

ŠIAULIŲ UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

Donatas Kavaliauskas

Informatikos specialybės IV kurso dieninio skyriaus studentas

**Dirbtinės bičių kolonijos algoritmai
ir jų taikymai maršrutų optimizavimo uždaviniams spręsti**
Artificial Bee Colony Algorithms and their Application to Route Optimisation Problems
BAKALAURO DARBAS

Darbo vadovas:

Doc. dr. G. Felinskas

Recenzentas:

Šiauliai, 2013

„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąrašą pateiktus informacinius šaltinius bei savo tyrimų duomenis“

Darbo autoriaus _____
(vardas, pavardė, parašas)

Darbo tikslai ir uždaviniai

Tikslas:

- Išanalizuoti dirbtinės bičių kolonijos algoritmus ir sukurti programinę realizaciją maršrutų optimizavimo uždaviniams spręsti.

Uždaviniai:

- Susipažinti su dalelių spiečių sistemų (angl. Swarm systems) algoritmais;
- Susipažinti su dirbtinės bičių kolonijos (angl. Artificial Bee Colony, ABC) algoritmo paradigmomis bei taikymais;
- Susipažinti su maršrutų parinkimo bei jų optimizavimo uždavinių formuluotėmis, interpretacijomis ir taikymais;
- Suprojektuoti ABC algoritmo funkcijas ir šio algoritmo procedūrų praktines interpretacijas;
- Sukurti dirbtinės bičių kolonijos algoritmo programinę realizaciją ir ją pritaikyti maršrutų optimizavimo uždaviniams spręsti;
- Atlikti sukurtos programinės įrangos testavimą, pateikti skaičiavimo rezultatus.

Darbo vadovo _____

(vardas, pavardė, parašas)

Turinys

IVADAS	5
1. ANALITINĖ DALIS	6
1.1 Temos analizė	6
1.2 Darbinės srities analizė	7
1.2.1 Uždavinio formulavimas	7
1.3 Darbinės srities modelis	7
1.3.1 Kolektyvinis intelektas	7
1.3.2 Bičių elgesys gamtoje	8
1.3.3 Bičių kolonijos optimizavimo algoritmas	12
1.3.4 Bičių poravimosi optimizavimo algoritmas	13
1.3.5 ABC algoritmas	14
2. PROJEKTINĖ DALIS	15
2.1 Įrankių ir priemonių pasirinkimas	15
2.2 Pradinis projektas	16
2.3 Darbų planas	17
3. REALIZACINĖ DALIS	18
3.1 Darbų atlikimo atskaita	18
3.2 Galutinis projektas	19
3.3 Problemų sąrašas	22
3.4 Darbo rezultatų analizė	22
3.5 Rekomendacijos	24
IŠVADOS	25
LITERATŪROS IR INFORMACINIŲ ŠALTINIŲ SĄRAŠAS	26
ANOTACIJA	28
SUMMARY	28
Priedai	29
1. CD turinys:	29
2. Skaičiavimai	30

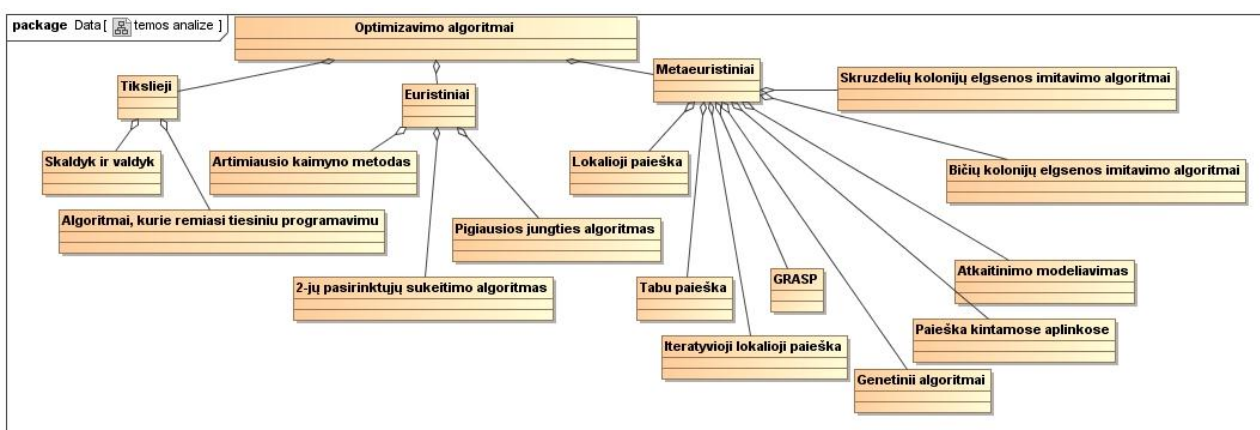
IVADAS

Viena iš aktualių informatikos mokslo sričių yra optimizavimo metodai ir algoritmai. Sprendžiant vis sudėtingesnius optimizavimo uždavinius yra svarbu panaudoti nestandartinius optimizavimo algoritmus, kurie remtųsi ne pavieniais sprendiniais. Vienas iš tokių algoritmų yra pamėgdžiotas gamtos. Tai yra dirbtinis bičių spiečiaus algoritmas. Jame yra naudojamas kolektyvinis intelektas. Bičių spiečiaus algoritmą ir jo principus pirmieji suformavo Yonezawa, Kikuchi, 1995; Lučić, Teodorović, 2001; Karaboga, 2005; Karaboga, Basturk, 2007 [1].

1. ANALITINĖ DALIS

1.1 Temos analizė

Optimizavimu vadinama tokio, uždavinio apibrėžtoje aibėje, elemento paieška, kuriam kriterijaus reikšmė būtų minimali (arba maksimali). Spręsti optimizavimo uždavinius yra sukurta daug algoritmų, kurie yra skirstomi į grupes (Žr. Pav. nr. 1). Tikslųjų optimizavimo grupė garantuoja tikslų sprendinį. Tačiau surasti sprendinį galima tik su mažu duomenų kiekiu, nes patikrinti kiekvieno duomens sprendinį laikas išauga eksponentiškai.[8],[9].



Pav. 1 Optimizavimo algoritmai [8].

Euristiniai algoritmai žymiai greičiau randa apytikslį sprendimą (dažniausiai besiskiriantį nuo optimalaus vos keliais procentais). Algoritmai remiasi taisyklių rinkiniais, kurie vadinami euristikomis. Taigi, euristikos pritaiko algoritmą prie konkretaus optimizavimo uždavinio su apribojimais ir prielaidomis [8],[7],[9].

Metaeuristiniai metodais yra aprašomi kurios nors klasės uždavinių sprendimo idėja, principas. Taigi, metaeuristiniai algoritmai yra virš euristinių algoritmų – kas yra išvystyta, modernizuota tradicinių euristinių algoritmų atžvilgiu. Optimizavimo uždaviniams spręsti yra naudojami metaeuristiniai algoritmai: lokalioji paieška, tabu paieška, genetiniai algoritmai, godžiosios randomizuotos adaptyvios paieškos procedūros (GRASP), skrudėlių kolonijų elgsenos imitavimo algoritmą, paieška kintamose aplinkose, bičių kolonijų elgsenos imitavimo algoritmai ir kt. (Pav. 1). Visi šie algoritmai yra skirti spręsti uždavinius: tvarkaraščių optimizavimui, keliaujančio pirklio uždavinys, tinklo paskirstymo optimizavimas, šviesoforų šviesų sinchronizavimui ir kt. [7], [8], [9].

Kadangi, optimizavimo metodika galima spręsti daug uždavinių, buvo nuspręsta dirbtinių bičių spiečiaus algoritmą realizuoti keliaujančio pirklio uždavinių grupei.

1.2 Darbinės srities analizė

Keliaujančio pirklio uždavinys (sutr. KPU; ang. „Traveling salesperson problem“ - TSP) yra gerai žinoma ir svarbi kombinatorinio optimizavimo problema. Šios problemos principas yra apkelti visus miestus (aplankant miestą tik vieną kartą), kuo trumpesniu keliu. Dažniausiai yra grįžtama į pradinį miestą [2],[5].

Keliaujančio pirklio uždaviniai yra naudojami daugelyje sričių:

- Tvarkaraščių sudarymas;
- Transporto organizavime. Pvz. autobuso maršruto suplanavimas, kai autobusas turi apvažiuoti tam tikras stoteles;
- GPS maršruto apskaičiavimas iš taško A į tašką B;
- Pilnajame svoriniame grafe surasti mažiausio svorio Hamiltono¹ ciklą.

1.2.1 Uždavinio formulavimas

Matematiškai šis uždavinys yra formuluojamas taip: turima n miestų, kurie sudaro $G=(V,E)$ viršūnių aibę $V = \{v_i : 1 \leq i \leq n\}$. Miestai yra sujungti keliais ir žinomas atstumas tarp šių miestų. Keliai sudaro grafo G briaunų aibę $E = \{e_{ij} : e_{ij} = (v_i, v_j), 1 \leq i, j \leq n\}$. Jeigu, grafas yra nepilnasis, tada trūkstamas briaunas yra pakeičiamos labai dideliu atstumu. Pirklys, išėjęs iš pirmo miesto, turi aplankyti visus miestus po vieną kartą ir grįžti į pradinį miestą [2][7].

1.3 Darbinės srities modelis

1.3.1 Kolektyvinis intelektas

Algoritmai, kurie yra grindžiami manipuliavimu tarp sprendinių rinkinių ar grupių, šiuolaikinėje literatūroje yra vadinama kolektyvinio intelekto (angl. swarm intelligence) algoritmais (Bonabeu ir kt., 1999; Karaboga, Akay, 2009). Išskiriami tokie algoritmai: dirbtinis bičių spiečiaus algoritmas (angl. Artificial bee colony algorithm), skruzdėlių kolonijos metodas (angl. Ant colony optimization), krioklio algoritmas (angl. Intelligent Water Drops algorithm), minios metodas (angl. Multi-swarm optimization) ir kt. Tokių metodų pagrindas paprastų gyventojų, agentų, bičių sąveikavimas vienas su kita ir su savo aplinka. Taigi, agentai sąveikaudami tarpusavyje, dalijasi informacija, ją apdoroja, pasiskirsto savo funkcijas kolektyve. Tai yra svarbiausi veiksniai, kurie leidžia siekti geriausių įmanomų rezultatų sprendžiant uždavinius [1],[3],[6].

¹ Grafas, kuris yra pilnasis, yra Hamiltono grafas.

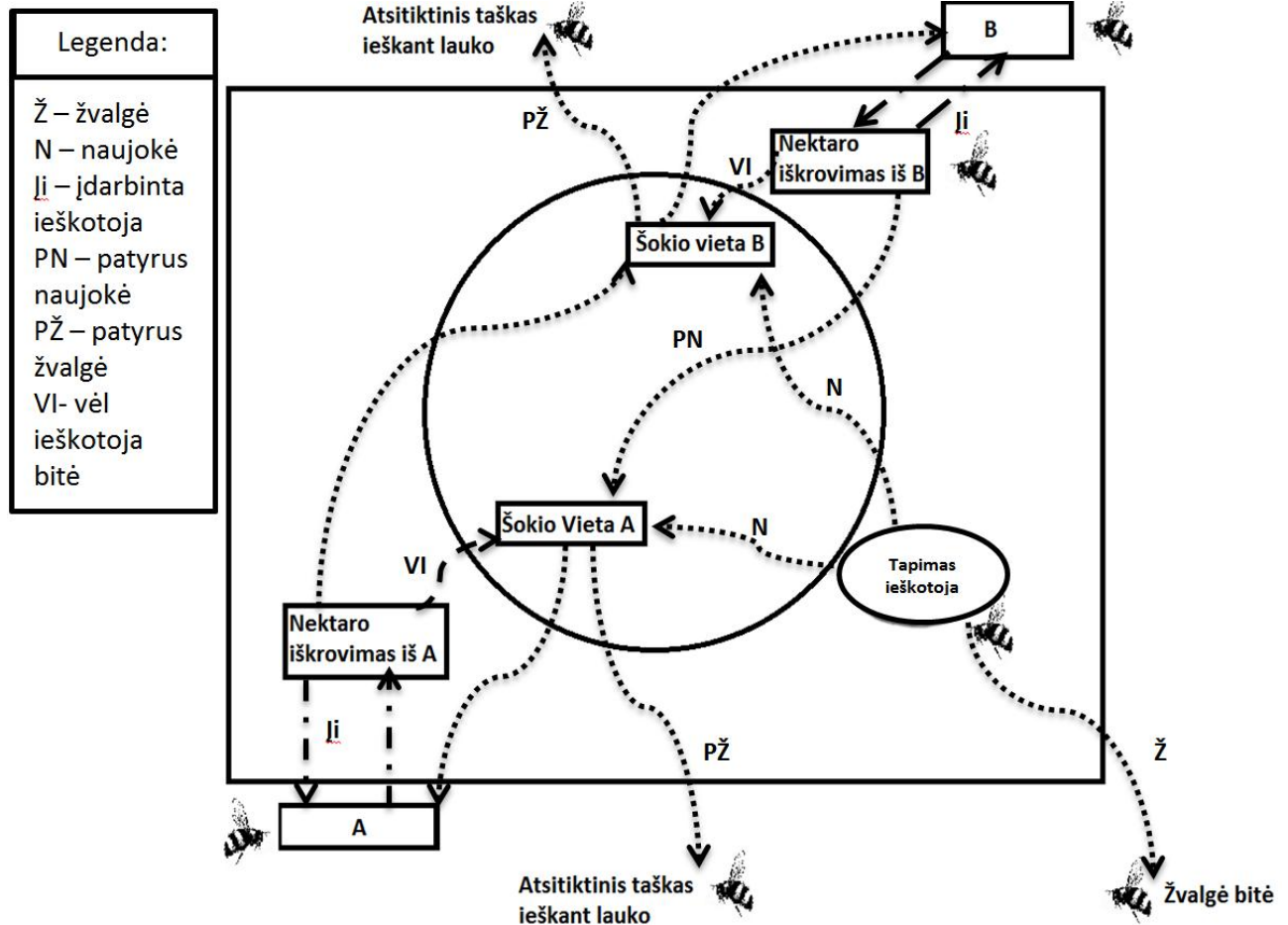
1.3.2 Bičių elgesys gamtoje

Bitės gamtoje veikia, kaip dinamiška sistema, kurios renka informaciją iš aplinkos ir pritaiko savo elgesį atsižvelgiant į gautą informaciją. Kol renkama informacija ir bandoma prisitaikyti prie aplinkos, bitės neatlieka visas savo grupės užduotis. Apskritai, visas socialinės vabzdžių kolonijos elgesys priklauso nuo pasidalijimo darbuose, susijusiuose su jų morfologija. Bičių kolonija susideda iš dviejų pagrindinių komponentų:

- Maisto šaltinių (angl. food sources). Maisto šaltinio vertė priklauso nuo įvairių parametru, kaip jo artumą prie lizdo, maisto kiekį;
- Ieškotoja (angl. foragers). Ieškotojos yra trijų rūšių:
 - Neįdarbintos ieškotojos (angl. unemployed foragers): jei manoma, kad bitė neturi informacijos apie maisto šaltinio vietą, ji inicijuoja save kaip neįdarbinta bitė. Yra dvi neįdarbintų bičių rūšys:
 - Žvalgė bitė (angl. scout bee): tai bitė, kuri pradeda ieškoti maisto šaltinio atsitiktinai skraidydama;
 - Naujokė bitė (angl. recruit): tai bitė, kuri neturėdama darbo skrenda į šokio vietą, kad gautų iš kitų bičių informacijos apie maisto šaltinius. Ši bitė nusikopijuoja informacija ir pagal ją pradės ieškoti maisto.
 - Įdarbinta ieškotoja (angl. employed foragers): kai naujokė bitė randa maisto šaltinį, ji yra paaugštinama iki įdarbintos ieškotojos, kurios įsimina kelią iki maisto šaltinio. Kai bitė paima dalį nektaro, kurį gali panešti, ji grįžta į avilį padėti nektarą maisto saugykloje. Yra trys galimi variantai, susiję su maisto šaltinio vieta:
 - Jei palikto nektaro kiekis yra mažas ar išsekęs, bitė pamiršta kelią į šį šaltinį ir tampa neįdarbinta bitė;
 - Jei palikto nektaro kiekis yra pakankamas, bitė gali skristi toliau rinkti maisto nesidalindama maisto šaltinio informacijos su kitomis bitėmis;
 - Arba bitė gali skristi į šokio vietą ir šokdama informuoti bites apie maisto šaltinį.
 - Patyrusios ieškotojos (angl. experienced foragers): šiuo tipo bitės naudoja savo istorinius atsiminimus.
 - Ji gali būti inspektorė, kuri kontroliuoja nesenai atrastų maisto šaltinių statusą;

- Ji gali tapti vėl ieškotoja bitė, kuri naudojasi informacija iš šokio. Ji bando iširti to pačio maisto šaltinio kokybę, pagal kitas bičių informaciją;
- Ji gali tapti žvalgė bitė ir ieškoti naujų kelių, jei visi maisto šaltiniai išseko;
- Ji gali tapti naujoke bite, kuri ieško naujų maisto šaltinių pagal kitų bičių šokį [1].

Visi šie komponentai atsispindi paveikslėlyje (Žr. Pav. nr. 2).



Pav. 2 Tipiškas bičių maitinimosi elgesys [1].

Pagal bičių elgseną gamtoje yra kuriami įvairūs optimizavimo algoritmai. Per pastarąjį dešimtmetį smarkiai pagausėjo tyrimų su bičių algoritmais literatūroje. Bičių algoritmai yra suskirstyti pagal jų elgsenas į:

- Maitinimosi elgsena;
- Santuokos elgsena;
- Bičių karalienė.

Visa tai atsispindi lentelėje (Žr. lentelė 1) [1].

Tipas	Bičių literatūra	Algoritmas	Taikymas
Maitinimosi elgsena (angl. Foraging Behaviour)	Yonezawa and Kikuchi (1996) Seeley and Buhrman (1999) Schmickl et al. (2005) Lemmens (2006) Sato and Hagiwara (1997)	Bičių sistema (angl. Bee System)	Biologiniai simuliacijai Genetinių algoritmų tobulinimas
	Karaboga (2005)	Dirbtinių bičių spiečiaus algoritmas (angl. Artificial Bee Colony Algorithm – ABC)	Testinė optimizacija
	Yang (2005)	Virtualus bičių algoritmas (angl. Virtual Bee Algorithm – VBA)	Testinė optimizacija
	Basturk and Karaboga (2006)	ABC	Testinė optimizacija
	Pham et al. (2006)	Bičių algoritmas (angl. Bees Algorithm – BA)	Testinė optimizacija
	Lucic and Teodorovic (2001) Bee System (BS) Travelling Salesman Problem(TSP) Lucic (2002)	Bičių sistema (angl. Bee System – BS)	Keliaujančio pirklio uždavinys (KPU) KPU ir stochastinis transporto priemonių maršruto uždavinys
	Lucic and Teodorovic (2002)	BS	KPU
	Lucic and Teodorovic (2003)	BS	KPU
	Lucic and Teodorovic (2003)	BS + neapibrėžta logika	Stochastinis transporto priemonių maršrutas
	Teodorovic and Dell'Orco (2005)	Bičių kolonijos optimizacija + neapibrėžta bičių sistema (angl. Bee Colony)	

		Optimization – BCO + Fuzzy Bee System – FBS)	
	Nakrani and Tovey (2003)	Medaus bičių algoritmas (angl. A Honey Bee Algorithm)	Dinaminis interneto paslaugų paskirstymas
	Wedde et al. (2004)	Bičių avilys (angl. BeeHive)	Telekomunikacijos tinklų maršrutas
	Bianco (2004)		Didelio tikslumo navigacija
	Chong et al. (2006)		Parduotuvės darbų planavimas
	Drias et al. (2005)	Bičių spiečius (angl. Bee Swarm)	Max-W-Sat Problem
	Pham et al. (2006)	BA	LVQ - neuroninis tinklas
	Pham et al. (2006)	BA	MLP - neuroninis tinklas
	Pham et al. (2006)	BA	Neuroninis tinklas
	Quijano and Passino (2007)		Dinaminiai ištekliai
	Markovic et al. (2007)	BCO pagrindais	Maks - maršrutų ir bangų paskirstymas
Santuokos elgsena (angl. Marriage Behaviour)	Abbass (2001) Teo and Abbass (2001, 2003) Bozorg Haddad and Af shar (2004) Bozorg Haddad et al. (2006) Chang (2006)	Bičių santuokos optimizacija (angl. Marriage in Honey-Bees Optimization – MBO) Modifikuotas MBO MBO Bičių poravimosi optimizavimo algoritmas (angl. Honey-bee mating optimization algorithm - HBMO) MBO Pagrindais	3-Sat užduotis 3-Sat užduotis Vandens resursų Vadovavimo užduotis Netiesiškai ribotas ir nesuvaržytas optimizavimas stochastinis dinaminis programavimas
	Afshar et al. (2007)	Patobulintas HBMO	Testinė optimizacija

Bičių karalienė (angl. Queen Bee)	Fathian et al. (2007)	HBMO pagrindais	Duomenų gavyba
	Koudil et al. (2007)	MBO pagrindais	Integruoto disko
	Benatchba et al. (2005)	MBO pagrindais	dalijimo/
	Sung (2003)	Bičių karalienės evoliucijos algoritmas (angl. Queen-Bee Evolution Algorithm – QBE)	tvarkaraščiai Duomenų gavyba Genetinio algoritmo tobulinimui
	Qin et al. (2004)	QBE Pagrindais	Ekonominės galios siuntimas
	Kara (2004)	Bičių kryžminimas (angl. Bee Crossover)	Genetinio algoritmo tobulinimui
	Azeem and Saad (2004)	Modifikuotas QBE	Genetinio algoritmo tobulinimui

Lentelė 1 Bičių algoritmai ir jų taikymai [1].

Darbe bus aptarti keli iš išvardytų algoritmų.

1.3.3 Bičių kolonijos optimizavimo algoritmas

Bičių kolonijos optimizavimo (angl. Bee Colony Optimization) algoritmas yra dirbtinių bičių populiacijos paieška kombinatorinio optimizavimo uždaviniams spręsti. Algoritmas susideda iš dviejų pakaitomis einančių etapų: priekinio paso (angl. forward pass) ir atgalinio paso (angl. backward pass). Kiekviename priekiniame pase visos bitės tyrinėja paieškos erdvę. Taip stengdamosi pagerinti sprendinį. Gavus naują dalinį sprendinį, bitės skrenda atgal į lizdą ir pradeda antrąjį etapą, kuris vadinasi atgalinis pasas. Šiame etape visos dirbtinės bitės dalinasi informacija apie gautą sprendinį. Bičių kolonijos optimizavimo algoritmą galima suformuoti taip:

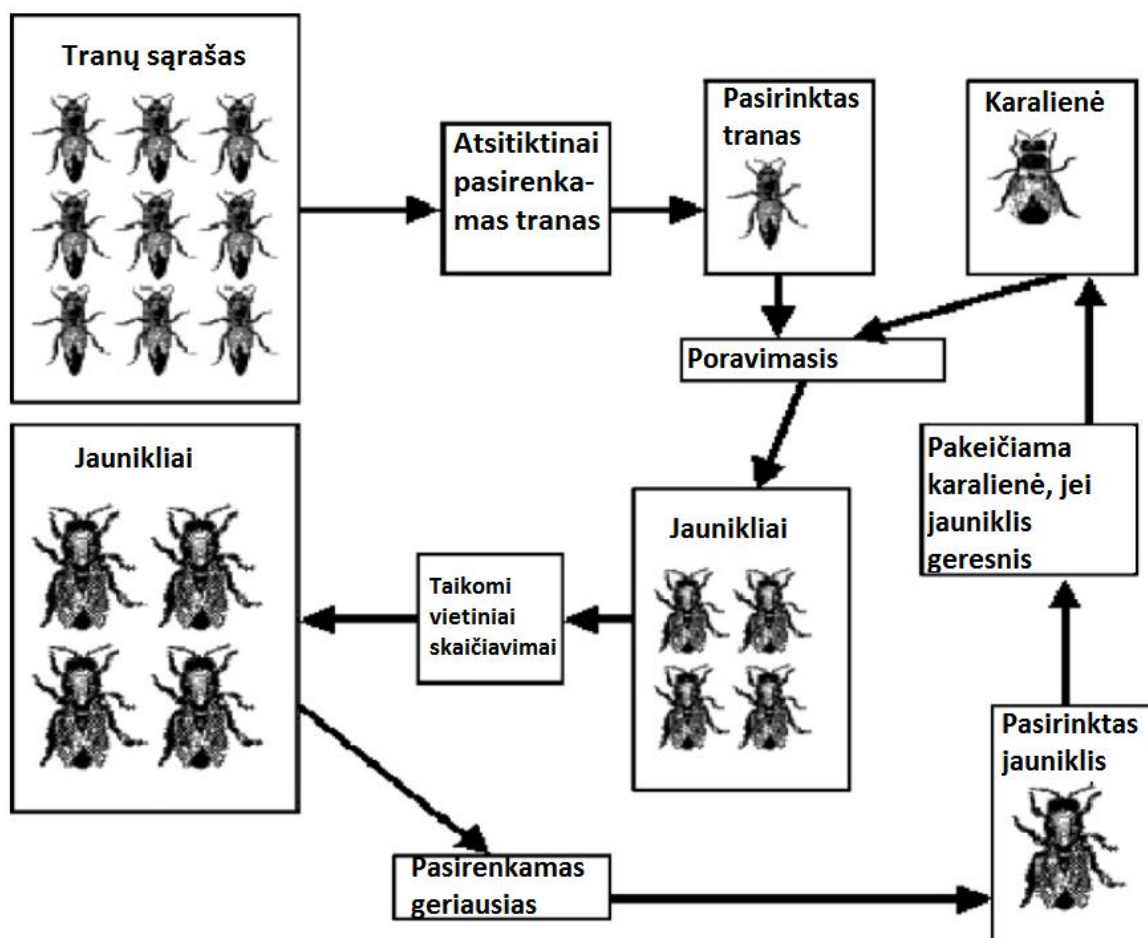
1. Inicializacija: kiekvienai bitei nustatomas tuščias sprendinys;
2. Kiekviena bitė atlieka priekinį pasą:
 - a. Įvertina visus galimus kelius.
 - b. Pagal įvertinimą pasirenkamas kelias.
3. Visos bitės grįžta į lizdą (atgalinis pasas);
4. Surūšiuojamos bitės pagal jų sprendinio reikšmę;
5. Kiekviena bitė atsitiktinai nusprendžia ar tęsti savo tyrinėjimą ir tapti darbininke ar tapti žiūrovėmis ir tyrinėti kitos bitės pasiektą geresnį sprendinį;
6. Grįžtama į antrą žingsnį, kol neįgyvendinama pabaigimo sąlyga [10].

1.3.4 Bičių poravimosi optimizavimo algoritmas

Bičių poravimosi optimizavimo algoritmas (angl. Honey-bee mating optimization algorithm) yra plačiai naudojamas kaip paieškos ir optimizavimo įrankis, įvairiose probleminėse srityse, kaip komercija, inžinerija, moksliniai darbai. Algoritmą sudaro viena karalienė (angl. queen), keletas tūkstančių tranų (angl. drone) ir darbininkės bitės (angl. workers) ir trano ir karalienės jaunikliai (angl. brood). HBMO algoritmas gali būti suformuotas iš šių etapų:

1. Algoritmas prasideda poravimosi skrydžiu, kur karalienė (geriausias sprendinys) ir atsitiktinai pasirinktas tranas suporuojami, kad siekiant gauti jauniklių;
2. Sukuriami nauji jaunikliai, kryžminant trano genus su karalienės;
3. Darbininkės atlieka vietinius skaičiavimus su jaunikliais;
4. Darbininkių bičių tinkamumas pritaikymas, grindžiamas pasiektų jauniklių tobulinimo suma;
5. Pakeičiamas karalienės prastesnis rezultatas geresniu jauniklio rezultatu.

Visas algoritmas pavaizduotas paveikslėlyje (Žr. Pav. Nr. 3).



Pav. 3 HBMO algoritmas

HBMO algoritmas yra panašus į genetinį algoritmą, nes Tranai su karaliene atlieka kryžminimo funkciją[11],[12].

1.3.5 ABC algoritmas

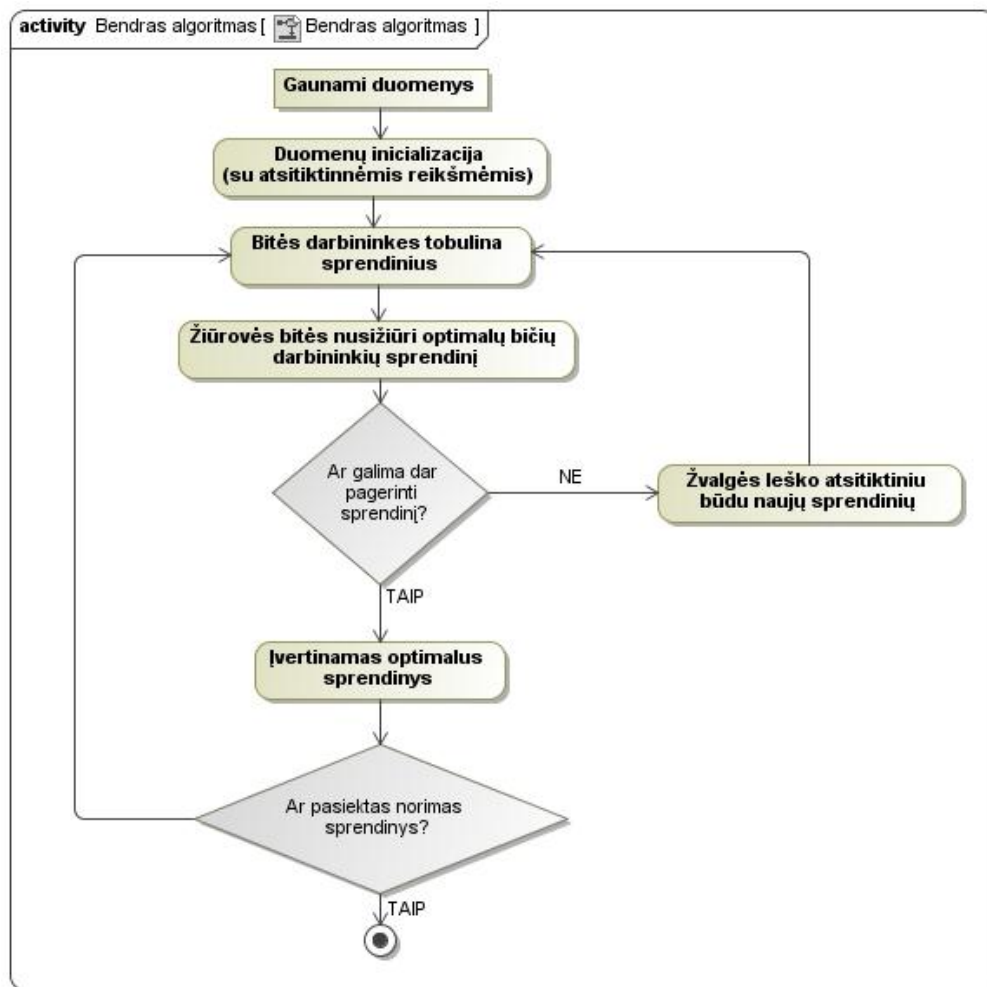
Dirbtinių bičių spiečiaus algoritmas (ABC) yra optimizavimo algoritmas grindžiamas tam tikru manipuliavimu sprendinių rinkiniais, grupėmis. Kadangi ABC algoritmas priklauso metaeuristinių optimizavimo metodų pogrupiui, jis negarantuoja, kad bus rasti optimalūs sprendiniai. Tačiau, šis algoritmas leidžia gerokai sumažinti skaičiavimų kiekį, o tai aktualu skaičiuojant uždavinius su dideliais kiekiais duomenų ir sutaupo kompiuterinių skaičiavimų išteklius. Taip pat dažniausiai gaunami sprendiniai aukštos kokybės. Tai yra arti optimalaus sprendinio [1].

Bičių spiečiaus algoritmas susideda iš trijų bičių grupių: darbininkės bitės (angl. employed bees), žiūrovės (angl. onlookers) ir žvalgės (angl. scouts).

Žiūrovės bitės ir žvalgės bitės taip pat gali būti vadinamos nedarbinėmis bitėmis. Darbininkės bitės ieško geresnio kelio iki maisto šaltinio, nei prieš tai žinojo. Jei rado geresnį maisto šaltinį jos įsimina kelią ir grįžta atgal į šokio vietą pasidalinti informacija apie maisto kiekį su žiūrovėmis, kurios laukia šokio vietoje [1],[3],[4],[5],[6].

Visos žiūrovės bitės laukia darbininkių bičių šokio, kad sužinotų maisto šaltinio vietą ir jo kiekį. Išanalizuoja visų šokusių bičių maisto šaltinius ir atsirenka geriausią šaltinį, kuri nusikopijuoja [1],[3],[4],[5],[6].

Bitės darbininkės, kai nebegali pagerinti kelio prie maisto, gali nuspręsti tapti žvalgėmis. Tada bando ieškoti atsitiktinai naujo kelio prie neatrastų maisto šaltinių. Pagal tai galima sudaryti ABC algoritmą (Žr. Pav.4) [1],[3],[4],[5],[6].



Pav. 4 ABC algoritmas.

2. PROJEK TINĖ DALIS

2.1 Įrankių ir priemonių pasirinkimas

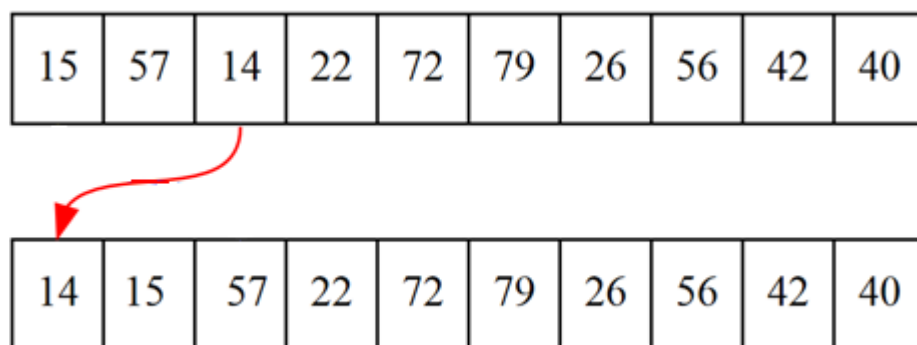
Įrankio pasirinkimą nulėmė, tai, jog ABC algoritmo realizacija pritaikyta interneto svetainėms, kad žmonės galėtų algoritmu naudotis, nepriklausomai nuo operacinės sistemos. Taigi, pasirinkimas buvo tarp PHP, C#, JavaScript, JAVA programavimų kalbų. C# ir PHP programavimo kalbos nebuvo pasirinktos, nes jų kodas yra vykdomas serveryje. JAVA buvo atsisakyta, nes daliai vartotojų bus reikalinga papildoma JAVA instaliacija kompiuteriuose iš kurių būtų bandoma pasinaudoti algoritmu. JavaScript kalbos kodas yra vykdomas kliento pusėje. Tai reiškia, kad kodas yra vykdomas vartotojo procesoriuje, o ne svetainės serveryje. Tai sutaupo pralaidumą ir įtampą serveryje. Taipogi, šioje kalboje yra realizuoti masyvai, If sakiniai, For ciklai, funkcijų kūrimas, kurie bus panaudojami įgyvendinant ABC algoritmą.

Notepad++ v6.3.2 teksto redaktorių pasirinkau kaip darbo atlikimo priemonę. Pasirinkimą nulėmė tai, jog rašant kodą JavaScript kalba, nėra reikalinga kokia nors programavimo aplinka, kaip pvz. Microsoft Visual Studio 2010. Taip pat Notepad++ išskiria kalbos bazinius žodžius, todėl kodą yra lengviau skaityti ir suprasti.

2.2 Pradinis projektas

Atlikus ABC algoritmo ir keliaujančio pirklio uždavinio analizę, toliau vykdomas projektavimas ir realizavimas.

- A. Pradiniai duomenys pateikiami iš tekstinio failo, nes ABC algoritmas yra skirtas dirbti su dideliu kiekiu duomenų;
- B. Pradiniai duomenys ABC algoritmui bus sugeneruojami artimiausio kaimyno metodu, kuris yra euristinis algoritmas;
- C. ABC algoritmo ir artimiausio kaimyno algoritmo geriausi sprendiniai bus atvaizduojami greta vienas kito, kad vartotojas galėtų įvertinti atsakymų skirtumus;
- D. Abu algoritmai bus atvaizduojami grafiškai, kad vartotojui būtų lengviau suprasti atsakymą;
- E. Vartotojui leidžiama keisti ABC algoritmo parametrus:
 - a. Bičių spiečiaus skaičių;
 - b. Pakeisti algoritmo iteracijų skaičių.
- F. ABC algoritmo, darbininkų bičių etape, realizuotas perkėlimo (angl. Shift) metodas (Žr. Pav. nr. 5), optimizuojant sprendinį. Tai yra vieno maršruto mazgo perkėlimas į kitą vietą. Į šokio vietą skris tik geriausią sprendinį turinti bitė.

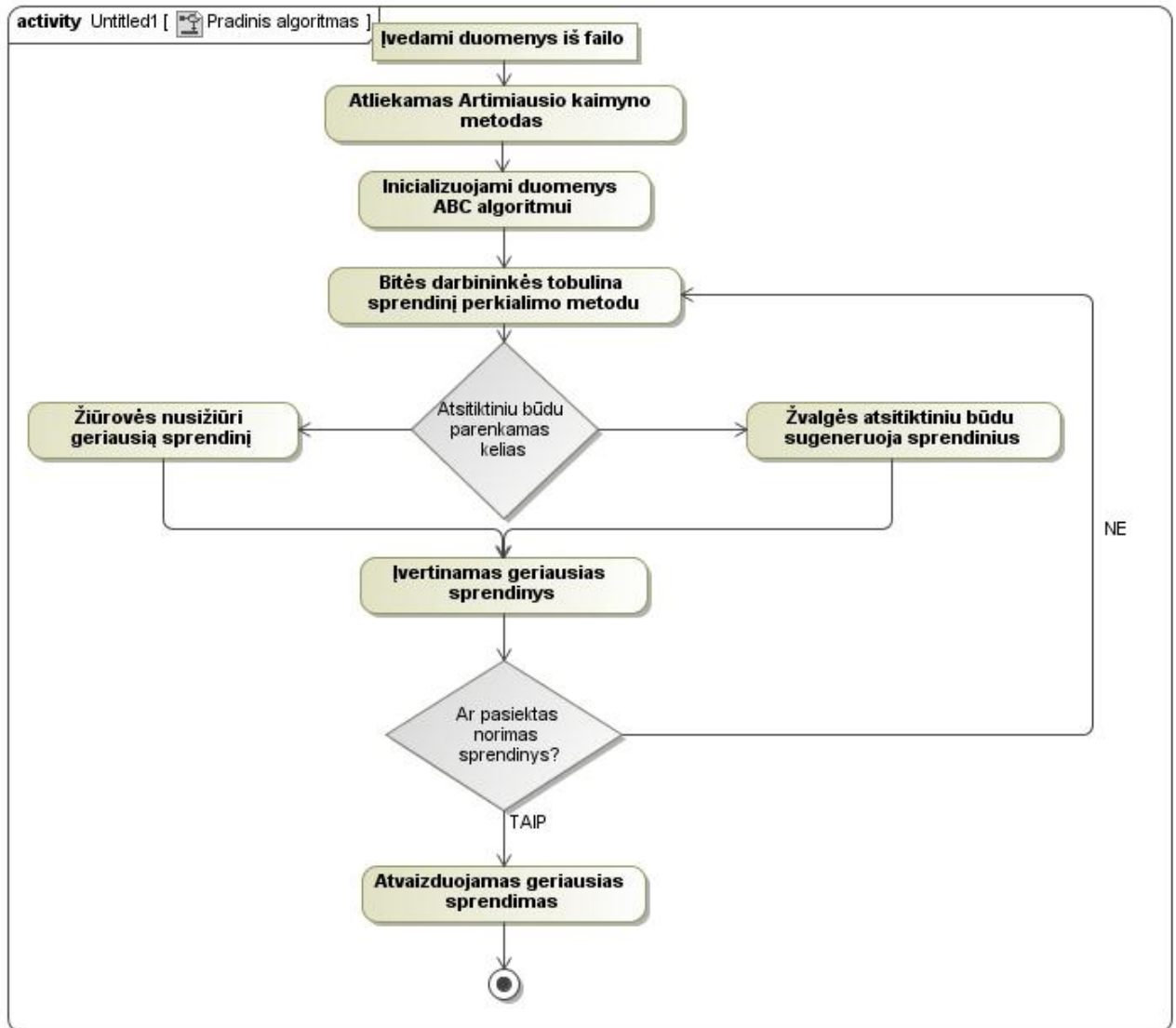


Pav. 5 Perkėlimo funkcija.

- G. Atsitiktiniu būdu parinkti kuo taps darbininkės bitės baigusios optimizuoti sprendinį, ar žvalgėmis, ar žiūrovėmis. Bitės, kurios tapo žvalgėmis, savo turėtą sprendinį pamiršta, ir atsitiktiniu būdu sugeneruoja naują sprendinį. Bitės, kurios tapo

žiūrovėmis, savo turėtą sprendinį pamiršta, ir nusižiūri geriausią sprendinį, kurį turėjo bitė darbininkė.

H. Sudaromas programos algoritmas (Žr. Pav. nr. 6).



Pav. 6 Pradinis programos algoritmas.

2.3 Darbų planas

		Vasaris				Kovas				Balandis				Gegužė	
		Savaitės													
Nr.	Valandų sk.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	40														

2	60														
3	40														
4	30														
5	60														
6	50														
7	40														
Iš viso:	320														

Lentelė 2 Darbų planas.

Darbų paaiškinimas:

1. Informacijos paieška;
2. Informacijos analizė;
3. Aptarimas su darbo vadovu;
4. Artimiausio kaimyno algoritmo realizavimas;
5. ABC algoritmo realizavimas;
6. ABC algoritmo testavimas;
7. Darbo aprašymas.

3. REALIZACINĖ DALIS

3.1 Darbų atlikimo atskaita

		Vasaris				Kovas				Balandis				Gegužė	
		Savaitės													
Nr.	Valandų sk.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	40														
2	55														
3	40														
4	23														
5	60														
6	50														
7	40														
8	12														

Iš viso:	320
----------	-----

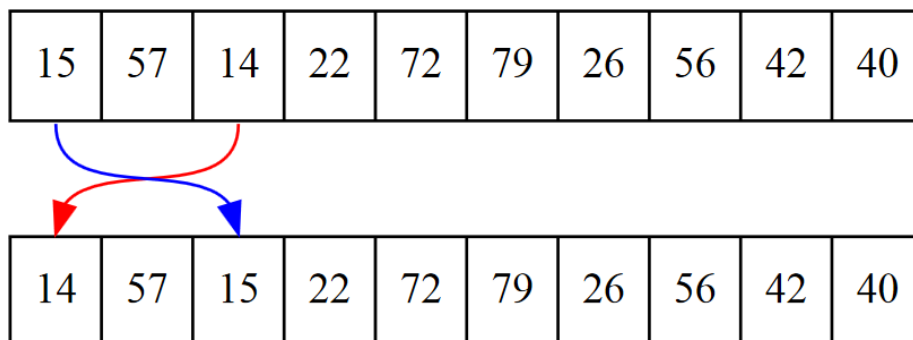
Lentelė 3 Darbų atlikimo atskaita.

Darbų paaiškinimas:

1. Informacijos paieška;
2. Informacijos analizė;
3. Aptarimas su darbo vadovu;
4. Artimiausio kaimyno algoritmo realizavimas;
5. ABC algoritmo realizavimas;
6. ABC algoritmo testavimas;
7. Klaidų taisymas;
8. Darbo aprašymas.

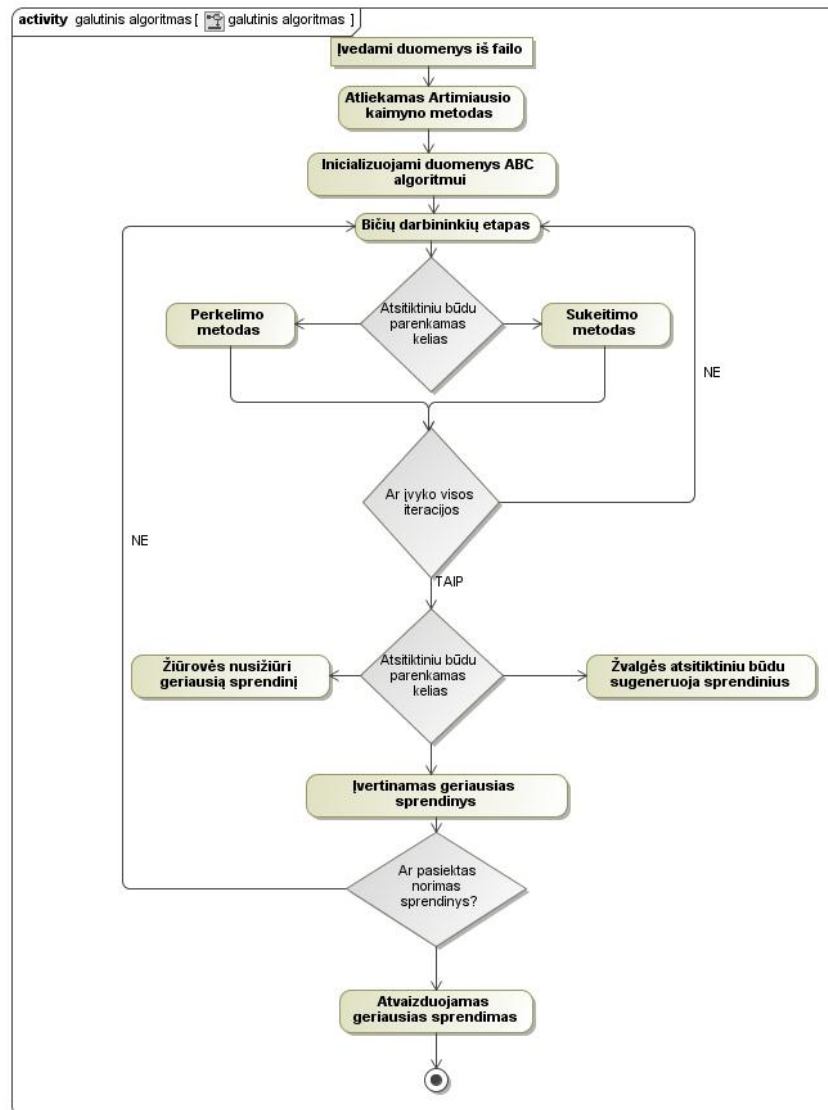
3.2 Galutinis projektas

Šiuo metu sukurta ABC algoritmo realizacija įvykdo visas funkcijas, kurios buvo aprašytos pradiniam projekte. Taip pat yra pridėta darbininkų bičių etape sukeitimo (angl. Swap) funkcija (Žr. Pav. nr. 7).



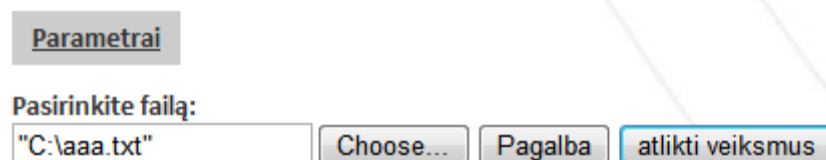
Pav. 7 Sukeitimo funkcija.

Tai yra du maršruto mazgai sukeičiami vietomis. Taigi, dabar darbininkės bitės atsitiktine tvarka atlieka perkėlimo ar sukeitimo funkciją. Be to pridėti papildomi parametrai, kuriais galima valdyti perkėlimo ir sukeitimo funkcijų iteracijų skaičių, bei pakeisti bičių spiečiaus procentą, kurios tampa žvalgėmis ar žiūrovėmis. Realizuotas bičių spiečiaus algoritmas pavaizduotas paveikslėlyje (Žr. Pav. nr. 8).



Pav. 8 Galutinis ABC algoritmas

Programos vartotojo sąsaja pavaizduota paveikslėlyje (Žr. Pav. nr. 9).



Pav. 9 Programos ABC vartotojo sąsaja.

ABC algoritmo parametrų nustatymas yra atskirame skirtuke (Žr. Pav. nr. 10), kuris atsiranda paspaudus mygtuką „Parametrai“.

Parametrai:

Iteracijų skaičius:

Bičių spiečiaus skaičius:

Bičių darbininkų etapas:

Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:

Uždaryti

Parametrai

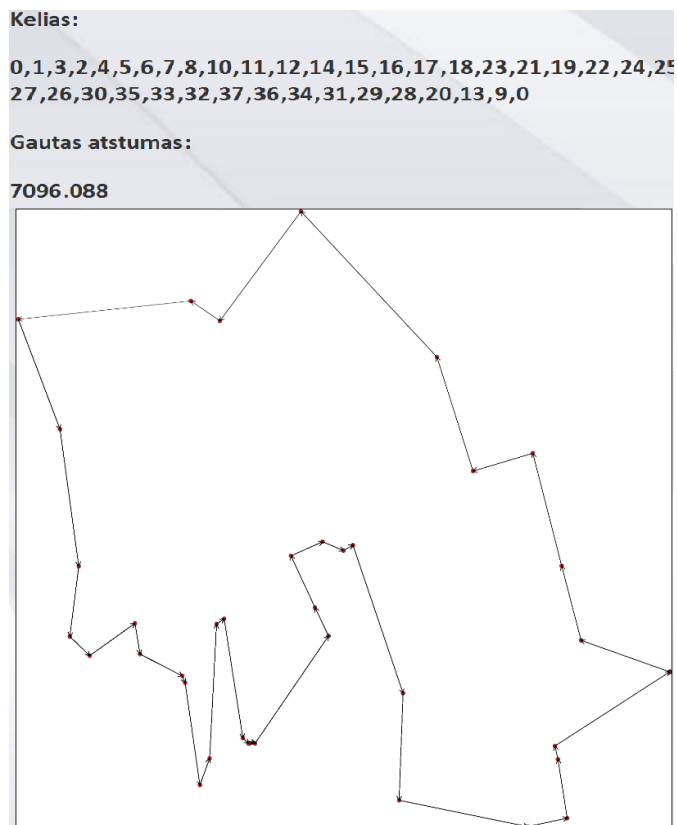
Pasirinkite fail:

Pasirinkti failą

Duomenys:

Pav. 10 Skaičiavimo parametrų įvedimas

Nustačius norimus parametrus, suskaičiuotų rezultatų atvaizdavimas pateiktas paveikslėlyje (Žr. Pav. nr. 11).



Pav. 11 Programos ABC rezultatų atvaizdavimas.

3.3 Problemų sąrašas

Realizuojant ABC algoritmą, susidurta su amžino ciklo problema. Norint ją panaikinti teko įsivesti papildomą pabaigimo sąlygą arba įvesti ribotą ciklų veikimų laiką. Buvo realizuota baigimo sąlyga per iteracijų skaičių.

Realizuojant grafinį maršrutų mazgų atvaizdavimą, susidurta su problema, kad atvaizdavimas netelpa fiksuoto dydžio paveikslėlyje. Šiai problemai spręsti buvo panaudotas CSS (Cascading Style Sheets) failas, kuriame nustatytas piešinio fiksuotas dydis, o mastelio dydis nustatomas kaskart piešiant maršruto mazgą pateikiant apskaičiuotas koordinatas.

3.4 Darbo rezultatų analizė

Sukūrus programą buvo analizuojami gaunami uždavinio rezultatai su žinomu optimaliu keliu keičiant programos parametrus.

Atlikus skaičiavimus (Priedas nr. 2; lentelė nr. 4), ABC algoritme keičiant tik iteracijų skaičiaus parametą, kas 50 vienetų, o kitus parametrus palikus numatytus, pastebėta, jog po 350 iteracijų priartėjama prie optimalaus sprendinio. Kad atispindėtų objektyvus rezultatas kiekvienas bandymas buvo kartotas 20 kartų ir paimtas rezultatų vidurkis. Taip pat dauguma rezultatų yra labai artimi žinomam optimaliam keliui. Tai atspindi diagramoje. (Žr. Diagrama 1).

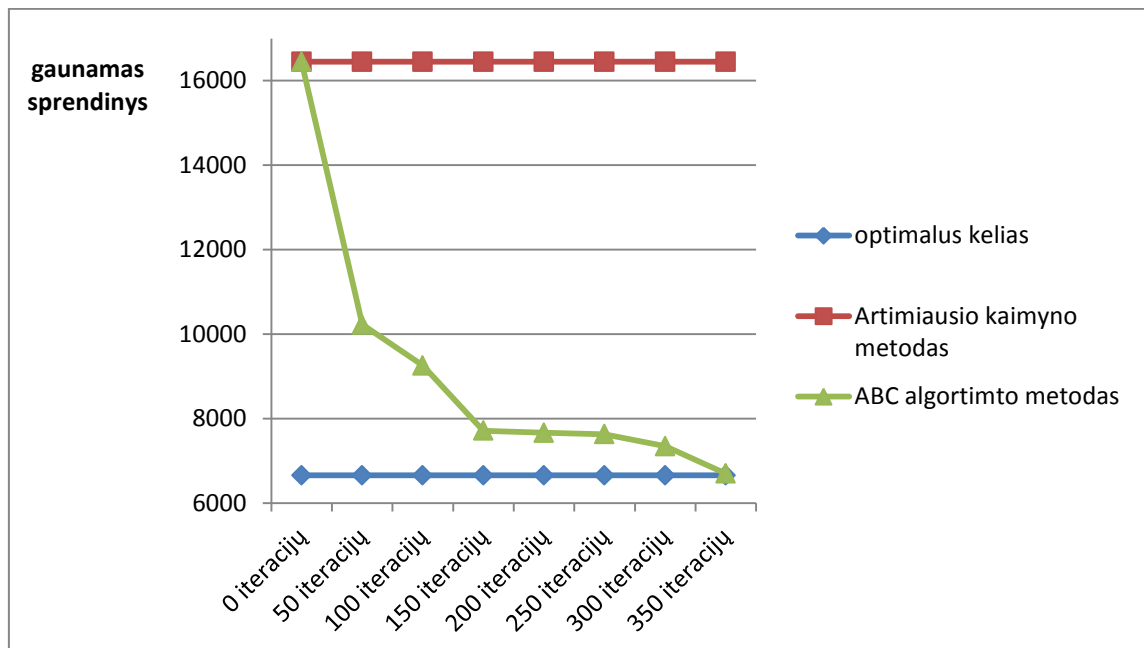


Diagrama 1 ABC algoritmo optimalaus kelio priklausomybė nuo iteracijų skaičiaus.

Atlikus skaičiavimus (Priedas nr. 2; lentelė nr. 5) su bičių kiekio parametru ir paėmus rezultatų vidurkį buvo pastebėta, jog numatytas parametras yra arti optimalios reikšmės (Žr. Diagrama 2). Tačiau, padidinus bičių skaičių iki 200 rezultatas atitolo nuo optimalaus sprendinio. Tai galėjo atsitikti dėl to, kad ABC algoritmas yra stochastinis algoritmas (kiekvieną kartą paleidus algoritmą

su pradinėmis reikšmėmis yra gaunamas skirtingas rezultatas) ir kiekvienas bandymas yra atliktas nuo artimiausio kaimyno metodo sprendinio, o nebandant pagerinti vieno bandymo rezultatą.

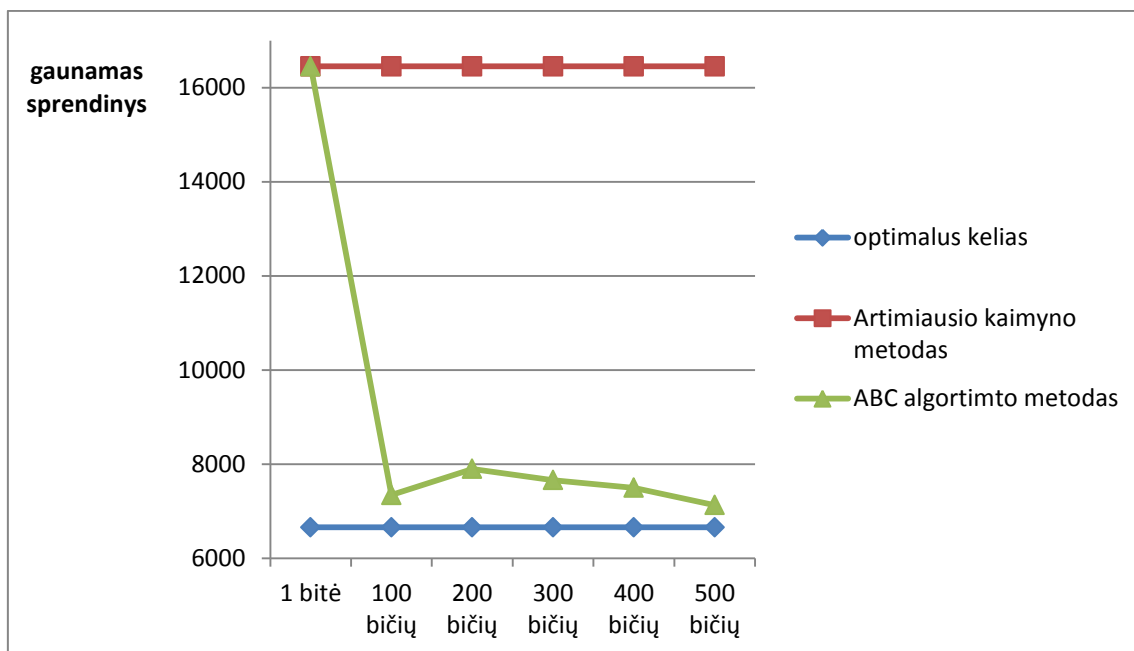


Diagrama 2 ABC algoritmo optimalaus kelio priklausomybė nuo bičių skaičiaus.

Užbaigus bandymus (Priedas nr. 2; lentelė nr. 6) su bičių darbininkų etapo parametru ir išvedus rezultatų vidurkį suprasta, kad didinant etapų skaičių nuo 55 gaunamas sprendinys mažai progresuoja prie optimalaus sprendinio. Gauti rezultatai matosi diagramoje (Žr. Diagrama 3).

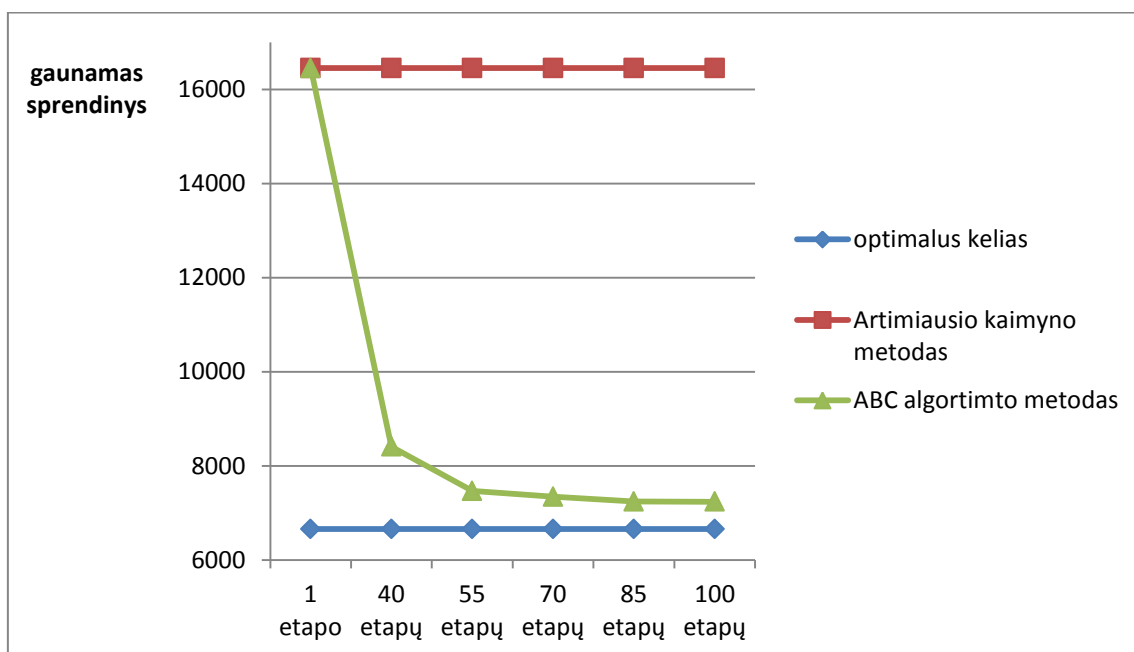


Diagrama 3 ABC algoritmo optimalaus kelio priklausomybė nuo bičių darbininkų etapo.

Įvykdžius ABC algoritmo priklausomybės bandymus (Priedas nr. 2; lentelė nr.7) nuo bičių procentų, kurios keičia darbo kryptį ir suskaičiavus rezultatų vidurkį pastebima, kad numatytas parametras yra arčiausiai optimalaus kelio. Jei šis parametras didinamas, tai gaunamas kelias tolsta nuo optimalaus kelio reikšmės, taip pat kaip ir mažinant šį parametrą.

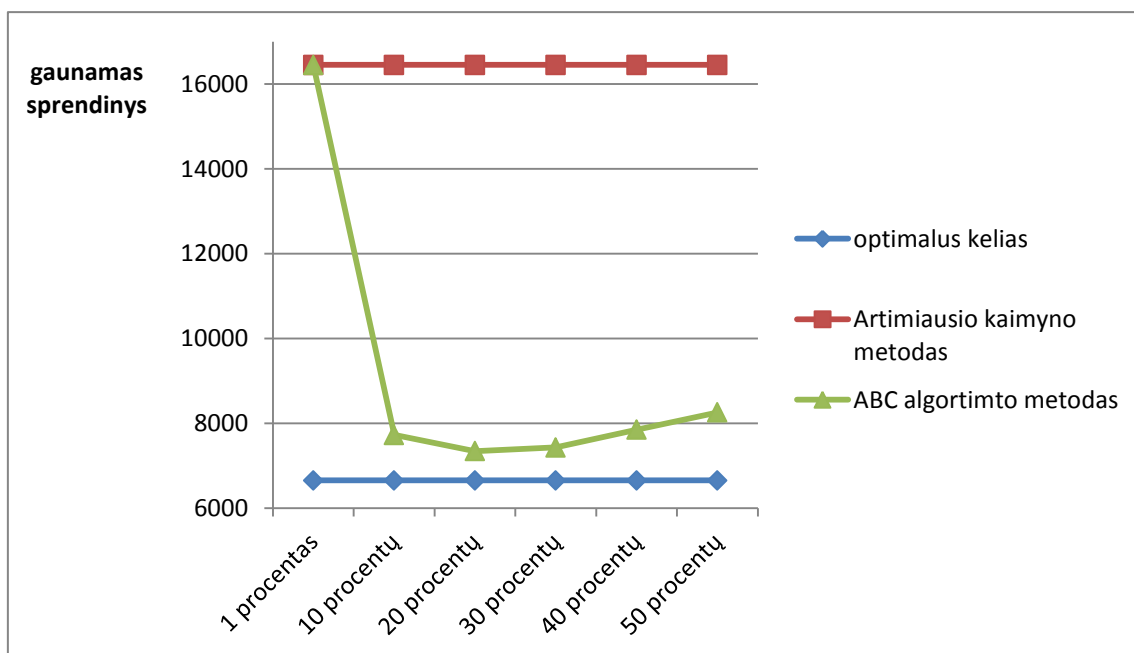


Diagrama 4 ABC algoritmo optimalaus kelio priklausomybė nuo bičių procentų, kurios keičia darbo kryptį.

3.5 Rekomendacijos

- Šiuo metu rezultatams gauti programa naudoja artimiausio kaimyno algoritmą. Patobulinus programą pradiniai duomenys galėtų būti sugeneruojami atsitiktiniu būdu. Tai leistų išanalizuoti ABC algoritmo rezultatų priklausomybę nuo pradinių duomenų;
- Taip pat reikėtų sukurti apsaugas nuo neteisingai įvestų parametrų. Tai užtikrintų teisingų ir korektiškų duomenų gavimą. Taipogi, tai apsaugotų programą nuo klaidų, kurios įvyksta dėl neteisingų duomenų įvedimo;
- Sukurti galimybę vartotojui tęsti gauto sprendinio optimizavimą, jei su pasirinktais parametrais gautas rezultatas netenkina vartotojo;
- Suderinti vartotojo sąsaja su mobiliaisiais įrenginiais, kad vartotojams būtų patogiau naudotis programa.

IŠVADOS

- Susipažinus su dalelių spiečių sistemų algoritmais paaiškėjo, jog šie metaeuristiniai algoritmai taikomi įvairiems optimizavimo uždaviniams spręsti;
- Bičių spiečiaus algoritmų analizė parodė, kad dažniausiai praktikoje jie taikomi įvairiems didelės dimensijos kombinatoriniams optimizavimo uždaviniams spręsti, veikia gana efektyviai, kai reikia taupyti skaičiuojamuosius resursus;
- Susipažinus su įvairiais maršrutų optimizavimo uždaviniais paaiškėjo, jog šie uždaviniai labai dažnai sutinkami praktikoje, su jais susiduria logistikos, turizmo ir įvairių paslaugų kompanijos;
- Kadangi ABC algoritmas nereikalauja sudėtingų kintamųjų klasių ar sudėtingu bibliotekų, programai kurti buvo naudojama JavaScript kalba. Didžiausias šios kalbos privalumas dirbant su masyvais yra realizuota perkėlimo funkcija, kuri palengvino programos realizaciją ir sumažino kodo eilučių skaičių;
- Atlikus testinius skaičiavimus su kontroliniais duomenimis paaiškėjo, jog programa veikia be klaidų, sugeba rasti sprendinius, artimus optimumui, todėl šią programą galima naudoti ir didesnės dimensijos uždaviniams spręsti;
- Išanalizavus ABC algoritmo gautus rezultatus pastebėta, kad prie optimalaus kelio priartėjama maždaug po 150 iteracijų, kai turime 39 mazgus;
- Atlikus skaičiavimų rezultatų analizę galima teigti, kad keliaujančio pirklio uždaviniams neužtenka klasikinio algoritmo, artimiausio kaimyno metodo, gaunamų rezultatų, nes ABC metodu gauti rezultatai yra maždaug 2 kartus geresni;
- Išanalizavus žiūrovių ir žvalgių procento parametą, paaiškėjo, jog artimiausias optimaliam sprendiniui rezultatas yra gaunamas, kai šis parametras yra intervale nuo 15 iki 30.

LITERATŪROS IR INFORMACINIŲ ŠALTINIŲ SĄRAŠAS

1. Felix T. S. Chan and Manoj Kumar Tiwari, (2007). *Swarm Intelligence Focus on Ant and Particle Swarm Optimization*. I-tech education and Publishing. Vienna, Austria.
2. Michael Hahsler, Kurt Hornik. *TSP – Infrastructure for the Traveling Salesperson Problem*. P. 15-37 ; 113-145. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://cran.r-project.org/web/packages/TSP/vignettes/TSP.pdf>.
3. Pei-Wei TSai, Jeng-Shyang Pan, Bin-Yih Liao, Shu-Chua Chu, (2009). *Enhanced artificial bee colony optimization*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://iiirc.hitsz.edu.cn/course/CI/Section5/ENHANCED%20ARTIFICIAL%20BEE%20COLONY%20OPTIMIZATION.pdf>.
4. Nadezda Stanarevic, Milan Tuba, Nebojsa Bacanin, (2011). *Modified artificial bee colony algorithm for constrained problems optimization*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://www.naun.org/multimedia/NAUN/m3as/20-418.pdf>.
5. Nishant Pathak, Sudhanshu Prakash Tiwari, (2012). *Traveling Salesman Problem Using Bee Colony With SPV*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://www.ijscce.org/attachments/File/v2i3/C0716052312.pdf>.
6. M. Faiza Abdulsalam, Azuraliza Abu Bakar, (2012). *A Cluster-Based Deviation Detection Task Using the Artificial Bee Colony (ABC) Algorithm*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://docsdrive.com/pdfs/medwelljournals/ijscmp/2012/71-78.pdf>.
7. Gražvydas Felinskas, (2007). *Euristinių metodų tyrimas ir taikymas ribotų išteklių tvarkaraščiams optimizuoti*. Daktaro disertacija. P. 58-72. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: http://www.mii.lt/files/mii_dis_07_felinskas.pdf.
8. Narimantas Listopadskis. *Kombinatorinio optimizavimo uždaviniai ir jų sprendimo algoritmai*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://www.fmf.lt/ft/studiju-programos/taikomoji-matematika/S-18508/straipsnis/Kombinatorinio-optimizavimo-uzdaviniai-ir-ju-sprendimo-algoritmai?name=S-18508&l=5&p=1>.
9. A. Misevičius, J. Blonskis, V. Bukšnaitis, (2007). *Kombinatorinis optimizavimas ir metaeuristiniai metodai: teoriniai aspektai*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: http://www.leidykla.eu/fileadmin/Informacijos_mokslai/42-43/213-219.pdf.
10. Dušan Teodorović. *Bee Colony Optimization (BCO)*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: <http://www.sf.bg.ac.rs/downloads/katedre/oi/1.BCO-Book-Chapter.pdf>.
11. A. Afshar, O. Bozorg Haddad, M.A. Marino, B.J. Adams (2006). *Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation*. [interaktyvus] [žiūrėta

2013-04-15]. Prieiga internete: http://robotics.csie.ncku.edu.tw/HCI_Project_2009/徐福瑜2.pdf

12. Hussein A. Abbass, Jason Teo (2001). *A True Annealing Approach to the Marriage in Honey-Bees Optimization Algorithm*. [interaktyvus] [žiūrėta 2013-04-15]. Prieiga internete: http://pdf.aminer.org/000/905/860/a_true_annealing_approach_to_the_marriage_in_honey_bees.pdf

ANOTACIJA

Autorius: Donatas Kavaliauskas

Tema: *Dirbtinės bičių kolonijos algoritmai ir jų taikymai maršrutų optimizavimo uždaviniams spręsti.*

Šiaulių universitetas 2013.

Šiame darbe yra trumpai apžvelgiami dalelių spiečių sistemų algoritmai, maršrutų optimizavimo uždaviniai ir jų formuluotės, bei praktinės interpretacijos. Plačiau apžvelgiami dirbtinių bičių kolonijų algoritmai ir jų pritaikymas keliaujančio pirklio uždaviniams spręsti. Taip pat šiame darbe galima rasti dirbtinių bičių kolonijų algoritmo pritaikymą keliaujančio pirklio uždaviniams spręsti, bei sukurtos programos skaičiavimo rezultatų analizę.

SUMMARY

Author: Donatas Kavaliauskas

Subject: *Artificial Bee Colony Algorithms and their Application to Route Optimisation Problems*

Siauliai University 2013

This paper consists of short description of swarm systems algorithms, route optimisation problems overview and longer description of artificial bee colony algorithms adaptation for traveling salesman problem. Moreover, you can find an artificial bee colony algorithm's application to traveling salesman problem and analysis of computational results.

Priedai

1. CD turinys:

- Kataloge „Bakalauro aprašymas“ yra bakalauro darbo elektroninis aprašo versijos: Donatas_Kavaliauskas.docx ir Donatas_Kavaliauskas.pdf
- Kataloge „Programa“ yra įdėtas index.html failas, kuriame realizuota ABC algoritmas, papildomi failai dėl programos išvaizdos. Be to yra įdėtas tekstinis failai pvz.txt, pvz1.txt ir kt., kuriuose yra testiniai duomenys programai.
- Programos vartotojo vadovas, kuris paaiškina , kaip naudotis ABC algoritmo realizacija, yra įdėtas kataloge „Programa“.

2. Skaičiavimai

Duomenys paimti:	http://www.tsp.gatech.edu/world/djtour.html						
Minimali reikšmė:	6656						
Sukurta programa:							
Artimiausio kaimyno metodas							
Reikšmė:	16450.369						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7096.088	6	7386.697	11	7078.146	16	6810.967
2	7664.804	7	7286.027	12	8164.579	17	8119.689
3	7534.637	8	7167.071	13	8120.164	18	6922.886
4	6810.967	9	7096.088	14	7167.071	19	7167.071
5	8166.073	10	6914.993	15	6952.18	20	6881.197
Vidurkis:	7.345.011						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						300	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7413.504	6	7784.613	11	6881.197	16	9080.685
2	8164.579	7	6881.197	12	7313.056	17	7096.088
3	6659.431	8	6659.431	13	7102.778	18	7102.778
4	7078.146	9	8397.929	14	9042.365	19	7167.071
5	6659.431	10	7879.458	15	6952.18	20	7242.826
Vidurkis:	7.452.977						
Parametrai:							

Iteracijų skaičius:						500	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7096.088	6	6881.95	11	6810.967	16	7658.114
2	7167.071	7	7129.884	12	9196.73	17	7096.088
3	6914.993	8	7096.088	13	8807.938	18	6778.313
4	7143.038	9	7129.884	14	7129.884	19	9037.577
5	7129.884	10	7096.088	15	6881.197	20	7408.008
Vidurkis:	7.306.172						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						100	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	8119.689	6	6881.19 7	11	7686.888	16	7096.088
2	7740.042	7	7129.88 4	12	8164.579	17	9042.365
3	7096.088	8	9042.36 5	13	7469.708	18	7096.088
4	9042.365	9	6881.95	14	7129.884	19	6659.431
5	9080.685	10	9125.89 1	15	7167.071	20	7096.088
Vidurkis:	7.782.442						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						50	

ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7368.243	6	7096.088	11	7096.088	16	9042.365
2	8871.975	7	6914.993	12	7368.243	17	7386.697
3	9132.001	8	9042.365	13	8120.164	18	6881.197
4	7527.947	9	7129.884	14	7143.038	19	7096.088
5	7129.884	10	8336.864	15	9138.388	20	8119.689
Vidurkis:	7.797.110						
Parametrai:							
Iteracijų skaičius:						250	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	6844.763	6	8164.579	11	6961.173	16	7273.471
2	6914.993	7	8161.151	12	6881.197	17	6844.763
3	9558.161	8	7367.003	13	7096.088	18	7808.259
4	8858.319	9	7879.58	14	7693.129	19	8119.689
5	7129.884	10	7096.088	15	7136.574	20	9056.635
Vidurkis:	7.629.785						
Parametrai:							
Iteracijų skaičius:						200	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	8487.864	6	7129.884	11	8712.742	16	7096.088
2	7949.688	7	8557.353	12	8591.595	17	9087.248
3	8336.864	8	8279.445	13	6959.458	18	8893.453
4	8648.06	9	7129.884	14	7204.356	19	7129.884
5	7096.088	10	7386.697	15	7129.884	20	9600.362
Vidurkis:	7.661.182						

Parametrai:							
Iteracijų skaičius:						150	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	8073.23	6	8477.368	11	9139.207	16	7275.12
2	8782.711	7	7167.071	12	6995.21	17	7881.161
3	7527.947	8	9399.382	13	8481.99	18	8076.374
4	8107.745	9	7371.223	14	6663.734	19	8774.755
5	7231.635	10	8618.271	15	6881.197	20	9063.278
Vidurkis:	7.713.328						
Parametrai:							
Iteracijų skaičius:						100	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	8700.954	6	10414.61	11	8928.816	16	9907.658
			7				
2	9934.803	7	9354.357	12	9425.033	17	8063.614
3	9621.941	8	9612.109	13	9820.944	18	8268.573
4	7895.658	9	9440.557	14	7604.94	19	9595.192
5	9640.595	10	8854.204	15	9081.922	20	10417.48
Vidurkis:	9.253.419						
Parametrai:							
Iteracijų skaičius:						50	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė

1	10504.29 8	6	10458.91 3	11	10871.69 9	16	10570.03 1
2	10251.76 8	7	9460.282	12	9748.422	17	11293.28
3	10818.44 9	8	9943.597	13	10464.53 7	18	8934.729
4	10487.45	9	10073.77 8	14	11065.20 5	19	10700.81 7
5	10360.94 5	10	10406.49	15	8994.523	20	10627.59 3
Vidurkis:	10.226.446						
Parametrai:							
Iteracijų skaičius:						350	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7096.088	6	8648.06	11	7096.088	16	7096.088
2	7096.088	7	7527.947	12	6659.431	17	7731.628
3	7569.252	8	6914.993	13	6881.197	18	7167.071
4	6844.763	9	7096.088	14	7129.884	19	7129.884
5	7129.884	10	8279.445	15	7167.071	20	6894.1
Vidurkis:	6.695.086						

Lentelė 4 ABC algoritmo optimalaus kelio priklausomybė nuo iteracijų skaičiaus skaičiavimai.

Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						200	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7096.088	6	7646.651	11	7096.088	16	9056.635

2	9166.086	7	7242.826	12	8279.445	17	6810.967
3	7849.531	8	7569.252	13	8487.471	18	8765.389
4	8682.63	9	7527.947	14	6659.431	19	7686.888
5	9076.861	10	7313.056	15	6952.18	20	8839.921
Vidurkis:	7.898.363						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						300	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7096.088	6	6881.197	11	7386.697	16	7096.088
2	9528.587	7	8692.314	12	7102.778	17	8164.579
3	7096.088	8	6659.431	13	7167.071	18	8119.689
4	7774.542	9	9528.587	14	8119.689	19	7527.947
5	7352.754	10	6810.967	15	6939.445	20	8164.579
Vidurkis:	7.660.456						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						400	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	6705.339	6	6810.967	11	8395.811	16	6810.967
2	7389.878	7	7334.012	12	8745.621	17	6914.993
3	6810.967	8	7096.088	13	7167.071	18	8839.921
4	7352.754	9	7990.948	14	6914.993	19	7601.588
5	7527.947	10	6881.95	15	8947.955	20	7096.088
Vidurkis:	7.497.574						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						500	

Bičių darbininkių etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7129.884	6	6844.763	11	7096.088	16	7078.146
2	6939.445	7	6914.993	12	6914.993	17	7167.193
3	6881.197	8	7129.884	13	7167.071	18	7313.056
4	7352.754	9	6952.18	14	7278.858	19	6810.967
5	7386.697	10	8119.689	15	7096.088	20	6881.197
Vidurkis:	7.131.735						

Lentelė 5 ABC algoritmo optimalaus kelio priklausomybė nuo bičių skaičiaus skaičiavimai.

Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						100	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7096.088	6	8692.314	11	7167.071	16	6902.153
2	7096.088	7	7375.318	12	7129.884	17	7096.088
3	8348.973	8	6952.18	13	6914.993	18	7129.884
4	6881.197	9	6914.993	14	7096.088	19	7096.088
5	7096.088	10	7527.947	15	6881.197	20	7129.884
Vidurkis:	7.129.884						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						85	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	6810.967	6	6952.18	11	7095.374	16	8120.685
2	7129.884	7	7286.027	12	7096.088	17	7055.621

3	8119.689	8	7143.038	13	8164.579	18	6810.967
4	7096.088	9	6914.993	14	7242.826	19	6659.906
5	7096.088	10	7686.888	15	6914.993	20	7167.071
Vidurkis:	7.242.725						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						55	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	9296.499	6	8119.689	11	9042.365	16	9132.001
2	9080.685	7	7897.838	12	9354.3	17	7801.967
3	8137.218	8	9054.259	13	8799.319	18	8244.02
4	8620.081	9	7697.473	14	7096.088	19	8648.06
5	7601.588	10	7096.088	15	8309.337	20	9042.365
Vidurkis:	7.466.469						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						40	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						20	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7299.622	6	8390.495	11	7626.287	16	6662.823
2	8874.501	7	9042.365	12	8594.085	17	8450.845
3	8856.898	8	8727.325	13	8700.637	18	8658.095
4	9211.523	9	8030.491	14	8765.625	19	8120.164
5	9114.658	10	6980.851	15	9042.365	20	9042.365
Vidurkis:	8.409.601						

Lentelė 6 ABC algoritmo optimalaus kelio priklausomybė nuo bičių darbininkų etapo, skaičiavimai.

Parametrai:	
Iteracijų skaičius:	300

Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						10	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	7004.532	6	7691.683	11	7129.884	16	7827.64
2	8804.234	7	8333.95	12	7516.737	17	9485.487
3	7144.61	8	8164.579	13	8066.902	18	7658.114
4	7167.193	9	7167.071	14	9042.365	19	7478.238
5	7368.243	10	7601.588	15	7339.725	20	6660.427
Vidurkis:	7.726.294						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						30	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	8799.319	6	8164.579	11	7096.088	16	9047.983
2	6881.197	7	7129.884	12	8119.689	17	7096.088
3	7242.826	8	7167.071	13	8225.161	18	9042.365
4	7129.884	9	6659.431	14	6659.431	19	7949.688
5	7164.02	10	8124.371	15	9092.579	20	7096.088
Vidurkis:	7.432.320						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						40	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	8164.579	6	9045.963	11	9080.685	16	7097.303
2	7167.071	7	7096.088	12	9092.579	17	9080.685
3	8870.213	8	6710.337	13	7143.038	18	7129.884

4	8433.289	9	9847.403	14	9296.621	19	9354.3
5	7774.542	10	6659.431	15	7096.088	20	9528.587
Vidurkis:	7.851.848						
Parametrai:							
Iteracijų skaičius:						300	
Bičių spiečiaus skaičius:						100	
Bičių darbininkų etapas:						70	
Bičių procentas, kurios taps žvalgiais arba nusižiūrinčios:						50	
ABC algoritmas							
Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė	Bandymas	Reikšmė
1	6881.95	6	10065.75	11	7129.884	16	6810.967
			7				
2	9080.685	7	7879.458	12	7693.129	17	9062.903
3	7167.071	8	8839.921	13	8138.515	18	8120.164
4	8943.802	9	8336.864	14	7601.588	19	8815.247
5	7686.888	10	8172.311	15	9042.365	20	8344.051
Vidurkis:	8.259.556						

Lentelė 7 ABC algoritmo optimalaus kelio priklausomybė nuo bičių procentų, kurios keičia darbo kryptį, skaičiavimai.