tie patys, norint patikrinti, ar didesnis individų kiekis turės didesnę įtaką rezultatams. Iš šios simuliacijos gauti du grafikai (žr. pav. 7, 8).



pav. 7. Visų trijų atrinkimo operatorių didžiausia atitiktis iteracijose Eksperimentas nr.3

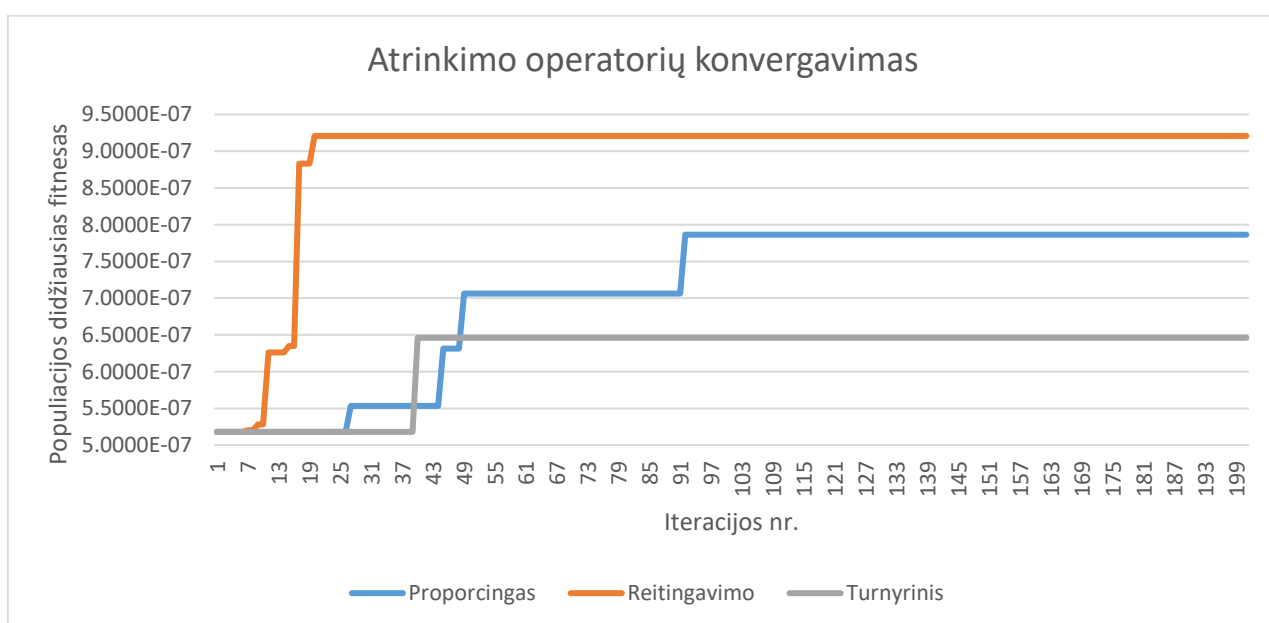


pav. 8. . Visų trijų operatorių populiacijų atitikties sumos kiekvienoje iteracijoje. Eksperimentas nr.3

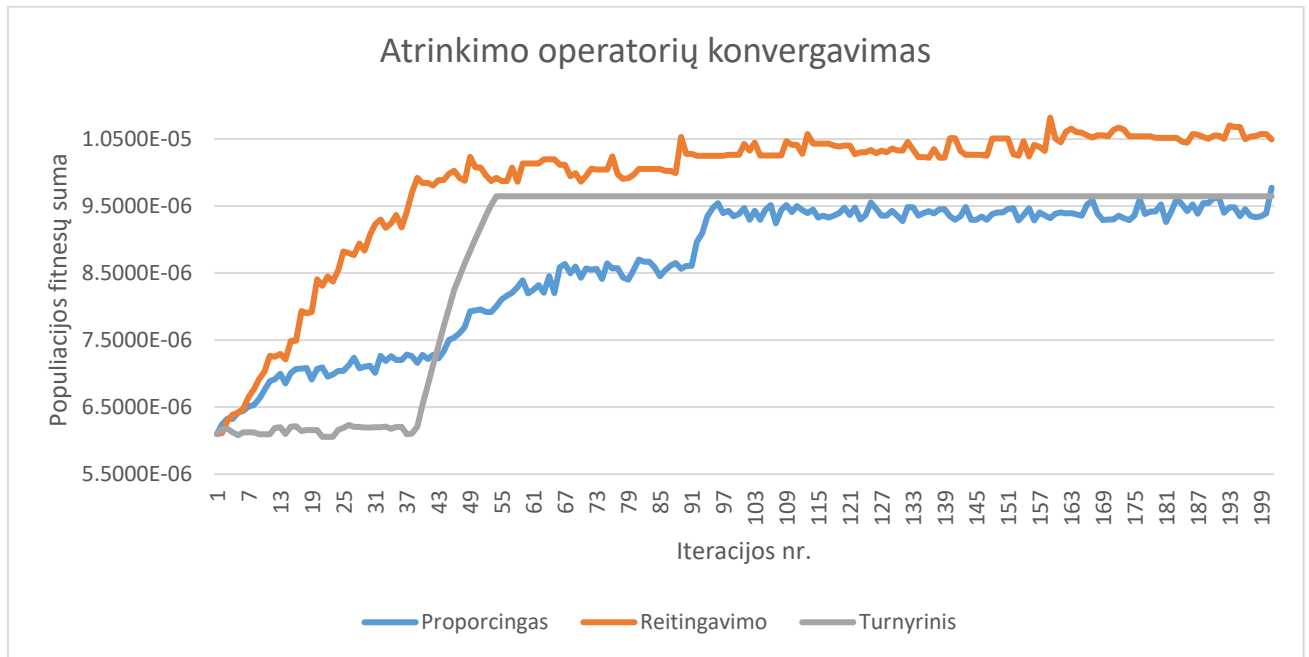
Iš šių grafikų galima matyti, kad esant didesnei nei 30 individų populiacijai, ne taip kaip ankstesniame tyrime, didžiausia atitikties suma gauta su reitingavimo operatoriaus ankstesne iteracija. Naudojant reitingavimo operatorių šiuo atveju buvo naudingiausia, ir jis anksčiausiai pasiekė didžiausią atitikimo reikšmę.

ketvirtajam tyrimo eksperimentui naudota mažesnė populiacija – 15 individų, bet žymiai daugiau iteracijų – 200, norint patikrinti ar mažesnis individų kiekis bet didesnis skaičius iteracijų gali padaryti teigiamą įtaką naudojant skirtingus atrinkimo operatorius.

Iš šio eksperimento gautos dvi duomenų lentelės (žr. pav. 9, 10).



pav. 9. Visų atrinkimo operatorių didžiausių atitikties funkcijų iteracijose palyginimas. Eksperimentas nr. 5



pav. 10. Visų atrinkimo operatorių atitikties funkcijų sumos. Eksperimentas nr. 5

Iš šių naujai gautų diagramų galima matyti, kad populiacijai esant perpus mažesnei nei ankstesniuose tyrimuose, bet esant 200 iteracijų didžiausios atitikties vertės suradimui tai nedaro įtakos – visų trijų operatorių didžiausios vertės pasiektos dar iki šimtosios iteracijos.

Šio tyrimo metu pagal konvergavo greitumą reitingavimo operatorius konvergavo greičiausiai, o prasčiausiai konvergavo proporcingas atrinkimo operatorius.

Pagal autoriaus pasirinktą eismo duomenų genetinę reprezentaciją ją simuliuojant bent trimis skirtingais atrinkimo operatoriais, galima pastebėti, kad:

- iš atsitiktinių pradinės aibės duomenų, net per trumpą laiką – 30-60 iteracijų – galima bent kažkiek optimizuoti eismo simuliaciją.
- Tyrimo atveju, greičiausiai geriausią populiacijos imtį savo funkcijos ribose randa reitingavimo atrinkimo operatorius.
- Tyrimo atveju, optimalus populiacijos dydis – 30-40 individų.
- Iteruojant daugiau nei 100 iteracijų, realios naudos nėra.

Rezultatai ir išvados

Atlikus tyrimo eksperimentus su pradine nurodyta duomenų aibės reprezentacija genetiniais duomenimis, matoma, kad genetinius algoritmus galima panaudoti eismo optimizavimo automatizavimui.

Duomenų aibės reprezentacija kis nuo atitinkamai norimos optimizuoti eismo srities. Tyrimo metu pasirinkta simuliuojamo eismo šviesoforų visų fazių trukmių reprezentacija gali netikti norint optimizuoti kitokią sritį.

Norint pritaikyti GA eismo optimizavimui reikia teisingai apsibrėžti pradinę duomenų genetinę reprezentaciją pagal norimą eismo modelio optimizavimo sritį.

Įmanoma sudaryti genetinę duomenų reprezentaciją iš vienos bei kelių skirtingų duomenų rūšių juos atvaizduojant (užkoduoiant) į pasirinktus ir pritaikytus narius tinkamumo funkcijos skaičiavimui.

Atitikties funkcija atitikimo skaičiavimui gali būti primityvi, ir kiekvienam optimizavimo variantui gali kisti.

Dėl pasirinkto populiacijos dydžio atitinkamai gali pagerėti/suprastėti optimizavimo rezultatas. Tyrimo metu su tyrime nurodytais pradiniais duomenimis, optimaliausias populiacijos dydis – 30 individų. Su mažiau individų – 5-10 vienetų optimizavimas vyko lėčiau ir pasiekta mažesnė atitikimo vertė.

Įmanoma integruoti eismo simuliacijos programą ir automatizuoti procesų iteracijų simuliacijų skaičiavimus. Tyrimo metu buvo integruota SUMO programos simuliacijos paleidimas, gautų duomenų išskaitymas, analizavimas, naujos duomenų aibės sukūrimas simuliacijai.

Norint gauti geresnę populiacijos konvergavimą reiktų ištirti ir kitus kryžminimo bei mutacijos operatorius. Šio tyrimo metu pagrinde buvo lyginami skirtingų atrinkimo – proporcingas, reitingavimo ir turnyrinis – operatorių gaunamų rezultatų konvergavimas su pradine nurodyta duomenų reprezentacija.

Conclusions

After the study was done, it shows that genetic algorithms can be used to automate traffic optimization. In order to apply genetic algorithms to traffic optimization it is necessary to correctly define the initial genetic representation of the data, make fitness function that may be not that complicated but rather quite simple, choose correct size of the population.

Also, the study shows that it is possible to integrate a simple traffic simulation program to automate the process of iteration calculations.

Further, other crossover and mutation operators should be investigated to achieve better convergence of the population.

Literatūros sąrašas

- [ADAC19] ADAC (Allgemeiner Deutscher Automobil-Club). Staubilanz 2018 – Neue Rekordlängen, 2019-01-19. Prieiga per internetą: <https://www.adac.de/der-adac/verein/aktuelles/staubilanz/>
- [TTI15] Published Jointly by The Texas A&M Transportation Institute and INRIX. URBAN MOBILITY SCORECARD, 2015. Prieiga per internetą: <https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-scorecard-2015.pdf>
- [P99] Matti Pursula. Simulation of Traffic Systems – An Overview, 1999. Prieiga per internetą: http://publish.uwo.ca/~jmalczew/gida_5/Pursula/Pursula.html
- [L08] John D. Leonard II, Ph.D. An Overview of Simulation Models in Transportation, 2008. Prieiga per internetą: https://web.archive.org/web/20081101205621/http://www.sisostds.org/webletter/siso/iss_79/art_429.htm
- [NER99] World Scientific Publishing Company. Annual Reviews of Computational Physics VII. (pp.151-202). Kai Nagel, Jörg Esser, Marcus Rickert. LARGE-SCALE TRAFFIC SIMULATIONS FOR TRANSPORTATION PLANNING, 1999. Prieiga per internetą: https://books.google.lt/books?hl=lt&lr=&id=kmjVCgAAQBAJ&oi=fnd&pg=PA151&dq=Transportation+planning+simulations&ots=FzExnoIUCO&sig=MWv2II8DyCRh4Z7iMNpUnpf4DzU&redir_esc=y#v=onepage&q&f=true
- [N06] Daiheng Ni. A Framework for New Generation Transportation Simulation. Proceedings of the 2006 winter Simulation Conference, 2006. Prieiga per internetą: <https://sci-hub.se/10.1109/WSC.2006.322920>
- [KSS98] Ulrich Klein, Thomas Schulze, Steffen Straßburger. Traffic Simulation Based on High Level Architecture, 1998. Prieiga per internetą: <https://sci-hub.se/10.1109/wsc.1998.745965>
- [CLOR03] Lianyu Chu, Henry X. Liu, Jun-Soek Oh, Will Recker. A Calibration Procedure for Microscopic Traffic Simulation. Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, 2003. Prieiga per internetą: <https://sci-hub.se/10.1109/ITSC.2003.1252749>
- [BKA05] Wilco Burghout, Haris N. Koutsopoulos, Ingmar Andreasson. Hybrid Mesoscopic-Microscopic Traffic Simulation. Transportation Research Record: Journal of the Transportation Research Board, 1934(1), 2005. Prieiga per internetą: <https://sci-hub.se/10.1177/0361198105193400123>

- [FMT01] Michael Florian, Michael Mahut, Nicolas Tremblay. A Hybrid Optimization – Mesoscopic Simulation Dynamic Traffic Assignment Model, 2001. Prieiga per internetą: <https://sci-hub.se/10.1109/ITSC.2001.948640>
- [TH99] Martin Treiber, Dirk Helbing. Macroscopic simulation of widely scattered synchronized traffic states, 1999. Prieiga per internetą: <https://sci-hub.se/10.1088/0305-4470/32/1/003>
- [YKB00] Qi Yang, Haris N. Koustopoulos, Moshe E. Ben-Akiva. Simulation Laboratory for Evaluating Dynamic Traffic Management Systems, 2000. Prieiga per internetą: <https://sci-hub.se/10.3141/1710-14>
- [PMO17] Andreas Pell, Andreas Meingast, Oliver Schauer. Trends in Real-time Traffic Simulation, 2017. Prieiga per internetą: <https://reader.elsevier.com/reader/sd/pii/S2352146517304684?token=964A7BC09A8E1723613DE0C2BDE30A1DBEC2A57C8A728347C7C953AA3A19041676269F68D722782A613043DD141FEEEE>
- [W94] Darrell Whitley. A Genetic Algorithm Tutorial, 1994. Prieiga per internetą: http://cobweb.cs.uga.edu/~potter/CompIntell/ga_tutorial.pdf
- [A10] Zakir h. Ahmed. Genetic Algorithm for the Traveling Salesman Problem usign Sequential Constructive Crossover Operator, 2010. Prieiga per internetą: <https://pdfs.semanticscholar.org/a1e6/50daed4ed9c6a403b08e5d50b3ea9f3b5de4.pdf>
- [GL08] Gen, M., & Lin, L. Genetic Algorithms. Wiley Encyclopedia of Computer Science and engineering, 2008. Prieiga per internetą: <https://sci-hub.se/10.1002/9780470050118.ecse169>
- [M99] Mitchell Melanie. An Introductuion to Genetic Algorithms, 1999. Prieiga per internetą: <http://www.boente.eti.br/fuzzy/ebook-fuzzy-mitchell.pdf>
- [Z06] Thomas Zeller. Driving Germany, The Landscape of the german Autobahn, 1930-1970, 2006. Psl.49. Prieiga per internetą: https://books.google.lt/books?id=8JaYz09bSBYC&pg=PP4&lpg=PP4&dq=978-1-84545-309-1&source=bl&ots=rISlxtMjhQ&sig=ACfU3U2OwmtZ_cSj8LKeCfQ-GyLiHdQbXw&hl=lt&sa=X&ved=2ahUKEwil5J_miL_iAhUDuIsKHdhUCQcQ6AEwAXoECAkQAQ#v=onepage&q=width&f=false
- [LL05] Fernando G. Lobo, Claudio F. Lima. A Review of Adaptive Population Sizing Schemes in Genetic Algorithms, 2005. Prieiga per internetą: https://www.cs.york.ac.uk/rts/docs/GECCO_2005/Workshop%20and%20tutorials/gecco05/papers/0228.pdf

- [GD91] Goldberg, D. E., & Deb, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. Foundations of Genetic Algorithms, 1991. Psl.69–93. Prieiga per internetą: <https://sci-hub.im/10.1016/b978-0-08-050684-5.50008-2>
- [J87] Jonh J. Grefenstette - Editor. Genetic algorithms and theri applications: proceedings of the second international conference on genetic algorithms, 1987. Psl.63. Prieiga per internetą: https://books.google.lt/books?hl=en&lr=&id=MYJ_AAAAQBAJ&oi=fnd&pg=PA9&dq=proportionate+reproduction&ots=XwuNyi5DIF&sig=sO12DTCyFd7qfMQz9c8K7vEHOHU&redir_esc=y#v=onepage&q=proportionate%20&f=false
- [LKMY96] Pedro Larrafiaga, Cindy M. H. Kuijpers, Roberto H. Murga, and Yosu Yurramendi. Learning Bayesian Network Structures by Searching for the Best Ordering with Genetic Algorithm, 1996. Psl.489-490. Prieiga per internetą: <https://www.cse.unr.edu/~sjose/Papers%20Referenced/Learning%20Bayesian%20Network%20Structures%20by%20Searching%20for%20the%20Best%20Ordering%20with%20Genetic%20Algorithm.pdf>

Priedas Nr. 1

Šioje lentelėje yra palyginamos įvairios eismo simuliacijos programos – tiek mokamos, tiek atviro kodo nemokamos programos. Jos lyginamos pagal įvairias grupes, tokias kaip eismo modelių tipų palaikomumas, modelių dydžių apribojimai sumanaus eismo funkcijos ir kitos.

Pagal šias kategorijas simulatoriai buvo sureitinguoti ir išskirti daugiausiai bei mažiausiai atitinkantys kriterijų.

lentelė 3. Eismo simuliacijos programų palyginimas [PMO17]

Key: O..... No X..... Yes ! (further) improvement [empty] not given	Aimsun 7	Cube Voyager/Avenue	DRACULA	DynaMIT	FLEXYT-II	FreeSim	Integration	MITSimLab	PELOPS	PLANSIM-T	Quadstone Paramics	S-Paramics	SITRA-B+	SUMO	TransModeler	TSIS-CORSIM	VISSIM
Flow model category																	
macroscopic model	X	O	O			X		O	O		O	O		O	X	O	O
mesoscopic model	X	X	O	X		O		O	O		O	O		X	X	O	O
microscopic model	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Model size restrictions																	
limit of crossings		X	O			O			X!		X	O	X	O	O	X	
limit of links		X	O			O			O		X	O	X	O	O	O	
limit of edges			O			O						O		O	O		
limit of street categories	X!		O			O			X!		X	O		O	O		
limit of lanes	X!		O			O			X		X	O		O	O	X	
limit of vehicle types		X	X			O			X		X	O		O	O		
limit of driver profiles			O			O			O		X	O		O	O		
limit of public transportation routes			O			O					X	O		O	O	X	
ITS functionalities																	
co-ordinated traffic signals	X		X		X	O	X	X	X!	X	O!	X	X	X!	X!	X	X
adaptive traffic signals	X	X	X		X	O	X	X	X!	X	X!	X	X	X!	X!	X	X
public transport priority	X	X	X		X	O	X	X	O	X	X	X	X	O!	X	X	X
ramp metering					X	O	X	X	X		X	X		O!	X	X	X
freeway flow control	X			X	X	X	X	X	X		X	X		O	X	X	X
adaptive cruise control						O			X!			O		O		X	
automated highway system						X			X!	X		O		O			
autonomous vehicles						O			X!			O		O			
v2v/v2i communication						X			X!		X!	O		X!	X!		X
automatic debiting & toll plazas					X	O	X		O		X	X		O	X	X	X
zone access control					X	O			O	X	X	X		X!	X!		
incident management	X		X	X	X	O	X		O		X	X	X	O	X	X	
variable message signs	X		X	X		O	X	X	X!	X	X	X		X!	X!		
static route guidance						X	X	X	X!	X	X	X	X	X!	X		X
dynamic route guidance	X	X		X		X	X	X	O!	X	X	X	X	X!	X		X
vehicle type specific barred turning movement and link/lane closures						O			O		X	X		X!	X	X	
multimodal traffic	X	X				O		X	O		X	X		X!	X		
parking guidance						O			O	X	X	X	X	O!	X		X
public transport information						O	X		O	X	X	O		O	O		X
probe vehicles						X	X		X!	X	X	O	X	X!	X		X
Street restrictions																	
speed limits	X	X	X		X		X	X			X		X	X	X		
weight	X				O			O			X		O	O			
vehicle height	X				O			O			X		O	O			
vehicle width	X				O		X	O			X		O	O			
vehicle type specific lane use (e.g. bus lane)	X										X		X	X	X		

