

VILNIAUS UNIVERSITETAS

Gintaras Vaira

GENETINIS ALGORITMAS TRANSPORTO MARŠRUTŲ SUDARYMO
UŽDAVINIAMS SPREŠTI

Daktaro disertacijos santrauka
Technologijos mokslai, informatikos inžinerija (07 T)

Vilnius, 2014

Disertacija rengta 2009-2013 m. Vilniaus universiteto Matematikos ir informatikos institute.

Mokslinė vadovė

doc. dr. Olga Kurasova (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Disertacija ginama Vilniaus universiteto Matematikos ir informatikos instituto Informatikos inžinerijos mokslo krypties taryboje:

Pirmininkas

prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Nariai:

prof. habil. dr. Juozas Augutis (Vytauto Didžiojo universitetas, fiziniai mokslai, matematika – 01 P),

prof. habil. dr. Rimantas Barauskas (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09 P),

prof. habil. dr. Artūras Kaklauskas (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07 T),

prof. dr. Julius Žilinskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Oponentai:

prof. dr. Eduardas Bareiša (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07 T),

dr. Virginijus Marcinkevičius (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties tarybos posėdyje 2014 m. birželio 10 d. 13 val. Vilniaus universiteto Matematikos ir informatikos institute, 203 auditorijoje.

Adresas: Akademijos g. 4, LT-08663 Vilnius, Lietuva.

Disertacijos santrauka išsiuntinėta 2013 m. gegužės 9 d.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje.

VILNIUS UNIVERSITY

Gintaras Vaira

GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM

Summary of Doctoral Dissertation

Technological Sciences, Informatics Engineering (07 T)

Vilnius, 2014

Doctoral dissertation was prepared at Institute of Mathematics and Informatics of Vilnius University in 2009-2013.

Scientific Supervisor

Assoc. Prof. Dr. Olga Kurasova (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

The dissertation will be defended at the Council of the Scientific Field of Informatics Engineering at the Institute of Mathematics and Informatics of Vilnius University:

Chairman

Prof. Dr. Habil. Gintautas Dzemyda (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Members:

Prof. Dr. Habil. Juozas Augutis (Vytautas Magnus University, Physical Sciences, Mathematics – 01 P),

Prof. Dr. Habil. Rimantas Barauskas (Kaunas University of Technology, Physical Sciences, Informatics – 09 P),

Prof. Dr. Habil. Artūras Kaklauskas (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07 T),

Prof. Dr. Julius Žilinskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Opponents:

Prof. Dr. Eduardas Bareiša (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07 T),

Dr. Virginijus Marcinkevičius (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

The dissertation will be defended at the public meeting of the Scientific Council of Science of Informatics Engineering in the auditorium number 203 at the Institute of Mathematics and Informatics of Vilnius University, at 1 p. m. on 10th of June 2014.

Address: Akademijos g. 4, LT – 08663 Vilnius, Lithuania.

The summary of the doctoral dissertation was distributed on 9th of May 2014.

A copy of the doctoral dissertation is available for review at the Library of the Vilnius University.

Įvadas

Tyrimų sritis

Transporto maršrutų sudarymo uždaviniai (angl., *vehicle routing problem* (VRP)) yra plačiai nagrinėjama kombinatorinė problema, kuria domisi egzistuojančių bei naujai kuriamų algoritmų tyrėjai. Paprastai VRP uždavinys yra formuluojamas kaip vieno krovinių paskirstymo punkto, aibės klientų bei kelių transporto priemonių uždavinys, kurio tikslas yra surasti tokį maršrutą, kuris minimizuotų bendrą kainą aptarnaujant visus klientus. Transporto maršrutų sudarymo uždaviniuose yra galimi apribojimai. Literatūroje nagrinėjami įvairūs transporto maršrutų sudarymo uždavinių tipai, kur uždaviniai yra sugrupuoti priklausomai nuo įtraukiamų apribojimų. Dažniausiai pasitaikantys transporto maršrutų sudarymo uždavinių tipai yra šie: VRP su talpos apribojimais (angl., *capacitated VRP* (CVRP)), kur transporto priemonių talpa yra apribota; VRP su laiko langais (angl., *VRP with time windows* (VRPTW)), kur klientai gali būti aptarnaujami tik nustatytu laikotarpiu ar laikotarpiais; VRP su keliais krovinių paskirstymo punktais, iš kurių klientams gali būti pristatytos prekės (angl., *VRP with multiple depots* (MDVRP)); VRP su surinkimais ir pristatymais (angl., *VRP with pick-up and delivery* (VRPPD)), kur yra apibrėžiamos sąlygos surinkti prekes iš vienu vietų bei pristatyti į kitas vietas. Minėtiems uždaviniams spręsti literatūroje yra siūlomi įvairūs būdai, tame tarpe ir euristiniai.

Pastaruoju metu VRP uždaviniai pritraukia daug dėmesio dėl populiarėjančių įvairių geografinių sistemų ir technologijų bei jų panaudojimo logistikos ir transportavimo srityje. Vis daugiau logistikos kompanijų stengiasi pasinaudoti egzistuojančiomis technologijomis tam, kad galėtų geriau organizuoti prekių pervežimus. Tai gali būti įvairios logistikos sistemos, apjungtos su plačiai naudojamomis pozicionavimo sistemomis. Sėkmingas transporto maršrutų sudarymo uždavinių sprendimas yra vienas iš transportavimo išlaidų sumažinimo būdų. Pavyzdžiui, tinkamesnis įvairiose srityse (pašto pristatyme, prekių pristatyme į parduotuves, degalų pristatyme į degalines ir t.t.) naudojamų transporto priemonių maršrutų planavimas gali padėti sutaupyti degalų, pinigų ir/arba laiko, kuris galėtų būti skirtas naujų klientų aptarnavimui. Maršrutų organizavimas versle gali padėti sumažinti taršą bei daryti įtaką ekologiniams aspektams, kurie yra labai plačiai aptarinėjami šiandieniniame pasaulyje.

Problemos aktualumas

Mokslinėje literatūroje galima rasti algoritmų, skirtų vienam arba kitam VRP uždavinių tipui. Tokie algoritmai yra sukurti siekiant išspręsti tik tam tikrą uždavinį su tam tikrais apribojimais. Nors minėti VRP uždaviniai ir atspindi specifines realias situacijas, šios pavienės situacijos neatspindi visos bendros problemos. Tokie VRP uždaviniai yra dažnai kritikuojami dėl to, kad yra per daug akcentuojami tam tikri modeliai, kurie yra paremti ne visai realiomis prielaidomis. Realūs VRP uždaviniai su įvairiais apribojimais apibendrina tipinį VRP uždavinį. Tokie uždaviniai yra vadinami bendri transporto maršrutų sudarymo uždaviniai (angl., *rich vehicle routing problem* (RVRP)). Jų sprendimas išlieka aktuali šiandieninių tyrimų sritis.

Yra kuriami ir tiriami įvairūs tikslūs ir euristiniai algoritmai tam, kad išspręstų transporto maršrutų sudarymo uždavinius, kurie yra NP sudėtingumo. Nors tikslūs algoritmai randa optimalų sprendinį, didėjant uždavinio apimčiai skaičiavimo laikas

ženkliai padidėja. NP sudėtingumo uždaviniams spręsti egzistuoja įvairūs euristiniai metodai, tačiau lokaliai paieškos ir euristinių algoritmų efektyvumas gali priklausyti nuo uždavinio duomenų, pavyzdžiui, apribojimų, arba gali reikėti papildomų duomenų algoritmui apmokyti. Taipogi siekiant didesnio efektyvumo skaičiavimuose yra nagrinėjami ir tiriami įvairūs hibridiniai algoritmai, susidedantys iš kelių algoritmų.

Metaeuristikos yra algoritmai, skirti spręsti sudėtingus uždavinius, kurių sprendimas kitais algoritmais tampa per daug sudėtingu arba reikalauja daug laiko. Vienas iš metaeuristikos algoritmų, nagrinėjamų VRP uždaviniams spręsti, yra genetinis algoritmas (angl., *genetic algorithm* (GA)), paremtas evoliucijos teorijos idėjomis. Genetiniai algoritmai remiasi principu „išlieka tik tinkamiausi“. Šie algoritmai dirba su subjektais, dar kartais vadinamais chromosomomis, kurių kiekvienas reprezentuoja galimą uždavinio sprendinį. Paprastai genetiniuose algoritmuose yra sukuriama pradinė sprendinių aibė, vadinama populiacija. Kiekvienoje sekančioje iteracijoje GA panaudodamas kryžminimo (angl., *crossover*) bei mutacijos (angl., *mutation*) operacijas iš ankstesnės generacijos parinktų sprendinių-tėvų sukuria naują potencialų sprendinių-palikuonį. Toks sprendinių generavimo ir įvertinimo procesas tęsiasi tol, kol sustojimo kriterijai nėra įvykdyti.

Uždaviniuose, kuriuose yra papildomi apribojimai ir sąlygos, standartinis genetinis algoritmo veikimas yra ribotas. Dėl stochastinių savybių uždaviniuose su apribojimais GA gali skaičiuoti labai ilgai, kol bus surastas tinkamas sprendinys. Leistinų sprendinių aibė tokiuose uždaviniuose yra mažesnė nei visų sprendinių aibė, tačiau GA generuoja sprendinius visoje sprendinių aibėje, įskaitant ir neleistinus sprendinius. Uždavinių su apribojimais sprendimui genetiniuose algoritmuose dažnai yra papildomai naudojami atstatymo (angl., *repair*) ir patobulinimo (angl., *improvement*) metodai. Tokie metodai yra pritaikyti išspręsti tam tikrus specifinius apribojimų neatitikimus sprendiniuose tam, kad gautų sprendinius, priklausančius leistinų sprendinių aibei. Visgi vienų apribojimų neatitikimo taisymas gali iššaukti kitų apribojimų neatitikimus. Tokio algoritmo taikymas kitokiems uždaviniams spręsti gali pateikti netinkamus sprendinius, nes algoritmas yra sunkiai papildomas naujų apribojimų sprendimu. Šie specializuoti algoritmai gali būtų sunkiai pritaikomi bendro transporto maršrutų sudarymo uždavinio sprendimui.

Darbo tikslas ir uždaviniai

Šios disertacijos tikslas yra sukurti naują, transporto maršrutų sudarymo uždaviniams spręsti skirtą genetinį algoritmą, kuriame apribojimų sprendimas būtų įtrauktas į genetinio algoritmo operatorius bei kuris efektyviai spręstų bendrą transporto maršrutų sudarymo uždavinį.

Šiam tikslui pasiekti yra iškelti tokie uždaviniai:

- Išnagrinėti egzistuojančius genetinius algoritmus, skirtus transporto maršrutų sudarymo uždaviniams spręsti.
- Išanalizuoti būdus, taikomus transporto maršrutų sudarymo uždavinių su apribojimais sprendimui genetiniuose algoritmuose, bei išnagrinėti paieškos intensyvavimo genetinių algoritmų operatoriuose būdus.
- Išanalizuoti egzistuojančius bendro transporto maršrutų sudarymo uždavinio apibrėžimus bei formulavimus.
- Pasiūlyti naują genetinį algoritmą bendro transporto maršrutų sudarymo uždavinio sprendimui, kuriame apribojimų sprendimas būtų įtrauktas į genetinio algoritmo operatorius ir išsprendžiamas kiekvienoje iteracijoje.

- Iširti Dijkstra trumpiausio kelio paieškos algoritmo pagreitinimo būdus siekiant pasiūlytą genetinį algoritmą efektyviai taikyti realių transporto maršrutų sudarymo uždavinių, į kuriuos yra įtraukti kelių tinklo duomenys, sprendimui.
- Įvertinti pasiūlytą genetinį algoritmą, testuojant jį su moksliniuose tyrimuose naudojamais testiniais uždaviniais bei palyginant gautus rezultatus su kitų žinomų genetinių algoritmų rezultatais.

Darbo rezultatų praktinė reikšmė

Darbo rezultatų praktinė reikšmė yra ši:

- Pasiūlytas genetinis algoritmas gali būti lanksčiai pritaikomas realiems transporto maršrutų sudarymo uždaviniams spręsti. Įmonėms, kurios susijusios su transportavimu, šio algoritmo taikymas gali padėti sumažinti kelionės kelią ir laiką, taip sumažinant ir išlaidas. Algoritmas taip pat gali būti pritaikytas dinaminiam transporto maršrutų sudarymo uždavinio perskaičiavimui tuo atveju, kai atsiranda papildomų duomenų (naujas užsakymas iš kliento, sugedo viena iš transporto priemonių vykdant transportavimo užduotį ir t. t.).
- Pasiūlytas algoritmas gali būti pritaikytas ir kitiems uždaviniams, kurie yra išreiškiami grafu bei kurių sprendinys priklauso nuo jo komponentų eiliškumo.
- Dalis tyrimo rezultatų panaudota vykdant 2010-07-28 inovacinio čekio gavimo trišalę sutartį Nr. 31V-79 tarp Matematikos ir informatikos instituto, UAB „AKTKC – Apsaugos centras“ ir Mokslo, inovacijų ir technologijų agentūros „Maršruto optimizavimo tarp N taškų algoritmas ir kuro lygio duomenų iš transporto priemonės matematinio vidurkinimo ir nuokrypių (kuro užpylimo/nupylimo) fiksavimui algoritmas“.

Tyrimo metodai

Siekiant surinkti ir apibendrinti kitų tyrimų rezultatus buvo naudojami apžvalginio tyrimo bei sisteminės analizės metodai. Siekiant įvertinti pasiūlytus metodus ir algoritmus, palyginant gautus rezultatus su kituose tyrimuose gautais rezultatais buvo naudojami eksperimentinio tyrimo bei apibendrinimo metodai.

Ginamieji teiginiai

1. Įterpimo euristika genetiniuose algoritmuose gali būti panaudota ne tik pradinės populiacijos sudarymui, bet ir leistinų dalinių sprendinių kūrimui genetiniuose operatoriuose įvertinant apribojimus. Pakartotinai taikant atsitiktinio įterpimo euristiką sprendiniai populiacijoje yra diversifikuojami ir generuojant tik leistinus sprendinius paieška neleistinų sprendinių aibėje nėra vykdoma. Taip yra išvengiama nebūtinų skaičiavimų ir padidinamas bendras skaičiavimo greitis.
2. Kryžminimo operatoriai, kurie paremti bendros elementų sekos tarp dviejų sprendinių-tėvų paieška ir išsaugojimu, gali intensyvinti optimalaus sprendinio paiešką. Skirtingai nuo tradicinių kryžminimo operatorių, kuriuose sprendinys-palikuonis yra sukuriamas iš sprendinių-tėvų dalių, nauji kryžminimo operatoriai nustato sprendinių suardymo lygį išsaugodami tas sprendinių-tėvų dalis, kurios yra bendros, taip išsaugodami sprendinių dalis, kurios turi didesnę tikimybę būti tinkamai sukonstruotos.
3. Genetinis algoritmas, paremtas leistino įterpimo metodu genetiniuose operatoriuose, kryžminimo operatoriais, kurie išsaugo bendras sprendinių-tėvų dalis, bei antros populiacijos naudojimu mutacijos operatoriuje, per trumpą laiką randa panašius arba geresnius sprendinius lyginant su kitais genetiniais algoritmais. Antros populiacijos naudojimas mutacijos operatoriuje didina

sprendinių diversifikavimą populiacijoje bei bendrą genetinio algoritmo efektyvumą. Pasiūlytas genetinis algoritmas yra pritaikomas bendram transporto maršrutų sudarymo uždaviniui spręsti.

Pasiūlyti sprendimai ir mokslinis naujumas

- Pasiūlyti genetiniai operatoriai, paremti gausios aplinkos paieškos (angl., *large neighbourhood search* (LNS)) išardymo ir atkūrimo (angl., *destroy and reconstruct*) naudojimu, kur atsitiktinio įterpimo euristika genetiniuose operatoriuose yra naudojama kaip atkūrimo metodas. Įterpimo euristikos metodas yra papildytas apribojimų įvertinimu tam, kad būtų išvengta sprendinių kūrimo neleistinų sprendinių aibėje bei pagreitinti skaičiavimai. Dėl atsitiktinio įterpimo euristikos naudojimo yra išsaugomos stochastinės genetinio algoritmo savybės ir diversifikuojami sprendiniai populiacijoje.
- Pasiūlyti kryžminimo operatoriai, kurie kuria sprendinius-palikuonis remdamiesi bendrų dalių paieška tarp sprendinių-tėvų. Skirtingai nuo tradiciškai naudojamų kryžminimo operatorių, kurie sukuria palikuonis iš sprendinių-tėvų dalių, pasiūlyti kryžminimo operatoriai identifikuoja ir išsaugo tas sprendinių dalis, kurios yra bendros abiejuose sprendiniuose-tėvuose, taip intensyvinant optimalaus sprendinio paiešką. Ilgiausios bendros didėjančios sekos paieškos (angl., *longest common increasing sequence* (LCIS)) naudojimas kryžminimo operatoriuje identifikuoja ir išsaugo pagrindinę transporto maršrutų savybę – maršruto seką.
- Pasiūlytas genetinis algoritmas, į kurį įtraukta įterpimo euristika, leistinių sprendinių kūrimas, bendrų dalių paieška kryžminimo operatoriuose bei antros populiacijos naudojimas mutacijos operatoriuje. Sprendiniai, gauti antroje populiacijoje, išlieka konkurencingi pagrindinėje sprendinių populiacijoje: tokie sprendiniai turi didesnę tikimybę būti parinkti naujų sprendinių generavimui, taip didinamas sprendinių diversifikavimas populiacijoje. Pasiūlytas algoritmas randa sprendinį per trumpą laiką ir sprendiniai yra geresni arba lygūs lyginant su kitų genetinių algoritmų sprendiniais. Naujo sukurto genetinio algoritmo pranašumas yra tas, kad šis algoritmas gali būti taikomas bendro transporto maršrutų sudarymo uždaviniui spręsti, kurio apibrėžimas bei formulavimas taip pat yra pateiktas šiame tyrime.

Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 3 recenzuojamuose periodiniuose moksliniuose leidiniuose, vienas iš jų yra referuojamas „Thomson Reuters Web of Science“ duomenų bazėje ir turintis citavimo indeksą, bei 2 kituose moksliniuose leidiniuose. Tyrimų rezultatai pristatyti ir aptarti 5 tarptautinėse konferencijose, vykusiose užsienyje.

Darbo apimtis

Disertaciją sudaro įvadas, 3 skyriai, išvados, literatūros sąrašas. Kiekvieno skyriaus, išskyrus įvadą ir išvadas, pabaigoje yra pateikiamas apibendrinimas. Bendra disertacijos apimtis – 140 puslapių, 26 paveikslėliai ir 19 lentelių.

Įvade aprašoma tyrimų sritis ir aktualumas, pateikiamas problemos apibrėžimas, aptariami darbo tikslai ir uždaviniai, tyrimo metodai, pateikiama darbo rezultatų praktinė reikšmė, mokslinis naujumas, ginamieji teiginiai ir darbo rezultatų aprobavimas.

I skyriuje pateikiama transporto maršrutų sudarymo uždavinių bei jų sprendimų apžvalga, taip pat detaliau apžvelgiami genetiniai algoritmai, skirti transporto maršrutų sudarymo uždaviniams spręsti.

II skyriuje aprašomas pasiūlytas, transporto maršrutų sudarymo uždaviniams spręsti skirtas genetinio algoritmas.

III skyriuje pateikiami eksperimentiniai pasiūlytų algoritmų įvertinimai.

Išvados pateikiamos darbo išvados.

1. Transporto maršrutų sudarymo uždaviniai: apžvalga

Šioje disertacijos dalyje yra apžvelgiami literatūroje nagrinėjami VRP uždaviniai bei analizuojami apribojimai. Analitinėje apžvalgoje yra analizuojami įvairūs euristiniai algoritmai, skirti VRP uždaviniams spręsti. Detaliau nagrinėjami genetiniai algoritmai ir jų savybės, genetinių algoritmų taikymas VRP uždavinių sprendimui bei tokių algoritmų modifikacijos, skirtos leistinių sprendinių genetiniuose algoritmuose radimui. Šioje disertacijos dalyje taip pat aptariami įterpimo euristikos taikymai genetinių algoritmų kryžminimo operatoriuose, generuojančiuose leistinus sprendinius. Be to šiame skyriuje apžvelgiami Dijkstra trumpiausio kelio paieškos algoritmo pagreitinimo būdai.

Literatūroje yra nagrinėjami skirtingi transporto maršrutų sudarymo uždaviniai. Šie uždaviniai skiriasi priklausomai nuo įtraukiamų specifinių apribojimų, kurie apibrėžia tam tikrą specifinę realią situaciją. Tokių uždavinių sprendimui yra kuriami įvairūs algoritmai, jų modifikacijos bei įvairūs tokių algoritmų hibridiniai variantai. Tik labai nedaug tyrimų yra susiję su algoritmu, kurie galėtų būtų taikomi didesnei aibei VRP uždavinių sprendimui, kūrimu.

Genetinis algoritmas yra metaeuristinis algoritmas, kuris taip pat yra taikomas VRP uždaviniams spręsti. Tai stochastinis algoritmas, kuris yra paremtas evoliucijos teorijos idėjomis. Genetinių algoritmų paieška priklauso nuo kelių faktorių: atrankos spaudimo (angl., *selective pressure*) bei populiacijos diversifikavimo (angl., *population diversity*). Šie faktoriai vaidina svarbų vaidmenį genetiniuose algoritmuose: atrankos spaudimas apibrėžia optimalaus sprendinio paieškos intensyvinimą kiekvienoje generacijoje reprodukcijai pasirenkant geresnius sprendinius iš populiacijos, o populiacijos diversifikavimu yra gaunama nehomogeninė sprendinių populiacija. Nauji sprendiniai yra gaunami iš ankstesnės generacijos parinktų sprendinių taikant genetinius operatorius. Sprendžiant uždavinius su apribojimais genetiniai operatoriai gali generuoti sprendinius visoje sprendinių aibėje, įskaitant ir neleistinių sprendinių aibę. Kai yra generuojami neleistiniai sprendiniai, reikalingos papildomos priemonės leistinam sprendiniui rasti. Genetiniuose algoritmuose uždavinių su apribojimais sprendimui dažniausiai yra naudojami baudos metodai (angl., *penalty methods*), tačiau baudos metodai vis tiek nedraudžia generuoti sprendinių visoje sprendinių aibėje. Kiti būdai apima atstatymo (angl., *repair*) ir patobulinimo (angl., *improvement*) metodų naudojimą genetiniuose algoritmuose sprendžiant uždavinius su apribojimais. Visgi tokie metodai dažnai yra sukurti specifiniams apribojimams spręsti ar specifiniam tikslui optimizuoti ir yra sunkiai pritaikomi kitai problemai su kitais apribojimais ar su kitu optimizavimo tikslu. Tad vis dar išlieka algoritmų, kurie galėtų būti taikomi spręsti įvairesnius transporto maršrutų sudarymo uždavinius, poreikis.

Trumpiausio kelio paieška yra svarbi, kai kalbama apie transporto maršrutų sudarymo uždavinių, kurie apima ir kelių tinklą, sprendimą. Priklausomai nuo skirtingų paieškos duomenų, pavyzdžiui, maksimalus leistinas svoris važiuojant tiltu ir transporto priemonės svoris, arba dinaminė spūsčių keliuose informacija atskirais laiko momentais, gali būti randami skirtingi trumpiausieji keliai. Disertacijoje yra nagrinėjami Dijkstra trumpiausio kelio paieškos algoritmo būdai siekiant perskaičiuoti trumpiausius kelius

realiame kelių tinkle, kuomet yra sprendžiamas realus transporto maršrutų sudarymo uždavinys. Viena iš Dijkstra algoritmo modifikacijų yra dvikryptis Dijkstra algoritmas. Šiame algoritme apibrėžti sustojimo kriterijai, kuriuos taikant trumpiausio kelio paieškoje rezultatas gali ženkliai skirtis nuo trumpiausio kelio.

2. Naujas algoritmas transporto maršrutų sudarymo uždaviniams spręsti

Šioje disertacijos dalyje yra pasiūlyti genetinio algoritmo, skirto transporto maršrutų sudarymo uždaviniams spręsti, komponentai, didelis dėmesys yra skiriamas sprendinių perdarymui genetinio algoritmo kryžminimo bei mutacijos operatoriuose. VRP uždavinys yra apibrėžiamas kaip grafas $G = (N, E)$, kuris susideda iš mazgų aibės $N = \{n_0, n_1, \dots, n_k\}$, žyminčios krovinių paskirstymo punktą n_0 bei reikalingas aplankyti vietas $N \setminus \{n_0\}$, ir aibės $E = \{e_{ij}\}$, žyminčios trumpiausius kelius tarp šių vietų. Sprendinys x pasiūlytame genetiniame algoritme yra išreikštas kaip maršrutų, kuriuose apribojimai nėra pažeidžiami, aibė R bei į maršrutus dėl pažeidžiamų apribojimų neįtrauktų mazgų aibė U :

$$\begin{aligned}
 x &= (R = \{r_1, \dots, r_t\}, U = \{n_1, \dots, n_u\}) \\
 r_i &= (N_i, A_i) \\
 N_i &= \{n_0, n_{i_1}, \dots, n_{i_p}\} \\
 A_i &= \{a_{i_1} = (n_0, n_{i_1}), a_{i_2} = (n_{i_1}, n_{i_2}), \dots, a_{i_p} = (n_{i_{p-1}}, n_{i_p}), a_{i_{p+1}} = (n_{i_p}, n_0)\} \\
 N_v &= \{N_1 \cup \dots \cup N_t\} \setminus n_0
 \end{aligned}$$

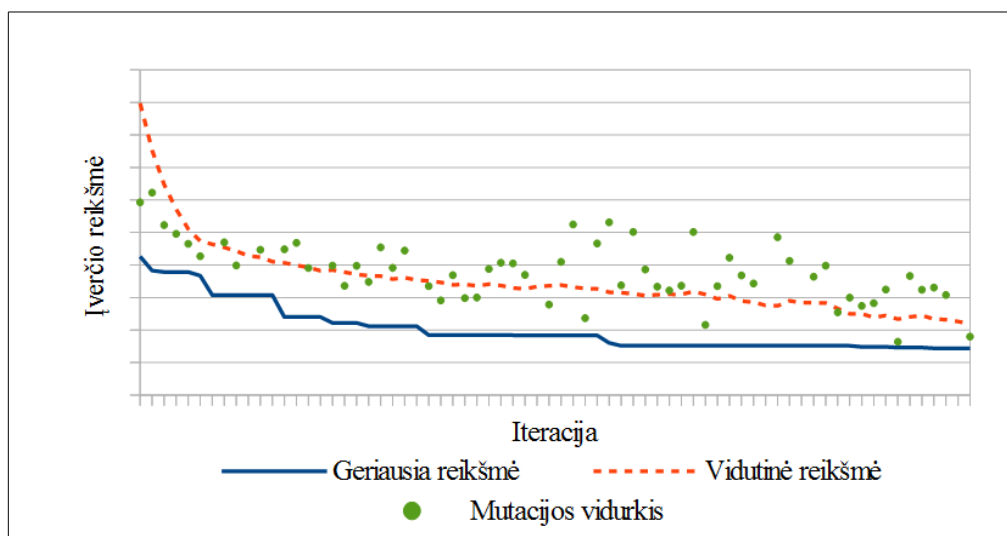
Atskiras maršrutas sprendinyje yra išreikštas kaip grafas, kuriame kiekvienas lankas apibrėžia trumpiausią kelią tarp kelių maršrute aplankomų vietų. Aibė U yra sprendinio x dalis, kur $N_v \cup U = N$ ir $N_v \cap U = \emptyset$. Jeigu aibė U yra tuščia, sprendinys x yra leistinas, priešingu atveju jis yra neleistinas. Disertacijoje yra pateikiama atsitiktinio įterpimo euristika, kuri yra papildyta apribojimų sprendimu. Naujų mazgų įterpimas į maršrutus yra vykdomas įvertinant apribojimus, o apribojimų sprendimui prieš kiekvieną įterpimą yra įvykdomas išankstinis leistinių įterpimų apskaičiavimas, taip yra sumažinamas leistino sprendinio sudarymo sudėtingumas. Vieno sprendinio sudarymo sudėtingumas yra $O(k^2)$, k – vietų, kurias reikia aplankyti uždavinyje, skaičius.

Naujas genetinis algoritmas

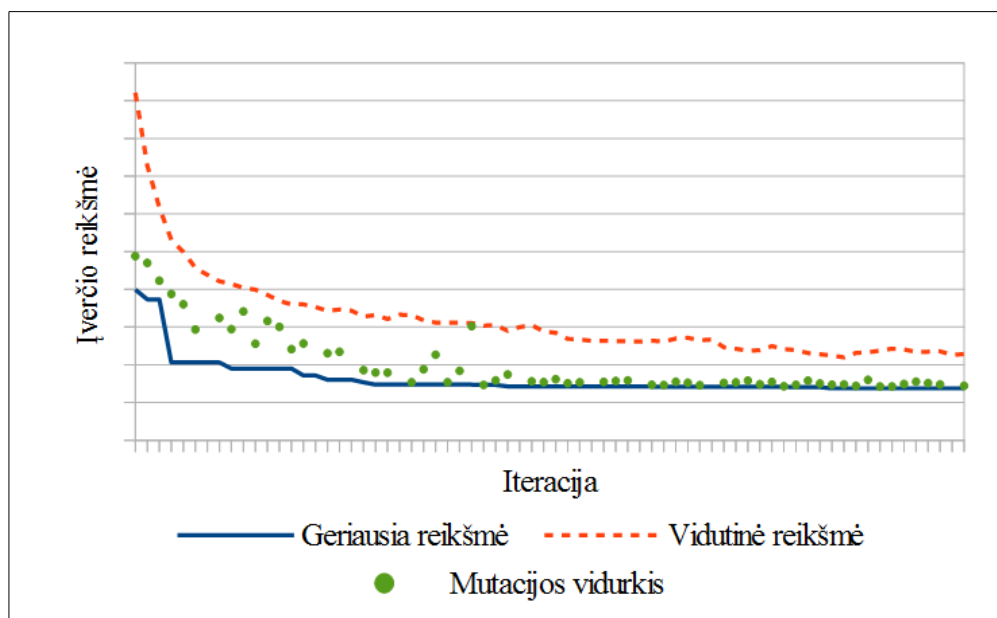
Pasiūlyto genetinio algoritmo kryžminimo bei mutacijos operatoriai remiasi principu – išardyti dalį sprendinio ir rekonstruoti jį pasinaudojant aprašytu atsitiktinio įterpimo metodu su apribojimų įvertinimu. Minėtas įterpimo metodas yra naudojamas ir pradinės populiacijos sudarymui.

Pasiūlytame algoritme mutacijos operatorius yra taikomas su tikimybe $MP = 0,1$, o kryžminimo operatorius yra taikomas visiems sprendiniams, kurie yra parinkti suporavimui. Kryžminimo operacijoje sprendiniai-palikuonys yra sugeneruojami iš dviejų parinktų sprendinių-tėvų, kurie yra parenkami ranguojant sprendinius populiacijoje. Sugeneruoti sprendiniai-palikuonys yra įtraukiami į populiaciją pakeičiant jais blogiausius populiacijos sprendinius ir išlaikant nekintantį populiacijos dydį. Mutacijos operacijoje sprendinių atkūrimui yra naudojama atsitiktinio įterpimo euristika, tad mutacijos operatoriaus sukurti sprendiniai, ypač vėlesnėse iteracijose, turi mažą tikimybę būti parinktais kryžminimui ir gali būti pakeičiami kitais sugeneruotais sprendiniais. Mutacijos operatoriaus sukurtų sprendinių gerumas gali būti įvertintas

lyginant sprendinį su kitais populiacijos sprendiniais. Jeigu mutacijos operacijoje sugeneruoto sprendinio įverčio reikšmė yra artima blogiausių populiacijos sprendinių įverčiams, yra didelė tikimybė, kad toks sprendinys nebus parinktas kryžminimui, o bus pašalintas iš populiacijos vienoje iš vėlesnių iteracijų. Tokių sprendinių generavimas nėra naudingas. Siekiant mutacijos operacijoje generuoti sprendinius, kurie galėtų būti parinkti kryžminimui bei padidinti diversifikavimą populiacijoje, yra sukuriama antra populiacija. Sugeneruotų sprendinių reikšmės lyginant su populiacijos vidutiniu ir geriausiu sprendiniu yra pateiktos 1 paveikslėlyje, kuriame pavaizduota mutacijos operatoriaus elgsena taikant jį tipiniu būdu, bei 2 paveikslėlyje, kuriame parodytas veikimas mutacijos operacijoje naudojant antrą populiaciją.



1 pav. Mucacijos operatoriaus elgsena, kai mutacijos operatorius taikomas tipiniu būdu



2 pav. Mucacijos operatoriaus elgsena, kai antra populiacija yra sukuriama ir skaičiuojama mutacijos operacijoje

Mucacijos operacijoje sukūrus antrą populiaciją bei optimizavus joje sprendinius gauti sprendiniai daugeliu atveju geresni už bendrą populiacijos vidurkį, tad jie yra

konkurencingi pirmoje populiacijoje bei yra didesnė tikimybė, kad tokie sprendiniai bus parinkti kryžminimui.

Pasiūlyto genetinio algoritmo pseudo kodas yra toks:

*Pop*₁ – initial population of size *PS*₁ //create a set of solutions using the feasible insertion method

while number of iterations without improvement < *IL*₁ and time < *TL*₁

sort(*Pop*₁) // sort individuals with a defined comparison function

 remove ($|Pop_1| - PS_1$) worst individuals from *Pop*₁

for *i*=1...*PL*₁

*x*_{p11}, *x*_{p12} – select parent solutions from *Pop*₁ by using ranking method

*x*_{c11} = *crossover*(*x*_{p11}, *x*_{p12}) // generate offspring

*x*_{c12} = *crossover*(*x*_{p12}, *x*_{p11}) // generate offspring

 add *x*_{c11} and *x*_{c12} to *Pop*₁

if *random*(0,1) < *MP* // apply the mutation with probability *MP*

 (*x*'_{m1}, *N*'_{m1}) – create partial solution *x*'_{m1} and node list *N*'_{m1} by

 mutating *x*_{c11}

*Pop*₂ – create population of size *PS*₂ by inserting *N*'_{m1} to *x*'_{m1}

PROCESS *Pop*₂

*x*_{m1} = select the best individual from *Pop*₂

 add *x*_{m1} to *Pop*₁

end if

end for

end while

best solution is *Pop*₁[1] // the best individual in the first population

PROCESS *Pop*₂

while number of iterations without improvement < *IL*₂

sort(*Pop*₂) // sort individuals with a defined comparison function

 remove ($|Pop_2| - PS_2$) worst individuals from *Pop*₂

for *j*=1...*PL*₂

*x*_{p21}, *x*_{p22} – select parent solutions from *Pop*₂

*x*_{c21} = *crossover*(*x*_{p21}, *x*_{p22})

*x*_{c22} = *crossover*(*x*_{p22}, *x*_{p21})

 add *x*_{c21} and *x*_{c22} to *Pop*₂

if *random*(0,1) < *MP*

*x*_{m2} – generate offspring by mutating *x*_{c21}

 add *x*_{m2} to *Pop*₂

end if

end for

end while

Mutacijos operacijoje nauja populiacija yra sukuriama dviem žingsniais. Visų pirma vienas iš pasiūlytų mutacijos operatorių yra taikomas sprendiniui *x*_{p11} tam, kad būtų sukurtas dalinis sprendinys *x*'_{m1} bei laisvų mazgų aibė *N*'_{m1}. Tuomet populiacijos *Pop*₂ sprendiniai yra sukuriami panaudojant maršrutus iš dalinio sprendinio *x*'_{m1} bei kiekvieną kartą įterpiant mazgus *N*'_{m1} pritaikius atsitiktinio įterpimo euristicą.

Skaiciavimai populiacijoje *Pop*₂ yra stabdomi, kai geriausias sprendinys šioje populiacijoje nėra pagerinamas per *IL*₂ iteracijų. Maksimalus skaiciavimo laikas populiacijoje *Pop*₂ nėra taikomas, o reikšmė *IL*₂ yra parenkama tokio dydžio, kad būtų

išvengta perteklinių skaičiavimų populiacijoje Pop_2 . Eksperimentiniuose tyrimuose buvo naudojamos tokios šio algoritmo parametrų reikšmės: $PS_1 = 100$, $PL_1 = 10$, $IL_1 = 50$, $TL_1 = 5\text{min}$, $MP = 0,1$, $PS_2 = 20$, $IL_2 = 5$, $PL_2 = 2$.

Kryžminimo operatoriai

Pasiūlyti kryžminimo operatoriai remiasi gausios aplinkos paieškos (angl., *large neighbourhood search* (LNS)) euristicos suardymo ir atkūrimo principu. LNS euristicos efektyvumas priklauso nuo sprendinio suardymo lygio: jeigu maža sprendinio dalis yra suardoma, LNS gali išanalizuoti tik nedidelę paieškos aibės dalį; jeigu yra suardoma didesnė sprendinio dalis, reikėtų atlikti daugiau operacijų ieškant geresnio sprendinio gautoje aplinkoje.

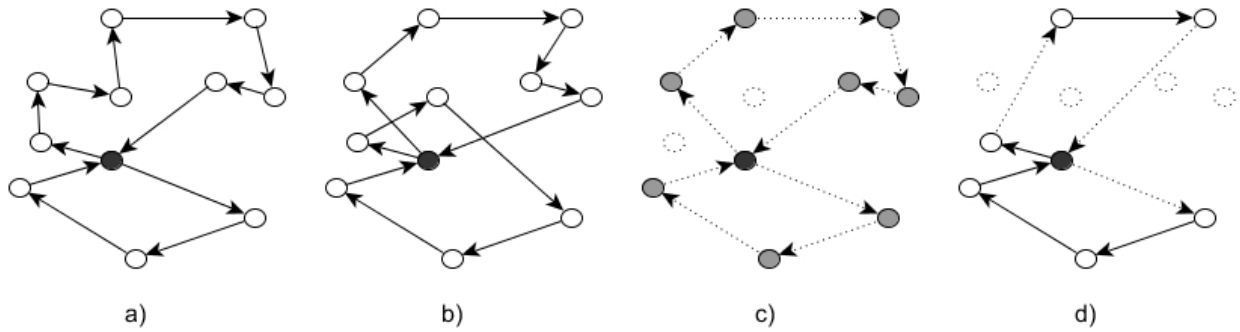
Pasiūlytų kryžminimo operatorių tikslas yra identifikuoti bendras sprendinių-tėvų dalis, išsaugoti jas tarpiniuose sprendiniuose bei taikant rekonstravimą sukurti sprendinius-palikuonis. Išardant tas sprendinių dalis, kurios nėra bendros, gaunama abiejų sprendinių kaimyninė aplinka. Kaimynystės dydis yra atvirkščiai proporcingas bendrų sprendinių dalių dydžiui. Kadangi pradinė populiacija yra sukuriamą atsitiktine tvarka, išsaugant tas sprendinių dalis, kurios yra bendros abiems sprendiniams-tėvams, yra išsaugomos tos dalys, kurios turi didesnę tikimybę būti tinkamai sukonstruotos. Yra didelė tikimybė, kad bus išardytos tos maršrutų dalys, kurios prailgina maršrutus, o esant ilgesniems maršrutams, sprendinyje bus didesnis maršrutų skaičius. Disertacijoje yra pasiūlyti trys kryžminimo operatoriai, kurie remiasi bendrumų sprendiniuose-tėvuose paieška. Kiekvienas kryžminimo operatorius iš dviejų sprendinių-tėvų x_i ir x_j sukuria vieną dalinį sprendinį x'_o bei aibę į sprendinį neįtrauktų mazgų N_{temp} . Sprendinys-palikuonis x_o yra sukuriamas į dalinį sprendinį x'_o apibrėžtu atsitiktinio įterpimo metodu įterpiančiais laisvus mazgus N_{temp} .

Bendrų mazgų kryžminimo operatorius (angl., *common nodes crossover* (CNX)) atlieka dviejų sprendinių x_i ir x_j sankirtą atsižvelgiant į maršrutuose esančių mazgų aibes N_i ir N_j ; CNX operatoriaus sudėtingumas yra $O(k^2)$.

Bendrų lankų kryžminimo operatorius (angl., *common arcs crossover* (CAX)) sprendinyje-palikuonyje išsaugo maršrutų lankus iš pirmo sprendinio, jeigu lankai su tokiomis pačiomis pradžiomis ir pabaigomis egzistuoja ir antrame sprendinyje. CAX operatorius išsaugo aibių $A_i \in x_i$ ir $A_j \in x_j$ sankirtą. Šio kryžminimo operatoriaus sudėtingumas yra $O(k)$, kur k uždavinyje esančių mazgų skaičius.

CNX bei CAX operatorių elgsena yra pateikta 3 paveikslėlyje, kuriame a) ir b) yra sprendiniai-tėvai, c) yra pateiktas tarpinis dalinis sprendinys, gautas su CNX, o pilka spalva pažymėti apskritimai vaizduoja mazgus, kurie yra identifikuoti ieškant didžiausio bendrų mazgų kiekio tarp maršrutų, d) yra tarpinis dalinis sprendinys, gautas su CAX, o ryškios linijomis yra pavaizduoti bendri lankai. Taškuoti apskritimai vaizduoja į sprendinį neįtrauktų mazgų aibę N_{temp} .

Trečias disertacijoje pasiūlytas kryžminimo operatorius yra ilgiausios bendros sekos kryžminimo operatorius (angl., *longest common sequence crossover* (LCSX)). LCSX visuose sprendinių-tėvų maršrutuose ieško ilgiausių bendrų sekų. Visų pirma yra randami priešingų sprendinių maršrutai, kurie turi didžiausią skaičių bendrų mazgų. Tuomet identifikuotuose maršrutuose yra ieškoma ilgiausios bendros didėjančios sekos (angl., *longest common increasing subsequence* (LCIS)). Kadangi tie patys maršrutai gali turėti kelias ilgiausias bendras didėjančias sekas, viena tokia seka yra parenkama atsitiktinai ir išsaugoma tarpiniame sprendinyje-palikuonyje.



3 pav. Kryžminimo operatorių elgsena: a) b) sprendiniai-tėvai; c) dalinis sprendinys, gautas su CNX; d) dalinis sprendinys, gautas su CAX

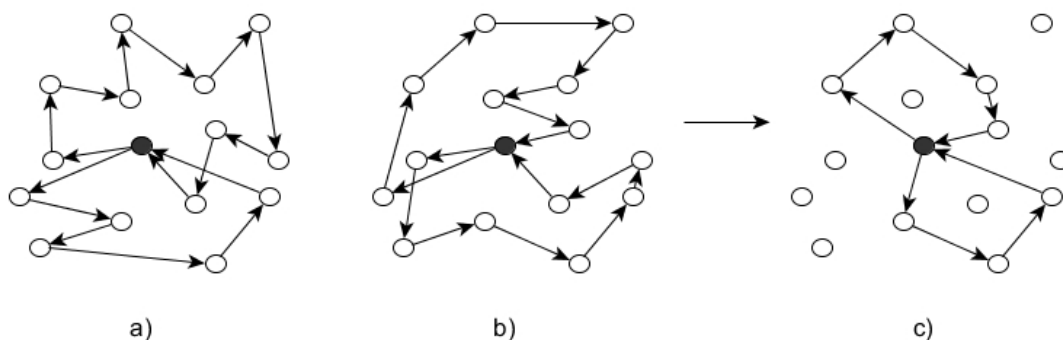
Atliekant ilgiausių didėjančių sekų paiešką tarp visų sprendinių-tėvų maršrutų yra gaunamas tarpinis dalinis sprendinys x'_o (4 pav.):

```

 $N_{temp} = U_i$ 
 $x'_o$  – offspring solution
for each route  $r_i \in R_i$ 
     $r_{int} = \text{find the intersecting route } r_{int} \in x_j \text{ that has the largest number of common nodes}$ 
     $SEQ = \text{find the longest common sequence between the routes } r_i \text{ and } r_{int}$ 
     $r_{new} = \emptyset$ 
    for each node  $n \in r_i$ 
        if  $n$  exist in the sequence  $SEQ$ 
            add the node  $n$  into  $r_{new}$ 
        else
            add  $n$  to  $N_{temp}$ 
        end if
    end for
    if  $|r_{new}| > 1$ 
        add  $r_{new}$  to  $x'_o$ 
    else
         $\forall n \in r_{new}$  add to  $N_{temp}$ 
    end if
end for
create  $x_o$  by inserting  $\forall n \in N_{temp}$  into  $x'_o$  using the defined random insertion

```

LCSX operatoriaus sudėtingumas yra $O(k^2)$. CNX operatoriaus bendrumų paieška nėra susijusi su sekomis maršrutuose, kai tuo tarpu CAX operatorius tarpiniame sprendinyje išsaugo bendrą dviejų iš eilės einančių mazgų seką, o LCSX operatorius ieško ilgiausių bendrų sekų tarp skirtingų sprendinių maršrutų. CAX ir LCSX operatoriai gali būti taikomi ir tuo atveju, kai sprendiniai-tėvai turi tik po vieną maršrutą, CNX operatoriaus gautas sprendinys-palikuonis šiuo atveju bus lygus vienam iš sprendinių-tėvų.



4 pav. Ilgiausios bendros sekos kryžminimo operatoriaus elgsena: a) ir b) du sprendiniai-tėvai; c) tarpinis dalinis sprendinys

Pasiūlytais kryžminimo operatoriais gauti tarpiniai sprendiniai gali turėti tą patį maršrutų skaičių kaip ir sprendiniai-tėvai. Siekiant minimizuoti maršrutų skaičių iš tarpinio sprendinio yra pašalinami maršrutai, kurie turi tik po vieną mazgą. Pašalinti mazgai yra perkelti į aibę mazgų, kurie yra iš naujo įterpiami į maršrutus rekonstruojant sprendinį-palikuonį. Visgi gali būti situacijų, kai tarpinis sprendinys turi tą patį skaičių maršrutų kaip ir vienas iš sprendinių-tėvų, tad visi pasiūlyti kryžminimo operatoriai yra papildyti tokiais veiksmiais:

1. kol $f_v(x'_o) > f_v(x_i) - \delta$, pašalinamas atsitiktinis maršrutas r_r iš x'_o įkeliant $\forall n \in r_r$ į N_{temp} ;
2. sprendinys-palikuonis x_o yra sukuriamas įterpiant $\forall n \in N_{temp}$ į x'_o panaudojus atsitiktinio įterpimo metodą.

Pirmame žingsnyje δ reikšmė gali būti nuo 0 iki $f_v(x_i) - 1$, kur x_i yra vienas iš sprendinių-tėvų, dalyvaujantis kryžminimo operacijoje. Jeigu $\delta = 0$, maršrutų skaičius sprendinyje nebus mažinamas. Pasiūlytuose kryžminimo operatoriuose δ yra atsitiktinė reikšmė iš aibės $\{0, 1\}$, jeigu $f_v(x_i) > 1$, ir $\delta = 0$, priešingu atveju.

Mutacijos operatoriai

Pasiūlyti mutacijos operatoriai remiasi jau aprašytu LNS suardymo bei atkūrimo principu, tačiau mutacijos operatoriai yra taikomi tik vienam sprendiniui x_i . Pasiūlyti mutacijos operatoriai išardo sprendinio dalis apibrėžtais būdais ir suformuoja naują sprendinį taikydami atsitiktinio įterpimo euristicą. Disertacijoje yra pasiūlyti penki mutacijos operatoriai. Kiekvienas iš jų išardo sprendinį atsižvelgdamas į skirtingas sprendinio savybes. Mutacijos operatoriai yra taikomi pasirenkant vieną iš jų atsitiktine tvarka vienos mutacijos operacijos metu.

Pirmasis mutacijos operatorius atsitiktine tvarka parenka ir išima iš sprendinio $0,5z|N|$ mazgų, kur z yra atsitiktinis skaičius intervale $(0, 1)$. Šio mutacijos operatoriaus sudėtingumas yra $O(k)$.

Antrasis mutacijos operatorius atsitiktinai parenką vieną mazgą n_r iš visų sprendinio x_i maršrutų. Tuomet iš sprendinio maršrutų yra išimami kiti mazgai minimizuojant atstumo funkciją $l(n_r, n_i)$, kur $\forall n_i \in R_i$. Tokiu būdu iš sprendinio yra išimama $0,5z|N|$ mazgų, kur z yra atsitiktinis skaičius intervale $(0, 1)$. Šio mutacijos operatoriaus sudėtingumas yra $O(k^2)$.

Trečiasis mutacijos operatorius atsitiktine tvarka iš sprendinio išima $0,5z|R_i|$ maršrutų, kur $R_i \in x_i$. Šio mutacijos operatoriaus sudėtingumas yra $O(k)$.

Ketvirtasis mutacijos operatorius iš sprendinio išima tuos mazgus, kuriems apvažiuoti maršrutuose yra daroma didžiausia apylanka. Yra ieškoma mazgų maksimizuojant funkciją $l_r(n_r) = l(n_{r-1}, n_r) + l(n_r, n_{r+1}) - l(n_{r-1}, n_{r+1})$. Iš sprendinio maršrutų pašalinamų mazgų skaičius yra ribojamas iki $0,5z|N|$, kur z yra atsitiktinis skaičius intervale $(0, 1)$. Šio mutacijos operatoriaus sudėtingumas yra $O(k^2)$. Genetiniame algoritme ketvirtasis mutacijos operatorius yra taikomas kartu su kitais mutacijos operatoriais.

Penktasis mutacijos operatorius ieško mazgų, kurie sprendinio maršrutuose yra aplankomi panašiu laiko momentu. Šis mutacijos operatorius yra panašus į antrą mutacijos operatorių. Iš pradžių iš sprendinio maršrutų yra parenkamas atsitiktinis mazgas n_r . Tuomet kiti mazgai yra išimami iš sprendinio minimizuojant funkciją $t_r(n_i) = |t_a(n_r) - t_a(n_i)|$, kur $\forall n_i \in R_i, R_i \in x_i$, o funkcija $t_a(n_i)$ įvertina atvykimo į n_i laiką sprendinio maršrute. Penktasis mutacijos operatorius yra taikomas, kai uždavinyje yra apibrėžti laiko apribojimai. Šio mutacijos operatoriaus sudėtingumas yra $O(k^2)$.

Genetiniai operatoriai, skirti bendram transporto maršrutų sudarymo uždaviniui spręsti

Nors mokslinėje literatūroje dažniausiai yra nagrinėjami specifiniai VRP uždaviniai, egzistuoja ir bandymų apibendrinti šiuos uždavinius. Šio tyrimo tikslas yra sukurti algoritmą, kuris galėtų būti taikomas bendram transporto maršrutų sudarymo uždaviniui spręsti. Literatūroje galima rasti tyrimų, kuriuose bendras transporto maršrutų sudarymo uždavinio apibrėžimas apima įvairias situacijas bei apribojimus, kylančius praktinėse situacijose planuojant transporto maršrutus.

Apibendrinant transporto maršrutų sudarymo uždavinius apibrėžiame keletą juos sudarančių komponentų:

- *duomenys*, kurie sudaro uždavinį;
- *užduotys*, kurios turi būti įgyvendintos sprendinyje;
- *apribojimai*, kurių reikia paisyti sprendinyje;
- *tikslas*, kurį reikia optimizuoti.

Uždavinio *duomenis* sudaro grafas $G = (N, E)$, kuris susideda iš mazgų N bei lankų aibės E . Uždavinio duomenims taip pat priskiriama transporto priemonių aibė $V = \{v_1, \dots, v_t\}$. Mazgų aibė N gali būti padalinta į poaibius: a) N_d – paskirstymo punktų, b) N_c – klientų, c) N_o – kitų mazgų, kurie gali būti padalinti į poilsio vietų, degalinių bei kitas vietas. Duomenys taip pat apibrėžia pradinę kiekvienos transporto priemonės v_i vietą n_i^{init} . Papildomai uždavinio duomenys gali apibrėžti vairuotojus ir jų parametrus, transporto priemonių parametrus, krovinių tipus ir t. t.

Užduotys apibrėžia pavestus transportavimo darbus, kurie turi būti atlikti sprendinyje. Sakykime, kad transportavimo užsakymų aibė yra $M = \{m_1, \dots, m_q\}$, sudaryta iš q užsakymų, ir aibė $T = \{t_1, \dots, t_k\}$, apibrėžianti k užduotis, kurias reikia atlikti, kad įgyvendinti užsakymus. Kiekvienas transportavimo užsakymas m_i gali būti išreikštas užduočių, skirtų atlikti užsakymui, aibe $m_i = \{t_{i1}, t_{i2}, \dots\}$, kur $\forall t_{ij} \in T, |m_i| > 0, \forall m_i \in M, m_1 \cap \dots \cap m_q = \emptyset$ ir $m_1 \cup \dots \cup m_q = T$. Pagrindinis skirtumas tarp užsakymo ir užduoties yra toks, kad viena transporto priemonė vienu metu gali atlikti tik vieną užduotį, o užsakymai gali būti vykdomi lygiagrečiai. Kiekvieną užduotį toliau galima būtų skirstyti į smulkesnes užduotis, taip didinant granuliškumą, tačiau VRP uždaviniams užduotys dalomos iki tokio lygio, kol atitinka savybę „vienu metu vykdoma viena užduotis“. VRP uždaviniuose kiekviena užduotis t_i apibrėžiama kaip

$t_i = (n_i^{start}, n_i^{end})$, kur mazgas $n_i^{start} \in N$ apibrēžia užduoties t_i vykdymo pradžios vietą, o mazgas $n_i^{end} \in N$ apibrēžia užduoties pabaigos vietą. Kad įvykdytų užduotį t_i , transporto priemonė turi atvykti į pradžios vietą n_i^{start} , kur paslaugos vykdymas yra pradedamas, ir užbaigti paslaugos vykdymą vietoje n_i^{end} .

VRP uždaviniuose, kurie susiję su krovinų transportavimu, krovinio užsakymas gali būti apibrėžiamas kaip $m_i = \{t_i^+, t_i^-\}$, kur m_i apibrėžia krovinio pristatymą iš vienos vietos į kitą, o užsakymo įgyvendinimui reikia įvykdyti užduotis t_i^+ (tam tikroje vietoje pakrauti krovinį) ir t_i^- (tam tikroje vietoje krovinį iškrauti). Krovinio parametrai yra apibrėžiami kiekvienai užduočiai. Sakykime, kad funkcija $w(m_i)$ įvertina krovinio dydžio vertę $w_i = w(m_i)$. Krovinų gabenimo užduotis galima apibrėžti kaip $t_i = (n_i^{start}, n_i^{end}, w_i)$, kur krovinys, kurio dydis w_i , yra pakraunamas/iškraunamas tam tikroje vietoje $n_i^{start} = n_i^{end}$. Paprastai VRP uždaviniuose yra numatytos grįžimo į paskirstymo punktus užduotys $T^{end} = \{t_1^{end}, \dots, t_t^{end}\} \subseteq M$, $\forall t_{vi} \in T$. Užduoties, kuri prasidėtų ir baigtųsi skirtingose vietoje, pavyzdys galėtų būti taksi paslauga, kur kiekvienas klientas yra paimamas iš vienos vietos ir pristatomas į kitą bei keli skirtingi klientai nėra vežami kartu.

VRP uždavinio sprendinyje užduotys yra vykdomos transporto priemonėmis. Apibrėžkime atskirą VRP uždavinio sprendinį kaip $x = \{s_1, \dots, s_t\}$, kur $\forall s_j = (t_{j1}, t_{j2}, \dots)$, $\forall t_{ji} \in T$, $s_1 \cap \dots \cap s_t = \emptyset$ ir $s_1 \cup \dots \cup s_t = T$. $\forall s_j$ apibrėžia seką užduočių, priskirtų transporto priemonei $v_j \in V$, o $|x| \leq |V|$. Sakykime, kad yra tokia funkcija $F_m(x)$, kuri įvertina įvykdytus užsakymus sprendinyje x , bei funkcija $f_m(x)$, kuri įvertina baigtų užduočių, susijusių su vienu užsakymu, įvykdymą.

$$F_m(x) = \sum_{m \in M} f_m(m, x)$$

$$f_m(m, x) = \sum_{t \in m} f_t(x, t)$$

$$f_t(x, t) = \begin{cases} 0 & \text{jeigu užduotis } t \text{ yra įvykdyta sprendinyje } x \\ 1 & \text{priešingu atveju} \end{cases}$$

Visi užsakymai sprendinyje x yra įvykdyti, jeigu $F_m(x) = 0$.

Apribojimai apibrėžia sąlygas, kurios atspindi tam tikras realias situacijas bei kurių negalima pažeisti sprendinyje x . Sakykime, kad visų apribojimų aibė yra C , o $c \in C$ apibrėžia tam tikrą vieną apribojimą. Apribojimai gali būti apibrėžti kiekvienai užduočiai (pvz., laiko momentu, kada užduotį galima vykdyti), transporto priemonei (pvz., maksimaliu leistinu krovinio svoriu/tūriu), transporto kroviniumi ir t. t. Vienas iš apribojimų VRPPD uždaviniuose apibrėžia tai, kad užduotis t_i^+ turi būti įvykdyta prieš užduotį t_i^- arba kad grįžimo į paskirstymo punktą užduotis turi būti atlikta įvykdžius visas kitas transporto priemonei priskirtas užduotis. Taigi, apribojimai gali apibrėžti tam tikrų užduočių vykdymo tvarką. Apribojimai taip pat gali būti taikomi ir vairuotojui, pavyzdžiui, poilsio laikas reikalingas kiekvienam vairuotojui, arba apribojimai gali apibūdinti transporto priemonių kuro bakų dydį. Sakykime, kad yra tokia funkcija $F_c(x)$, kuri įvertina apribojimų pažeidimus sprendinyje x , o funkcija $f_c(x)$ įvertina vieno apribojimo pažeidimą $c \in C$ sprendinyje x :

$$F_c(x) = \sum_{c \in C} f_c(x)$$

$$f_c(x) = \begin{cases} 0 & \text{jeigu apribojimas } c \text{ nepažeistas} \\ z, \text{ kur } z \in \mathbb{R}, z > 0 & \text{priešingu atveju} \end{cases}$$

Sprendinys x nepažeidžia uždavinijoje nustatytų apribojimų, jeigu $F_c(x) = 0$.

Tikslas tipiniame VRP uždavinijoje yra apibrėžiamas kaip visų maršrutų bei bendro kelio ilgio minimizavimas. Taigi, tikslas visų pirma yra maršrutų kiekio įvertinimo funkcijos $f_v(x)$ minimizavimas, o po to – kelio ilgio įvertinimo funkcijos $f_d(x)$ minimizavimas, kur sprendinys taip pat turi atitikti lygybes $F_c(x) = 0$ ir $F_m(x) = 0$:

$$f_v(x) = \sum_{j=1}^{|x|} f_a(s_j), \forall s_j \in x$$

$$f_d(x) = \sum_{j=1}^{|x|} d_l(s_j), \forall s_j \in x$$

$$f_a(s_j) = \begin{cases} 1 & \text{jei } d_l(s_j) > 0 \\ 0 & \text{priešingu atveju} \end{cases}$$

$$d_l(s_j) = \sum_{i=1}^{|s_j|} l(n_{j_i}^0, t_{j_i}), \text{ kur } \forall t_{j_i} \in s_j, n_{j_i}^0 = n_{j_{i-1}}^{end}, n_{j_0}^{end} = n_j^{init}$$

$$l(n_i, t_j) = l(n_i, n_j^{start}) + l(n_j^{start}, n_j^{end})$$

$$l(n_i, n_j) - \text{atstumas tarp mazgų } n_i \text{ ir } n_j$$

Skirtingos tikslo funkcijos, kurios minimizuotų/maksimizuotų tam tikrą reikšmę, galėtų būti taikomos transporto maršrutų sudarymo uždaviniuose.

Pasiūlyto genetinio algoritmo sprendinys yra leistinas, jeigu jis atitinka visus uždavinio apribojimus, tačiau toks sprendinys gali būti ne pilnas, t. y. gautame sprendinyje įvykdytos ne visos užduotys. Tokiu atveju bendro transporto maršrutų sudarymo uždavinio tikslas turi būti išplėstas: tikslas yra rasti sprendinį, kuriame įvykdoma daugiausiai užduočių. RVRP uždavinio sprendinys x yra apibrėžiamas taip:

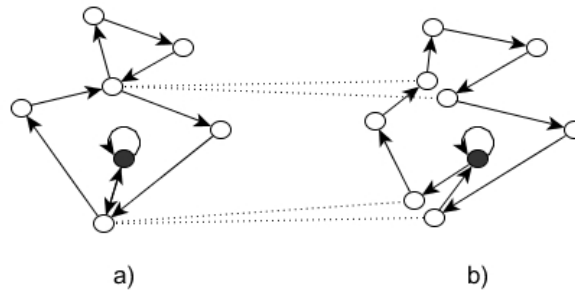
$$x = (R = \{r_1, \dots, r_t\}, T_u = \{t_1, \dots, t_u\})$$

Sakykime, kad maršrutas $r_r \in x$ sprendinyje yra išreikštas kaip grafas $G_r^T = (T_r, A_r)$, kur aibė $T_r = \{t_r^{init}, t_{r1}, \dots, t_r^{end}\}$ apibrėžia aibę užduočių, priskirtų maršrutui, o t_r^{init} reiškia maršruto pradžią vietoje n_r^{init} . Aibė T_u apibrėžia sprendinyje neįgyvendintų užduočių aibę, kur $\{T_1, \dots, T_t\} \cup T_u = T$ ir $\{T_1, \dots, T_t\} \cap T_u = \emptyset$. Kai maršrutas sudarytas iš užduočių, tai panaikina galimybę, kad sprendinyje bus pasikartojantys elementai, kai tuo tarpu sprendiniuose, sudarytuose iš aplankomų vietų, ta pati vieta gali būti aplankyta kelis kartus, jei ji numatyta keliose užduotyse. Sprendinio, sudaryto iš aplankomų vietų bei iš vykdomų užduočių, pavyzdys yra pateiktas 5 paveikslėlyje. Lankai iš aibės A_r jungia užduotis: $\forall a_{ri} \in A_r, a_{ri} = (t_{ri}, t_{r(i+1)})$, kur t_{ri} yra lanko pradžia, o $t_{r(i+1)}$ yra lanko pabaiga. Naujos užduoties įterpimas į maršruto lanką a_{ri} reiškia:

1. pašalinti lanką a_{ri} iš aibės A_r ;
2. įtraukti t_m į T_r ;
3. įtraukti du naujus lankus (t_{ri}, t_m) ir $(t_m, t_{r(i+1)})$ į aibę A_r .

Tuščias maršrutas (transporto priemonei nėra priskirta vykdomų užduočių) yra grafas, susidedantis iš $T_r = \{t_r^{init}, t_r^{end}\}$ ir $A_r = \{(t_r^{init}, t_r^{end})\}$. 5 paveikslėlyje toks grafas yra pavaizduotas lanku, prasidedančiu ir pasibaigiančiu pradiniam taške. Naujos užduoties įterpimas į maršrutą gali būti suskirstytas į šiuos žingsnius:

1. parinkti užduotį įterpimui į maršrutą;
2. parinktai užduočiai parinkti poziciją maršrute.



5 pav. Sprendinys sudarytas iš: a) aplankomų vietų; b) vykdomų užduočių

Parinkti poziciją maršrute užduoties įterpimui reiškia parinkti lanką įterpimui iš aibės A_{all} (aibė apibrėžia lankus visuose maršrutuose daliniame sprendinyje x'). Kad įterpus naują užduotį į maršrutą dalinis sprendinys būtų leistinas, reikia įvertinti gauto dalinio sprendinio atitikimą apribojimams. Sakykime, kad funkcija $F_c(a_i, t_m)$ įvertina sprendinio, gauto įterpus užduotį t_m į lanką $a_i \in x'$, atitikimą apribojimams, o $x'^{i,m}$ yra dalinis sprendinys x' , kuriame užduotis t_m yra įterpta į lanką a_i :

$$F_c(a_i, t_m) = \Delta F_c^{i,m}(x') = F_c(x'^{i,m}) - F_c(x')$$

$$F_c(a_i, t_m) = \sum_{c \in C} f_c(a_i, t_m)$$

Funkcija $f_c(a_i, t_m)$ įvertina dalinio sprendinio atitikimą vienam apribojimui įterpiant užduotį t_m į lanką a_i . Naujos užduoties įterpimas nepažeidžia apribojimų, jei $F_c(a_i, t_m) = 0$. Lankai įterpimui turi būti parinkti atsižvelgiant į nustatytą tikslo funkciją. Taigi lankas užduoties įterpimui yra parenkamas minimizuojant funkcijas $f_v(a_i, t_m)$ ir $f_d(a_i, t_m)$, kurios įvertina tikslo funkcijų gautos reikšmės skirtumą įterpiant užduotį t_m į lanką a_i . Visų pirma yra įvertinamas maršrutų skaičiaus pokytis, po to yra įvertinamas viso maršrutų kelio ilgio pokytis:

$$f_v(a_i, t_m) = \Delta f_v^{i,m}(x') = f_v(x'^{i,m}) - f_v(x')$$

$$f_d(a_i, t_m) = \Delta f_d^{i,m}(x') = f_d(x'^{i,m}) - f_d(x')$$

Genetiniai operatoriai, kurie buvo pasiūlyti šioje disertacijoje VRP uždaviniui, sudarytam iš aplankomų vietų, spręsti taip pat gali būti taikomi RVRP uždavinio, sudaryto iš vykdomų užduočių, sprendimui. Bendras pasiūlytų genetinių operatorių taikymo principas yra toks pat. Pasiūlyti kryžminimo operatoriai nustato sprendinių suardymo lygį išsaugant tas sprendinių-tėvų dalis, kurios yra bendros, taip išsaugant sprendinių dalis, kurios turi didesnę tikimybę būti tinkamai sukonstruotos. RVRP

uždaviniui CNX operatoriumi ieškoma bendrų užduočių aibių maršrutuose, CAX operatoriumi ieškoma bendrų lankų, kurie jungtų tas pačias užduotis, o LCSX operatoriumi ieškoma ilgiausių bendrų sekų, sudarytų iš maršrutuose vykdomų užduočių. Identifikuotos bendros dalys yra išsaugomos sprendiniuose-palikuonyse, o likusios, maršrutams nepriskirtos užduotys atkūrimo fazėje yra įterpiamos į dalinį sprendinį naudojant konstravimo euristiką. Siekiant išlaikyti stochastines genetinio algoritmo savybes užduočių įterpimui į sprendinį yra naudojamas atsitiktinio įterpimo euristikos metodas, kuris parenką sprendinį atsitiktine tvarka iš maršrutams nepriskirtų užduočių aibės ir įterpia į maršrutą įvertindamas visus apribojimus bei minimizuodamas įterpimo įvertinimo funkcijas $f_v(a_i, t_m)$ ir $f_d(a_i, t_m)$. Paieška genetiniame algoritme yra intensyvinama išsaugant bendras sprendinių-tėvų dalis, o naudojant atsitiktinio įterpimo euristiką sprendiniai populiacijoje yra diversifikuojami. Taikant atskirus pasiūlytus mutacijos operatorius RVRP uždavinio sprendimui dalinis sprendinys-palikuonis yra gaunamas skirtingais būdais pašalinant užduotis iš sprendinio. Sprendinys-palikuonis gaunamas minėtu atsitiktinio įterpimo metodu įterpiant atgal sprendinio dalis, kurios buvo pašalintos. Antros populiacijos naudojimas mutacijos operacijoje didina tikimybę, kad mutacijos operacijoje gautas sprendinys bus konkurencingas pirmoje populiacijoje.

Genetinis algoritmas su pasiūlytais kryžminimo bei mutacijos operatoriais gali būti taikomas bet kokiam uždaviniui, kurį galima išreikšti per pateiktą RVRP uždavinio formulotę ir kurio sprendinys priklauso nuo sprendinio dalių sekos. Atsitiktinio įterpimo metodas gali būti papildytas ir kitų uždavinyje apibrėžtų apribojimų įvertinimu, pavyzdžiui, sprendžiant periodinių transporto maršrutų sudarymo uždavinius nauja užduotis bus įterpiama į maršrutą, jeigu atitiks apribojimą, kad atitinkamos užsakymo užduotys bus vykdomos per nustatytus laiko periodus. Uždaviniai, kurie apima maršrutų sudarymą apskaitant tam tikrus grafo lankus, pvz., uždaviniai susiję su gatvių valymo planavimu, taip pat gali būti išreikšti nauja pasiūlyta forma bei sprendžiami pasiūlytu genetiniu algoritmu. Tokiuose uždaviniuose kiekvienos užduoties pradžia bei pabaiga sutaps su lanko pradžia bei pabaiga.

Lygiagretus dvikryptis trumpiausio kelio paieškos algoritmas

Disertacijoje taip pat yra pasiūlytas lygiagretus Dijkstra algoritmas trumpiausių kelių paieškai kelių grafe. Lygiagretus Dijkstra algoritmas skirtas paspartinti trumpiausio kelio paiešką tarp dviejų vietų kelių grafe. Šis algoritmas gali būti panaudotas perskaičiuojant trumpiausius kelius genetinio algoritmo vykdymo metu, jeigu dėl atsiradusių papildomų duomenų trumpiausio kelio reikšmė pasikeičia, pvz., skirtingu metu važiuojant tuo pačiu keliu, važiavimo laikas gali skirtis dėl transporto spūsčių. Dvikrypčiam algoritmui yra apibrėžtas sustojimo kriterijus, kurį naudojant yra išvengiama netikslumų trumpiausio kelio paieškoje. Disertacijoje yra pasiūlytas lygiagretus dvikryptis Dijkstra algoritmas siekiant paspartinti trumpiausių kelių paiešką. Pasiūlytas algoritmas remiasi transakcinės atminties (angl., *transactional memory*) naudojimu tam, kad būtų išvengta didelės apimties duomenų sinchronizacijos.

3. Eksperimentiniai tyrimai

Šioje disertacijos dalyje yra pateikiami eksperimentiniai pasiūlytų algoritmų tyrimai. Disertacijos antroje dalyje pasiūlyti kryžminimo operatoriai (CNX, CAX, LCSX) bei analitinėje apžvalgoje analizuoti kryžminimo operatoriai (BCRC, RBX, LRX), kurie remiasi įterpimo euristika, yra suprogramuoti Java programavimo kalba

siekiant juos palyginti tarpusavyje. Kiekvieno kryžminimo operatoriaus atkūrimo metodas yra įgyvendintas toks, kaip yra išnagrinėta apžvalginėje dalyje. Kitos genetinio algoritmo dalys yra bendros visuose eksperimentuose:

- Pradinė populiacija yra sukurta atsitiktinai parenkant uždavinio aibės mazgus ir įterpiančios juos į sprendinį. Įterpimas yra vykdomas įvertinant apribojimus ir minimizuojant tikslo funkciją. Ta pati populiacija yra naudojama atliekant eksperimentą su visais kryžminimo operatoriais. Populiacijos dydis yra 100.
- Atrankai yra naudojamas *k-tournament selection* metodas, kur $k = 2$. Kiekvienos iteracijos metu yra sukuriama 10 naujų sprendinių palikuonių.
- Tam, kad geriau būtų įvertintos kryžminimo operatorių savybės, buvo atliktos dvi skirtingų eksperimentų grupės, kurių viena apima eksperimentus, į kuriuos nebuvo įtraukta mutacija, o antra apima eksperimentus, kuriuose buvo naudojama mutacija. Naudojama mutacijos operacija atsitiktine tvarka išimdavo 0,5z maršrutų mazgų ir juos įterpdavo atgal naudodama tokį atstatymo metodą, kokį naudoja kryžminimo operatorius, čia z yra atsitiktinė reikšmė intervale (0,1). Mutacija eksperimentuose taikyta su tikimybe 0,15.
- Skaičiavimai eksperimentuose atliekami tol, kol geriausias populiacijos sprendinys nepagerinamas per 300 iteracijų arba skaičiavimai pasiekia maksimalų skaičiavimo laiką (5 minutes).

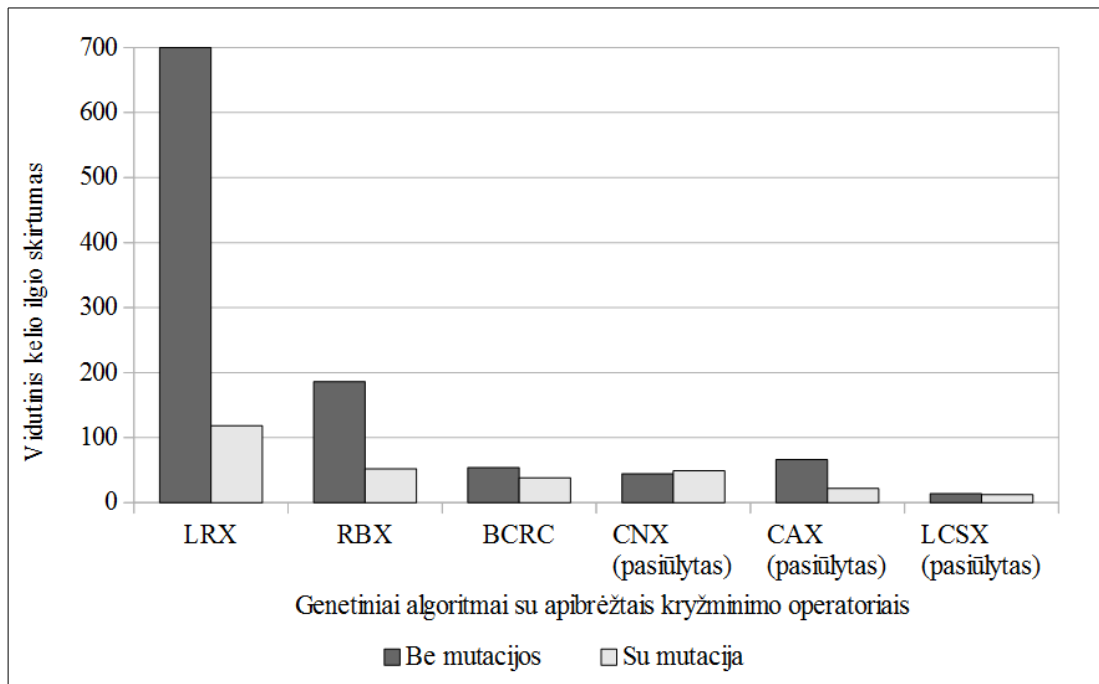
Eksperimentai buvo atlikti su Solomon VRPTW uždavinių pavyzdžiais [7]. Genetinis algoritmas su skirtingais kryžminimo operatoriais kiekvienam uždavinio pavyzdžiui buvo vykdomas 10 kartų. Kiekvieną kartą buvo sukuriama nauja pradinė populiacija.

Gauti rezultatai parodė, kad sprendiniai, rasti LCSX operatoriumi, buvo geresni už kitų kryžminimo operatorių rastus sprendinius ~92 % atvejų. Kryžminimo operatoriais rastų sprendinių vidutiniai maršrutų kelio ilgio skirtumai nuo geriausių žinomų sprendinių pateikti 6 paveikslėlyje, o vidutiniai rastų maršrutų kiekio skirtumai nuo geriausių žinomų sprendinių pateikti 7 paveikslėlyje. Iš paveikslėlių matyti, kad LCSX operatoriumi gauti rezultatai buvo tiksliausi lyginant su kitais. Taip pat verta paminėti, kad daugeliu atveju LCSX operatoriumi gauti rezultatai, gauti netaikant mutacijos operatoriaus, yra geresni net ir už rezultatus, gautus kitais kryžminimo operatoriais taikant mutaciją. Visais pasiūlytais kryžminimo operatoriais rastas sprendinių vidutinis maršrutų kiekis yra geresnis už kitų kryžminimo operatorių sprendinius.

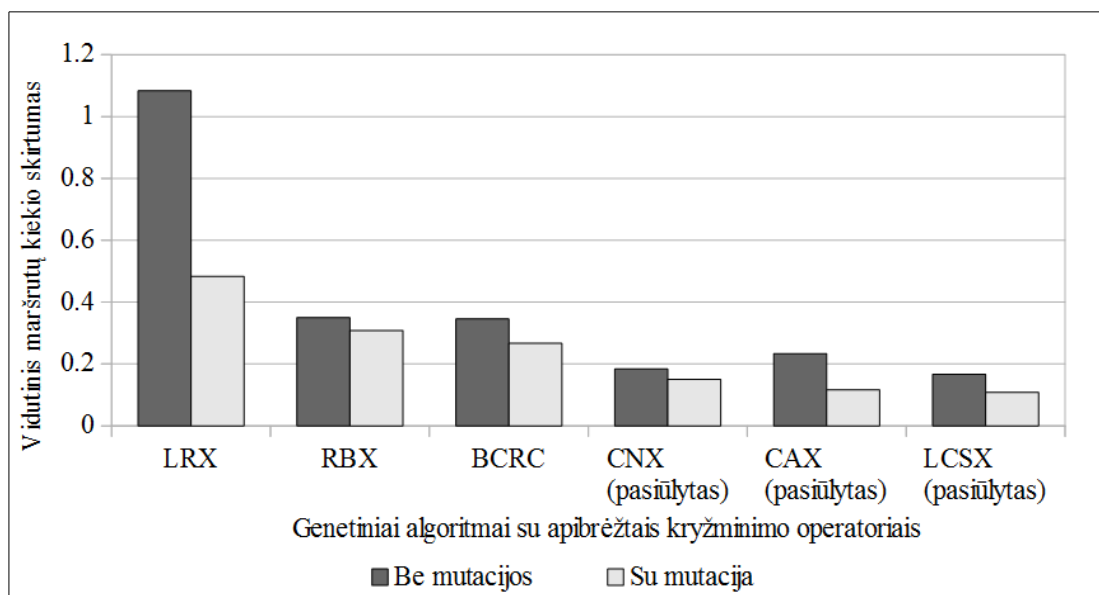
Disertacijoje pasiūlytas genetinis algoritmas buvo testuojamas atliekant eksperimentus su dviem uždavinių aibėmis: Solomon uždavinių aibe [7], kurią sudaro VRPTW uždavinių pavyzdžiai (C1, C2, R1, R2, RC1, RC2), bei Li ir Lim uždavinių aibe [5], kurią sudaro VRPPD uždavinių pavyzdžiai. Pasiūlytas genetinis algoritmas yra įgyvendintas Java programavimo kalba. Skaičiavimai atlikti naudojant asmeninį kompiuterį (Intel Core 2 Duo 2.2 GHz CPU, 4GB RAM). Eksperimentuose buvo naudojami šios genetinio algoritmo parametrų reikšmės:

- $PS_1 = 100$ – populiacijos dydis;
- $PL_1 = 10$ – pertvarkymo operacijų vienoje iteracijoje skaičius;
- $IL_1 = 50$ – maksimalus iteracijų skaičius be pagerinimų;
- $TL_1 = 5$ min. – maksimalus skaičiavimo laikas;
- $MP = 0,1$ – mutacijos taikymo tikimybė;
- $PS_2 = 20$ – antros populiacijos dydis;

- $IL_2 = 5$ – maksimalus iteracijų skaičius be pagerinimų antroje populiacijoje;
- $PL_2 = 2$ – pertvarkymų operacijų vienoje iteracijoje skaičius antroje populiacijoje.



6 pav. Vidutiniai kryžminimo operatoriais rastų sprendinių maršrutų kelio ilgio skirtumai nuo geriausių žinomų sprendinių



7 pav. Vidutiniai kryžminimo operatoriais rastų sprendinių maršrutų kiekio skirtumai nuo geriausių žinomų sprendinių

Gauti sprendiniai yra palyginti su geriausiais sprendiniais, gautais kitais algoritmais iš [1],[2],[3],[4],[5],[6],[7],[8],[9],[10]. Vidutinis rastų maršrutų atstumas bei vidutinis maršrutų skaičius skirtingoms uždavinių kategorijoms yra pateiktas 1 lentelėje, kurioje stambesniu šriftu yra pažymėti geriausi rezultatai. Iš lentelėje pateiktų rezultatų matyti, kad pasiūlytas algoritmas rado geresnius vidutinius rezultatus 4-ioms iš 6-ių uždavinių kategorijų.

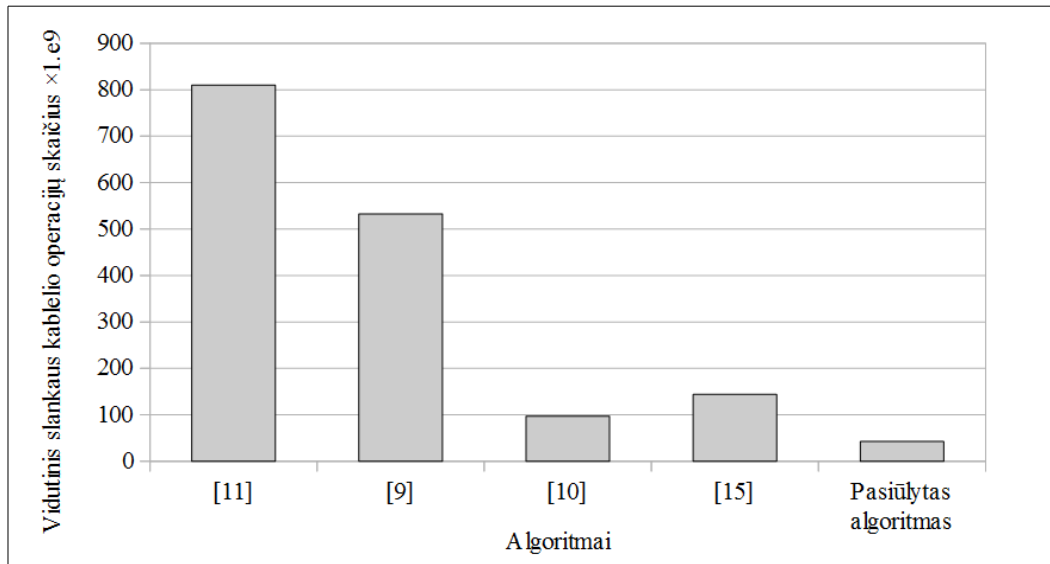
1 lentelė. Skirtingų uždavinių kategorijų gautų sprendinių maršrutų vidutinis kelio atstumas bei vidutinis maršrutų skaičius

	[8]	[9]	[4]	[10]	[6]	[2]	[1]	[3]	Pasiūlytas algoritmas
C1	861 /10,1	860,62 /10,1	833,32 /10	828,9 /10	828,48 /10	828,38 /10	828,38 /10	828,38 /10	828,38 /10
C2	619 /3,3	624,47 /3,3	593 /3	589,86 /3	590,6 /3	589,93 /3	590,9 /3	591,74 /3	589,86 /3
R1	1227 /13,2	1314,79 /14,4	1203,32 /12,6	1242,7 /12,8	1220,92 /12,5	1221,1 /11,92	1224 /11,92	1187,32 /13,08	1213,66 /12,08
R2	980 /5	1093,37 /5,6	951,17 /3,2	1016,4 /3	938,75 /3,1	975,43 /2,73	1012 /2,73	897,95 /4	961,44 /2,73
RC1	1427 /13,5	1512,94 /14,6	1382,06 /12,8	1412 /13	1386,35 /12,12	1389,89 /11,5	1417 /11,5	1348,22 /12,63	1370,01 /11,75
RC2	1123 /5	1282,47 /7	1132,79 /3,8	1201,2 /3,7	1132,12 /3,38	1159,37 /3,25	1195 /3,25	1036,65 /5,63	1126,75 /3,25

Verta paminėti, kad VRPTW sprendiniai buvo rasti pasiūlytu genetiniu algoritmu vidutiniškai per 38,97 sekundės atliekant skaičiavimus asmeniniu kompiuteriu su Intel Core 2 Duo 2,2 GHz procesoriumi (vienos šerdies sparta – 1,09 Gflops/s). Straipsnyje [1] pateikti eksperimentiniai rezultatai buvo gauti per 15 minučių kompiuteriu Pentium IV 2,4 GHz (0,9 Gflops/s), straipsnyje [10] pateikti rezultatai buvo gauti per 592 sekundes kompiuteriu Pentium IV 2,4 GHz (0,9 Gflops/s) bei straipsnyje [2] pateikti rezultatai buvo gauti per 30 minučių kompiuteriu Pentium 400 MHz (54 Mflops/s). Straipsnyje [3] pateikti eksperimentiniai rezultatai buvo gauti per 117 sekundžių naudojant kompiuterių klasterį, sudarytą iš AMD Opteron 2,6 GHz (vienos šerdies sparta – 1,23 Gflop/s) procesorių. Vidutinis slankaus kablelio operacijų, atliktų ieškant VRPTW uždavinių rezultatų, skaičius yra pateiktas 8 paveikslėlyje. Iš jo matyti, kad pasiūlytas genetinis algoritmas sprendiniams rasti panaudojo vidutiniškai 2 kartus mažiau slankaus kablelio operacijų lyginant su kitų algoritmų geriausiu slankaus kablelio operacijų skaičiumi.

VRPPD uždavinių rezultatai parodė, kad pasiūlytu genetiniu algoritmu rasti geriausi sprendiniai ~96 % atvejų yra lygūs geriausiems žinomiems rezultatams, o rasti vidutiniai rezultatai ~80 % atvejų yra lygūs geriausiems žinomiems rezultatams.

Disertacijos eksperimentinių tyrimų dalyje taip pat yra pateikiami lygiagretaus Dijkstra trumpiausio kelio paieškos algoritmo eksperimentinių tyrimų rezultatai. Tyrimai apima du skirtingus eksperimentus su realaus Lietuvos kelių tinklo duomenimis. Pirmame eksperimente buvo skaičiuojami visi trumpiausi keliai tarp parinktų 5 vietų kelių tinkle. Antrame eksperimente buvo ieškoma trumpiausių kelių tarp atsitiktinai parinktų 10, 20, 30, 40, 50 vietų kelių tinkle. Tyrimai parodė, kad dvikryptis Dijkstra algoritmas yra ~2 kartus greitesnis už standartinį, o lygiagretus Dijkstra algoritmas yra ~2,9 karto greitesnis už standartinį bei 1,4 karto greitesnis už nuoseklų dvikryptį algoritmą. Dvikrypčio lygiagretaus Dijkstra algoritmo efektyvumas yra 0,7.



8 pav. Vidutinis slankaus kablelio operacijų skaičius $\times 1.e9$

Išvados

Šioje disertacijoje atlikti tyrimai leido padaryti tokias išvadas:

1. Skirtingai nuo kitų kryžminimo operatorių, kur sprendiniai-palikuonys yra sukonstruojami iš sprendinių-tėvų dalių, pasiūlyti kryžminimo operatoriai, identifikuojantys ir išsaugantys tas sprendinių-tėvų dalis, kurios yra bendros, daugeliu atvejų randa tikslesnius rezultatus nei kiti kryžminimo operatoriai. Kai kurie sprendiniai yra lygūs geriausiems žinomiems sprendiniams net ir tais atvejais, kai mutacijos operatorius nėra taikomas.
2. VRPTW uždavinių sprendimo rezultatai rodo, kad pasiūlytas algoritmas, kuris paremtas leistino įterpimo metodu genetiniuose operatoriuose, kryžminimo operatoriais, išsaugančiais bendras sprendinių-tėvų dalis, bei antros populiacijos naudojimu mutacijos operatoriuje, randa geresnius sprendinius 4-ioms iš 6-šių uždavinių grupių lyginant su kitų genetinių algoritmų sprendiniais.
3. Pakartotinai pritaikant atsitiktinio įterpimo euristicą sprendiniai populiacijoje yra diversifikuojami ir kuriant tik leistinus sprendinius neleistinų sprendinių aibė yra neanalizuojama. Taip yra išvengiama nereikalingų operacijų bei pagreitinamas bendras skaičiavimas. Sprendiniai buvo rasti vidutiniškai per 38,97 sekundes. Pasiūlytas genetinis algoritmas sprendiniams rasti panaudojo vidutiniškai 2 kartus mažiau slankaus kablelio operacijų lyginant su kitų algoritmų geriausiu slankaus kablelio operacijų skaičiumi.
4. Geriausi VRPPD uždavinio sprendiniai, gauti pasiūlytu algoritmu, ~97% atvejų yra lygūs geriausiems žinomiems sprendiniams. Gauti vidutiniai rezultatai ~80% atvejų yra lygūs geriausiems žinomiems sprendiniams.
5. Trumpiausio kelio paieškos eksperimentų rezultatai parodė, kad modifikuotas dvikryptis Dijkstra algoritmas yra 2 kartus greitesnis už standartinį, o lygiagretus Dijkstra algoritmas yra beveik 2,9 kartus greitesnis už standartinį ir beveik 1,4 karto greitesnis už nuoseklų dvikryptį Dijkstra algoritmą.
6. Pasiūlyto genetinio algoritmo mutacijos ir kryžminimo operatoriai yra paremti atsitiktinio įterpimo euristika. Operatoriai nėra pritaikyti tik tam tikriems specifiniams uždaviniams ir gali būti panaudoti kitiems uždaviniams spręsti. Tai leidžia taikyti pasiūlytą genetinį algoritmą bendram transporto maršrutų sudarymo

uždaviniui spręsti. Genetiniame algoritme nėra naudojami jokie atstatymo bei patobulinimo metodai, kurie galėtų būti kliūtimi pritaikant algoritmą naujų apribojimų bei sąlygų sprendimui. Pasiūlyti genetiniai operatoriai nesuardo pagrindinių genetinio algoritmo principų, tad skirtingos tikslo funkcijos, taip pat ir daugiakriterinio optimizavimo funkcijos, gali būtų taikomi sprendiniams populiacijoje įvertinti.

Santraukoje cituota literatūra

- [1] Alvarenga, G. B., de A. Silva, R. M., Sampaio, R. M. (2005). A Hybrid Algorithm for the Vehicle Routing Problem with Time Window, *INFOCOMP Journal of Computer Science*, 4(2), 9–16.
- [2] Berger, J., Barkaoui, M. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 31, 2037–2053.
- [3] Garcia-Najera, A., Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & OR*, 38, 287–300.
- [4] Ho, W.-K., Ang, J. C., Lim, A. (2001). A Hybrid Search Algorithm for the Vehicle Routing Problem with Time Windows. *International Journal on Artificial Intelligence Tools*, 10 (3), 431–449.
- [5] Li, H., Lim, A. (2003). A Metaheuristic for the Pickup and Delivery Problem with Time Windows. *International Journal on Artificial Intelligence Tools*, 12(2), 173–186.
- [6] Ombuki, B. M., Ross, B., Hanshar, F. (2006). Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. *Applied Intelligence*, 24(1), 17–30.
- [7] Solomon, M.M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35(2), 254–265.
- [8] Tan, K. C., Hay, L. L., Ke., O. (2001). A hybrid genetic algorithm for solving vehicle routing problems with time window constraints. *Asia-Pacific Journal of Operational Research*, 18(1), 121–130.
- [9] Tan, K. C., Lee, L. H., Zhu, K. Q., Ou, K. (2001a). Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15(3), 281–295.
- [10] Zhu, K. Q. (2003). A Diversity-Controlling Adaptive Genetic Algorithm for the Vehicle Routing Problem with Time Windows. *Proceedings of the 15th IEEE International Conference on Tools for Artificial Intelligence (ICTAI 2003)*, 176–183.

Trumpos žinios apie autorių

Gintaras Vaira gimė 1983 m. rugsėjo 28 d. Klaipėdoje. 2002 m. baigė Klaipėdos „Ažuolyno“ gimnaziją. 2006 m. Vilniaus universitete įgijo informatikos bakalauro laipsnį, 2009 metais Vilniaus Gedimino technikos universitete įgijo informatikos inžinerijos magistro laipsnį. 2009–2013 m. doktorantas Vilniaus universiteto Matematikos ir informatikos institute.

GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM

Research context and motivation

The vehicle routing problem (VRP) is a well known combinatorial problem that attracts researchers to investigate it by applying the existing and newly created optimization algorithms. Traditionally, the VRP is defined as a routing problem with a single depot, a set of customers, multiple vehicles and the objective to minimize the total cost while servicing every customer. A set of constraints can be defined for the VRP. In literature we can find different kinds of vehicle routing problems (VRPs) that are grouped according to the specific constraints. The well known constrained VRPs are as follows: VRP with capacity limitations (CVRP), where vehicles are limited by the carrying capacity; VRP with time windows (VRPTW), where a customer can be serviced within a defined time frame or time frames; VRP with multiple depots (MDVRP), where goods can be delivered to a customer from a set of depots; VRP with pick-up and delivery (VRPPD), where rules are defined to visit pick-up places and later to deliver goods to the drop-off location. Many researches on different heuristic approaches can be found for the solution of the above mentioned problems.

In recent years, VRP attracts much attention due to the increased interest in various geographical solutions and technologies as well as their usage in logistics and transportation. More and more logistic companies are trying to organize deliveries of goods better by enabling various today's proposed technologies. They can be various logistic systems coupled with widely used positioning systems, etc. The important part in reduction of transportation costs is a better organization of routes by solving a vehicle routing problem. For example, a better organization of fleet routes in various distribution areas – delivery of post, supply delivery to markets, fuel delivery to gasoline stations, etc. – can save fuel, money and/or time that can be used for servicing new customers. Also, a better organization of routes in business deliveries can affect ecological aspects by reducing pollution that is important problem of these days.

Problem statement

In literature we can find algorithms that are designed for one or another VRP, where the algorithms are designed to deal with a specific subject or specific constraints. Although mentioned VRP variants mimic some real world situations, these situations do not reflect the whole problem. The mentioned VRPs are criticized for being too focused on specific models that involve non-realistic assumptions. Real-world VRP with various constraints generalizes traditional VRP and is usually called a rich vehicle routing problem (RVRP). Solving RVRP has been a challenging today's task.

A number of different exact and heuristic methods have been studied to solve the VRP that is known to be NP-hard. Although the exact methods give the optimal solution, their computation time considerably increases with the increasing size of the problem. Various heuristic methods exist for solving problems that are known to be NP-hard. Local searches and heuristic approaches may be sensitive to the given data sets (i.e., constraints) or require additional training data during the learning process. Also hybrid combinations of various algorithms are designed while seeking for higher efficiency in the computation.

Metaheuristic is another approach for solving a complex problem that may be too difficult or time-consuming for other techniques. One of the metaheuristics that are investigated for solving VRP is a genetic algorithm (GA). Genetic algorithms are based

on ideas of evolution theory. The main principle here is that only the fittest entities survive. Genetic algorithms work with individuals, sometimes also called chromosomes, each representing a possible solution to a given problem. GA typically works with the initial population of solutions; together with each new generation GA creates a new potential offsprings, based on the selected individuals from the previous generation using a set of stochastic transition operators (crossover and mutation). The iterative process of generations and evaluation of individuals continues until a sufficient stopping criterion is met.

The standard genetic algorithm has limitations in the constrained environment. Due to a stochastic characteristic, genetic algorithms can continue very long until the acceptable solution has been found for a constrained problem. For a constrained problem, the feasible search space is smaller than the whole search space and genetic algorithm operators generate solutions in the whole search space including the infeasible space. The common approaches for constraint handling in genetic algorithms involve additional repair and improvement methods that are designed for a specific constraint to keep the generated solutions in the feasible search space. The repair of one constraint can involve the violation of another constraint. Such approaches can produce an inadequate result when they are applied to different problems and are hardly extendable with new constraints. Specialized algorithms usually are hardly applicable to RVRP.

Tasks and objectives of the research

The objective of the thesis is to design a new genetic algorithm for vehicle routing problem that handles constraints in genetic operators and that can be efficiently applied for solving rich vehicle routing problem.

In order to achieve the objective, the following tasks are stated:

- To study existing genetic algorithms for solving vehicle routing problems.
- To analyze approaches in genetic algorithms for dealing with constraints in vehicle routing problems and investigate search intensification approaches in genetic algorithm operators.
- To analyze the existing formulations of rich vehicle routing problem and detail them.
- To propose a new genetic algorithm for rich vehicle routing problem, where genetic operators handle constraints in solutions in each iteration.
- To investigate Dijkstra's shortest path algorithm speed up techniques in order to efficiently apply the proposed genetic algorithm to the real vehicle routing problems taking into account the road network.
- To evaluate the proposed genetic algorithm by applying it on public available benchmark instances and compare it with other known genetic algorithms.

Practical significance of the results

The practical significance of the thesis is as follows:

- The proposed genetic algorithm can be applied to real-world vehicle routing problem more flexibly and in such way reduce costs for various companies that deal with delivery by reducing overall traveling path and/or traveling time. The algorithm also can be applied to dynamic re-computation of the VRP depending on new data (new requests came from customers; some accident happened for one of the vehicles during the delivery; etc.).
- The proposed algorithm can be applied to any problem that can be defined as a graph and which solution depends on the sequence of the elements.

- A part of the research was used in the project “Algorithm for optimizing the route between N points and algorithm for fixing deviations and mathematical averaging of fuel level data from transport means (fuel filling pouring off)” based on the agreement between “Institute of Mathematics and Informatics”, JSC “AKTKC – Apsaugos centras” and “Agency for Science, Innovation and Technology” for achievement of innovation voucher (agreement No 31V-79, 2010-07-28).

Research design and research methods

Both the exploratory research and systematic review have been used to collect and summarize the results of other researches. Experimental research and generalization method have been used to evaluate the proposed methods and algorithms in comparison with the obtained results in other researches.

Defensive propositions

1. Insertion heuristic in genetic algorithms is suitable not only to produce initial solutions, but also can be incorporated in genetic operators for constraint handling, i.e. for generation of feasible partial solution in each iteration by evaluating constraints. By repeatedly applying destroy method and random insertion heuristic, diversification is enabled in the population and, by dealing only with feasible solutions, infeasible search space is not examined, thus avoiding unnecessary computation and increasing overall computation speed.
2. The crossover operators that preserve common sequence from two parent solutions can intensify the search towards the optimal solution. In contrast to traditional crossover approaches, where offspring solutions are constructed from the parts of parent solutions, new crossovers define the degree of the destruction by preserving the parts that are common in both parent solutions, thus preserving the parts that have a higher probability to be optimally constructed than the other ones.
3. A genetic algorithm, based on the feasible reinsertion approach in genetic operators, on crossovers preserving common parts, and on second population in mutation operator, produces similar or better solutions than other genetic algorithms in short computation time. The usage of the second population in the mutation operator increases diversification in the population and overall efficiency of the genetic algorithm. Overall genetic algorithm is applicable to the rich vehicle routing problem.

Proposed solutions and contributions of the scientific novelty

- Operators of the genetic algorithm, that involves the destroy and reconstruct approach of large neighborhood search (LNS) usage in crossover and mutation operators, are proposed, where random insertion heuristic in genetic operators is used as a reconstruction method. Insertion heuristic is adjusted with evaluation of constraints to avoid generation in infeasible search space, thus speeding-up the computation. Random insertion heuristic preserves stochastic characteristics of the genetic algorithm thus involving the diversification in the population.
- New crossover operators that are based on the search of common parts in the parent solutions for generation of the offspring are proposed. In contrast to traditional crossover approaches, where offspring solutions are constructed from parts taken from the parents, the proposed crossovers identify and preserve parts of the solution that are common in both parents, thus intensifying a search towards the optimal solution. The usage of the longest common increasing

sequence (LCIS) search in crossover operator preserves the sequence of elements from parent solutions, where the sequence is important characteristic of vehicle routing problem solutions.

- The genetic algorithm is proposed that involves insertion heuristic, feasibility preservation, a search of common parts in the crossover operators and the second population used in the mutation operator. Solutions obtained in the second population remain competitive in the main population: they have a higher probability to be selected for reproduction and involve the diversification in the population. The proposed algorithm produces solution in short time and solutions are better or equal to results obtained by other genetic algorithms. The advantage of a new developed genetic algorithm is that it can be applied to rich vehicle routing problem and the formulation of the rich vehicle routing problem is also defined in this research.

Approbation and publications of the research

The main results of the thesis were presented at the following international conferences:

- 9th Conference on Databases and Information Systems, DB&IS 2010, July 5 - 7, 2010 – Riga, Latvia;
- 1st International Conference of EURO Working Group on Vehicle Routing and Logistic Optimization (VeRoLog 2012), June 18 - 20, 2012 – Bologna, Italy;
- 25th European Conference on Operation Research (EURO-2012), July 8 - 11, 2012 – Vilnius, Lithuania;
- 26th European Conference on Operation Research (EURO-2013), July 1 - 4, 2013 – Rome, Italy;
- 2nd International Conference of EURO Working Group on Vehicle Routing and Logistic Optimization (VeRoLog 2013), July 7 - 10, 2013 – Southampton, United Kingdom.

List of Publications

Articles in the reviewed scientific periodical publications:

- G. Vaira, O. Kurasova. Parallel Bidirectional Dijkstra's Shortest Path Algorithm. *Databases and Information Systems VI*, Volume 224 of Frontiers in Artificial Intelligence and Applications, p. 422–435, IOS Press, 2011, ISSN 0922-6389 (print), ISSN 1879-8314 (online).
- G. Vaira, O. Kurasova. Genetic algorithms and VRP: the behaviour of a crossover operator. *Baltic Journal of Modern Computing*, 1(3–4), p. 161–185, 2013, ISSN 2255-8942 (print), ISSN 2255-8950 (online).
- G. Vaira, O. Kurasova. Genetic Algorithm for VRP with Constraints based on Feasible Insertion. *Informatika*, 25(1), p. 155–184, 2014, ISSN 0868-4952.

Articles in other editions:

- G. Vaira, O. Kurasova. Modified bidirectional shortest path Dijkstra's algorithm based on the parallel computation. In *Proceedings of the 9th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2010)* (J. Barzdins, M. Kirikova (eds.)), p. 205-217 Riga: University of Latvia Press, 2010, ISBN 978-9984-45-199-2.
- G. Vaira, O. Kurasova. Feasible Insertion Genetic Algorithm for VRP with Constraints. In *Proceedings of 2nd International conference of EURO Working*

Outline of the Dissertation

The text of the thesis consists of introduction, 3 main chapters, conclusions and references. Each chapter is provided with the summary (except introduction and conclusions). The total scope of this thesis is 140 pages, 26 figures and 19 tables.

Introduction describes research context and motivation, presents the statement of the problem, discusses tasks and objectives of the research, methodology of research, presents practical significance of results, scientific novelty, defending propositions and approbation of obtained results.

Chapter 1 provides overview of vehicle routing problems and solutions, reviews genetic algorithms for solving vehicle routing problems in details.

Chapter 2 describes the proposed genetic algorithm for vehicle routing problem.

Chapter 3 provides experimental evaluation of the proposed algorithms.

Conclusions present the main conclusions of the thesis.

Conclusions

The research completed in this thesis has led to the following conclusions:

1. In contrast to crossover operators, where solutions are constructed from parts of the parent solutions, the proposed crossover operators, that search and preserve parts of the solution that are common to both parents, find the results that in most of the cases are more accurate than the ones found by other crossover operators. Some solutions are equal to the best known solutions even in the cases, where mutation was not applied.
2. As results of VRPTW instances show, the proposed algorithm, based on feasible reinsertion approach in genetic algorithm operators, on crossovers preserving common parts, and on the second population in mutation operator, finds better solutions for 4 out of 6 problem instance groups in comparison with other genetic algorithm approaches.
3. By repeatedly applying random insertion heuristic, the diversification is enabled in the population and, by dealing only with feasible solutions, infeasible search space is not examined, thus avoiding unnecessary computation and increasing overall computation speed. The solutions are found on average in 38.97 seconds. The proposed genetic algorithm performs ~2 times less floating point operations to find the results comparing to the best value of other algorithms.
4. The best solutions for VRPPD instances, obtained by the proposed algorithm, are equal to the best known solutions in ~96% cases. The found average solutions are equal to the best known solutions in ~80% cases.
5. The results of the shortest path search experiments indicate that the modified bidirectional Dijkstra's algorithm is 2 times faster than the standard one and the parallel Dijkstra's algorithm is almost 2.9 times faster than the standard one and 1.4 times faster than the bidirectional Dijkstra's algorithm.
6. Mutation and crossover operators in the proposed genetic algorithm are based on a random insertion heuristic. The operators are not designed to a certain specific problem and can be applied to different problems. The proposed algorithm can be applied for the rich vehicle routing problem. No additional repair or improvement methods are used that could be a problem for extending scheme with a new

constraint handling. Proposed genetic operators do not break main genetic algorithm principles, so different objective functions can be applied to ranking solutions in the population, including multi-objective approaches.

About the Author

Gintaras Vaira was born in Lithuania in Klaipėda on 28 August 1983. In 2002, he finished Klaipėda “Ažuolynas” Gymnasium. He obtained Bachelor’s degree from Vilnius University in 2006, and Master’s degree from Vilnius Gediminas Technical University in 2009, both in the field of informatics. Since 2009 to 2013 he has been a doctorate at Vilnius University Institute of Mathematics and Informatics.