

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Bitininkystės procesų modeliavimas

Modelling of Beekeeping Processes

Magistro baigiamasis darbas

Atliko:	Vytautas Vėgėlė	(parašas)
Darbo vadovas:	asist. dr. Tomas Plankis	(parašas)
Recenzentas:	asist. dr. Vytautas Valaitis	(parašas)

Vilnius – 2020

TURINYS

ĮVADAS	4
1. LITERATŪROS APŽVALGA	6
1.1. BEEHAVE modelis	6
1.1.1. Modelio įvadas	6
1.1.2. Modelio dalys	7
1.1.2.1. Bitės ir bičių šeimos modelis.....	7
1.1.2.2. Parazitų modelis	7
1.1.2.3. Maisto paieškos modelis	8
1.1.2.4. Aplinkos modelis	8
1.1.2.5. Spietimosi modelis	9
1.1.2.6. Simuliacijos apžvalga.....	10
1.1.3. Modelių palyginimas su BEEHAVE	10
1.2. Kiti procesai	12
1.2.1. Sezoninis klimatas ir jo kaita	12
1.2.2. Bičių migracija tarp šeimų	12
1.2.3. Varozės plitimas tarp šeimų	13
1.2.4. Bičių spiečiaus naujo lizdo paieška	14
1.2.5. Bičių mirtingumas skirtingu metų laiku	15
2. MODELIŲ SUJUNGIMAS	17
2.1. Procesai	18
2.1.1. Žiemojimas	18
2.1.2. Šeimos įkūrimas.....	19
2.1.3. Kiaušinėlių dėjimas	19
2.1.4. Bitininkystės procesai	20
2.1.5. Varozė ir kitos ligos	20
2.1.6. Bičių mirtingumas pagal metų laiką.....	21
2.2. Jungtinio modelio tikrinimas	21
2.2.1. Lyginimas su Beehave modeliu	21
2.3. Jungtinio modelio aptarimas.....	24
3. PATOBULINTAS MODELIS	26
3.1. Bičių migracija tarp šeimų.....	26
3.2. Spietimosi procesas ir algoritmas	27
3.2.1. Dirbtinės bičių šeimos algoritmas	27
3.2.2. Dirbtinio spietimosi algoritmas.....	30
3.2.3. Dirbtinių bičių šeimų algoritmų lyginimas.....	31
3.3. Spietimosi proceso įgyvendinimas	37
REZULTATAI IR IŠVADOS	41
ŠALTINIAI	42
PRIEDAI	45
1 priedas. Konvergavimo grafikai su iqABC ir iqsABC	46
2 priedas. Kiti iqsABC ir sABC algoritmų tyrimo rezultatai	53
3 priedas. Konvergavimo grafikai su ABC, qABC, iqABC, iqsABC algoritmais	58
4 priedas. Landšafto tikslo funkcijos grafikai	65
5 priedas. Pseudokodai	66

Anotacija

Darbe siekiama sukurti patobulintą bičių populiacijos kompiuterinį modelį. Atlikus esančių modelių analizę apjungti BEEHAVE ir Bumble-BEEHAVE modeliai, prie jų pridėtas spietimosi procesas ir aprašytas bičių migracijos procesas. Sukurtas ir išanalizuotas patobulintas greitas besispyčiančios bičių šeimos algoritmas. Atliktas šio ir kitų dirbtinių bičių šeimų algoritmų palyginimas naudojant 25 skirtingas tikslo funkcijas ir Wilcoxon testą. Pateikta ir kita šio algoritmo versija, kuri buvo pritaikyta patobulinto modelio spietimosi procese.

Raktiniai žodžiai: dirbtinis bičių šeimos algoritmas, kompiuterinis modeliavimas, bitininkystė, optimizavimo užduotys.

Summary

The purpose of this paper was to create an improved bee population computer model. After researching current bee models, two current models (BEEHAVE and Bumble-BEEHAVE) were combined. This joint model was expanded by adding a swarming process and an additional bee drifting process implementation was described. An improved quick swarming artificial bee colony algorithm was introduced and compared to currently existing artificial bee colony algorithms. 25 different goal functions for fitness evaluation and Wilcoxon signed-rank test was used to determine any statistical significance between results. Another variation of said algorithm was adopted and used in the new improved bee model to implement a swarming process.

Keywords: artificial bee colony algorithm, computer modelling, beekeeping, optimization problems.

Įvadas

Bičių populiacijos kaita Europos šalyse [JLC⁺17] yra aktuali tema. Naujausi Europos komisijos tyrimai nagrinėja spartų bičių populiacijos mažėjimą bei bando paaiškinti įvairių aplinkos poveikių įtaką bičių sveikatai [EFS16]. Didelis dėmesys skiriamas pesticidų poveikiui bitėms [EFS18] ir varozei [EFS16]. Dalis šių tyrimų ir projektų remia bičių stebėjimo būdus, o kompiuterinių modelių naudojimas yra pigesnė alternatyva nei visų bitynų stebėjimas. Bitininkystės procesams simuliuoti yra sukurti įvairių kompiuterinių modelių skirtingais laikotarpiais [BOT⁺13]. Vienas senesnių modelių yra BEEPOP [DRL⁺89], modeliuojantis vienos bičių šeimos (viena motinėlė su darbininkėmis ir tranais) avilyje (lizde). Šiame modelyje bičių šeima gyvena be išorinių veiksnių, galinčių pakenkti bičių šeimos populiacijai: daroma prielaida, jog aplinkui yra maisto, jų nepuola kiti gyvūnai. Šis modelis buvo papildytas varozės procesais [DC05] tik 2004 m. Vienas sudėtingiausių bičių šeimos tipo modelių yra HoPoMo [SC07], naudojantis 65-ias diferencialines lygtis. Šio modelio trūkumas šeimos atžvilgiu yra fiksuotas bičių amžius, modelis paskiria bitėms darbus neatsižvelgiant į brandą. Šį trūkumą sprendžia Khoury modelis [KBM13], keisdamas bičių atliekamus darbus pagal jų amžių. Naujesni modeliai įprastai remiasi senesniais modeliais, sujungdami jų tiriamus procesus, pavyzdžiui, Torres'o [TRR15] sukurtas modelis simuliuoja vienos bičių šeimos populiaciją pasiremdamas HoPoMo ir Khoury modeliais. Nors minėti modeliai daug dėmesio skiria vidiniams procesams bičių avilyje, yra ir modelių, teikiančių prioritetą kitiems procesams. Atliktoje modelių apžvalgoje pastebėta [BOT⁺13], jog modelius galima grupuoti pagal procesų grupes, kurias plačiausiai padengia modeliai: bičių šeimos vidiniai procesai (avilyje), žiedadulkių ir maisto paieškos procesai, varozės ir kitų ligų sąveikos su šeimomis procesai. Tačiau tik vieną sritį apimantis modelis negali visapusiškai modeliuoti realios bičių šeimos sistemos ir todėl atsiranda nukrypimai tarp modelių ir realių bičių stebėjimo. Bandant kartu apjungti šias procesų grupes buvo sukurtas „BEEHAVE“ modelis [BGT⁺14], galintis modeliuoti vienos bičių šeimos populiaciją ir atsižvelgiantis tiek į vidinius avilio procesus, tiek ir į išorinius. Modelis buvo praplėstas naujais plėtiniais ir modeliais, su jais atlikti įvairūs tyrimai bičių populiacijos atžvilgiu, įskaitant skirtingas bičių rūšis (modelis „Bumble-BEEHAVE“) [BTP⁺17], pesticidų poveikį [RBT⁺17] ir t.t. Prognozuojami klimato pokyčiai [AAA⁺19], chemikalų naudojimas, plėtojama žemdirbystė turėtų skatinti atlikti bičių populiacijos prognozes ir nustatyti galimą šių procesų žalą. Šias prognozes būtų prasminga atlikti tiek pavienių bitynų, tiek valstybiniu atžvilgiu. Pavyzdžiui, purkštų laukų informacija Lietuvoje kaupiama APIS sistemoje, o bitynai žymimi geografinės informacijos sistemoje bityno paso išdavimo metu jau nuo 2016 m. Taigi, yra empirinių duomenų, kurie leistų modeliuoti dabartinę bičių populiaciją ir prognozuoti jos kitimus. Yra pagrindo manyti, jog verta sukurti modelį, galinti modeliuoti bitininkystės procesus didesnės teritorijos ar net šalies lygiu. Dabartiniai modeliai įprastai skirti tik vienos bičių šeimos modeliavimui, todėl reikia nustatyti ir į kurią modelį įtraukti daugiau procesų:

- Bakterinių, grybelinių, virusinių ligų plitimo procesai (nozematozė, puvinys, t.t)
- Kelių bičių šeimų įtaka viena kitai (maisto rinkimo, ligų plitimo atžvilgiu)
- Bičių spiečių susidarymas

- Pavienių bičių migracija

Įvedus naujus procesus ir įgyvendinus naują modelį, reikės patikrinti jo gaunamas populiacijos prognozes su empirinėmis žiniomis apie bičių populiacijos kaitą bei neatitikimus lyginti su kitų modelių gaunamais rezultatais.

Taigi, šio darbo tikslas – sukurti patobulintą modelį, leidžiantį simuliuoti kelių bičių šeimų populiacijos augimą nurodytoje teritorijoje.

Darbo metu bus sprendžiami šie uždaviniai:

1. Procesų analizės atlikimas: plačiau išnagrinėti bitininkystės procesus ir jų sąryšius. Palyginti esamus bitininkystės procesų modeliavimo įrankius bei jų naudojamus procesus. Nuspręsti, kokie bitininkystės procesai, į kuriuos neatsižvelgia dabartiniai modeliai, padėtų tiksliau modeliuoti bičių populiaciją.
2. Modelio kūrimas: sukurti naują modelį, kuris būtų patobulintas naujais pasirinktais procesais.
3. Modelio tikrinimas: palyginti modelį su kitais sukurtais modeliais ir palyginti modelio gaunamus rezultatus su prieinamomis empirinėmis žiniomis

1. Literatūros apžvalga

Sudarant tobulesnį bičių populiacijos modelį verta išvardinti bitininkystės procesus, kurie yra nagrinėjami dabartiniuose modeliuose, bei procesus, kuriuos verta įtraukti bandant simuliuoti bičių populiacijos augimą šalies mastu. Šioje apžvalgoje aptarsime BEEHAVE modelį, jo modeliuojamus procesus, kaip tuos pačius procesus modeliuoja kitų modelių autoriai, bei procesus, kurie neįeina į BEEHAVE modelį ir pagal kuriuos bus grindžiamas patobulinto BEEHAVE modelio veikimas.

1.1. BEEHAVE modelis

Šiame poskyryje aptarsime BEEHAVE modelio ypatumus ir bitininkystės procesus, kuriuos padengia modelis ir kaip juos padengia kiti modeliai ir autoriai.

1.1.1. Modelio įvadas

Šio modelio bičių populiacijos skaičiavimai buvo lyginami su empirinėmis žiniomis apie realių bičių šeimų populiacijos kaitą. Šio modelio trūkumai buvo nustatyti ir paviešinti Europos maisto saugos tarnybos [EFS15]. Esminiai trūkumai buvo: pesticidų procesų trūkumas, primityvus landšafto (topografijos, augalijos) modelis, ribota klimato konfigūracija (modelis buvo pritaikytas centrinės Europos klimato zonai) ir apribotas ligų kiekis, tiriant tik varozę. Dalis trūkumų buvo pašalinti plečiant modelio funkcionalumą moduliais, pavyzdžiui, leidžiant naudoti išsamesnius landšafto modelius pasitelkiant geografinių informacinių sistemų informaciją su „Bee-Steward“ moduliu. Su modelio plėtiniais buvo atlikti įvairūs tyrimai bičių populiacijos atžvilgiu, įskaitant skirtingas bičių rūšis (modelis „Bumble-BEEHAVE“) [BTP⁺17], pesticidų poveikį [RBT⁺17] ir t.t. Šis modelis remiasi stochastiniais skaičiavimais šioms sritims:

- Lizdo suradimas.
- Tikimybė bitei mirti ieškant lizdo, maisto.
- Tikimybė motinėlei išgyventi žiemos miegą.
- Nustatant genetinę informaciją, kurią motinėlė laiko surinktoje spermoje.
- Bičių pabudimo dienos pradžioje laikas.
- Naujos motinėlės kiaušinėlio padėjimo laikas.
- Maisto šaltinio paieška, kaip nusprendžiama ar dabartinis maisto šaltinis tenkina bitę ir kito šaltinio nustatymas (arčiau ar toliau).

BEEHAVE modelis remiasi trimis pagrindiniais modeliais, kuriuos modelio autoriai atrinko savo modelių apžvalgoje [BOT⁺13]. Šie modeliai bus BEEPOP [DRL⁺89], HoPoMo [SC07] ir Khoury [KBM13] modeliai.

1.1.2. Modelio dalys

BEEHAVE modelį sudaro daug smulkesnių modelių, atvaizduojančių įvairius bičių ir jų šeimų procesus. Čia bus išvardijami pagrindiniai modeliai ir jų paaiškinimai ir galimos alternatyvos.

1.1.2.1. Bitės ir bičių šeimos modelis

Kiekvieną bitę modelyje atitinka individas. Bitės skirstomos į grupes (darbininkės, motinė-lės, tranai) bei veiklas (pvz. rinkėjos), bei nurodomas jų dydis, kuris nulemia jų liežuvio ilgį ir didžiausią maisto kiekį, kurį gali nešti. Bitėms taip pat priskiriama rūšys ir šeima. Rūšis nulemia hibernacijos laiką, kiaušinėlių dėjimo dažnį, brandos stadijų trukmę, liežuvėlio dydį. Bičių šeimos modelį sudaro kiaušinėliai, lervos, lėliukės ir subrendusios bitės. Šios stadijos taikomos darbinin-kėms, tranams ir motinėlėms. Kiekvienai bitei taikomas identifikuojantis numeris, amžius. Taip pat sekama varozės ir virusinių ligų įtaka bitei. Sekamas avilyje laikomų žiedadulkių ir medaus kie-kis, jiems ir pačioms bitėms išskirta vieta, vieta perams. Nusakomas motinėlės kiaušinėlių dėjimo dažnis sezono metu ir šį dažnį koreguoja motinėlės amžius bei šeimos dydis.

Kiaušinėlių dėjimas BEEHAVE modelyje remiasi HoPoMo dėjimo modeliu pagal normalųjį skirstinį, atvaizduojanti kintantį dėjimo dažnį skirtingų sezonų metu (2). Duotai dienai d šis dydis f randamas pagal formulę:

$$f = 1600(1 - \alpha) \quad (1)$$

Kur dydis α randamas pagal:

$$\alpha = \max\left(1 - \frac{1}{1 + x_1 e^{\frac{-2d}{x_2}}}, \frac{1}{1 + x_3 e^{\frac{-2(d-x_4)}{x_5}}}\right) \quad (2)$$

Čia konstantos $x_1 = 385$, $x_2 = 25$, $x_3 = 36$, $x_4 = 155$, $x_5 = 30$ paimtos iš HoPoMo modelio, kuriame eksperimento būdu buvo parinkti parametrai, jog skirstinio kreivė atitiktų empirines žinias.

Perų vystymasis BEEHAVE modelyje remiasi empiriniais duomenimis. Kiaušinėlių vysty-mosi į lėliukes laikotarpis trunka 9 dienas, įskaitant pirmas 3 dienas, po kurių iš kiaušinėlio išsiritą lerva. Šie skaičiai gauti iš empirinių eksperimentų, atliktų 1968 ir 1977 m. [FO77; FS68]. Šie duo-menys naudojami ir kituose modeliuose, įskaitant ir vėliau aptariamą sezoninį matematinį modelį [YYY19]. Šiuo atžvilgiu BEEHAVE modelis sutampa su HoPoMo modeliu.

1.1.2.2. Parazitų modelis

Varozės erkių populiacijos modelis pagrįstas Martin [Mar01] modeliu. Skaičiuojamos prie subrendusių bičių prisikabinusios erkės, jas nusako jų dydis ir proporcija, nusakanti kiek iš tų erkių nešioja virusines ligas. Erkės įsikuria korio akelėse prieš jas uždarant. Bičių darbininkių akelėse gali gyventi iki aštuonių erkių, o tranų akelėse iki šešiolikos erkių. Erkės atsitiktinai išskirstomos po tinkančias akeles.

1.1.2.3. Maisto paieškos modelis

Modelis numato skirtingus nektaro paieškos metodus bitėms lankuolėms ir rinkėjoms. Rinkėjas reikia suprasti kaip bites, kurios tiesiog skrenda į žinomą maisto šaltinį ir renka maistą, bet tai darydamos nebando ieškoti kitų maisto šaltinių. Maisto paieškos metu naudojama tikimybė, nusakanti mirčių dažnį per sekundę paieškos metu. Čia pateikiami modelio paieškos režimai:

- Pradinis: nėra iš anksto žinoma, kur teks skristi. Palikusi avilį, bitė pradeda ieškoti nektaro aplink avilį. Tai yra įprastas paieškos režimas naujai šeimai arba bitei, kuriai nebuvo perduota informacija apie rastus nektaro šaltinius.
- Žinomo šaltinio individualus: bitė atsitiktinai pasirenka pačios rastą nektaro šaltinį, į kurį skris. Dažniau renkamas skristi į susigrupavusius, tankesnius šaltinius (kiekviena gėlė su nektaru gėlių lauke yra galima kandidatė). Šis režimas labiau tinka kamanėms ir bičių rūšims, kurios tarpusavyje neperduoda nektaro šaltinių informacijos.
- Žinomo šaltinio grupės: kaip ir individualaus žinomo šaltinio atžvilgiu, tik renkamas bet kurį tarp bičių žinomą nektaro šaltinį.
- Toliausio žinomo šaltinio individualus: pasirenkamas toliausiai nuo avilio nutolęs šaltinis. Toks režimas naudojamas jeigu arti esantys šaltiniai neturi pakankamai nektaro.
- Aplankytos vietos: Atsitiktinai parenkama bet kokia, bet kurios bitės aplankyta vieta. Naudojama, kai šaltiniai yra laikini arba sunku perduoti tikslas šaltinių vietas.

Modelio dokumentacijoje aptariami paieškos modelio trūkumai. Vienas šių trūkumų yra bitės gebėjimas aptikti maisto šaltinį, kadangi bitės geba pastebėti gėlių lauką per atstumą, nors modelyje jos turi tiesiog įlėkti į maisto šaltinio lauką. Ši ieškojimo dalis nebuvo įgyvendinta, kadangi, pagal žinomus modelio autoriams šaltinius, bitė pastebėtų 500 m. skersmens gėlių lauką tik būdama apie 21 metrą nuo to lauko, kadangi bitės regos laukas (5 laipsniai) ir skrydžio aukštis (apie 1,5 m) nėra dideli. Taip pat modelis neatsižvelgia į kvapų sklaidimą dėl empirinių žinių trūkumo ir dalelių skaičiavimų sudėtingumo.

1.1.2.4. Aplinkos modelis

Modelis naudoja landšafto aplink bičių šeimą informaciją. Šioje informacijoje yra pateikiami maisto šaltiniai (kurių nektaro ir žiedadulkių kiekiai kinta sezoniškai ir priklauso nuo šaltinio tipo, pvz. augalo rūšies). Modelis nesiremia konkrečia klimato informacija, bet yra nurodytas leistinas maisto paieškos ir rinkimo laikas dienos atžvilgiu, o klimatas netiesiogiai aprašomas landšafto informacijoje, nurodant kiek kiekvieną dieną sekundžių gali bitės paskirti maisto paieškai. Šias sekundes galima nurodyti empirinių duomenų rinkiniu arba pasirenkant, jog leidžiamas paieškai laikas būtų paskirstytas pagal konstantą arba normalųjį skirstinį. Normalusis skirstinys remiasi HoPoMo modeliu ir $t_{paieškos}$ laikas sekundėmis pasirinktai dienai d apskaičiuojamas pagal (3):

$$t_{paieškos} = 12 \times 3600 \times (1 - \alpha). \quad (3)$$

Kur α yra HoPoMo sezoną nusakantis dydis, apibrėžtas (2).

1.1.2.5. Spietimosi modelis

Spiečių susidarymas BEEHAVE modelio vykdomoje simuliacijoje yra išjungtas, tačiau šis procesas yra įgyvendintas ir jį galima įtraukti. BEEHAVE modelis seka tik vieną iš likusių spiečių, pasirenkant arba buvusią šeimą, arba sudarytą naują šeimą spiečiuje. Šią galimybę verta išnaudoti kuriant patobulintą modelį.

BEEHAVE modelis remiasi spiečiaus susidarymo modeliu [FS06]. Šiame straipsnyje siūlomi kriterijai ir jų formulės, pagal kurias nusprendžiama ar susidarys bičių spiečius:

1. Kai darbininkės gali prižiūrėti ir išmaitinti daugiau kiaušinėlių nei motinėlė gali jų padėti:

$$\frac{M_{C_i-21}}{N} \geq qp. \quad (4)$$

2. Kai šeimos dydis viršija ribą:

$$C_i \geq T. \quad (5)$$

3. Jeigu per paskutines 7 dienas užaugusios darbininkės sudaro daugiau nei trečdalį visų šeimos aktyvių darbininkių skaičių:

$$\frac{\sum_{m=i-30}^{i-22} E_m}{C_i} \geq \frac{1}{3}. \quad (6)$$

4. Kai šeimoje susikaupia per daug kiaušinėlių ir lervučių:

$$\sum_{m=i-21}^i E_m \geq 17000. \quad (7)$$

Šiose formulėse C_i yra pradinės bičių šeimos dydis, M_{C_i} yra vidutinis energijos kiekis, kurį išskyrė darbininkės i -tają dieną, q santykis, jog nesubrendusi bitė išgyvens iki brandos, p nurodo kiek kiaušinėlių padeda motinėlė, T yra bičių riba, nuo kurios vyksta spietimasis, lygi 16000 bičių, E_i yra padėtų kiaušinėlių kiekis i -tają dieną, N – energijos kiekis, reikalingas kiaušinėlio padėjimui ir bitės išsiritimui iš kiaušinėlio.

BEEHAVE modelis tikrina tik ketvirtą sąlygą spiečiaus susidarymui bei turi savo papildomą kintamąjį, kurį naudoja kaip laikmatį spiečiaus susidarymui. Patobulintame modelyje vertėtų įtraukti ir likusias sąlygas, kadangi visus reikalingus dydžius BEEHAVE modelis suskaičiuoja vykdamas simuliaciją.

Darant tyrimus buvo pastebėta, jog antroji sąlyga ir (5) formulė neatitinka realios situacijos bitynuose, kadangi pateikta konstanta apibūdina didžiausios laukinės bičių šeimos dydį Šiaurės Amerikoje. Laukinių bičių šeimos dydį riboja vieta, kurioje šeima įrengė savo lizdą, dažnai tai būna medžio drevė su ribotu tūriu, tuo tarpu bitynų aviliuose paliekama daug korių ir stipri bičių

šeima turi ir apie 60 tūkstančių bičių šeimoje, tačiau tokiu atveju šeima taip pat bandys sudaryti spiečių, jeigu to nebandys sustabdyti bitininkai.

1.1.2.6. Simuliacijos apžvalga

Prieš simuliacijos vykdymą nurodoma aplinkos informacija, empirinių duomenų rinkiniai ar metodai nusakantys leistinas dienos valandas paieškai, kiaušinėlių dėjimą, bičių mirtingumą maisto paieškos metu ir panašiai. Yra apie 18 loginės reikšmės kintamųjų, 26 skaitinių kintamųjų ir 2 pasirinkimai metodams originaliajame BEEHAVE modelyje, tačiau su išplėtimais tenka nurodyti ir daugiau kintamųjų ar net dokumentų, pavyzdžiui, nurodant aplinkos modelyje BEEScout moduliui. Simuliacija pradedama nuo metų pirmos dienos su pradiniu šeimos dydžiu. Visos pradinės bitės yra lankuolės, su 130 dienų gyvenimo trukme, tačiau kartu atlieka ir darbininkių pareigas buveinės viduje. Motinėlė deda kiaušinėlius pagal anksčiau aptartą dažnį, lankuolės suranda ir atneša maistą pagal paieškos modelį. Augimo stadija pakinta kai praeina pakankamas minimalus laikas ir sukaupiama pakankamai daug energijos inkubacijos metu (darbininkės palaiko šilumą). Jeigu dar nesuaugusi bitė negauna pakankamai maisto arba negauna pakankamai energijos inkubacijos metu, ji miršta. Darbininkės tampa lankuolėmis pasiekusios tam tikrą amžių, kuris priklauso nuo avilyje esamo maisto ir žiedadulkių kiekio bei darbininkių ir lankuolių santykio. Simuliacija vykdoma dienomis (simuliacijos laiko žingsniai yra nurodomi dienomis, visa šeimos informacija atnaujinama su metodu, pavadintu kasdieniniu atnaujinimu), tačiau tikimybėmis grįstos mirtys paieškos metu ir kiti panašūs procesai remiasi veiksmo vykdymu sekundėmis (dienos metu buvo praleista tiek sekundžių vykdant kurį nors veiksmą ir jam pritaikomos tikimybės pagal praleistas sekundes). Priklausomai nuo parametrų ir simuliacijos našumo reikalavimų, dalis bičių, atliekančių tą pačią funkciją yra sugrupuojamos ir modeliuojamos kaip vienas agentas. Bitės-agentai pasirenka užduotis pagal sąlygas bičių šeimoje, maisto paieška atsižvelgia į energijos atžvilgiu efektyviausius maršrutus, kurie atneša daugiausiai maistingų medžiagų reikalaujant mažiausio kiekio pastangų. Šie pasirinkimai remiasi agentų atmintimi: bitė atsimena aplankytas vietas ir reikalingą laiką, norint pasiekti jas nuo avilio ir grįžti. Jos taip pat turi suvokimą kiek ir kokių maisto atsargų laiko šeima, bei sugeba sekti dienos laiką. Be minėtų procesų dalyvauja ir pasirenkamasis bitininkystės procesas, kai bitininkai renka medų, maitina cukraus sirupu, gydo šeimą nuo varozės, pakeičia šeimos motinėlą.

1.1.3. Modelių palyginimas su BEEHAVE

Šio poskyrio apibendrinimui pateikta 1 lentelė, kuri nurodo kaip įvairius bitininkystės procesus modeliuoja skirtingi modeliai. BEEPOP ir jo varianto asmeniniams kompiuteriams PC BEEPOP [BDO⁺91] straipsniai nėra viešai prieinami, už juos reikia mokėti arba tartis su Jungtinių Amerikos Valstijų aplinkosaugos agentūra. Literatūros apžvalgos metu buvo įsigyta prieiga tik prie PC BEEPOP straipsnio. Torres [TRR15] modelis pasižymi tuo, jog prie HoPoMo modelio pridėdami hormonų sukeliami procesai, kurių įtaką nulemia fiksuojamas hormonų kiekis šeimoje. Tai pakeičia bičių šeimos sprendimus turėti kitokį darbininkių ir lankuolių santykį.

1 lentelė. Modelių ir jų procesų palyginimas

Procesas	BEEPOP	HoPoMo	Khoury	BEEHAVE	Torres
Kiaušinėlių dėjimas	Procedūra pagal motinėės amžių, dienos ilgį, darbininkų populiaciją	Normalusis skirstinys su atsitiktine variacija	Be jokios variacijos	HoPoMo be atsitiktinės variacijos, naudoja BEEPOP modelį įtraukiant motinėės amžių	Keičiama pagal sezoną
Perų vystymasis	21 d. nuo kiaušinėlio iki darbininkės, 42 d. iki lankuolės.	Iš empirinių duomenų [FO77; FS68]	3 savaitės iki darbininkės	Iš empirinių duomenų [FO77; FS68]	Išsamus, pagrįstas hormonų skaičiavimu
Maisto paieškos modelis	Maistas visada prieinamas, kol $>12^{\circ}\text{C}$ vidutinė dienos temperatūra, $<34\text{ km/h}$ vėjas, $<0.5\text{cm/d}$ krituliai	Konstanta nurodo kiek atnešama per dieną	Skaičiuoja tik lankuolių/rinkėjų pasiskirstymą	Penkių režimų paieška priklausomai nuo žinomų šaltinių ir paieškos grupės (žr. 1.1.2.3.)	Skaičiuoja tik lankuolių/rinkėjų pasiskirstymą ir sezono įtaką
Maisto sunaudojimo modelis	?	Savas pagal empirines žinias.	Diferencialinė lygtis pagal sumodeliuotą vidutinį sunaudojimą	HoPoMo, neatsižvelgiant į lervos amžių	Pagal HoPoMo
Spiečiaus modelis	Nėra	Spiečiaus sudarymas pagal laiko konstantą	Nėra	Simuliuojamas tik arba pradinė arba nauja šeima (žr. 1.1.2.5)	Nėra
Varozės modelis	Su VARROA-POP išplėtimu [DC05]	Nėra	Netiesiogiai	Pagal Martin modelį [Mar01]	Netiesiogiai
Aplinkos modelis	Atsižvelgiama į tinkamą temperatūrą, vėjo greitį ir kritulius	Temperatūra ir krituliai nulemia paieškos efektyvumą	Nėra	Yra landšafto modelis nuo BEEHAVE, išplėstas su BEESCOUT	Nėra, galima keisti tik paieškos efektyvumą

1.2. Kiti procesai

Šiame poskyryje aptarsime kitus procesus, kurių nepadengia BEEHAVE modelis ir kuriuos prasminga įtraukti kuriant patobulintą modelį.

1.2.1. Sezoninis klimatas ir jo kaita

Straipsnyje [SCT⁺17] buvo surinktas didelis kiekis empirinių duomenų apie Austrijos bičių populiaciją (dalyvavo 4983 bitininkai ir informacija buvo renkama tarp 2009 ir 2014 metų, iš viso sukaupiant informaciją apie 106675 bičių šeimų peržiemojimus) ir buvo kreipiamas dėmesys į bičių mirtingumą žiemos laikotarpiu esant skirtingai temperatūrai ir drėgmei. Straipsnis grafiškai pateikia mirtingumą priklausomai nuo temperatūros ir kritulių. Matomos palankiausios sąlygos pagal fiksuotus duomenis peržiemojimui ties 0–4 °C metine temperatūra ir 2,5 – 3 metrais metinių kritulių, kai mirtingumas siekia 12 procentų (tiek miršta bičių žiemos laikotarpiu bičių šeimoje), o blogiausi su 10–12 °C metine temperatūra ir mažiau nei 5 cm metinių kritulių. Taigi, daroma išvada, jog bitėms palankiausias peržiemojimas yra esant vėsesniam ir drėgnesniam klimatui. Nors šios išvados statistiškai yra patikimos, straipsnio autoriai taip pat išvelgia galimus klimato poveikius pašaliniais veiksniais, nuo kurių tiesiogiai priklauso bičių mirtingumas, kadangi tikėtina, jog pati temperatūra ir drėgmė nedaro tiesioginės įtakos bičių šeimos populiacijai. Šie veiksniai yra:

- Pesticidų naudojimas. Pesticidai Austrijoje dažniau naudojami šiltesnėse ir sausesnėse vietovėse. Autoriai negali patvirtinti šios teorijos, kadangi informacija apie pesticidų vartojimą nėra viešai prieinama.
- Augalijos kaita. Besikeičiančios klimato sąlygos keičia tiek žemdirbystei skiriamus plotus, tiek augaliją esamuose plotuose.
- Parazitai ir kitos ligos. Šiltesnis oras suteikia geresnes sąlygas parazitų dauginimuisi ir užkrečiamumui. Papildomai vaistai nuo parazitų ir ligų ilgiau išsilaiko esant vėsesniam ir drėgnesniam orui.

Kita dalis tyrimo trūkumų yra landšafto nenaudojimas, nebuvo įtraukta vėjo ir šešėlio įtaką bičių šeimos populiacijai. Šio straipsnio nauda empirinių duomenų atžvilgiu nėra labai didelė bandant prognozuoti bičių peržiemojimo galimybes Lietuvoje, tačiau teikiamos išvados duoda svarbią tyrimo sritį atsižvelgiant į klimatą ir jo įtaką bitėms bei jų ligoms. Tobulinamame BEEHAVE modelyje verta pakeisti peržiemojimo mirties tikimybes bitėms pagal temperatūrą. Daugiau informacijos apie procesus, darančius įtaką bičių peržiemojimui, nurodo straipsnis [DFG15], kuriame pateikiamos nuorodos ir į kitus tyrimus ir empirinių žinių kaupimą apie temperatūros įtaką bitėms Vokietijoje ir Jungtinėse Amerikos Valstijose.

1.2.2. Bičių migracija tarp šeimų

Šiame straipsnyje [PC98] aptariamas bičių migracijos reiškinys kai viena bitė palieką savo šeimą ir prisijungia prie kitos, paprastai dėl klaidų grįžtant po maisto ieškojimo ir įskrendant į

ne tą avilį. Tai dažniausiai įvyksta tarp bičių avilių bitynuose, kur atstumai tarp bičių šeimų yra maži. Papildomai nagrinėtas straipsnis [BPC⁺17] nurodo tokių bičių migravimo kiekius ir atstumus (žr. pav. 2), pagal kuriuos atitinkamai būtų galima nustatyti bičių migracijos kiekį kuriamame sprendime.

Bumble-BEEHAVE modelis leidžia įkelti landšaftą atvaizduojantį žemėlapią bei pateikti jo mastelį. Visų bičių šeimų lizdai turi koordinates xcor ir ycor, kurios leidžia įvertinti ne tik atstumus iki maisto šaltinių, bet ir atstumus tarp skirtingų šeimų. Kitame straipsnyje nurodyta [BPC⁺17], jog eksperimento metu atstumai tarp avilių buvo 1 metras. Bumblee-Beehave pateiktame pavyzdyje modeliuojamas 5 kvadratinų kilometrų plotas, pats modelis jį padalina į 300x210 langelių. Aki-vaizdu, jog atstumas tarp dviejų atsitiktinai įkurtų šeimų bus per didelis, jog galėtume atsižvelgti į straipsnyje nurodyto eksperimento rezultatus, taigi reikės dirbtinai sudėti bičių šeimų pozicijos, jog šios atitiktų eksperimento sąlygas.

2 lentelė. Pateikti bičių migravimo eksperimento rezultatai iš straipsnio [BPC⁺17]

		Stebėtų bi- čių kiekis	Migruojančių bi- čių kiekis	Migracijų kiekis	Kontrolinė šeima	Nozematoze užkrėstos bitės	Kontrolinė šeima
Šeima 1 (kraštinė)	133	138	10 (7,52%)	6 (4,35%)	142	30	
Šeima 2 (centrinė)	142	142	32 (22,54%)	23 (16,20%)	358	348	
Šeima 3 (kraštinė)	147	137	7 (4,76%)	6(4,38%)	45	115	
Bandydas 1 (Balandis)	145	143	20(13,79%)	14(9,79%)	233	191	
Bandydas 2 (Gegužė)	133	140	13(9,77%)	11(7,86%)	248	145	
Bandydas 3 (Birželis)	144	134	16(11,11%)	10(7,46%)	64	157	

1.2.3. Varozės plitimas tarp šeimų

Straipsnyje [ID16] aptariamas varozės plitimas tarp skirtingų bičių šeimų. Atkreipiamas dėmesys į atvejus, kai vyksta bičių migracija tarp šeimų. Straipsnyje aprašomas atliktas tyrimas su tikrais aviliais ir bičių šeimomis, kai į vieną šeimą įkeliami 300 varozės erkių. Eksperimentas atliktas Jungtinėse Amerikos Valstijose, Džordžijos valstijoje, aviliai nudažyti ta pačia spalva su landomis tose pačiose pusėse, kad bitės dažniau keliautų iš vieno avilio į kitą. Bičių šeimos įleistos 2012 m. liepos 14-15 d., tuo tarpu varozės erkės nuo liepos 31 – rugpjūčio 9-tos dienos. Eksperimentai atlikti įvairiais laikotarpiais tikrinant du kartus trijų dienų pradžioje ir pabaigoje, nuo rugpjūčio 27-29, rugsėjo 10-12, rugsėjo 25-27, spalio 8-10, spalio 24-26, lapkričio 13-15 dienų. Tarp eksperimente naudotų avilių buvo sudarytos poros avilių, tarp kurių atstumai buvo 0 metrų, 10 metrų, 100 metrų. Atlikus stebėjimus didžiausias erkių kiekis buvo pastebėtas spalio 24-26 avilių porose, kurių atstumas buvo 10 m, ir labai panašų kiekį turėjo 0 m atstumo pora. Paskutinio

mėnesio stebėjimai (žr. 3 lentelę) užfiksavo mažesnę erkių kiekį, o šį rezultatą autoriai aiškino bičių veiklumo sumažėjimu ateinant žiemos sezonui. Papildomai straipsnyje pastebima, jog varozė taip pat daro įtaką tiek pačių bičių gebėjimui orientuotis erdvėje ir suvokti ar avilys priklauso tai pačiai šeimai, tiek ir sargyboje dirbančioms bitėms, kurios privalo atpažinti svetimas bites, įsibrovusias į jų avilį. Straipsnyje nurodomi ir kiti atlikti tyrimai, tarp kurių ir anksčiau minėto T.D. Seeley [SS15] darbas ta pačia tematika, kuriame papildomas dėmesys skiriamas erkių populiacijos kitimui po bičių spietimosi. Tarp retai išdėstytų avilių buvo pastebėtas smarkus erkių populiacijos kritimas, palyginus su spiečiais, susidariusiais glaustai išdėstyta bityne. Visos bičių šeimos, kurios nesudarė naujų spiečių, išmirė nuo varozės.

3 lentelė. [ID16] straipsnio paskutinio stebėjimo rezultatai

	0 m atstumu	10 m atstumu	100 m atstumu
Suaugusios bitės	7504 ±802	7956 ±719	8855 ±1139
Pera	669 ±479	786 ±307	1288 ±557
Visa erkių populiacija	319 ±63	589 ±114	453 ±118
Erkių kiekis bičių šeimoje procentais	7% ±5	10% ±5	14% ±7

1.2.4. Bičių spiečiaus naujo lizdo paieška

Šiame straipsnyje [JMB06] nusakomi principai, kuriais remiasi bičių spiečius, ieškodamas naujos vietos lizdui įrengti. Nurodomos pagrindinės paieškos stadijos kiekvienai spiečiaus bitei:

- Poilsis. Kai bitė aktyviai neprisideda prie lizdo paieškos. Pasiruošimas. Kai bitė ieško kitos bitės šokio, pagal kurį galėtų ieškoti lizdo. Šioje būsenoje bitė išbūna pagal pateiktą formulę (8):

$$r_{\Theta}(t) = \frac{t^2}{t^2 + \Theta^2} \quad (8)$$

Kur $r_{\Theta}(t)$ nurodo tikimybę rasti šokį, t nurodo laiką sekundėmis nuo paieškos pradžios. Θ reikšmė parinkta 4000.

- Sekimas. Kai bitė susiranda šokį ir pagal jį pradeda lizdo paiešką. Sėkmingas paieškas patyrusių bičių kiekį s apibūdina formulė (9):

$$s(w) = \frac{w^2}{w^2 + \Theta^2} \quad (9)$$

Kur w nusako šokių skaičių tai vietai, $\Theta = 60$.

- Paieška. Kai bitė nusprendžia ieškoti vietovės lizdui be informacijos iš kitų bičių šokių.
- Įvertinimas. Kai bitė suranda vietovę pagal šokį ir ją įvertina. Šis straipsnis nenurodo, pagal kokius kriterijus bitė nusprendžia ar aplankyta vieta yra tinkama lizdui. Atliekant bandymus buvo naudojamas normalus pasiskirstymas ir tinkamumo skalė nuo 0 iki 100.

- Klaida. Kai bitė neteisingai interpretuoja šokį ir po nesėkmingų paieškų grįžta į spiečių. Nesėkmingas paieškas patyrusių bičių kiekį u apibūdina formulė (10):

$$u(w) = 1 - \frac{1}{\sqrt{w+1}} \quad (10)$$

Kur w nusako šokių skaičių tai vietai.

- Šokis. Kai bitė perduoda informaciją apie surastą, jai įtikusią lizdo vietą.
- Kelionė. Kai bitė keliauja į geriausią vietą lizdui arba grįžta iš jos. Ar paieška įvyko sėkmingai skaičiuojama pagal formulę (11):

$$P_{sėkmingaPaieška}(w) = s \frac{w}{1,5u(w) + s(w)} \quad (11)$$

Autoriai nurodo, jog pagal atliktus stebėjimus [SB99] tik apie 2% - 4% spiečiaus bičių atlieka šokius. Šie procentai nenusako kiek iš viso bičių dalyvavo paieškoje ir prisidėjo prie galutinio sprendimo, kadangi šokius atlieka tik sėkmingą paiešką atlikusios bitės. Autoriai 10000-15000 bičių dydžio spiečiui išskirdavo 500 bičių, kurios atlikdavo paiešką ir iš kurių 80 procentų bičių dirbdavo paieškos režimu, o likusios ilsėjosi. Šis modelis supaprastina didelį kiekį spiečiaus susidarymo procesų, tačiau realybėje šie procesai yra sudėtingesni ir nėra pilnai suprasti. Spiečių susidarymo ir lizdo ieškojimo santrauką pateikia T. Seeley knygoje „Naminių bičių demokratija“ [See10], tačiau šioje knygoje nėra nurodymo kokiose vietovėse bitės labiausiai linkusios įsikurti lizdą, tėra tik nurodoma, jog bitės dažniausiai renkasi lizdus ir avilius, kuriuose jau anksčiau buvo įsirengusios kitos bitės [VMS85].

1.2.5. Bičių mirtingumas skirtingu metų laiku

Straipsnyje aprašomas matematinis modelis sezoninei bičių gyvenimo trukmei [YYY19] Šis modelis apskaičiuoja $L(t)$ reikšmę, kurią autoriai įvardija kaip numatomą bičių šeimos gyvenimo trukmę. Atlikus pakankamai daug stebėjimų įmanoma išsiskaičiuoti šią reikšmę. Straipsnyje pateikiamos šios formulės:

$$a(t) = \int_{T_{lėliukė}}^{T_{lėliukė}+L(t)} u(t-s)p(t-s)ds \quad (12)$$

$$b(t) = \int_0^{T_{lėliukė}} u(t-s)[p(t-s)]^{\frac{s}{T_{lėliukė}}} ds \quad (13)$$

$$u(t) = v(t - T_{kiauš}) = \begin{cases} u_0(t < t_0) \text{ prieš eksperimentą} \\ u_1(t_0 \leq t \leq t_1) \text{ nuo pradinio stebėjimo iki pirmo} \\ u_2(t_1 \leq t \leq t_2) \text{ nuo pirmo stebėjimo iki antro} \\ \vdots \\ u_k(t_{k-1} \leq t \leq t_k) \text{ nuo } k-1 \text{ iki } k \text{ stebėjimo} \\ \vdots \\ u_{N-1}(t_{N-2} \leq t \leq t_{N-1}) \text{ nuo priešpaskutinio iki paskutinio stebėjimo} \end{cases} \quad (14)$$

$$p(t) = \begin{cases} p_0(t < t_0) \\ p_1(t_0 \leq t \leq t_1) \\ p_2(t_1 \leq t \leq t_2) \\ \vdots \\ p_k(t_{k-1} \leq t \leq t_k) \\ \vdots \\ p_{N-1}(t_{N-2} \leq t \leq t_{N-1}) \end{cases} \quad (15)$$

$$b(tk) = \int_0^{T_{lėliukė}} u(t-s)[p(t-s)]^{\frac{s}{T_{lėliukė}}} ds | k = 0, \dots, N-1 \quad (16)$$

Šiose formulėse $T_{kiauš}$ kiaušinėlio ir lervos vystymosi laikotarpis. Bitei ši reikšmė lygi 9 dienoms. $T_{lėliukė}$ uždarytos akelės, lėliukės išsivystymo laikotarpis, trunkantis 12 dienų. Tarp stebėjimų skaičiuojami dydžiai yra $a(t)$ – visų suaugusių bičių skaičius, $v(t)$ – motinėlės padėtų kiaušinėlių skaičius intervalo metu, $b(t)$ – visų uždarytų akelių skaičius, $u(t)$ naujai uždarytų akelių skaičius intervalo metu. $u(t) = v(t - 9)$, nes $T_{kiauš} = 9$. Straipsnis taip pat pateikia santrauką žinomų bičių gyvenimo trukmių sąrašą, sudarytą remiantis kitais straipsniais. Nurodoma, jog bitės gyvena 15-40 dienų pavasarį, 15-38 dienų vasarą, 50-60 dienų rudenį, ir 150-304 dienų jeigu buvo išsiritusios žiemos sezonui.

2. Modelių sujungimas

Modelio įgyvendinimas remiasi dviejų kitais dvejais bičių procesų modeliais: Beehave [BGT⁺14] ir Bumble-Beehave [BTP⁺17]. Beehave modelis skirtas vienos bičių šeimos procesų tyrimams, Bumble-Beehave skirtas kelių kamanių motinėlių stebėjimui duotoje teritorijoje bei jų sukurtų šeimų stebėjimui. Pagal atliekamą užduotį reikėjo bičių modelio skirto kelioms šeimoms, taigi buvo kuriamas naujas Bumble-Beehave modelio variantas skirtas naminėms bitėms, tačiau kamanių šeimos procesus pakeičiant į bičių šeimos procesus.

Abu minėti modeliai yra atviro kodo ir remiasi atviro kodo NetLogo paketu, skirtu agentinių modelių kūrimui. Dabartinis sprendimas taip pat įgyvendintas naudojant NetLogo, naudojant 6.1.0 versijos modeliavimo paketą, todėl teko pakeisti NetLogo 5.3 versijai skirtą Beehave ir Bumble-Beehave.

Bumble-Beehave ir Beehave modeliai modeliuoja skirtingas bičių šeimas, ir šių šeimų procesai, nors ir atrodytų panašūs, turi daug esminių skirtumų. Atitinkamai šiuose modeliuose įgyvendinami skirtingi procesai (žr. 4 lentelę). Bumble-Beehave negali būti laikomas esminiu šio darbo sprendimu (kaip ir Beehave be papildymų) dėl skirtumų tarp kamanių ir naminių bičių, dėl kurių kamanių modelis nėra išbaigtas naminių bičių atžvilgiu:

- Kamanės, kitaip nei naminės bitės, nežiemoja, o tai reiškia:
 - Žiemą išgyvena tik motinėlės.
 - Pavasarį iš miego pabudusios kamanių motinėlės pradeda šeimą iš naujo susirasdamos lizdą, jį paruošdamos ir sudėdamos kiaušinėlius ir pradeda šeimą be darbininkių.
 - Kamanėms negalioja naminių bičių bitininkystės procesai, kadangi kamanių šeimos nekaupia papildomų medaus atsargų žiemai.
- Kamanių šeimos yra žymiai mažesnės už naminių bičių, Bumble-Beehave modelis turi labai paprastą kiaušinėlių dėjimo modelį, kai kiaušinėliai dedami pagal vieną konstantą.
- Varozė labiau veikia naminių bičių šeimos, nors vykdomi ir tyrimai, bandantys nustatyti naminių bičių varozės poveikį kamanėms [MTD⁺19]. Bumble-beehave šių procesų neturi.

Tačiau dėl kamanių modeliavimo Bumble-Beehave turi privalumų, kurie atitinka atliekamas darbo užduotis:

- Kamanės užaugina daug motinėlių, kurios ieško naujų lizdų. Kelių šeimų modeliavimas ir naujų šeimų kūrimo modeliavimas yra esminė tokio modelio dalis.
- Bumble-Beehave modelyje leidžiama turėti skirtingų rūšių kamanių. Tačiau dėl šaltinių trūkumo modelio kūrėjai daugumą procesų pagrindė B. terrestris kamanės procesais ir empirinėmis žiniomis.

4 lentelė. Bičių šeimos procesai Beehave ir Bumble-Beehave modeliuose

Nr	Procesas	Beehave	Bumble-Beehave
1	Bičių žiemojimas	Bitės žiemoja	Motinėlė užmiega, visos darbininkės miršta
2	Šeimos įkūrimas	Motinėlė su spiečiumi atsikrausto į naują vietą	Peržiemojusi motinėlė ieško naujo lizdo, šeima susidaro tik iš motinėlės
3	Spietimasis	Yra spietimasis, bet nesudaroma nauja šeima (arba nebemodeliuojama senoji šeima, pagal pasirinkimą)	Nėra spietimosi, bet sudaromos naujos šeimos
4	Kiaušinėlių dėjimas	HoPoMo [SC07] sezoninis pagal bičių kiekį	Statinis
5	Bitininkystės procesai	Medaus paėmimas, sirupo ir žiedadulkių pildymas, varozės prevencija	Nėra
6	Varozė ir kitos ligos	Varozės modelis, nėra perdavimo	Nėra
7	Bičių migracija tarp šeimų	Nėra	Nėra
8	Bičių mirtingumas pagal metų laiką	Nėra	Atskiras procesas motinėlių žiemojimui

2.1. Procesai

2.1.1. Žiemojimas

Bumble-Beehave modelio programos logikoje įgyvendinta, jog po žiemos ne vienoje bičių šeimoje neliktų darbininkių, tik motinėlės. Metų pabaigoje tikrinamos sukurtos bitės, ir jeigu jos nėra motinėlės, jos miršta. Papildomai po šio veiksmo atliekami patikrinimai ar iš tiesų modelyje neliko kitų bičių be motinėlių. Šios funkcijos buvo pašalintos naujame modelyje. Pašalinus šią kliūtį buvo rasta ir išspręsta kita problema: medaus trūkumas žiemos metu. Praeitame poskyryje buvo minėta, jog kamanės nekaupia medaus atsargų žiemai. Bumble-Beehave modelyje bičių sprendimus nulemia procesas, parenkantis bitės dabartinį veiksmą, pagal šią veiksmų eilę:

- Kiaušinėlių dėjimas
- Kiaušinėlių priežiūra
- Žiedadulkių nešimas
- Nektaro nešimas

Kiekvienai bitei nusprendžiama, kuriuos veiksmus ji turėtų atlikti. Tai priklauso nuo bičių lyties, darbo ir kitų parametrų vertinimų, taikomų kiekvieno šio veiksmo atžvilgiu. Jeigu pasirenkamos kelios veiklos, priskiriama tik paskutinė pagal eilę, t.y. jeigu bitė turėtų nešti žiedadulkes ir nektarą, ji nuspręš nešti tik nektarą. Žiemojimo atžvilgiu modelyje svarbios tik bitės darbininkės, o medaus paruošimui reikalingas nektaras. Nektaro nešimas sprendžiamas pagal atitinkamus parametrus:

- Reikalingo medaus kiekį dienai
- Medaus atsargų kiekį pagal laiką, per kurį užtektų minėtų atsargų dienomis d
- Papildomų empiriniu būdu nustatytų konstantų
- Minimalų atsargų kiekį, kurį būtina palaikyti, jei įmanoma $E_{\text{minimalus}}$

Galutinis sprendimas N dėl nektaro paieškos priimamas pagal formulę (17):

$$N = \begin{cases} 1, & \text{jei } \frac{E_{\text{idealus}} - E_{\text{atsargos}}}{E_{\text{idealus}}} > T \\ 0, & \text{jei } \frac{E_{\text{idealus}} - E_{\text{atsargos}}}{E_{\text{idealus}}} \leq T \end{cases} \quad (17)$$

Kur T yra empiriniu būdu nustatyta konstanta, E_{atsargos} yra turimo medaus energetinė vertė kJ, idealios energijos atsargos E_{idealus} paskaičiuojamos pagal (18):

$$E_{\text{idealus}} = (c_{\text{idealus}} E_{\text{dienos}} d) + E_{\text{minimalus}} \quad (18)$$

Atliekant bandymus buvo nustatytos atitinkamos konstantos, jog bitės intensyviai ieškotų maisto ir sukauptų apie 40 kilogramų medaus atsargų žiemai.

Užtikrinus medaus kiekį žiemojimo metu bitės neišgyvendavo žiemojimo laikotarpio dėl jų amžiaus. Šie pakeitimai nurodomi bičių mirtingumo pagal metų laiką poskyryje.

Kita kritinė bičių žiemojimo sąlyga yra pakankamas bičių kiekis šeimoje. Šio proceso pakeitimas bus aptartas kiaušinėlių dėjimo poskyryje.

2.1.2. Šeimos įkūrimas

Tiek simuliacijos pradžioje, tiek po žiemojimo Bumblee-Beehave modelyje egzistuoja tik motinėlės. Naminių bičių šeimos prasideda nuo spiečiaus, kurio dydis gali smarkiai skirtis. Sukurtame sprendime pradinis bičių kiekis šeimoje (t.y. tik viena motinėlė) papildomas jau užaugusių darbininkių skaičiumi. Jos jau gali iš karto pradėti ieškoti maisto šaltinių, nešti žiedadulkes ir nektarą.

2.1.3. Kiaušinėlių dėjimas

Bumble-Beehave modelyje kiaušinėliai dedami pagal konstantą, kiekvieną kartą padedant nurodytą kiekį kiaušinėlių. Šį procesą riboja žiedadulkių kiekis ir laikas, kurio prireikia motinėlei dedant kiaušinėlius.

BEEHAVE modelyje remiasi HoPoMo dėjimo modeliu pagal normalųjį skirstinį, atvaizduojanti kintantį dėjimo dažnį skirtingų sezonų metu (7). Duotai dienai d šis dydis randamas pagal formulę f_1 (19):

$$f_1(d) = 1600(1 - \alpha), \quad (19)$$

$$\alpha = \max(\alpha_1, \alpha_2), \text{ kur } \alpha_1 = 1 - \frac{1}{1+x_1 e^{\frac{-2d}{x_2}}} \text{ ir } \alpha_2 = \frac{1}{1+x_3 e^{\frac{-2(d-x_4)}{x_3}}}. \quad (20)$$

Čia konstantos $x_1 = 385$, $x_2 = 25$, $x_3 = 36$, $x_4 = 155$, $x_5 = 30$ paimtos iš HoPoMo modelio [SC07], kuriame eksperimento būdu buvo parinkti parametrai, jog skirstinio kreivė atitiktų empirines žinias. Šis modelis pritaikytas ir naujame sprendime, kartu atsižvelgiant ir į maisto ieškančių ir avilyje liekančių bičių skaičių (21).

$$f_2(d) = \frac{(n_{lizde} + n_{lankuolės} C_{lankuolės}) C_{santykis}}{t_{užaugimo}} \quad (21)$$

$$f(d) = \begin{cases} f_2(d), & \text{jei } f_2(d) < f_1(d) \\ f_1(d) \end{cases}. \quad (22)$$

Kur n_{lizde} yra lizde pasiliekančių bičių kiekis, $n_{lankuolės}$ maisto ieškančių bičių kiekis, $C_{lankuolės}$ bičių, ieškančių maisto, prisidėjimo prie perų priežiūros koeficientas, $C_{santykis}$ koeficientas pagal kurį visos darbininkės prisideda prie perų priežiūros ir $t_{užaugimo}$ dienos, per kurias užauga bitė darbininkė iki suaugusios bitės.

Šis papildymas reikalingas tam, jog nusilpusi, mažai bičių turinti šeima neišmirtų dėl maisto trūkumo bebandydama išmaitinti per daug perų. Jeigu šeima išauga, ši formulė nebeatitinka realių motinėlių galimybių ir tuo atveju vėl naudojamas HoPoMo modelis (22).

2.1.4. Bitininkystės procesai

Jungtiniame modelyje buvo panaudoti bitininkystės procesai, perkelti iš Beehave modelio:

- Nektaro pridėjimas – pridedamas nektaro kiekis (realybėje sirupas), jog bitės galėtų susidaryti pakankamą medaus atsargų kiekį žiemai.
- Žiedadulkių pridėjimas – medžiagų, reikalingų lizdo plėtimui ir naujų kiaušinėlių dėjimui.
- Medaus paėmimas.

Kitas svarbus bitininkystės procesas yra varozės gydymas. Šis procesas bus aptartas kitame poskyryje apie varozę.

2.1.5. Varozė ir kitos ligos

Jungtiniame sprendime panaudotas BEEHAVE modelio varozės modelis. Prie šio proceso prisideda ir varozės gydymo procesas iš BEEHAVE bitininkystės procesų. Šį procesą būtų galima paplėsti varozės plitimu tarp kelių šeimų, tačiau šiame darbe to atlikti dar nepavyko. Šio proceso realizacija remtųsi sprendime jau įgyvendintu migracijos tarp bičių šeimų procesu bei varozės plitimą tarp kelių šeimų nurodantį straipsnį [ID16].

2.1.6. Bičių mirtingumas pagal metų laiką

Bumble-Beehave modelis taiko papildomą mirtingumo modelį motinėlėms žiemojimo metu pagal jos amžių ir masę, tačiau tokie skaičiavimai nėra taikomi darbininkėms, nes kamanių darbininkės nežiemoja. Bičių darbininkės priklausomai nuo metų laiko ir darbų intensyvumo gyvena trumpiau ar ilgiau. Vasaros ir medunešio metu darbininkės išgyvena apie 6 savaites. Prieš žiemą užaugusios bitės, pasiruošusios žiemojimui, gali išgyventi ir iki 6 mėnesių. Bumble-Beehave modelis remiasi konstanta ir bitės miršta iškart pasiekusios maksimalų amžių pagal šią konstantą. Dabartinis sprendimo įgyvendinimas remiasi nurodytais [YYY19] intervalais, dėl to darbininkės išgyvena žiemą. Bumble-Beehave klimato modelį įgyvendina pagal valandas, per kurias bitė dienos metu gali ieškoti maisto. Pavyzdžiui, jeigu pasitaiko lietinga diena, tos valandos atimamos iš bendro laiko. Šį modelį vertėtų praplėsti identiškai pagal temperatūrą ir drėgmę, jog būtų nustatomas tikslesnis sunaudojamas energijos kiekis perų ir šeimos šildymui, tačiau tai nebuvo įgyvendinta šio darbo metu.

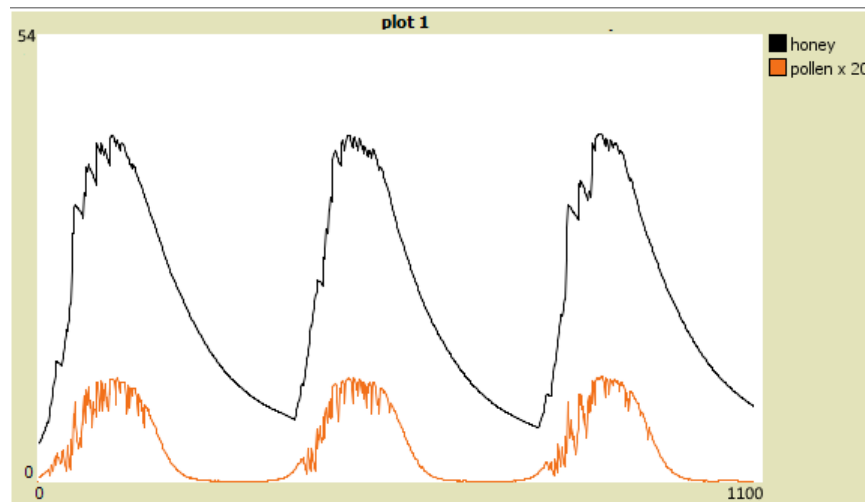
2.2. Jungtinio modelio tikrinimas

Sprendimas buvo kuriamas ir tikrinamas inkrementiniu būdu, pridėdant naują bitininkystės procesą ir lyginant jį su buvusiais modeliais, juose irgi atitinkamai pakeitus modeliuojamos procesus. Pateiktuose eksperimentuose nebuvo atsižvelgiama į tranų dauginimąsi, netikrinant tranų populiacijos motinėlių apvaisinimo metu. Tranai buvo nereikalingi tikrinant žiemojimo procesus, kadangi ir naminių bičių šeimose tranai yra išvaromi iš lizdo ir miršta rudenį, kaip ir kamanių šeimose. Papildomai nebuvo modeliuojamas motinėlių augimas, kadangi Bumble-Beehave modelis su visomis užaugusiomis motinėlėmis pavasarį bando įrengti naujas bičių šeimas, kas galioja kamanėms, tačiau naminėms bitėms galioja tik tuo atveju, jeigu pradinė šeima ruošiasi spiestis. Lyginant gautą sprendimą su Beehave modeliu nebuvo kuriamos naujos šeimos ar spietimosi procesai, kadangi Beehave modelis modeliuoja tik vieną šeimą. Šie procesai Bumble-Beehave modelyje vyksta jeigu modelyje atsiranda peržiemojusių motinėlių, taigi buvo sustabdytas motinėlių auginimas. Tai buvo atlikta Bumble-Beehave modelyje nurodant labai didelį svorį kiaušinėliams, lervoms ir lėliukėms, jog šį išsivystytą į motinėlę. Taigi šiame etape naujų motinėlių atsiradimas nebuvo modeliuojamas.

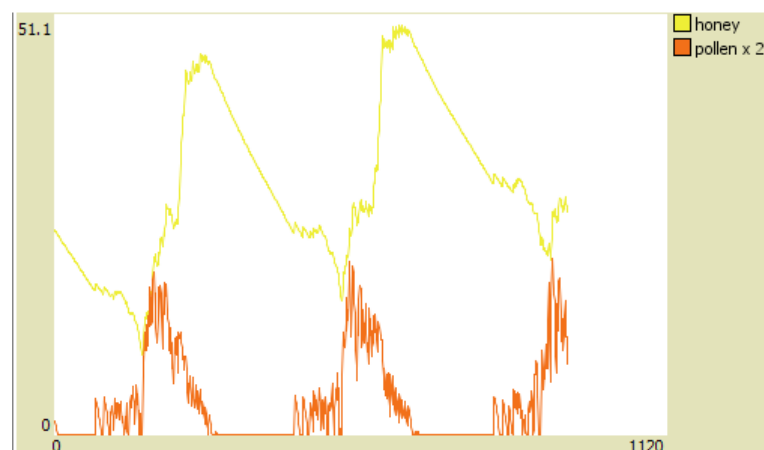
2.2.1. Lyginimas su Beehave modeliu

Beehave populiacijos skaičiavimai buvo lyginami su empirinėmis žiniomis apie realią bičių šeimų populiacijos kaitą. Modelio trūkumai buvo nustatyti ir paviešinti Europos maisto saugos tarnybos [EFS15]. Esminiai trūkumai buvo pesticidų procesų trūkumas, primitivus landšafto (topografijos, augalijos) modelis, ribota klimato konfigūracija (modelis buvo pritaikytas centrinės Europos klimato zonai) ir apribotas ligų kiekis, tiriant tik varozę. Kadangi Bumble-Beehave modelis ištaisė klimato ir landšafto trūkumus, su idealiomis sąlygomis naujas modelis, atitinkantis empirines žinias, turėtų atitikti ir Beehave modelio gaunamus rezultatus. Tikrinant kuriamą modelį buvo perkelti procesai iš Beehave modelio ir tikrinami parametrai ir daromi bandymai kol abiejų mo-

delių gaunami bičių šeimų dydžiai, kiaušinėlių, sukaupito medaus ir žiedadulkių kiekiai pasidarė panašūs. Šie rezultatai buvo lyginamai pagal modelio kuriamų resursų grafikus (žr. 1 ir 2 pav.).

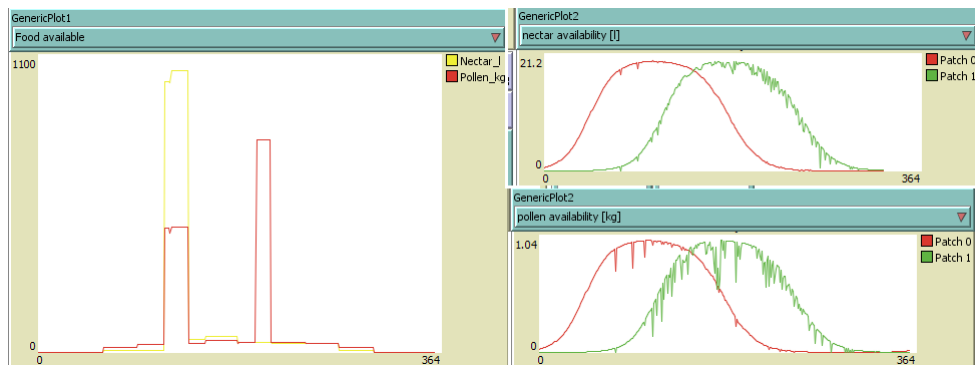


1 pav. Medaus kiekis (kg) ir žiedadulkių kiekis dabartiniame sprendime

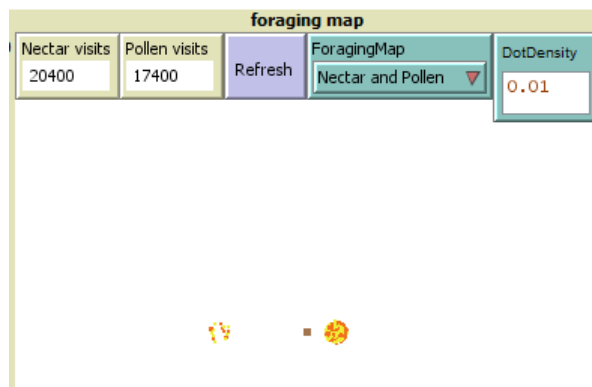


2 pav. Medaus kiekis (kg) ir žiedadulkių kiekis Beehave modelyje

Yra kiti skirtumai tarp modelių, kurie matosi nagrinėjant populiacijų ir sukaupito maisto (1 ir 2 pav.) skirtumus. Viena pagrindinių šio skirtumo priežasčių – maisto prieinamumas modeliuose. Beehave modelis turi tik du gėlių laukus, kurių nektaro ir žiedadulkių kiekis nurodomas normaliuoju skirstiniu (3 pav.), bet abu laukai turi skirtingus laikotarpius, kada bitės gali surinkti iš jų didžiausią kiekį maisto. Bumble-Beehave modelis gėlių ir augalų rūšis bei jų parametrus gauna pagal vieną iš įvesties failų, nurodančių skirtingų rūšių gėles ir augalus. Beehave modelyje bitės gali pasiekti abu gėlių laukus (4 pav.), tuo tarpu Bumble-Beehave modelyje atsižvelgiama į paties lauko atstumą nuo šeimos pagal pateiktą žemėlapi (5 pav.) bei reikalingo bitės liežuvelio ilgį, jog atskridusi bitė galėtų surinkti iš augalo nektarą.



3 pav. Maisto pasiekiamumas naujajame modelyje (kairėje) ir Beehave (dešinėje)

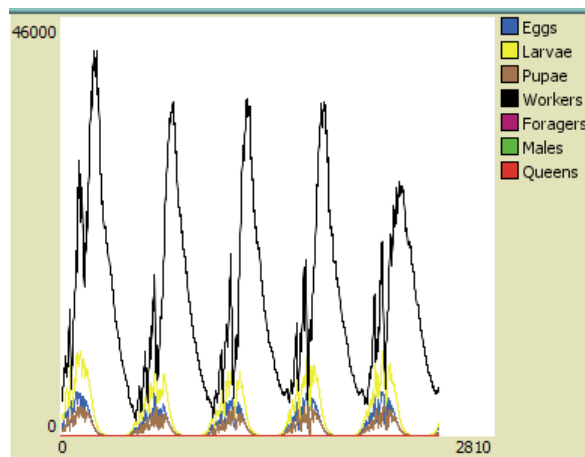


4 pav. Landšafto atvaizdavimas Beehave modelyje

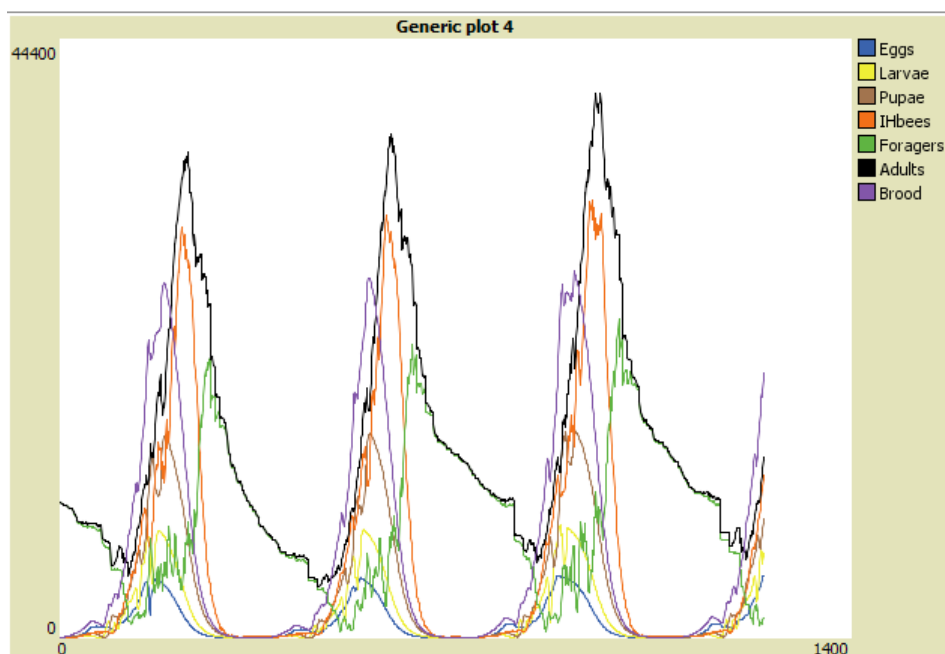


5 pav. Landšafto atvaizdavimas Bumble-Beehave modelyje

Šeimos sandara (6 ir 7 pav.) išlieka panaši. Vasarą bičių šeima išauga iki 40-50 tūkstančių bičių, pavasarį sumažėja iki 6-4 tūkstančių.



6 pav. Šeimos struktūra dabartiniame sprendime



7 pav. Šeimos struktūra Beehave modelyje

Kiti skirtumai susiję su bičių mirtingumu. Dabartiniame sprendime atsižvelgiama tik į bičių mirtingumą paieškos metu ir jos didžiausią gyvenimo trukmę. Jog bitės galėtų peržiemoti, proporcingai buvo pakeista rudenį išsiritusių bičių gyvenimo trukmė, atitinkanti realius bičių šeimos procesus, kadangi žiemojančios bitės neieško maisto ir nenaudoja papildomų medžiagų nektaro vertimui į medų. Beehave modelyje prie šio proceso prisideda ir bitės nukeliant atstumą, kadangi bičių sparneliai nusidėvi intensyvaus darbo metu ir bitė miršta anksčiau, tačiau tai nesudaro didelio skirtumo, kadangi bitei ieškant maisto visais atvejais išauga tikimybė mirti.

2.3. Jungtinio modelio aptarimas

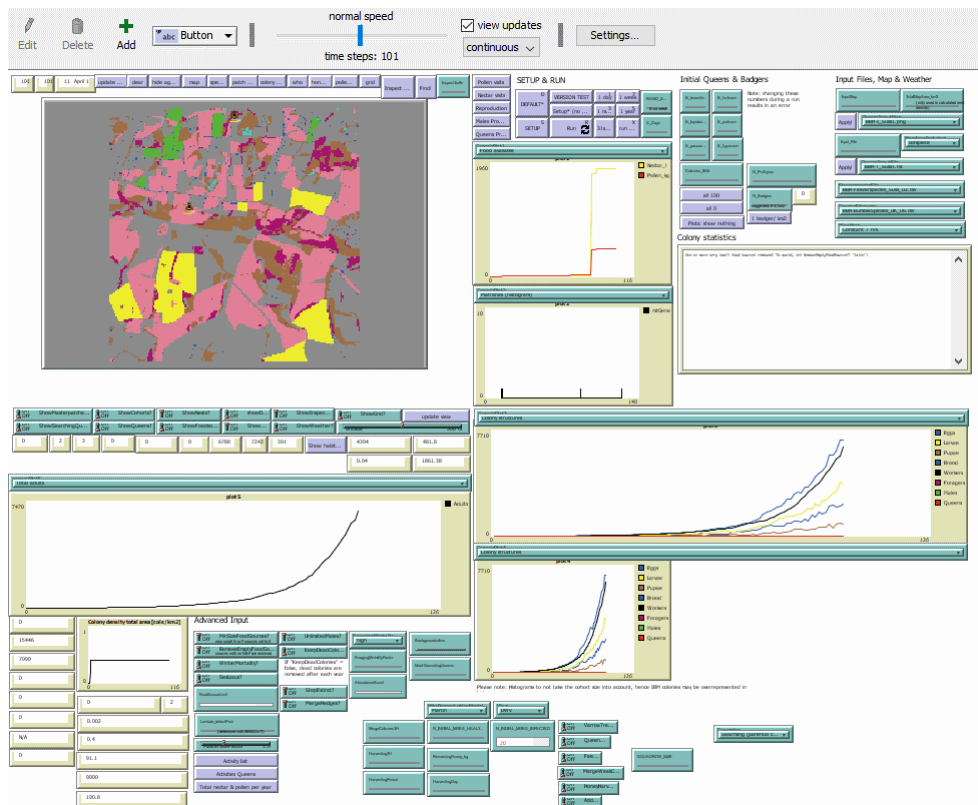
Darbo esminis sprendimas remiasi dvejais bitininkystės procesų modeliais, kurie turi daug esminių logikos skirtumų bei skirti modeliuoti skirtingą kiekį objektų ir skirtingoje aplinkoje. Tačiau kartu šie modeliai nusako didžiąją dalį procesų, kurie atspindi naminių bičių šeimos gyvenimą.

Iš šio skyriaus pateiktų grafikų matome, jog jungtiniame modelyje bičių populiacijos ir maisto kreivės yra panašios į BEEHAVE modelio kreives. Palyginkime gautus rezultatus su empirinėmis žiniomis.

Bičių šeimos išauga jeigu bitės gali rasti daug žiedadulkių. Tai labai priklauso nuo aplinkinės floros, kuri taip pat priklauso nuo vietovės. Pavyzdyje [Oli15] teigiama, jog bičių šeimos dydis pasiekia viršūnę birželio ir liepos mėnesiais su 50 tūkstančių suaugusių bičių, o tarp lapkričio ir kovo laikosi ties 8 – 12 tūkstančių bičių riba. Tačiau populiacijos atžvilgiu panašius skaičius teikia ir lietuviški vadovėliai. Teigiama, jog stipri šeima užauga iki 60-65 tūkstančių darbininkių [Kar00]. Šie rezultatai buvo pasiekiami ir dabartiniame sprendime, priklausomai nuo keičiamų landšafto duomenų. Tačiau šie vadovėliai nepateikia tikslios informacijos, tame pačiame šaltinyje stiprios šeimos nurodomos, turinčios 60-70 tūkstančių bičių vasarą ir 25-30 tūkstančių žiemojančių bičių, o tai viršija kitų šaltinių informaciją.

Kitas nagrinėtas modelis, be nagrinėtų literatūros apžvalgos metu, teikia panašius rezultatus į Beehave bei remiasi HoPoMo modeliu [BWZ16; SC07]. Jame bičių šeimos kiekis sumažėja ir iki 4 tūkstančių bičių kiekio šeimoje, šis sumažėjimas įvyksta pavasario pradžioje, kai vėl dedami kiaušinėliai ir pirmosios bitės, ieškančios maisto, turi didžiausią mirtingumą. Vasarą pasiekia 40-50 tūkstančių bičių ribą.

Taigi, sukurtas jungtinis modelis, veikiantis agentinio modeliavimo principu ir Net-Logo paketu, atitinkantis empirines žinias ir BEEHAVE modelio gaunamus rezultatus. Trečiame skyriuje aptarsime naujus įgyvendintus procesus.



8 pav. Grafinė jungtinio modelio grafinė sąsaja iš BEEHAVE ir Bumble-BEEHAVE elementų

3. Patobulintas modelis

Praeitame skyriuje pristatytas jungtinis BEEHAVE ir Bumble-BEEHAVE modelis, bei procesai, kuriuos pavyko apjungti tarp modelių, jog būtų įmanoma modeliuoti kelias naminių bičių šeimas vienu metu. Dabar pabandykime praplėsti šį modelį naujais procesais.

3.1. Bičių migracija tarp šeimų

Jungtinį sprendimą papildykime bičių migracijos procesu, pagal kurį vėliau galėsime įgyvendinti ir bičių spietimąsi.

Literatūros analizės metu rasti empiriniai duomenis apie bičių migraciją [BPC⁺17; PC98], tačiau šiam procesui nebuvo pateiktas joks migracijos apskaičiavimo modelis. Galime remtis dar vienu straipsniu, pateikiančiu lygtį migruojančių bičių santykiui [FNF⁺15]:

$$n_{stebėtos} = 0,716 + \frac{13,96}{i_{kaimyno}} \quad (23)$$

Kur $n_{stebėtos}$ yra tyrime stebėtų migruojančių bičių kiekis į konkrečią stebimą šeimą iš kitų šeimų, esančių atstumu $i_{kaimyno}$. Nurodomo eksperimento metu 14 avilių buvo sustatyti 18 metrų ilgio eilėje, taigi $n_{kaimyno}$ beveik atitinka metrą, tačiau šį atstumą reiktų suprasti kaip pozicijos eilėje skirtumą arba indeksą. Jei $n_{kaimyno} = 1$, tai reiškia, jog bitės į stebimą šeimą atkeliavo iš kairėje ir dešinėje esančių artimiausių kaimynų. Pateikti koeficientai atitinka tik eksperimente stebėtų bičių kiekį. Bendru atveju bitės imigrantės sudaro $17 \pm 4\%$ bičių šeimos populiacijos [BPC⁺17].

Šie eksperimentai atlikti rudenį, tačiau pateikia panašius rezultatus kaip ir kiti stebėjimai [BPC⁺17; PC98], ir galima teigti, jog panašus migracijos pasiskirstymas vyksta ir vasaros metu. Taigi jungtiniame modelyje migracija įgyvendinama persikrstant bites, pasirinkus šeimą ir priskiriant kaimyninių šeimų bites pasirinktai šeimai. Atsižvelgiant į intensyviausio medaus nešimo laikus, kai dažniausiai pasitaiko migracijos reiškinys, mes galime pasirinktai bičių šeimai priskirti imigruojančių bičių iš kiekvienos kaimynės kas 6 savaites, kadangi darbininkės tuo laikotarpiu tiek laiko ir gyvena. Tuo atveju pasiskaičiuojame pagal modelį sumą, kiek bičių $n_{migruojančios}$ migruotų pagal stebėjimą ir kaimynų didžiausią atstumą $N_{kaimynai}$.

$$n_{migruojančios} = \sum_{i=1}^{N_{kaimynai}} = 0,716 + \frac{13,96}{i} \quad (24)$$

Ir pagal duotą santykį 17% santykį perskaičiuotame iš kiekvieno $i_{kaimyno}$ atstumu esančio kaimyno gaunamą bičių kiekį $m_{migruojančios}$:

$$m_{migruojančios} = (0,716 + \frac{13,96}{i_{kaimyno}}) \frac{0,17 n_{visos}}{n_{migruojančios}} \quad (25)$$

Taigi, jungtinį modelį galima būtų praplėsti bičių migracijos procesu, tačiau jis galioja tik labai ribotomis sąlygomis. Tyrimo metu nebuvo pritaikyti varozės plitimo procesai tarp šeimų, o šiame procese ir pateikto straipsnio [BPC⁺17] modelyje pateikiamas idealizuotas atvejis, jei bitės

migruoja nepaveiktos ligų. Be to, šis procesas galioja tik jei atstumas tarp bičių šeimų yra labai mažas ir atitinka bityną. Be to, aptariamas modelis remiasi eksperimentu, atliktu su bičių migraciją skatinančiomis sąlygomis: aviliai buvo nudažyti identiška, avilių lakos (pro kur bitė įskrenda į avilį), buvo nukreiptos į tą pačią pusę. Taigi, nors pristatytas migracijos procesas remiasi empiriniais duomenimis, jį būtų verta perdaryti prijungus varozės procesus.

3.2. Spietimosi procesas ir algoritmas

Įprastai rastą maisto šaltinį bitės nurodo kitoms bitėms šokio metu, o priklausomai nuo maisto šaltinio kokybės, šis šokis vyksta trumpiau arba ilgiau, jog šoki stebinės bitės turėtų daugiau progų pasirinkti geresnį šaltinį ilgesnio šokio metu.

Kai spiečiasi bitės, šeima susispiečia į kamuolį aplink motinėlą, o apie 300-500 bičių žvalgų atlieka naujo būsto paiešką. Tačiau suradusios naują lizdą ir iš naujo šokdamos ir tikrindamos rastą vietą, šios bitės kiekvieną kartą, nepaisant nepakitusios lizdo kokybės, trumpina savo šokio ilgį. Pagal stebėjimus nurodoma, jog geros kokybės lizdą apibūdinantis šokis trunka apie šimtą ciklų ir su kiekvienu pakartotiniu patikrinimu sumažinamas 10-15 ciklų [See10]. Taigi, geros kokybės lizdai po kelių pakartotinių apsilankymų nusakomi tarsi turėtų mažesnę kokybę. Toks geriausio lizdo atsisakymas leidžia stebėtojoms pasirinkti kitus lizdus dažniau ir išbandyti daugiau variantų.

Šį slopinimo reiškinį verta ištirti plačiau. Toliau aptarsime dirbtinės bičių šeimos algoritmus, kurie remiasi maisto šaltinio paieška, bei pabandysime šiuos algoritmus pritaikyti spietimosi procesui.

3.2.1. Dirbtinės bičių šeimos algoritmas

Dirbtinės bičių šeimos algoritmas [KB07] priklauso kolektyvinio intelekto algoritmų šeimai ir yra metaeuristinių optimizavimo metodų pogrupis [MBB11]. Šie algoritmai skirti spręsti sudėtingus optimizavimo uždavinius, kurie turi daug optimizavimo kriterijų. Optimizavimo uždavinius galime apibrėžti pora (S, f) ; čia S žymi sprendinių aibę, o f – skaliarinę tikslo funkciją, kurios apibrėžimo sritis – S , o reikšmių aibė – realieji skaičiai. Tuomet išspręsti uždavinį (S, f) reiškia surasti sprendinį s :

$$s \in S = \{s^\nabla | s^\nabla = \operatorname{argmin}_{s \in S} f(s)\}. \quad (26)$$

Čia s yra globaliai optimalaus sprendinys. Ieškosime sprendinio, kuriame tikslo funkcija įgyja minimalią reikšmę, t.y. minimizuojama. Panagrinėkime kelias klasikinės tikslo funkcijas, naudojamas metaeuristinių algoritmų efektyvumo lyginimui. 5-oje lentelėje nurodomos darbe naudojamos klasikinės funkcijos ir jų tipai. M - funkcija multimodalinė, turinti daugiau nei vieną modą (nors šiame kontekste laikoma, kad turi vieną minimumą), U - unimodalinė, S - atskiriama (duotai funkcijai $h(x, y)$ galioja $h(x, y) = h_1(x)h_2(y)$), N - neatskiriama. Verta paminėti, jog Schwefel funkcija [Sch81], pateikta [ABK19] šaltinyje ir naudojama šiame darbe, įprastai apibrėžiama $f(x_1 \dots x_n) = \sum_{i=1}^D (-x_i \sin(\sqrt{|x_i|})) + 418,982887 * D$ ir jos minimumas lygus 0, tačiau bus naudojamas variantas, nurodytas 5-oje lentelėje.

5 lentelė. Klasikinės tikslo funkcijos optimizavimo uždaviniais

Funkcija	Ti- pas	Di- men- sija (D)	Sprendinių aibė	Minimumas	Formulė
Sferos	US	30	$[-100,100]^D$	0	$f(x) = \sum_{i=1}^D (x_i^2)$
Rosenbrock	UN	30	$[-30,30]^D$	0	$f(x) = \sum_{i=1}^D (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$
Rastrigin	MS	30	$[-5.12,5.12]^D$	0	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos 2\pi x_i + 10)$
Dixon Pri- ce	UN	30	$[-10,10]^D$	0	$f(x) = (x_1 - 1)^2 \sum_{i=2}^D (i(2x_i^2 - x_{i-1})^2)$
Ackley	MN	30	$[-32,32]^D$	0	$f(x) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D (x_i^2)}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i)$
Schwefel	MS	30	$[-500,500]^D$	-12569,5	$f(x) = \sum_{i=1}^D (-x_i \sin(\sqrt{ x_i }))$
Griewank	MN	30	$[-600,600]^D$	0	$f(x) = \frac{1}{4000} (\sum_{i=1}^D (x_i^2)) - (\prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})) + 1$
Kupra- nugario (camel)	MN	2	$[-5,5]^D$	-1,03163	$f(x) = 4x_1^2 - 2,1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
Branin	MN	2	$[-5,10]^D$	0,39789	$f(x) = (x_2 + 5 - \frac{5,1}{4\pi^2} - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$

Šios funkcijos ir jų testavimas implementuotas Python kalba.

Dirbtinį bičių šeimos algoritmą galime aprašyti šiais žingsniais:

1. Inicializacija. Nustatomi algoritmo parametrai. Spiečiaus populiacijos dydis, algoritmo ciklų vykdymo kiekis, algoritmo baigimo kriterijai. Visos populiacijos individualioms bitėms pritaikomas atsitiktinis sprendinys sprendinių aibės ribose $[k, j]$:

$$x_{m,i} = k_i + rand(0,1)(j_i - k_i) \quad (27)$$

2. Bičių darbininkų fazė. Kiekvienas sprendinys gerinamas pagal nustatytą lokalojo gerinimo procedūrą, ir, radus geresnį sprendinį, dabartinis bitės sprendinys pakeičiamas geresniu. Naujas sprendinys $v_{m,i}$ ieškomas keičiant $x_{m,i}$:

$$v_{m,i} = x_{m,i} + \phi_{m,i}(x_{m,i} - x_{k,i}) \quad (28)$$

Čia $\phi_{m,i}$ atsitiktinis skaičius tarp $[-1,1]$.

3. Stebinčių bičių fazė. Praeitame žingsnyje rasti ir gerinami bičių sprendiniai perduodami kitoms bitėms, apskaičiuojant tikimybes kiekvienam sprendiniui, jog stebinčioji bitė pasirinks gerinti tą sprendinį. Ši tikimybė proporcinga skaičiuojamo sprendinio kokybei pagal tinkamumo funkcijos reikšmę:

$$p_m = \frac{fit(x_m)}{\sum_{m=1}^{CS} fit(x_m)} \quad (29)$$

Kur CS yra šeimos dydis, o tinkamumo funkcija $fit(x_m)$ apskaičiuojama pagal tikslo funkciją $f(x)$: 1

$$fit(x_m) = \begin{cases} 1 + f(x_m), & jei f(x_m) > 0 \\ \frac{1}{1+|f(x_m)|}, & jei f(x_m) \leq 0 \end{cases} \quad (30)$$

4. Bičių žvalgų fazė. Jei konkretus sprendinys antroje fazėje buvo nesėkmingai gerintas nustatyta kiekį kartų l , jis yra atmetamas ir pakeičiamas kitu atsitiktinai parenkamu sprendiniu pagal (27).
5. Išimename geriausias rezultatas. Jeigu rezultatas netenkina baigimo sąlygų, grįžtama į antrą žingsnį. Baigimo sąlygos įprastai yra algoritmų iteracijų riba, tačiau dėl algoritmo lyginimo tikslų arba dėl ribotų resursų, šie kriterijai taip pat gali būti tikslo funkcijos skaičiavimų kiekis arba bendras vykdymo laikas. Paprastoje šio algoritmo versijoje tai yra iteracijų kiekis, kuris šiame žingsnyje didinamas vienetu.

Šis algoritmas turi įvairius galimus pakeitimus, gerinančius jo našumą. Vienas taikomų metodų inicializacijos žingsnyje pritaikyti papildomas sprendinių lokoliojo pagerinimo ar išbarstymo procedūras. Aptarkime keletą tokių algoritmo pakeitimų.

Pradiniame Karabogos algoritme [KB07] bitės stebėtojos pasirinkdavo naują sprendinį pagal tą patį algoritmą kaip ir dirbančios bitės. Realybėje stebinčios bitės ieškotų naujo maisto šaltinio dirbančios bitės nurodytos vietos kaimynystėje. Tuo pagrindu sukurtas greitis dirbtinės bičių kolonijos algoritmas [KG14] (angl. Quick Artificial Bee Colony algorithm arba qABC), kuris remiasi naujo sprendinio ieškojimu pagal pasirinkto sprendinio kaimynystėje esančiais sprendiniais. Kaimynystė šaltinyje pateikiama kaip euklidinis atstumas tarp sprendinių:

$$v_{N_m, i}^{geriausias} = x_{N_m, i}^{geriausias} + \phi_{m, i}^{geriausias} (x_{N_m, i}^{geriausias} - x_{k, i}) \quad (31)$$

Verta pastebėti, jog naujasis sprendinys remiasi geriausio kaimyno sprendinio vektoriumi, o ne buvusio sprendinio vektoriumi kaip paprastame dirbtinės bičių šeimos algoritme.

Šio algoritmo pagrindu sukurtas dar kitas jo patobulinimas, vadinamas patobulintu greituoju dirbtinės bičių šeimos algoritmu (angl. Improved Quick Artificial Bee Colony algorithm arba iqABC) [ABK19]. Patobulintas greitis dirbtinės bičių šeimos algoritmas panašią formulę (31) pritaiko ir stebėtojoms(32), ir darbininkėms(33):

$$v_{N_m, i}^{isimintas} = x_{N_m, i}^{isimintas} + \phi_{m, i}^{isimintas} (x_{N_m, i}^{geriausias} - x_{k, i}) \quad (32)$$

$$v_{N_m, i}^{isimintas} = x_{N_m, i}^{isimintas} + \phi_{m, i}^{isimintas} (x_{N_m, i}^{geriausias} - x_{j, i}) \quad (33)$$

Tačiau čia $x_{N_m, i}^{isimintas}$ yra ne $x_{k, i}$ kaimynas, o geriausias įsimintas sprendinys. Jis tikrinamas $l_{darbininkiu}$ kiekį kartų darbininkų žingsnyje ir tiek pat kartų $l_{stebinciu}$ tikrinamas stebinčiųjų bičių žingsnyje. Bendrai vadinkime šią ribą $l_{geriausias}$. Viršijus šią ribą, $l_{geriausias}$ riba pakeičiama į l ribą atitinkamam žingsniui, kartu pakeičiant naudojamas formules į paprasto bičių šeimos algoritmo formules (28) (29).

Taigi iqABC algoritmas turi keturis režimus:

- Kai darbininkų žingsniui taikoma (32) ir stebėtojų žingsniui (33).
- Kai darbininkų žingsniui taikoma ABC formulė (28), o stebėtojų žingsniui (33).
- Kai darbininkų žingsniui taikoma (33), o stebėtojų žingsniui ABC formulė (29).
- Kai algoritmas veikia pagal ABC formules (28) (29)

Šis algoritmas su kiekvienu tikslo funkcijos įvertinimu greičiau konverguoja link globalaus minimumo, nei įprastas ABC arba qABC. Kadangi daugiau sprendinių ieškoma aplink geriausią pradinį žinomą sprendinį, jei pradinėje sprendinių aibėje pasitaikė geras sprendinys, šis algoritmas konverguoja žymiai greičiau.

Dirbtinis bičių šeimos algoritmas remiasi bičių maisto paieškos procesais realiame pasaulyje. Tačiau iš atliktų stebėjimų matome, jog bičių elgesys maisto paieškos ir naujo lizdo paieškos metu skiriasi. Besispijamos bitės stengiasi kaip galima greičiau rasti geriausią naujo lizdo vietą. Toliau aptarsime spietimosi principus.

3.2.2. Dirbtinio spietimosi algoritmas

Poskyrio pradžioje minėjome spietimosi procese pastebimą slopinimo reiškinį. Dirbtinių bičių šeimų algoritmai slopinimą įgyvendina tik dalinai, turi šaltinio tikrinimo ribas, tačiau pačios ribos tėra konstantos.

Atsižvelgiant į pagreitintus dirbinių bičių algoritmus galime pakeisti vieno sprendinio tikrinimo ribą pagal jo tikslo funkcijos reikšmę, o ne pagal konstantą. Vienintelis iššūkis šio principo pritaikyme – mes iš anksto nežinome geriausios tikslo funkcijos reikšmės iš anksto vykdydami algoritmą. Šiuo atveju santykinį sprendinio tinkamumą vertinsime pagal didžiausią ir mažiausią žinomą tikslo funkcijos reikšmę. Šį principą išbandysime taikydami jį patobulinto greitojo bičių algoritmo žingsniams (iqABC).

Patobulinto greitojo bičių algoritmo atžvilgiu keisime stebinčių bičių žingsnio tikimybės formulę ir visas tikrinimo ribas. Stebinčių bičių žingsnyje sprendinio tikrinimo tikimybės p_m formulė (34) atsižvelgs į sprendinio prieš tai buvusių bandymų kiekį(35):

$$p_{m, santykinis} = \frac{g(x_m)}{\sum_{m=1}^{CS} g(x_m)} \quad (34)$$

$$g(x_m) = \frac{fit(x_m)}{norm(x_m)} bandymai(x_m) \quad (35)$$

Kur $norm(x_m)$ funkcija priskiria 2 mažiausiai rasti reikšmei ir 1 didžiausiai rasti reikšmei, kadangi bandome minimizuoti tikslo funkciją:

$$norm(x_m) = 1 + \frac{fit(x_{blogiausias}) - fit(x_m)}{fit(x_{blogiausias}) - fit(x_{geriausias})} \quad (36)$$

Bendrą iqABC ribą $l_{geriausias}$ keičiame į naują ribą $l_{santykinis}(x)$, kuri nėra konstanta, o apskaičiuojama pagal geriausio ir prasčiausio rasto sprendinio santykį (37). Viršijus šią ribą, $l_{santykinis}(x)$ pakeičiamas į $l_{naujas}(x)$ ribą atitinkamam žingsniui, kartu pakeičiant naudojamas formules į paprasto bičių šeimos algoritmo formules (28) (29), tačiau $l_{naujas}(x)$, palyginus su ABC ketvirto žingsnio l riba, taip pat perskaičiuojamas pagal geriausią ir prasčiausią sprendinį.

$$l_{geriausias}(x) = bandymai(x_m) norm(x_m) l_{geriausias} \quad (37)$$

$$l_{naujas}(x) = bandymai(x_m) norm(x_m) l \quad (38)$$

Ciklo pabaigoje be geriausio sprendinio $x_{geriausias}$ įsimenamas ir blogiausias $x_{blogiausias}$ sprendinys.

Bičių žvalgų ir dirbančiųjų žingsnyje jei konkretus sprendinys (x_m) buvo nesėkmingai gerintas $k(x_m)$ kartų, jis yra atmetamas ir pakeičiamas kitu atsitiktinai parenkamu sprendiniu:

$$k(x_m) = fit(x_{geriausias}) norm(x_m) bandymai(x_m) \quad (39)$$

Algoritmą, veikiantį su šiais pakeitimais, pavadinkime patobulintu greituoju besispiečiančios šeimos algoritmu (angl. Improved Quick Swarming Artificial Bee Colony arba iqsABC). Šio algoritmo bičių žvalgų ir dirbininkų žingsnių pseudokodas pateikiamas 5 priede. Šis algoritmas buvo įgyvendintas su Python programavimo kalba, besiremiant įprasto ABC algoritmo implementacija HIVE [Wui17], tačiau ją pakeitus, jog būtų kaupiama informacija pagal kiekvieną tikslo funkcijos įvertinimą ir algoritmas būtų baigiamas pasiekus nustatytą tikslo funkcijos tikrinimų ribą. Tuo pačiu principu įgyvendinti ir iqABC bei qABC algoritmai, kurie bus naudojami kitame skyriuje.

3.2.3. Dirbtinių bičių šeimų algoritmų lyginimas

Dirbtinių bičių šeimų algoritmai nėra determinuoti ir turi daug versijų, taigi šių algoritmų lyginimui yra nurodyti principai, kurių rekomenduojama laikytis [MLK⁺15], nustatant šių algoritmų efektyvumą

Dirbtinių bičių šeimų algoritmų lyginimui reikia nubrėžti:

- Ribotą tikslo funkcijos įvertinimų kiekį
- Rezultatų konvergavimo diagramas su vienu funkcijos įvertinimo kiekiu

Dirbtinių bičių šeimos algoritmo sudėtingumas pagal pateiktas rekomendacijas skaičiuojamas pagal tikslo funkcijos įvertinimo kiekį. Paprastojo bičių šeimos algoritmo šaltinyje pateikiama, jog to algoritmo sudėtingumas yra $2 \times CS \times L$, kur CS bičių kiekis naudojamas du kartus dėl bičių darbininkų ir stebėtojų žingsnio, o L yra iteracijų riba. Papildomai inicializacijos žingsnyje sukuriant pradinis sprendinius atliekamas CS kiekis tikslo funkcijos įvertinimų. Tobulinant algoritmus kartais pasitelkiama daugiau tikslo funkcijos įvertinimų, tačiau atsižvelgiama tik į bendrą iteracijų kiekį. Taigi kai kurie algoritmai, nors ir daugiau kartų skaičiuoja tikslo funkciją, šaltiniuose pristatomi tarsi turėtų tą patį sudėtingumą. Kuriant naująjį spietimosi algoritmą buvo padaryta ta pati klaida, kadangi darbininkų fazėje naujas rastas sprendinys (28) gali būti atmestas ir perskaiciuotas žvalgų fazėje (27), o tai naudoja papildomą tikslo funkcijos skaičiavimą. Todėl būtina algoritmo baigimo kriterijų iš iteracijų ribos pakeisti į tikslo funkcijos įvertinimų kiekį. Toliau pateikiami rezultatai remsis tikslo funkcijų įvertinimų riba.

Atsižvelgiant į prieš tai pateiktus algoritmus [ABK19; KG14], prasminga nustatyti panašias sprendinio tikrinimo ir kitų parametrų vertes, kurias pateikia kiti šaltiniai (žr. 6 lentelę).

6 lentelė. iqsABC algoritmo vertinime naudojamos parametrų reikšmės klasikinėms funkcijoms

Kintamasis	Prasmė	Reikšmė
CS	Bičių šeimos dydis (pradinių sprendinių kiekis)	50
D	Dimensijos (kintamųjų kiekis)	30
l	Sprendinio tikrinimo riba	$\frac{CS \times D}{2}$
$l_{geriausias}$	Sprendinio tikrinimo riba geriausiam sprendiniui	$CS \times D$
-	Tikslo funkcijų vertinimo riba	500000
-	Bandymų kiekis	30

Pasirinkus atitinkančius parametrus ABC, qABC, iqABC ir iqsABC algoritmai buvo išbandyti su klasikinėmis tikslo funkcijomis (žr. 5 lentelę) bei jų gaunami rezultatai ir konvergavimo grafikai palyginti su prieš tai buvusių algoritmų šaltiniais (šie grafikai pateikiami 3 priede).

Plačiau analizuosime skirtumus tarp iqABC algoritmo ir pagal jį sukurto naujo iqsABC algoritmo.

Kadangi analizuojami algoritmai yra nedeterminuoti, gautų rezultatų tikrinimas nesuteikia prasmingo algoritmo įvertinimo. Rezultatų skirtumui įvertinti pasitelkiamas Wilcoxon testas [Wil45], skirtas įrodyti statistinėms hipotezėms (40) duomenims, kai neturime išankstinių prielaidų apie duomenis. Hipotezę H_0 atmesime tik tuo atveju, jei testo duodama reikšmė $p < 0.05$.

$$\begin{aligned}
 H_0 &: skirstinys1 = skirstinys2 \\
 H_1 &: skirstinys1 \neq skirstinys2
 \end{aligned}
 \tag{40}$$

Wilcoxon testas atliekamas su $2N$ reikšmių tarp 2 skirtingų vienodo dydžio imčių. Randaime skirtumą tarp kiekvienos poros reikšmių, įsimename ženklą ir atmetame reikšmių poras, kurių skirtumas lygus 0. Likusias reikšmes žymime N_r . Išrikiuojame skirtumus nuo mažiausio iki didžiausio ir priskiriame numerį R pagal eilės tvarką, pradedant nuo vieneto. Suskaičiuojame testo

statistiką W pagal formulę(42):

$$W = \sum_{i=1}^{N_r} [\text{sgn}(x_{2,1} - x_{1,1})] R_i \quad (41)$$

Tuo atveju, jeigu hipotezė H_0 teisinga, W reikšmės patenka į savą skirstinį. Jei $N_r \geq 20$, standartinis įvertis z apskaičiuojamas pagal formulę(42):

$$z = \frac{W}{\sqrt{\frac{N_r(N_r+1)(2N_r+1)}{6}}} \quad (42)$$

H_0 atmetama, jei $z_{kritis} < |z|$, kur z_{kritis} yra kritinė W skirstinio reikšmė. Pagal tą patį skirstinį skaičiuojamos ir p-reikšmės.

Bandymai buvo atliekami dvejais aspektais:

- Algoritmo veikimas rezultatų konvergavimo atžvilgiu. Kiekvieno bandymo metu atliekami tik 2000 tikslo funkcijos įvertinimų. Iš šių rezultatų po kiekvieno funkcijos įvertinimo gauto geriausio rezultato brėžiami konvergavimo grafikai.
- Algoritmo veikimas ilgalaikio tikslumo atžvilgiu. Kiekvieno bandymo metu atliekami 500000 tikslo funkcijos įvertinimų. Konvergavimo grafikai nebuvo sudaryti dėl labai mažų geriausių rastų reikšmių ir jų skirtumo artėjant prie algoritmo užbaigimo kriterijaus.

Analizuojant algoritmų rezultatus su klasikinėmis tikslo funkcijomis nepastebėti reikšmingi konvergavimo ar gaunamų rezultatų skirtumai. Rezultatų vidurkių skirtumus ir Wilcoxon testo rezultatus jiems matome 7 ir 8 lentelėje. Konvergavimo diagramos pateikiamos 1 priede. Su 500000 tikslo funkcijos vertinimų iqabc geriau pasirodė su Griewank funkcija, tačiau su mažesniu nei 10^{-16} skirtumu. Su 2500 tikslo funkcijos vertinimų iqsABC geriau pasirodė su Branin, Dixon ir Schaffer funkcijomis, tuo tarpu iqABC geresnius rezultatus davė Schwefel funkcijai.

7 lentelė. iqsABC algoritmo efektyvumas su klasikinėmis tikslo funkcijomis kai $D = 30$ (500000 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Statistiškai geresnis algoritmas	Atsakymo vidurkio skirtumas ABC-sABC	p reikšmė
Ackley	-	−0,000 000	0,051 9
Branin	-	−0	1,000 0
Camel	-	0,000 000	0,479 5
Dixon	-	0,689 147	0,382 0
Griewank	iqabc	−0,000 000	0,019 6
Rastrigin	-	0	1,000 0
Rosenbrock	-	0,001 196	0,416 5
Schaffer	-	−0,000 000	0,926 3
Schwefel	-	0,000 000	1,000 0
Sphere	-	0,000 000	0,749 9

8 lentelė. iqsABC algoritmo efektyvumas su klasikinėmis funkcijomis kai $D = 30$ (2500 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Statistiškai geresnis algoritmas	Atsakymo vidurkio skirtumas ABC-sABC	p reikšmė
Ackley	-	0,039 341	0,893 6
Branin	iqsabc	0,000 004	0,035 0
Camel	-	-0,000 000	0,797 1
Dixon	iqsabc	394 271,844 461	0,030 0
Griewank	-	-0,002 681	0,643 5
Rastrigin	-	-0,414 406	0,184 6
Rosenbrock	-	-0,963 613	0,797 1
Schaffer	iqsabc	0,000 326	0,035 0
Schwefel	iqabc	-1 188,677 962	0,000 0
Sphere	-	-1,239 227	0,909 9

Išsamesnei analizei pasitelktos papildomos tikslo funkcijos [CLZ⁺14] (žr. 9. lentelę). Funkcijos nuo f_{10} iki f_{15} yra sudėtinės, t.y. $F(x) = f(z_1) + \dots + f(z_n)$, kur n yra sudedamųjų funkcijų kiekis, o z yra kintamųjų x poaibis, kur x kintamieji paslinkti, jog sprendinių ieškojimas vykėtų intervale $[-100,100]^D$, o globalų minimumą duodantys kintamieji būtų intervale $[-80,80]^D$. Šis paslinkimas naudojamas ir likusioms funkcijoms. Šių funkcijų pritaikymui naudota pateikto šaltinio [CLZ⁺14] funkcijų implementacija C kalba, į kurią realizuoti dirbtinių bičių šeimų algoritmai kreipėsi per Python ctypes biblioteką.

9 lentelė. Papildomos tikslo funkcijos optimizavimo uždaviniais

Nr.	Funkcija	Nr.	Funkcija
f_1	Lenkto cigaro	f_9	Schaffer f_6
f_2	Disko	f_{10}	Elipsės su f_8
f_3	Weierstrass	f_{11}	f_3 su f_8 ir f_9
f_4	Schwefel	f_{12}	f_5, f_6, f_8 , Ackley, Schwefel
f_5	Katsuura	f_{13}	Elipsės su Rosenbrock ir f_1
f_6	HappyCat	f_{14}	Schwefel, Rastrigin, elipsės
f_7	HGBat	f_{15}	f_7, f_3 , Rastrigin, elipsės
f_8	Griewank + Rosenbrock		

Tiriamų algoritmų atžvilgiu šioms papildomoms funkcijoms atitinkamai naudojami kitokie parametrai (žr. 10 lentelę). Algoritmų gaunamų rezultatų skirtumai pateikti 11 ir 12 lentelėse. Su 1500 tikslo funkcijos įvertinimų iqsABC algoritmas pranašesnis su f_1, f_3, f_6, f_{13} funkcijomis, tačiau matome ir funkcijų, kurios statistiškai galėtų būti laikomos ir iqABC naudai, pvz. f_9 . Su 500000 tikslo funkcijos vertinimų iqABC buvo pranašesnis su f_3 papildoma funkcija, tuo tarpu iqsABC geriau pasirodė su f_5 funkcija.

10 lentelė. iqsABC algoritmo vertinime naudojamos parametrų reikšmės papildomoms funkcijoms

Kintamasis	Prasmė	Reikšmė
CS	Bičių šeimos dydis (pradinių sprendinių kiekis)	20
D	Dimensijos (kintamųjų kiekis)	30
l	Sprendinio tikrinimo riba	$\frac{CS \times D}{2}$
$l_{geriausias}$	Sprendinio tikrinimo riba geriausiam sprendiniui	$CS \times D$
-	Tikslo funkcijų vertinimo riba	1500, jei $D = 30$ 500, jei $D = 10$
-	Bandymų kiekis	20

11 lentelė. iqsABC algoritmo efektyvumas su papildomomis tikslo funkcijomis, kai $D = 30$ (500000 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Statistiškai geresnis algoritmas	Atsakymo vidurkio skirtumas iqABC-iqsABC	p reikšmė
f1	-	-358,247	0,120
f2	-	-5 679,153	0,213
f3	iqabc	-0,199	0,039
f4	-	-4,003	0,504
f5	iqsabc	0,066	0,043
f6	-	0,000	0,992
f7	-	-0,008	0,245
f8	-	-0,717	0,781
f9	-	-0,023	0,719
f10	-	-37 376,602	0,299
f11	-	0,374	0,254
f12	-	-6,637	0,959
f13	-	6,016	0,382
f14	-	-2,291	0,079
f15	-	-4,671	0,057
f14	-	-2,291	0,079

12 lentelė. iqsABC algoritmo efektyvumas su papildomomis funkcijomis kai $D = 30$ (1500 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Statistiškai geresnis algoritmas	Atsakymo vidurkio skirtumas iqsABC-iqsABC	p reikšmė
f1	iqsabc	105 413 763,522	0,030
f2	-	7 375,536	0,526
f3	iqsabc	2,675	0,006
f4	-	-17,516	0,970
f5	-	0,189	0,455
f6	iqsabc	0,087	0,014
f7	-	0,178	0,412
f8	-	-227,336	0,455
f9	-	-0,180	0,062
f10	-	-373 565,302	0,526
f11	-	-8,551	0,654
f12	-	-106,109	0,433
f13	iqsabc	27,769	0,023
f14	-	-9,343	0,970
f15	-	24,251	0,218

Taigi bandydami gerinti iqsABC algoritmą pagal spietimosi lizdo paieškos logiką negavome statistiškai reikšmingo geresnio ar blogesnio rezultato išbandę naująjį iqsABC algoritmą.

Pabandykime atskirai patikrinti prielaidą, jog kintančios sprendinių tikrinimo ribos gali padėti bitėms greičiau aptikti geresnį sprendinį. Pasirenkime ABC algoritmu, tačiau stebinčių bičių žingsnio tikimybei naudokime (34), o žvalgų žingsnyje naudokime ribą iš formulės (38). Pavadiname šį algoritmą besispiečiančios dirbtinės bičių šeimos algoritmu (angl. Swarming Bee Colony Algorithm arba sABC). Atlikus bandymus ($D = 30$, tikslo funkcijų įvertinimų po 1500) ir palyginus ABC su sABC, iš tiesų matome, jog sABC statistiškai turi reikšmingą pranašumą su Ackley ir Rastrigin funkcija bei papildomomis f_4 , f_5 , f_6 , f_7 funkcijomis (žr. 13 lentelę). Konvergavimo grafikai pateikiami 3 priede, o pilna Wilcoxon testo lentelė pateikiama 2 priede.

13 lentelė. sABC algoritmo efektyvumas kai $D = 30$ (1500 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Statistiškai geresnis algoritmas	Atsakymo vidurkio skirtumas ABC-sABC	p reikšmė
Ackley	sabc	1,643	0,000
Rastrigin	sabc	45,017	0,002
f4	sabc	840,895	0,015
f5	sabc	0,643	0,010
f6	sabc	1,198	0,002
f7	sabc	71,969	0,000

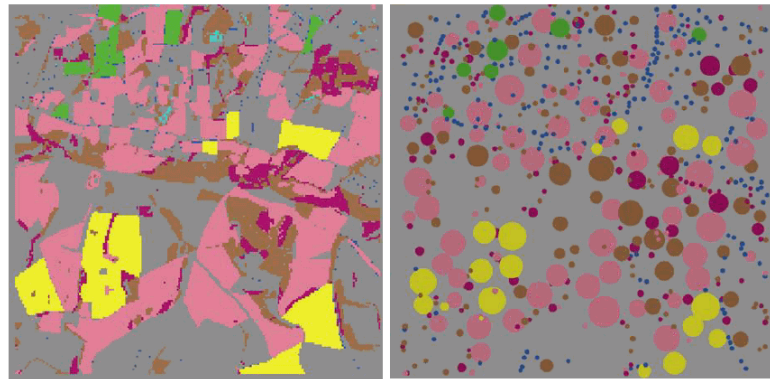
Taigi, nepavyko sukurti greičiau į minimumą konverguojančio algoritmo už iqsABC, tačiau

pasiūlyti iqsABC ir sABC algoritmai labiau tinka kuriamam spietimosi procesui tobulinamame modelyje, kadangi jie remiasi bičių naujo lizdo paieškos procesu.

3.3. Spietimosi proceso įgyvendinimas

Pirmas pasiūlytas iqsABC algoritmas remiasi paties geriausio sprendinio tikrinimo riba, kas neatitinka aptartų žinių apie bičių spietimąsi. Šiuo atveju geriau tinka sABC algoritmas, kurį būtų galima su keliais pakeitimais taikyti dalykinėje srityje, net jei ir jis konverguoja iki minimumo lėčiau. Pirmiausia pradėkime nuo tikslo funkcijos.

Tikslo funkcija šiuo atveju turėtų nurodyti geriausią bičių šeimos naujo lizdo vietą pateiktoje aplinkoje. Iš ankstesnio skyriaus prisimename modelio landšafto modelį. Išsamiau panagrinėjus jungtinio modelio kodą iš Bumble-Beehave dalies, matome, jog pateikto landšafto informacija atvaizduojama žmogui lengviau suprantamu pavidalu, tačiau iš tikrųjų šie duomenys turi persidengiančius gėlių ir kitų augalų laukus. Šiems laukams nurodomas jų užimamas plotas ir koordinatės, žiedadulkių ir nektaro kiekis, žydėjimo laikas ir t.t.



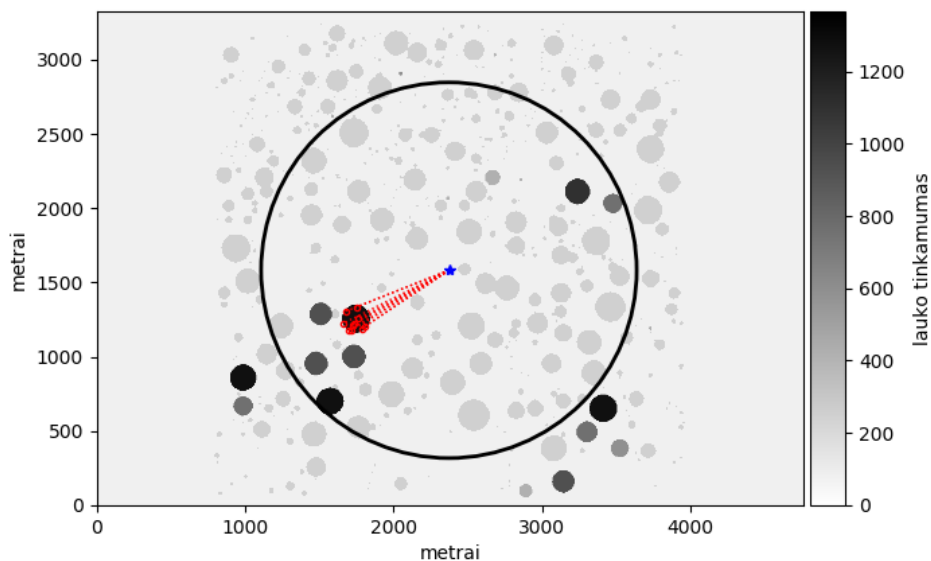
9 pav. Landšafto žemėlapis (kairėje) ir jo informacijos interpretacija modelyje

Sukurta tikslo funkcija, skaičiuojanti maistingumo reikšmę duotam landšafto žemėlapio pikseliui $x = (x_x, x_y)$ (43). Be maistingumo reikšmės matome, jog tikslo funkcijoje naudojamos sinuso ir kosinuso funkcijos. Kadangi landšafto informacija yra vienoda kiekvieno gėlių lauko plote, ši variacija padeda algoritmui išvengti situacijų, kai visi duoti sprendiniai yra vienodi. Papildomai naudojamas labai mažas skaičius ϵ , jog ši variacija nedarytų didelės įtakos galutiniam atsakymui. Duotai landšafto informacijai apskaičiuotos tikslo funkcijos reikšmės vaizdiškai pateiktos 4 priede.

$$f(x) = \epsilon \sin(x_x) \cos(x_y) - x_{\text{žydėjimo trukmė}}(x_{\text{žiedadulkės}} x_{\text{žiedadulkių koncentracija}} + x_{\text{nektaras}} x_{\text{nektaro maistinė vertė}}) \quad (43)$$

Kadangi naująjį algoritmą taikome dalykinėje srityje, o ne bendroms optimizavimo užduotims, turime pakeisti algoritmo parametrus ir veikimą, jog šis tiksliau atitiktų tikrų bičių elgesį.

Iš spietimosi stebėjimų [See10] žinome, jog tik dalis bičių šeimos ieško naujo lizdo spietimosi metu, apie 300-500 bičių dalyvauja procese, todėl taikyme naudosime $CS = 500$. Kadangi bitės gali nuskristi tik ribotą atstumą, visuose žingsniuose bitės sprendinių ieškos mažesnėje sprendinių



10 pav. Rezultatai gauti pritaikius iqsABC su 500 bičių ir 10000 tikslo funkcijos tikrinimų.

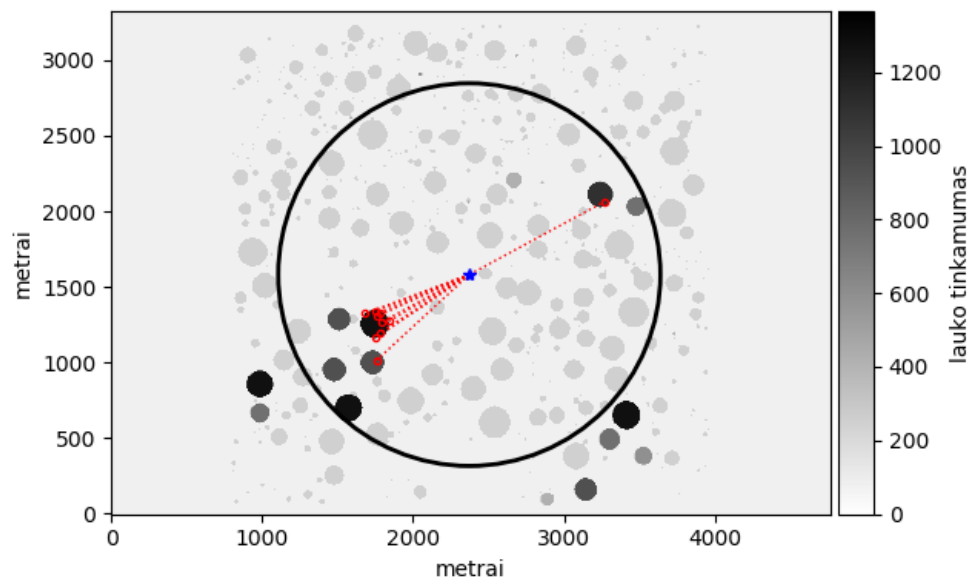
aibėje, sudarytoje iš viso žemėlapių pikselių poaibio, kur pikseliai yra nutolę nuo pradinio lizdo pikselio ne daugiau nei per fiksuotą atstumą d .

Panaudoję šią tikslo funkciją su iqsABC algoritmu gauname naują bičių lizdo vietą, į kurią spiesis bitės. Įvykdyti keli bandymai su skirtingais parametrais tikrinant, ar bitės išsirinks palankiausią augalų lauką nektaro ir žiedadulkių atžvilgiu. Rezultatai pateikti 10-ame ir 11-ame paveikslėliuose, čia apskritimu nurodomas paieškos plotas pagal parametą d , o kiekvienas mažesnis raudonas apskritimas yra algoritmo rastas sprendinys kiekvieno bandymo metu. Besispiečiančios šeimos pradinės koordinatės nurodytos su mėlyna žvaigžde. 10-ame paveikslėlyje algoritme naudojamos pradinės šeimos kiekis yra 500, tikslo funkcijų įvertinimų riba 10000. Šiuo atveju šeimos spiečiasi prie geriausiai įvertinto maisto šaltinio.

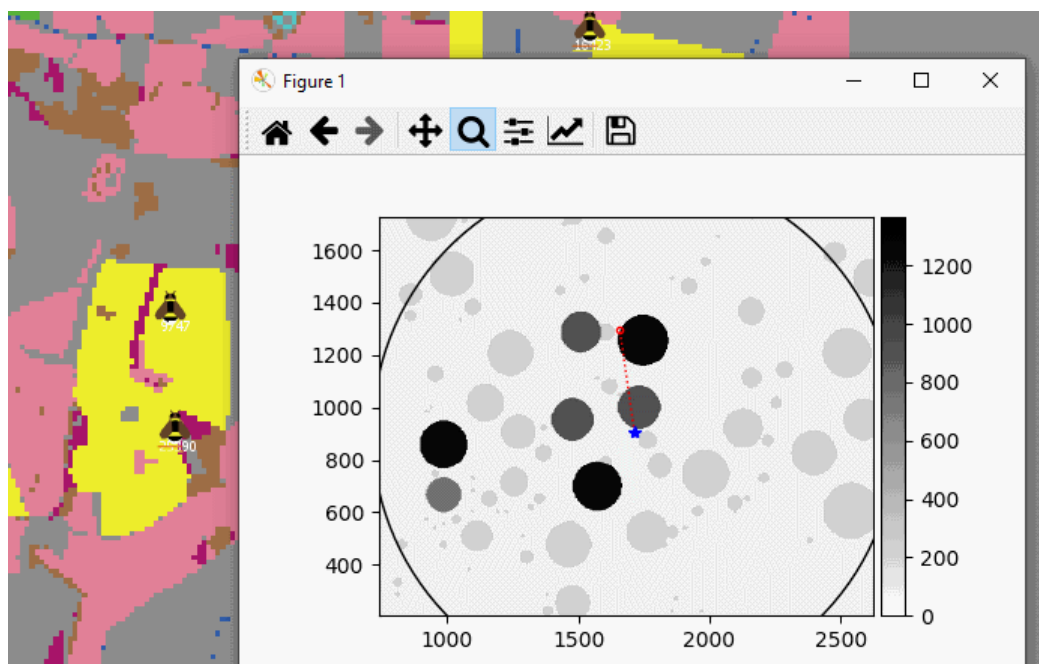
Realybėje spietimosi metu gali pasikeisti oro sąlygos. Jeigu spietimosi metu pradeda lyti, bitės keliaus į lizdą anksčiau, išnagrinėjusios mažiau vietų lizdui. Tokių elgesį galime atkartoti sumažinę tikslo funkcijos įvertinimų kiekį. Algoritmo veikimas su 100 bičių ir tik 1000 tikslo funkcijos įvertinimų pavaizduotas 11 paveikslėlyje, matome, jog bitės pasirenka ir kitus maisto šaltinius, įskaitant esančius arčiau lizdo.

Išbandykime šią tikslo funkciją ir algoritmą kartu su likusiu jungtiniu modeliu. Patenkinus vieną iš spietimosi sąlygų (6), bičių šeima spiečiasi ir ieško naujo lizdo. Iš modelio sABC algoritmas gauna pradines bičių šeimos koordinatas ir grąžina naujo lizdo vietą. Pritaikius naująjį algoritmą jungtiniame modelyje matome, jog besispiečiančios bitės iš pasiekiamų laukų pasirenka geriausiai tikslo funkcijos įvertintą augalų lauką (žr. 12 pav.). Taip naujasis sABC algoritmas įgyvendina naująjį spietimosi procesą patobulintame modelyje.

Bitės spiečiasi sudarydamos naują šeimą rastoje vietoje, palikdamos dalį bičių senojoje šeimoje pagal santykius, naudojamus BEEHAVE modelyje. Kartu spiečius išsineša ir reikalingą mais-



11 pav. Rezultatai gauti pritaikius iqsABC su 100 bičių ir 1000 tikslo funkcijos tikrinimų.



12 pav. Bičių šeimos spietimasis jungtiniame modelyje

to atsargų kiekį irgi pagal BEEHAVE modelį, tačiau nauja bičių šeima jau modeliuojama kitoje pozicijoje ir turi prieigą prie kitų maisto šaltinių.

Realybėje bitės atsižvelgtų į papildomus kriterijus, ieškodamos tinkamų medžių drevių arba ieškodamos kitų, anksčiau apleistų, avilių. Gavus išsamesnių landšafto pavyzdžių būtų įmanoma praplėsti tikslo funkciją ir ją pritaikyti išsamesnei informacijai.

Sukurto modelio išeigos kodas patalpintas GitHub kodo talpykloje [Vég20].

Rezultatai ir išvados

Darbo metu išanalizavus esamus bičių populiacijos modelius pastebėta, jog jie neapima visų bitininkystės ir bičių šeimos procesų, tačiau tarp jų yra daug procesų apjungiantis BEEHAVE modelis. Šis modelis gali simuliuoti tik vieną šeimą. Rastas panašus Bumble-BEEHAVE modelis, galintis modeliuoti kelias kamanių šeimas. Jungtinis modelis gautas kelių kamanių šeimų modeliavimą papildžius naminių bičių procesais. Tai buvo atlikta perrašant NetLogo kodą, jog Bumble-Beehave metodai atitiktų naminių bičių, o ne kamanių procesus. Taip buvo sukurtas naujas modelis galintis modeliuoti kelias naminių bičių šeimas vienu metu.

Sukurtas jungtinis modelis toliau buvo tobulinamas procesais, kurie remiasi keliomis arba sukuria naujas bičių šeimas. Įgyvendinti du nauji procesai: bičių migravimas ir bičių spietimasis. Bičių spietimasis jau egzistavo BEEHAVE modelyje, tačiau jis nebuvo susijęs su lizdo pakeitimu ir modeliavo tik vieną šeimą (arba naująją, arba senąją). Pristatytas proceso pakeitimas leidžia modeliuoti spiečiaus naujo lizdo pasirinkimą pagal bičių lizdo ieškojimo ir sprendimo priėmimo procesus.

Kuriant spietimosi procesą sukurti ir išbandyti nauji dirbtinių bičių šeimų algoritmai, paremti natūraliu bičių spietimosi procesu. Vienas iš jų (iqsABC) buvo sukurtas iqABC algoritmo pagrindu, kitas (sABC) ABC pagrindu. Nebuvo statistiškai įrodyta, jog pasiūlyti algoritmai pateiktų geresnius rezultatus nei iqABC algoritmas, tačiau parodyta, jog spietimosi principus pritaikius paprastam ABC algoritmui gauname statistiškai greičiau konverguojantį algoritmą už ABC algoritmą. Jungtinis modelis sėkmingai papildytas sABC algoritmu, sudarius tikslo funkciją iš aplinkos modelio informacijos ir pakeitus sprendinių aibę.

Iš gautų rezultatų daromos šios išvados:

1. Iš dabartinių BEEHAVE ir Bumble-BEEHAVE bitininkystės procesų modelių įmanoma sudaryti kelias naminių bičių šeimas simuliuojantį modelį.
2. Kelias bičių šeimas apimantis modelis suteikia galimybę jį toliau tobulinti pridedant tarpšeimyninius bičių procesus, tačiau šie procesai tarpusavyje gali būti reikšmingai susiję. Pagal empirines žinias ir tyrimus bičių migracija keičiasi pagal varozės procesus.
3. Dirbtinių bičių šeimų algoritmai gali būti sėkmingai tobulinami bičių spietimosi principu, o ne vien maisto paieškos principu.
4. Dirbtinių bičių šeimų algoritmai sėkmingai pritaikomi bitininkystės modeliavimo srityje.

Darbo metu sukurtą modelį galima būtų tobulinti literatūros analizės metu išskirtu varozės plitimo tarp šeimų procesu ir išsamesniu klimato įtakos bitėms modeliu. Pritaikius varozės plitimo procesus būtų galima patobulinti pristatyto modelio bičių migracijos procesą.

Sukurtą naują iqsABC algoritmą būtų galima išbandyti su kitais skirtingais parametrais ir nustatyti kitokius sprendinio tikrinimo kriterijus. Taip pat būtų galima palyginti naujus iqsABC ir sABC algoritmus su kitais dirbtinių bičių šeimos algoritmais. Pavyzdžiui, iš konvergavimo diagramų matome įdomų qABC ir sABC konvergavimo santykį, kuris nebuvo tirtas šiame darbe. Pristatytus algoritmus galima ištirti ir naudojant daugiau skirtingų tikslo funkcijų.

Šaltiniai

- [AAA⁺19] M. Allen, P. Antwi-Agyei, F. Aragon-Durand, M. Babiker ir k.t. Technical summary: global warming of 1.5 °c. an ipcc special report on the impacts of global warming of 1.5 °c above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty, 2019. URL: <http://pure.iiasa.ac.at/id/eprint/15716/>. Tikrinta 2020-04-25.
- [ABK19] Selcuk Aslan, Hasan Badem ir Dervis Karaboga. Improved quick artificial bee colony (iqabc) algorithm for global optimization. *Soft Computing*, 23, 2019.
- [BDO⁺91] Jerry J. Bromenshenk, Jiri Dosek, Gerald J. Olbu, Gloria Degrandi-Hoffman ir Stephan A. Roth. Pc beepop, an ecotoxicological simulation model for honey bee populations. *Environmental Toxicology and Chemistry*, 10(4):547–558, 1991.
- [BGT⁺14] Matthias A. Becher, Volker Grimm, Pernille Thorbek, Juliane Horn, Peter J. Kennedy ir Juliet L. Osborne. Beehave: a systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure. *Journal of Applied Ecology*, 51(2):470–482, 2014.
- [BOT⁺13] Matthias A. Becher, Juliet L. Osborne, Pernille Thorbek, Peter J. Kennedy ir Volker Grimm. Review: towards a systems approach for understanding honeybee decline: a stocktaking and synthesis of existing models. *Journal of Applied Ecology*, 50(4):868–880, 2013.
- [BPC⁺17] Célie Bordier, Maryline Pioz, Didier Crauser, Yves Le Conte ir Cédric Alaux. Should i stay or should i go: honeybee drifting behaviour as a function of parasitism. *Apidologie*, 48(3):286–297, 2017.
- [BTP⁺17] Matthias Becher, Grace Twiston-Davies, Tim Penny, Dave Goulson, Ellen Rotheray ir Juliet Osborne. Bumble-beehave: a systems model for exploring multifactorial causes of bumblebee decline at individual, colony, population and community level. *Journal of Applied Ecology*, 55, 2017.
- [BWZ16] M.I. Betti, L.M. Wahl ir M. Zamir. Age structure is critical to the population dynamics and survival of honeybee colonies. *Royal Society Open Science*, 3, 2016.
- [CLZ⁺14] Q Chen, B Liu, Q Zhang, JJ Liang, PN Suganthan ir BY Qu. Problem definition and evaluation criteria for cec 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, China and Nanyang Technological University, Singapore, Technical report*, 178, 2014.
- [DC05] Gloria Degrandi-Hoffman ir R. Curry. The population dynamics of varroa mites in honey bee colonies: part i - the varroapop program. *American Bee Journal*, 145:592–595, 2005.

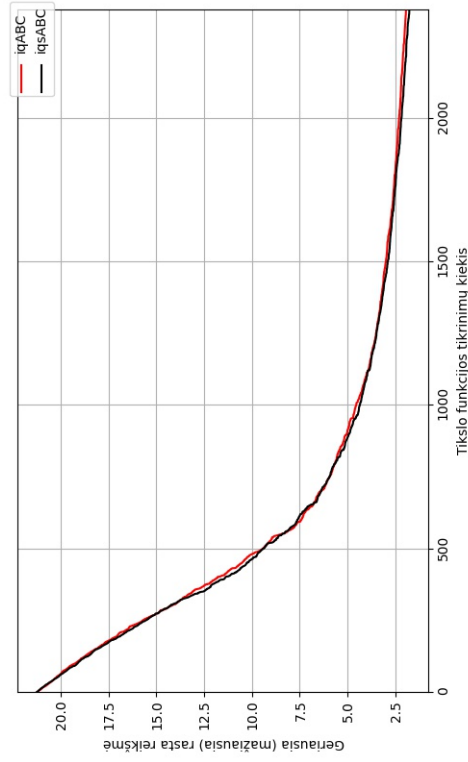
- [DFG15] Mehmet Ali Döke, Maryann T. Frazier ir Christina M. Grozinger. Overwintering honey bees: biology and management. *Current opinion in insect science*, 10:185–193, 2015.
- [DRL⁺89] G. DeGrandi-Hoffman, S.A. Roth, G.L. Loper ir E.H. Erickson. Beepop: a honeybee population dynamics simulation model. *Ecological Modelling*, 45(2):133–150, 1989.
- [EFS15] EFSA Panel on Plant Protection Products and their Residues (PPR). Statement on the suitability of the beehave model for its potential use in a regulatory context and for the risk assessment of multiple stressors in honeybees at the landscape level. *EFSA Journal*, 13(6):4125, 2015.
- [EFS16] EFSA Panel on Animal Health and Welfare (AHAW). Assessing the health status of managed honeybee colonies (healthy-b): a toolbox to facilitate harmonised data collection. *EFSA Journal*, 14(10):e04578, 2016.
- [EFS18] European Food Safety Authority (EFSA). Peer review of the pesticide risk assessment for bees for the active substance clothianidin considering the uses as seed treatments and granules. *EFSA Journal*, 16(2):e05177, 2018.
- [FNF⁺15] Nadège Forfert, Myrsini Natsopoulou, Eva Frey, Peter Rosenkranz, Robert Paxton ir Robin Moritz. Parasites and pathogens of the honeybee (*apis mellifera*) and their influence on inter-colonial transmission. *PloS one*, 10:e0140337, 2015.
- [FO77] Hiromi Fukuda ir Takeshi Ohtani. Survival and life span of drone honeybees. *Researches on Population Ecology*, 19(1):51–68, 1977.
- [FS06] Nina Fefferman ir P. Starks. A modeling approach to swarming in honey bees (*apis mellifera*). *Insectes Sociaux*, 53:37–45, 2006.
- [FS68] Hiromi Fukuda ir Shôichi F. Sakagami. Worker brood survival in honeybees. *Population Ecology*, 10(1):31–39, 1968.
- [ID16] Maxcy Iv ir Keith Delaplane. Distance between honey bee *apis mellifera* colonies regulates populations of *varroa destructor* at a landscape scale. *Apidologie*, 48, 2016.
- [YYY19] Yasuhiro Yamada, Toshiro Yamada ir Kazuko Yamada. A mathematical model to estimate the seasonal change in apparent longevity of bee colony. *Scientific Reports*, 9:Article number: 4102 (2019), 2019.
- [JLC⁺17] Antoine Jacques, Marion Laurent, EPILOBEE Consortium, Magali Ribière-Chabert, Mathilde Saussac, Stéphanie Bougeard, Giles E. Budge, Pascal Hendrikx ir Marie-Pierre Chauzat. A pan-european epidemiological study reveals honey bee colony survival depends on beekeeper education and disease control. *PLOS ONE*, 12(3):1–17, 2017.
- [JMB06] Stefan Janson, Martin Middendorf ir Madeleine Beekman. Searching for a new home—scouting behavior of honeybee swarms. *Behavioral Ecology*, 18(2):384–392, 2006.

- [Kar00] Pranciškus Karosas. *Bitininkystė kiekvienam*. Ūkininko patarėjas, Kaunas, Lietuva, 2000. 14 p.
- [KB07] Dervis Karaboga ir Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007.
- [KBM13] David S. Khoury, Andrew B. Barron ir Mary R. Myerscough. Modelling food and population dynamics in honey bee colonies. *PLOS ONE*, 8(5):1–7, 2013.
- [KG14] Dervis Karaboga ir Beyza Gorkemli. A quick artificial bee colony (qabc) algorithm and its performance on optimization problems. *Applied Soft Computing*, 23:227–238, 2014.
- [Mar01] Stephen J. Martin. The role of varroa and viral pathogens in the collapse of honeybee colonies: a modelling approach. *Journal of Applied Ecology*, 38(5):1082–1093, 2001.
- [MBB11] Alfonsas Misevičius, Jonas Blonskis ir Vytautas Bukšnaitis. Bičių spiečių imitavimas sprendžiant optimizavimo uždavinius. *Informacijos mokslai*, 56:163–173, 2011.
- [MLK⁺15] Marjan Mernik, Shih-hsi Liu, Dervis Karaboga ir Matej repinšek. On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. *Information Sciences*, 291:115–127, 2015.
- [MTD⁺19] Robyn Manley, Ben Temperton, Toby Doyle, Daisy Gates, Sophie Hedges, Michael Boots ir Lena Wilfert. Knock-on community impacts of a novel vector: spillover of emerging dwv-b from varroa-infested honeybees to wild bumblebees. *Ecology Letters*, 2019.
- [Oli15] Randy Oliver. Understanding colony buildup and decline – part 7b minimizing swarming. <http://scientificbeekeeping.com/understanding-colony-buildup-and-decline-part-7b>, 2015. Tikrinta 2020-04-24.
- [PC98] K. J. Pfeiffer ir K. Crailsheim. Drifting of honeybees. *Insectes Sociaux*, 45(2):151–167, 1998.
- [RBT⁺17] Jack Rumkee, Matthias Becher, Pernille Thorbek ir Juliet Osborne. Modeling effects of honeybee behaviors on the distribution of pesticide in nectar within a hive and resultant in-hive exposure. *Environmental Science & Technology*, 51, 2017.
- [SB99] Thomas Seeley ir Susannah Buhrman. Group decision making in swarms of honey bees. *Behavioral Ecology and Sociobiology*, 45:19–31, 1999.
- [SC07] Thomas Schmickl ir Karl Crailsheim. Hopomo: a model of honeybee intracolony population dynamics and resource management. *Ecological Modelling*, 204:219–245, 2007.
- [Sch81] H.-P. Schwefel. *Numerical optimization of computer models*. Wiley&Sons, Chichester, 1981.

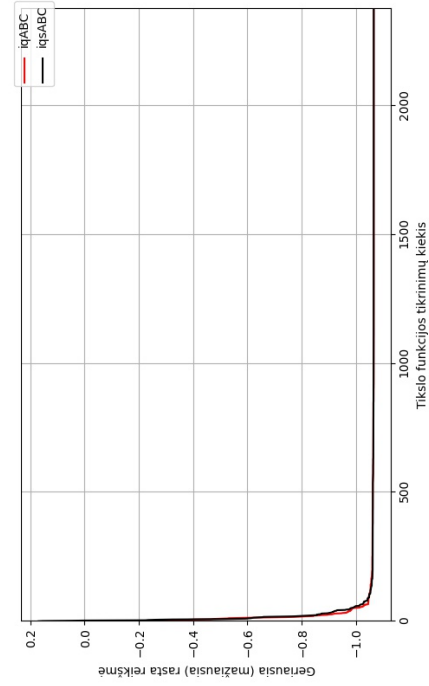
- [SCT⁺17] Matthew Switanek, Karl Crailsheim, Heimo Truhetz ir Robert Brodschneider. Modeling seasonal effects of temperature and precipitation on honey bee winter mortality in a temperate climate. *Science of The Total Environment*, 579:1581–1587, 2017.
- [See10] T.D. Seeley. *Honeybee Democracy*. Princeton University Press, 2010.
- [SS15] Thomas Seeley ir Michael Smith. Crowding honeybee colonies in apiaries can increase their vulnerability to the deadly ectoparasite varroa destructor. *Apidologie*, 46, 2015.
- [TRR15] David J. Torres, Ulises M. Ricoy ir Shanae Roybal. Modeling honey bee populations. *PLOS ONE*, 10(7):1–28, 2015.
- [Vég20] Vytautas Vėgelė. Bitininkystės procesų modeliavimo talpykla. https://github.com/vytautasvegele/vv_mtd, 2020. Tikrinta 2020-05-24.
- [VMS85] P. K. Visscher, R. A. Morse ir T. D. Seeley. Honey bees choosing a home prefer previously occupied cavities. *Insectes Sociaux*, 32(2):217–220, 1985.
- [Wil45] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.
- [Wui17] Romain Wuilbercq. Hive, an artificial bee colony optimization in python. <http://scientificbeekeeping.com/understanding-colony-buildup-and-decline-part-7b>, 2017. Tikrinta 2020-05-24.

1 priedas

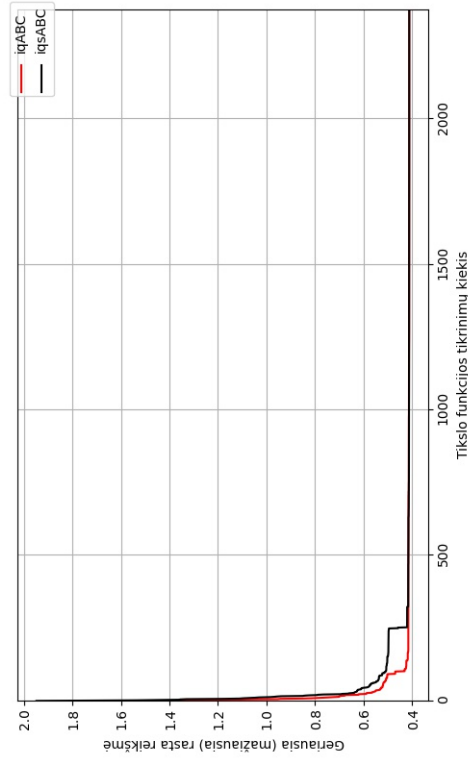
Konvergavimo grafikai su iqABC ir iqsABC



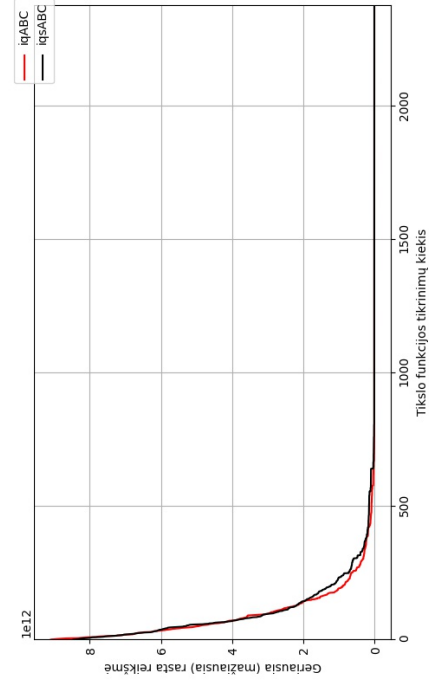
13 pav. Algoritimų konvergavimas su Ackley tikslo funkcija



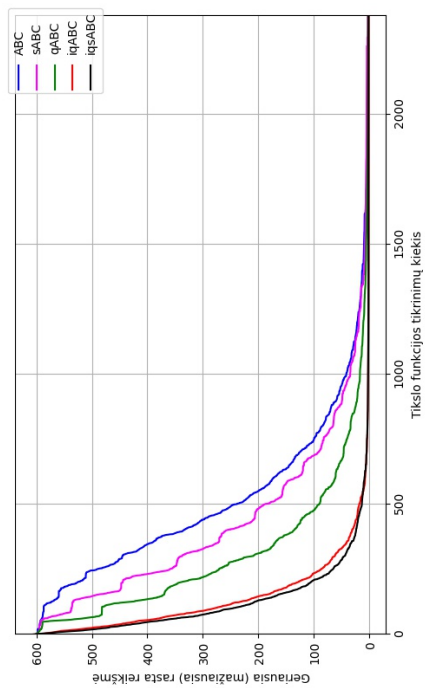
15 pav. Algoritimų konvergavimas su kupranugario tikslo funkcija



14 pav. Algoritimų konvergavimas su Branin tikslo funkcija



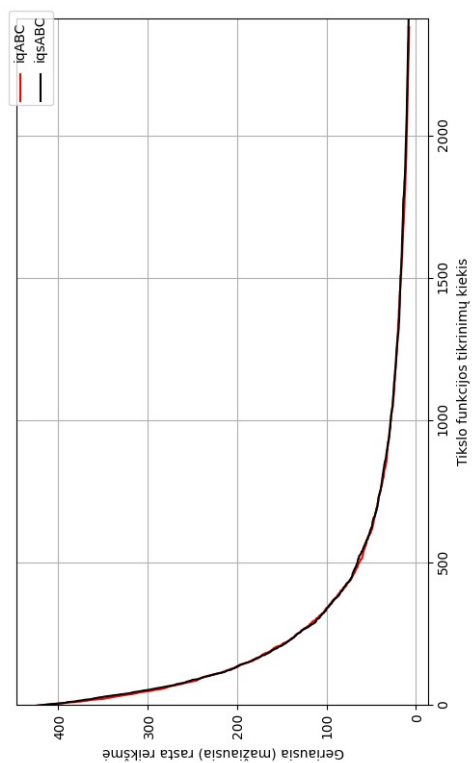
16 pav. Algoritimų konvergavimas su Dixon tikslo funkcija



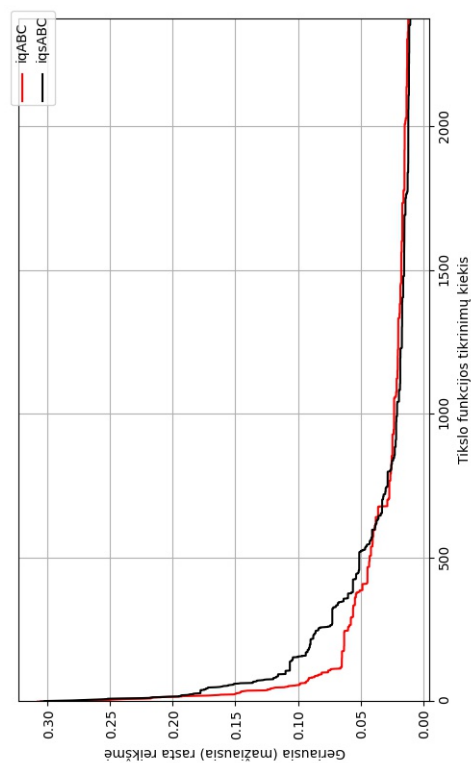
17 pav. Algoritmų konvergavimas su Griewank tikslo funkcija



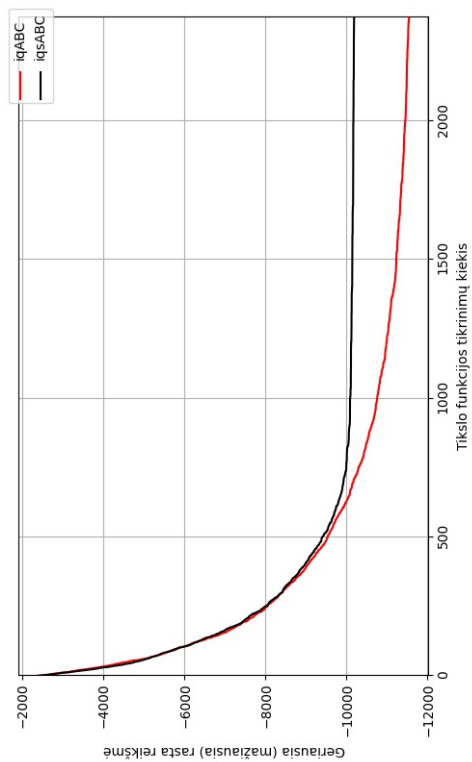
19 pav. Algoritmų konvergavimas su Rosenbrock tikslo funkcija



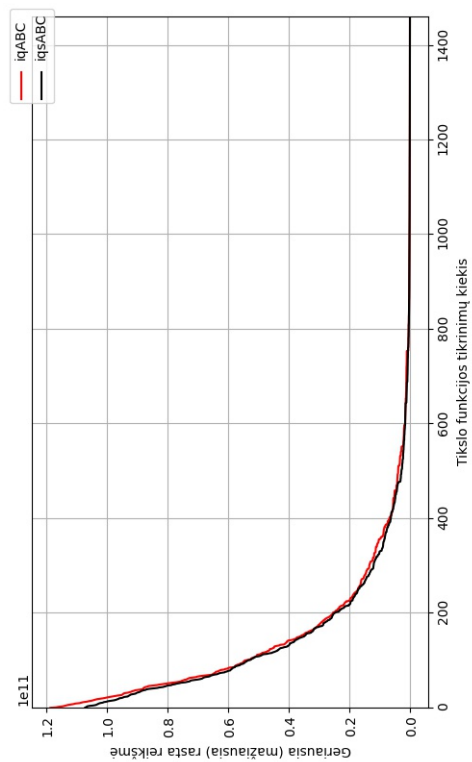
18 pav. Algoritmų konvergavimas su Rastrigin tikslo funkcija



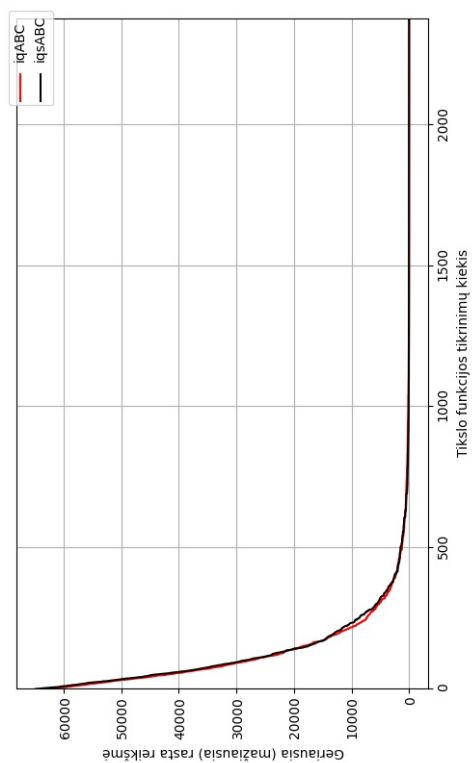
20 pav. Algoritmų konvergavimas su Schaffer tikslo funkcija



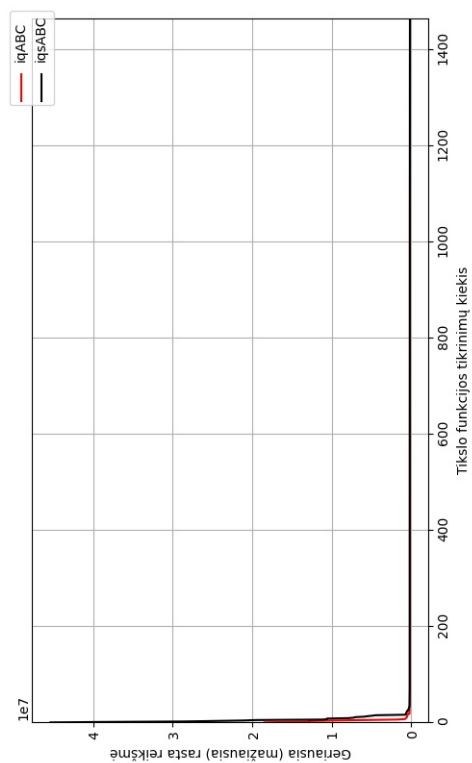
21 pav. Algoritmų konvergavimas su Schwefel tikslo funkcija



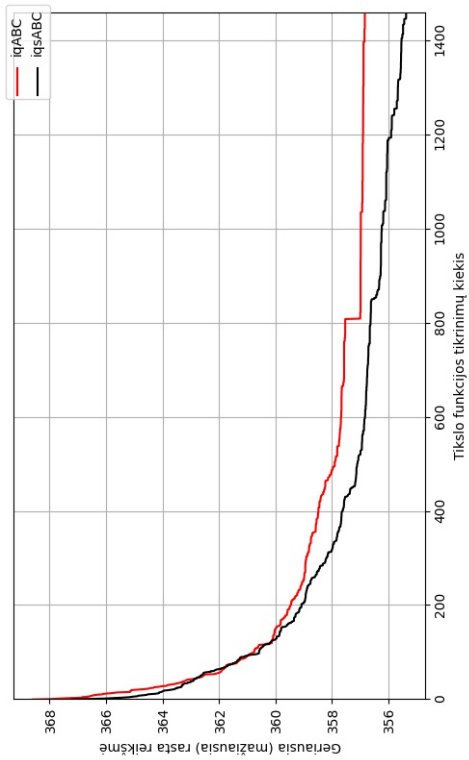
23 pav. Algoritmų konvergavimas su f_1 tikslo funkcija



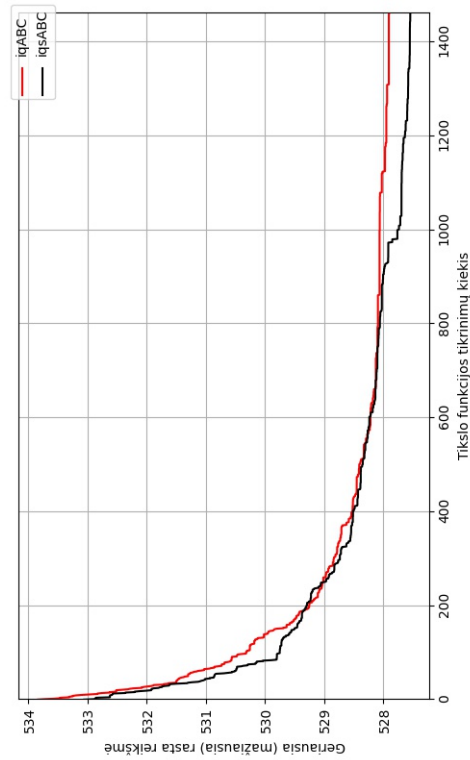
22 pav. Algoritmų konvergavimas su sferos tikslo funkcija



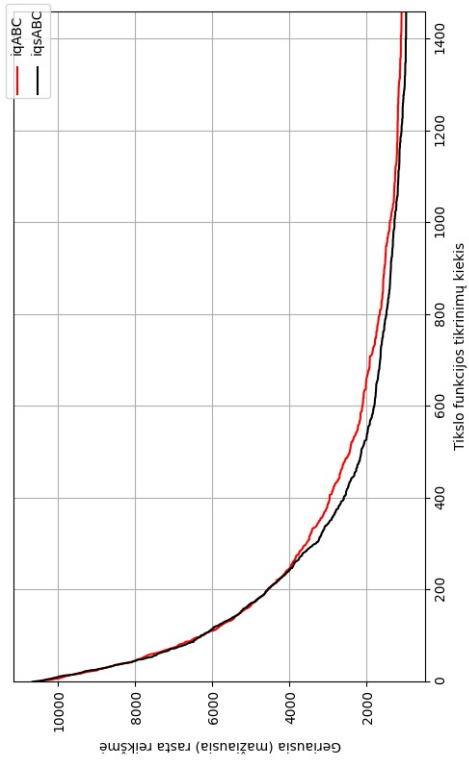
24 pav. Algoritmų konvergavimas su f_2 tikslo funkcija



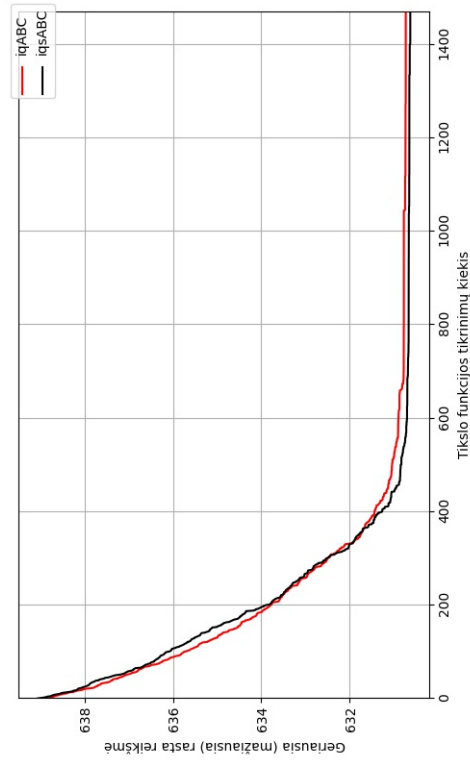
25 pav. Algoritmų konvergavimas su f_3 tikslo funkcija



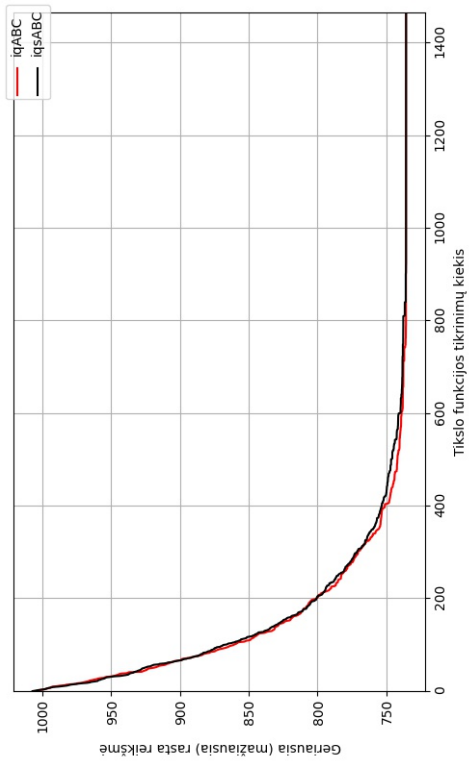
27 pav. Algoritmų konvergavimas su f_5 tikslo funkcija



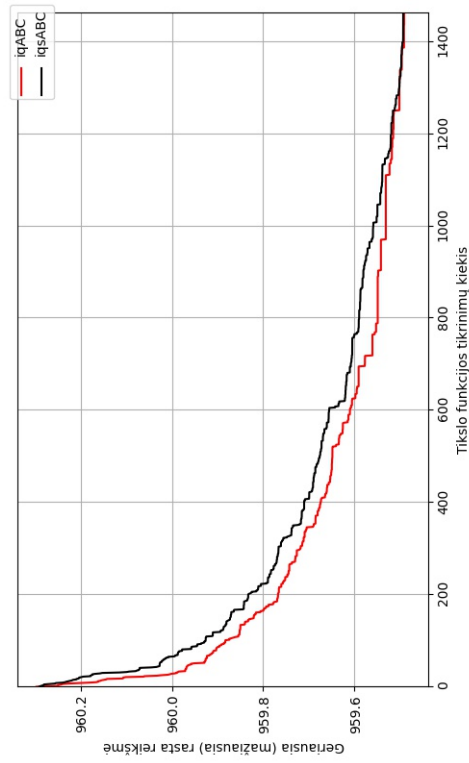
26 pav. Algoritmų konvergavimas su f_4 tikslo funkcija



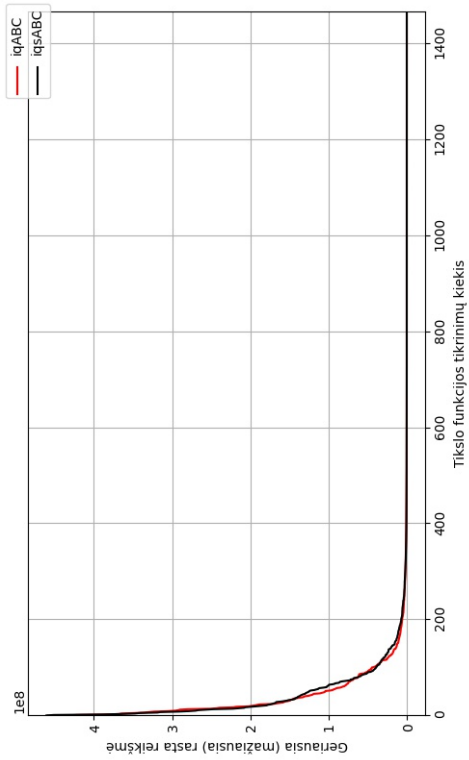
28 pav. Algoritmų konvergavimas su f_6 tikslo funkcija



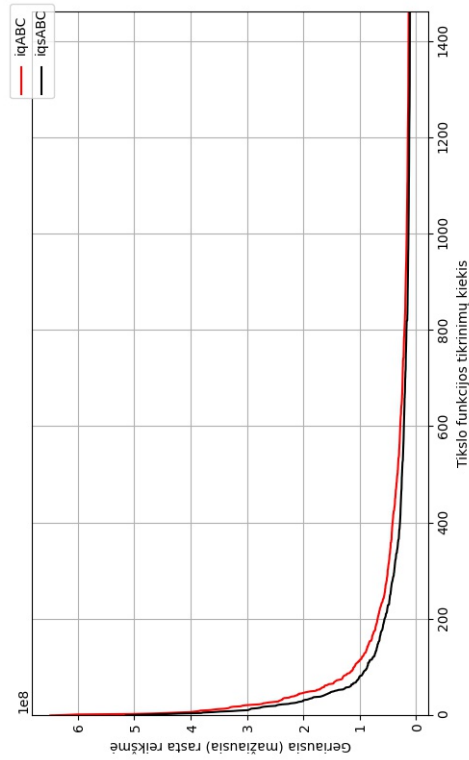
29 pav. Algoritmų konvergavimas su f_7 tikslo funkcija



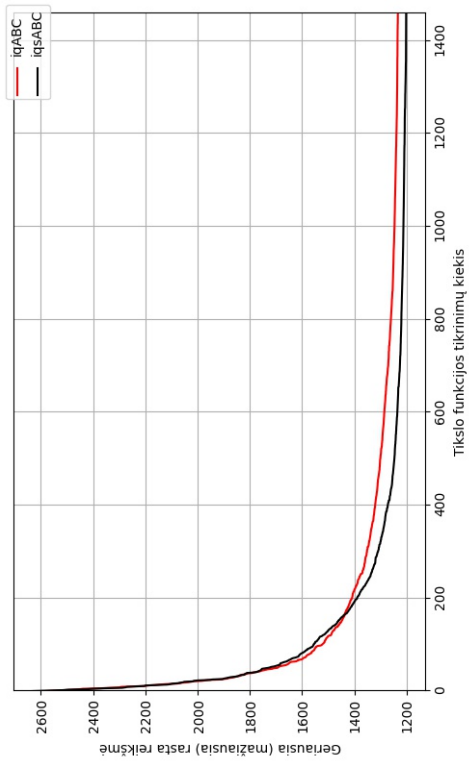
31 pav. Algoritmų konvergavimas su f_9 tikslo funkcija



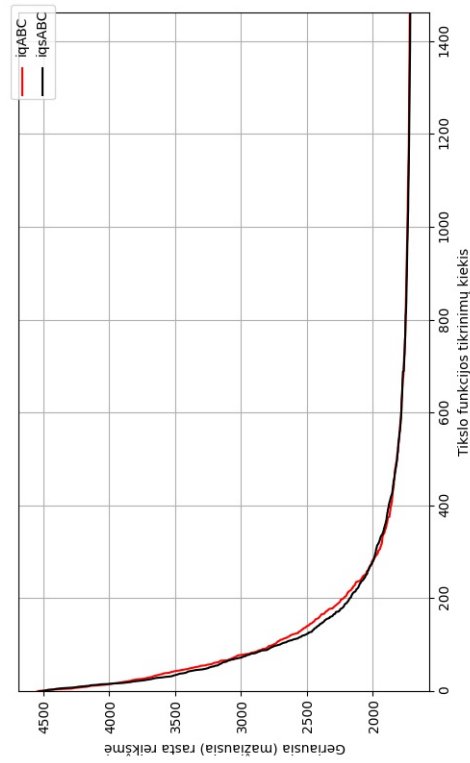
30 pav. Algoritmų konvergavimas su f_8 tikslo funkcija



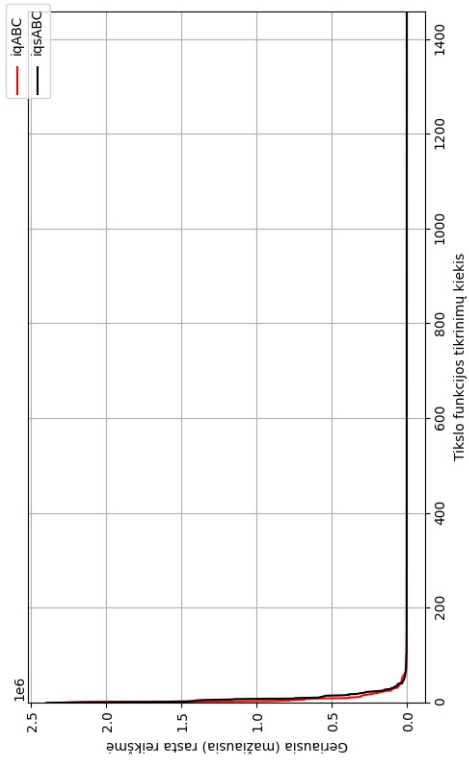
32 pav. Algoritmų konvergavimas su f_{10} tikslo funkcija



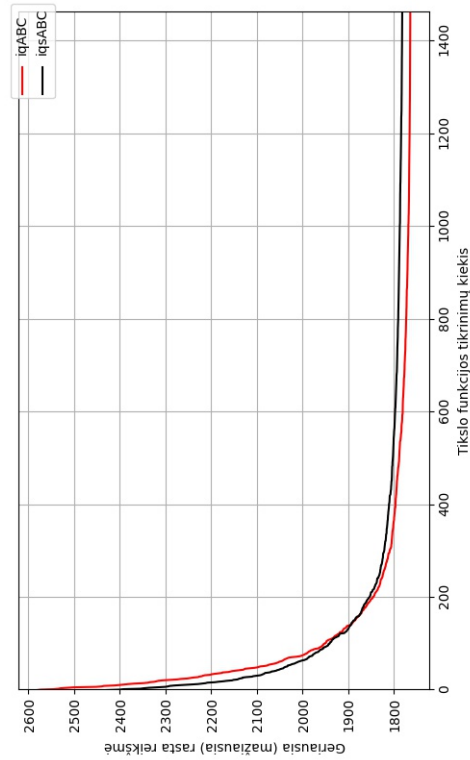
33 pav. Algoritmų konvergavimas su f_{11} tikslo funkcija



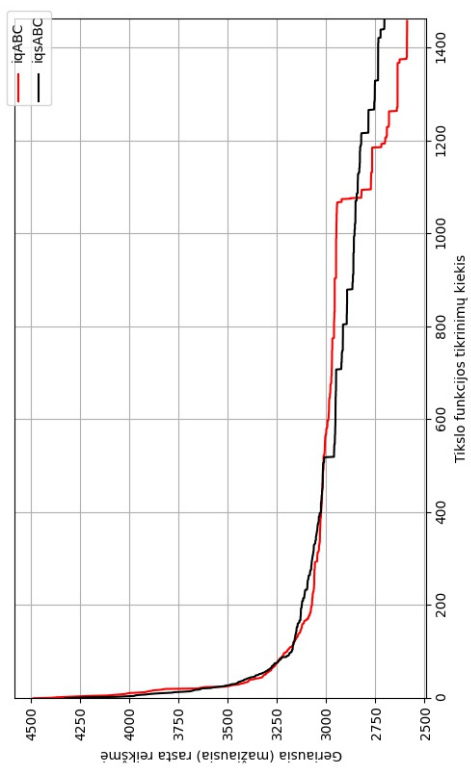
35 pav. Algoritmų konvergavimas su f_{13} tikslo funkcija



34 pav. Algoritmų konvergavimas su f_{12} tikslo funkcija



36 pav. Algoritmų konvergavimas su f_{14} tikslo funkcija



37 pav. Algoritmų konvergavimas su f_{15} tikslo funkcija

2 priedas

Kiti iqsABC ir sABC algoritmų tyrimo rezultatai

Paskutiniai rezultatai

14 lentelė. iqsABC algoritmo efektyvumas kai $D = 30$ (2500 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Algoritmas	Min	Max	Vidurkis
Ackley	iqABC	1,082	2,496	1,827
Ackley	iqsABC	0,784	2,734	1,788
Branin	iqABC	0,398	0,398	0,411
Branin	iqsABC	0,398	0,399	0,411
Camel	iqABC	-1,032	0,000	-1,032
Camel	iqsABC	-1,032	-1,032	-1,032
Dixon	iqABC	14 205,860	7 904 341,353	1 090 818,529
Dixon	iqsABC	43 717,752	1 400 632,952	696 546,685
Griewank	iqABC	1,024	1,186	1,105
Griewank	iqsABC	1,027	1,153	1,108
Rastrigin	iqABC	3,857	10,849	7,824
Rastrigin	iqsABC	2,902	11,345	8,238
Rosenbrock	iqABC	0,031	30,962	7,567
Rosenbrock	iqsABC	0,187	38,058	8,531
Schaffer	iqABC	0,002	0,029	0,011
Schaffer	iqsABC	0,000	0,037	0,011
Schwefel	iqABC	-11 405,912	-9 331,275	-11 330,279
Schwefel	iqsABC	-11 390,170	-9 127,375	-10 241,601
Sphere	iqABC	2,204	16,943	7,972
Sphere	iqsABC	2,514	55,343	9,211

15 lentelė. iqsABC algoritmo efektyvumas kai $D = 30$ (1500 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Algoritmas	Min	Max	Vidurkis
f1	iqABC	29 404 854,7271	513 001 999,886	211 996 760,472
f1	iqsABC	16 585 381,394	952 639 554,963	106 582 996,950
f2	iqABC	92 986,675	202 215,868	154 055,691
f2	iqsABC	93 251,461	197 176,826	146 680,154
f3	iqABC	334,112	345,546	357,626
f3	iqsABC	330,845	343,259	354,951
f4	iqABC	618,040	2 218,136	1 163,505
f4	iqsABC	633,432	2 253,315	1 181,021
f5	iqABC	501,378	503,534	527,596
f5	iqsABC	501,143	503,670	527,407
f6	iqABC	600,461	600,925	630,684
f6	iqsABC	600,262	600,887	630,597
f7	iqABC	700,294	702,911	735,771
f7	iqsABC	700,250	701,214	735,593
f8	iqABC	843,178	1 827,023	1 173,750
f8	iqsABC	842,578	3 402,838	1 401,085
f9	iqABC	912,628	914,021	959,327
f9	iqsABC	913,011	914,422	959,507
f10	iqABC	3 270 279,422	36 192 496,932	11 595 189,822
f10	iqsABC	1 671 791,610	33 707 495,648	11 968 755,123
f11	iqABC	1 124,374	1 308,211	1 222,097
f11	iqsABC	1 121,905	1 296,625	1 230,648
f12	iqABC	1 478,852	2 895,249	2 187,481
f12	iqsABC	1 659,334	2 834,963	2 293,590
f13	iqABC	1 640,435	1 768,302	1 788,314
f13	iqsABC	1 643,304	1 780,428	1 760,545
f14	iqABC	1 641,564	1 772,549	1 775,504
f14	iqsABC	1 635,805	1 866,642	1 784,847
f15	iqABC	1 909,802	2 980,930	2 407,681
f15	iqsABC	1 903,386	2 980,155	2 383,430

16 lentelė. iqsABC algoritmo efektyvumas kai $D = 30$ (500000 tikslo funkcijos skaičiavimų)

Tikslo funkcija	Algoritmas	Min	Max	Vidurkis
Ackley	iqABC	$2,842 \cdot 10^{-14}$	$3,908 \cdot 10^{-14}$	$3,328 \cdot 10^{-14}$
Ackley	iqsABC	$2,132 \cdot 10^{-14}$	$3,908 \cdot 10^{-14}$	$3,529 \cdot 10^{-14}$
Branin	iqABC	0,398	0,398	0,398
Branin	iqsABC	0,398	0,398	0,398
Camel	iqABC	-1,032	-1,032	-1,032
Camel	iqsABC	-1,032	-1,032	-1,032
Dixon	iqABC	0,005	14,299	1,664
Dixon	iqsABC	0,002	7,846	0,975
Griewank	iqABC	0,000	$1,110 \cdot 10^{-16}$	$1,850 \cdot 10^{-17}$
Griewank	iqsABC	0,000	$1,110 \cdot 10^{-16}$	$4,071 \cdot 10^{-17}$
Rastrigin	iqABC	0,000	0,000	0,000
Rastrigin	iqsABC	0,000	0,000	0,000
Rosenbrock	iqABC	0,000	0,036	0,009
Rosenbrock	iqsABC	$1,247 \cdot 10^{-05}$	0,034	0,008
Schaffer	iqABC	0,000	$6,206 \cdot 10^{-08}$	$9,214 \cdot 10^{-09}$
Schaffer	iqsABC	0,000	$8,629 \cdot 10^{-08}$	$1,052 \cdot 10^{-08}$
Schwefel	iqABC	-12 569,487	-12 569,487	-12 569,487
Schwefel	iqsABC	-12 569,487	-12 569,487	-12 569,487
Sphere	iqABC	$2,722 \cdot 10^{-16}$	$4,963 \cdot 10^{-16}$	$3,909 \cdot 10^{-16}$
Sphere	iqsABC	$2,715 \cdot 10^{-16}$	$4,956 \cdot 10^{-16}$	$3,724 \cdot 10^{-16}$

17 lentelė. iqsABC algoritmo efektyvumas kai $D = 30$ (500000 tikslo funkcijos skaičiavimų)

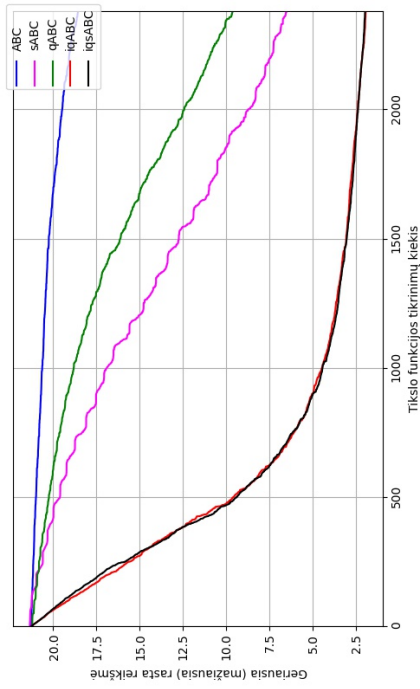
Tikslo funkcija	Algoritmas	Min	Max	Vidurkis
f1	iqABC	132,1804	2491,6314	684,3385
f1	iqsABC	145,4542	2030,1986	681,7975
f2	iqABC	63 437,4466	111 112,0483	85 671,3353
f2	iqsABC	59 309,2892	105 971,3519	91 350,4882
f3	iqABC	310,4773	332,2717	338,7921
f3	iqsABC	302,5181	333,4492	338,9914
f4	iqABC	279,7012	400,2707	409,4791
f4	iqsABC	400,0625	400,2498	413,4826
f5	iqABC	500,3881	500,8814	500,5514
f5	iqsABC	500,1786	500,7328	500,5893
f6	iqABC	600,0894	600,2150	620,1818
f6	iqsABC	600,1223	600,2282	620,1815
f7	iqABC	700,1478	700,2335	723,5319
f7	iqsABC	700,1408	700,2481	723,5401
f8	iqABC	804,4190	820,8463	842,9471
f8	iqsABC	806,1474	822,1962	843,6636
f9	iqABC	911,7378	913,0026	942,9914
f9	iqsABC	912,1972	912,9107	943,0146
f10	iqABC	122 092,6763	809 639,2767	410 473,7762
f10	iqsABC	78 003,0059	835 041,7021	447 850,3781
f11	iqABC	1114,8876	1119,5221	1118,1538
f11	iqsABC	1113,8905	1119,2604	1117,2986
f12	iqABC	1284,1456	1651,6250	1487,9772
f12	iqsABC	1287,3009	1666,8171	1494,6138
f13	iqABC	1574,6144	1627,6445	1663,5083
f13	iqsABC	1572,8179	1627,6440	1657,4919
f14	iqABC	1609,1849	1623,9450	1670,4659
f14	iqsABC	1611,4128	1629,6785	1672,7574
f15	iqABC	–39 791,0019	1912,2345	554,6452
f15	iqsABC	1900,0900	1906,1715	1964,7869

18 lentelė. sABC algoritmo efektyvumas kai $D = 30$ (1500 tikslo funkcijos skaičiavimų)

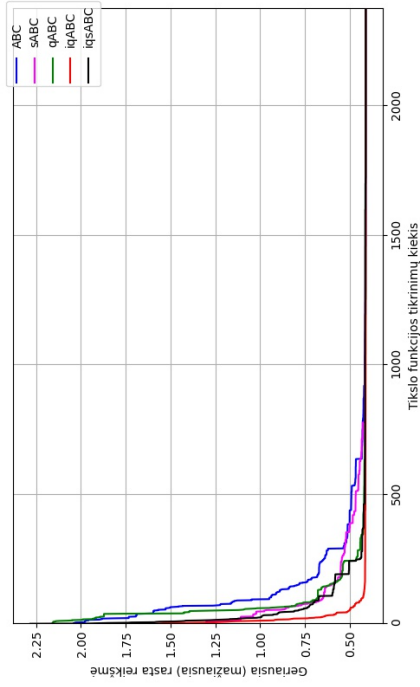
Tikslo funkcija	Statistiškai geresnis algoritmas	Atsakymo vidurkio skirtumas ABC-sABC	p reikšmė
Ackley	sabc	1,643	0,000
Branin	-	0,008	0,911
Camel	-	0,000	0,970
Dixon	-	-432 110 355 794,410	0,247
Griewank	-	23,421	0,852
Rastrigin	sabc	45,017	0,002
Rosenbrock	-	1 099,333	0,079
Schaffer	-	0,011	0,370
Schwefel	-	-96,477	0,576
Sphere	-	4 298,017	0,167
f1	-	12 278 488 139,420	0,191
f2	-	7 246,278	0,455
f3	-	-0,852	0,550
f4	sabc	840,895	0,015
f5	sabc	0,643	0,010
f6	sabc	1,198	0,002
f7	sabc	71,969	0,000
f8	-	-502 981,061	0,970
f9	-	-0,038	0,941
f10	-	3 977 891,570	0,941
f11	-	2,840	0,455
f12	-	8 024,587	0,247
f13	-	44,956	0,941
f14	-	-11,955	0,970
f15	-	30,641	0,911

3 priedas

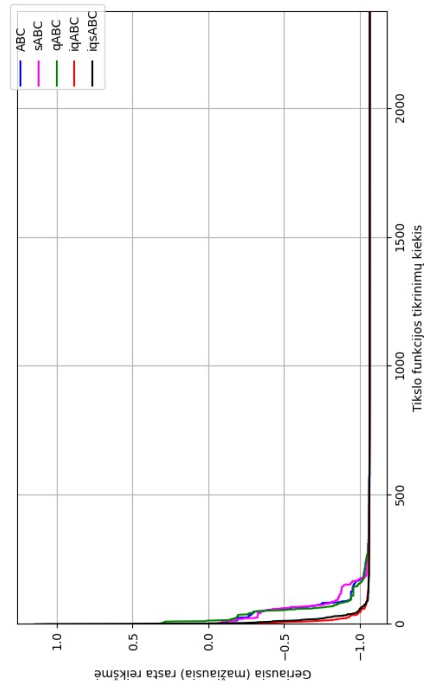
Konvergavimo grafikai su ABC, qABC, iqABC, iqsABC algoritmais



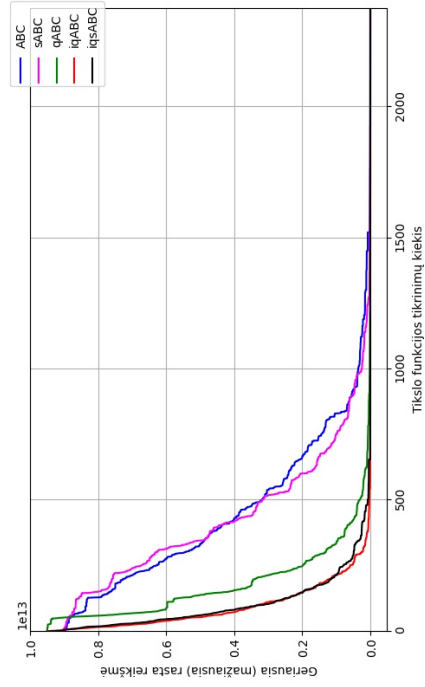
38 pav. Algoritmų konvergavimas su Ackley tikslo funkcija



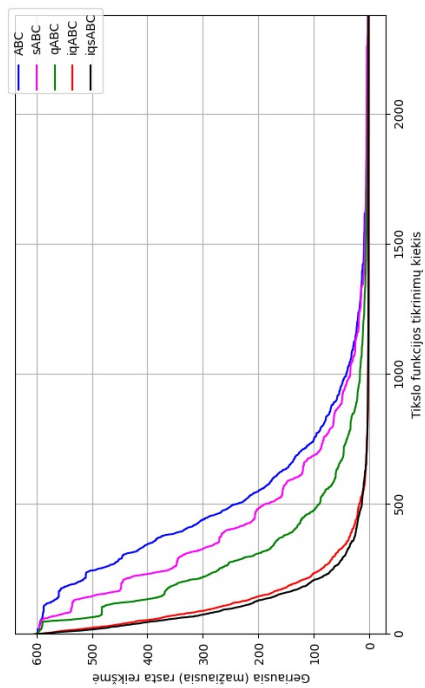
39 pav. Algoritmų konvergavimas su Branin tikslo funkcija



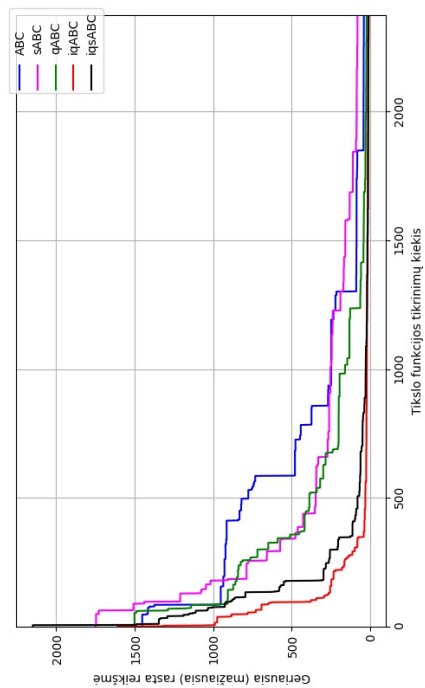
40 pav. Algoritmų konvergavimas su kuprunagario tikslo funkcija



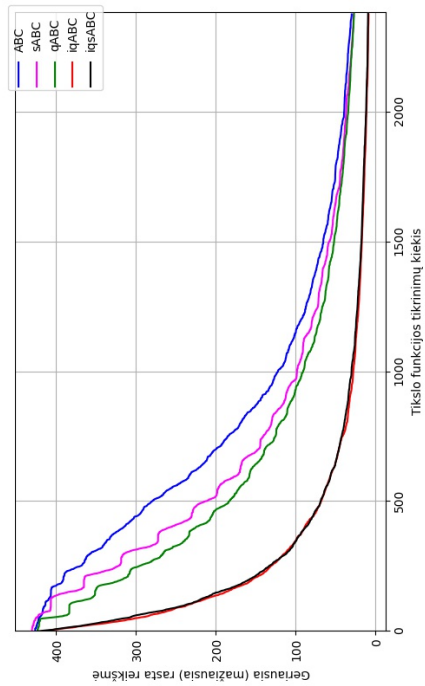
41 pav. Algoritmų konvergavimas su Dixon tikslo funkcija



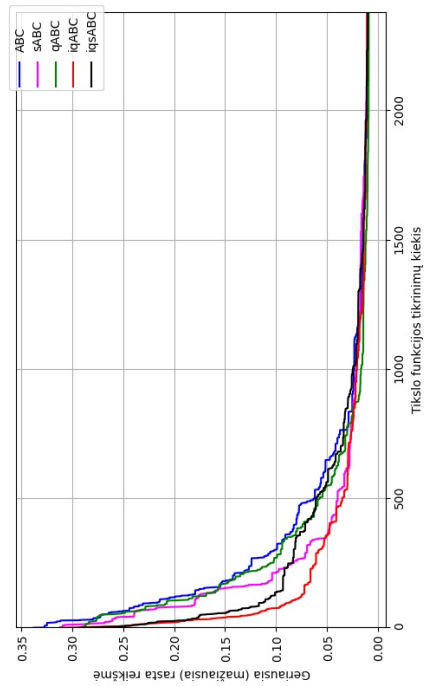
42 pav. Algoritmų konvergavimas su Griewank tikslo funkcija



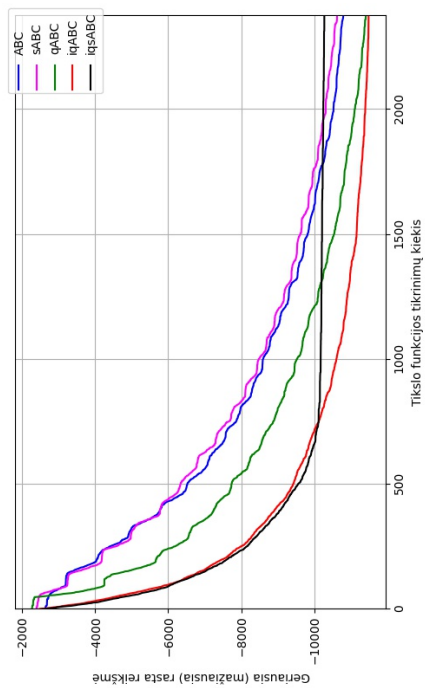
44 pav. Algoritmų konvergavimas su Rosenbrock tikslo funkcija



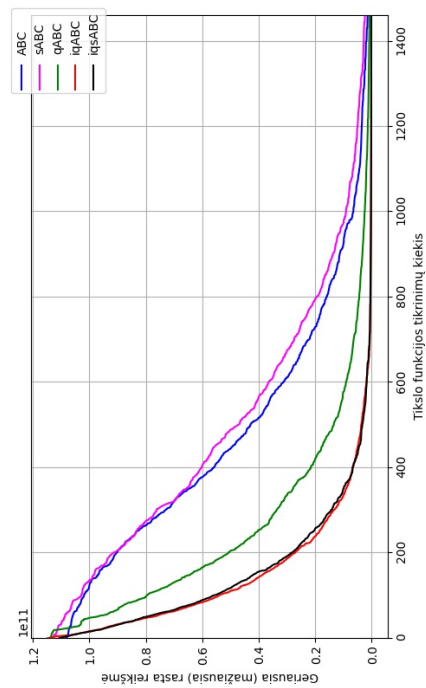
43 pav. Algoritmų konvergavimas su Rastrigin tikslo funkcija



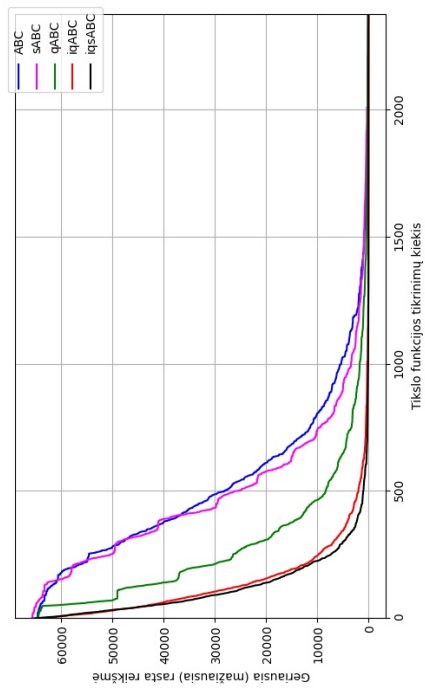
45 pav. Algoritmų konvergavimas su Schaffer tikslo funkcija



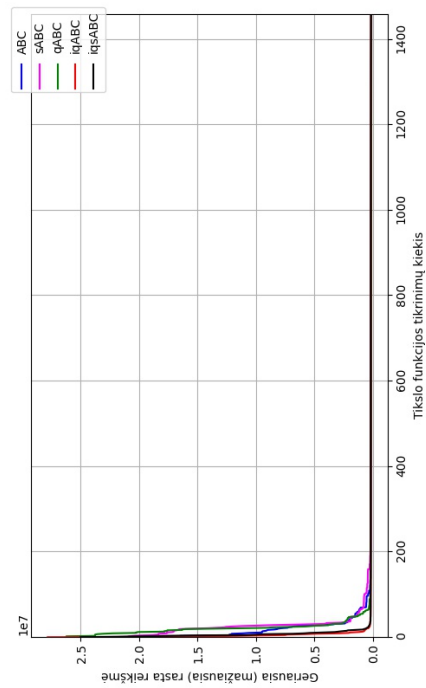
46 pav. Algoritmų konvergavimas su Schwefel tikslo funkcija



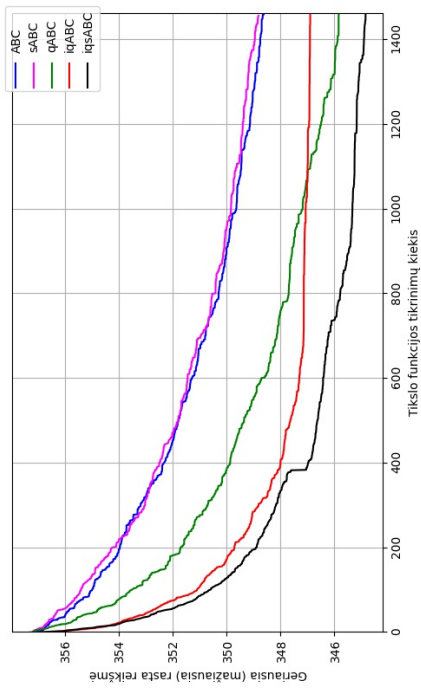
48 pav. Algoritmų konvergavimas su f_1 tikslo funkcija



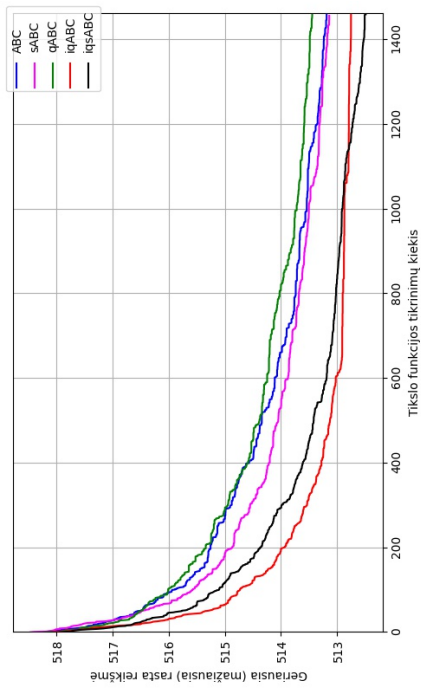
47 pav. Algoritmų konvergavimas su sferos tikslo funkcija



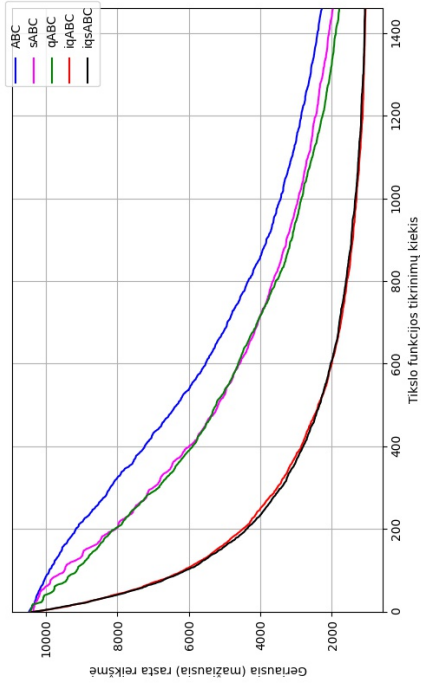
49 pav. Algoritmų konvergavimas su f_2 tikslo funkcija



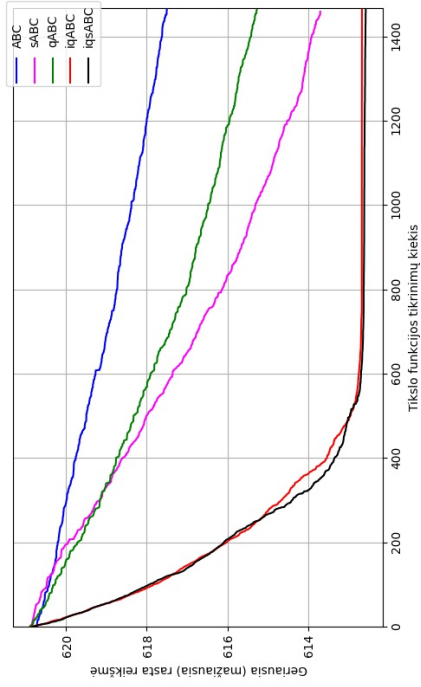
50 pav. Algoritimų konvergavimas su f_3 tikslo funkcija



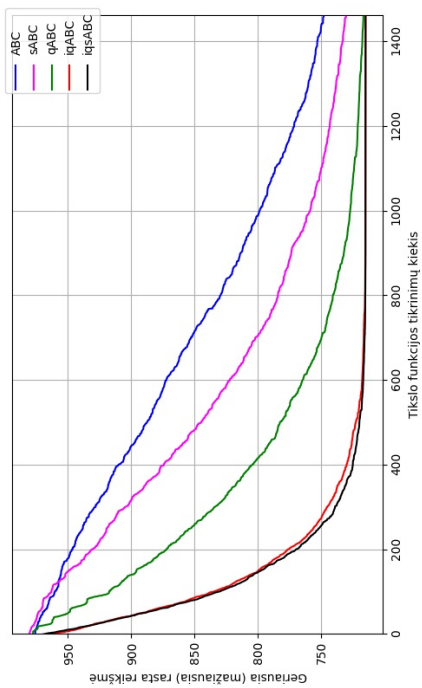
52 pav. Algoritimų konvergavimas su f_5 tikslo funkcija



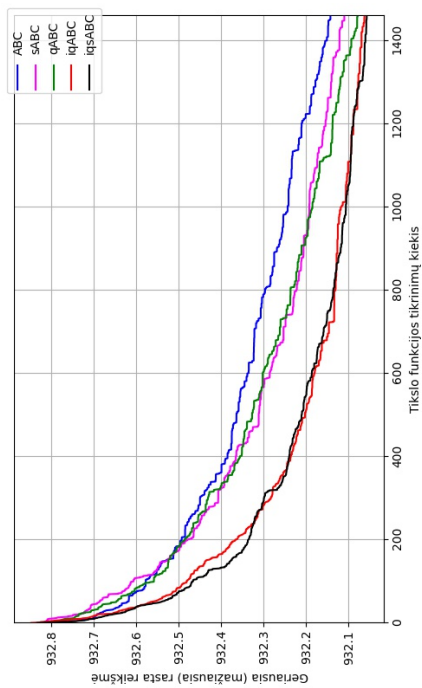
51 pav. Algoritimų konvergavimas su f_4 tikslo funkcija



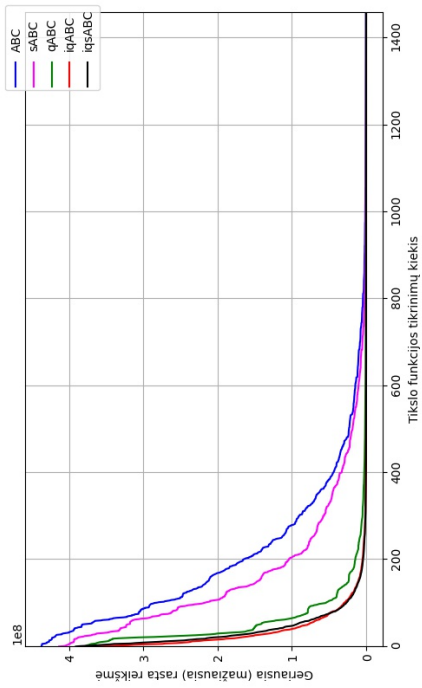
53 pav. Algoritimų konvergavimas su f_6 tikslo funkcija



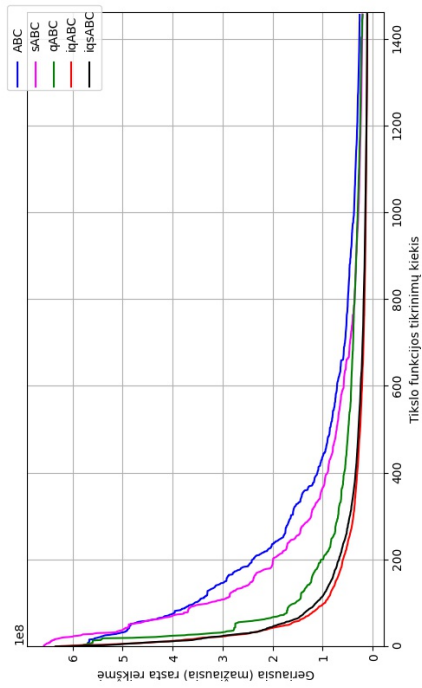
54 pav. Algoritimų konvergavimas su f_7 tikslo funkcija



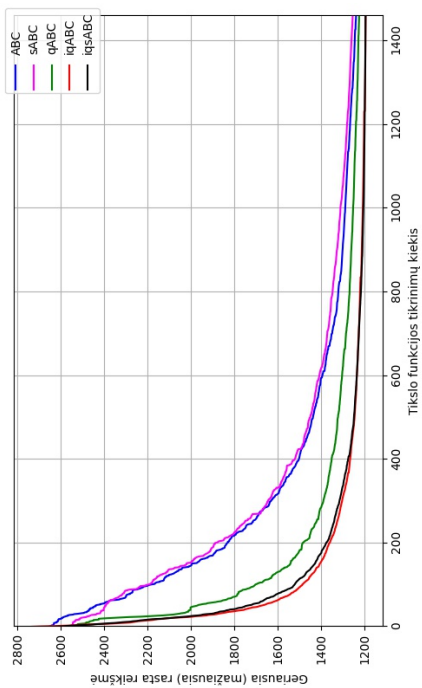
56 pav. Algoritimų konvergavimas su f_9 tikslo funkcija



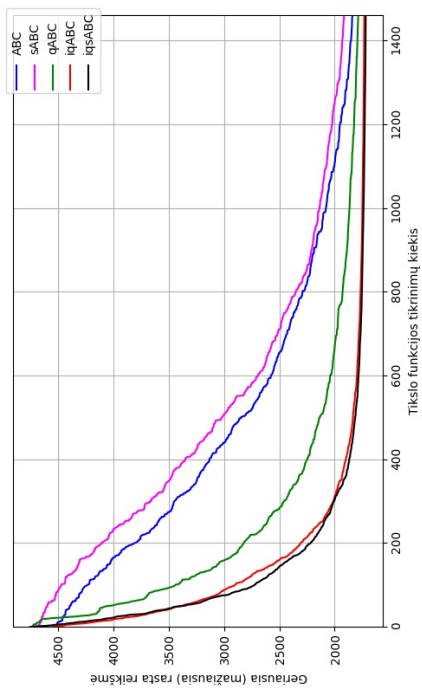
55 pav. Algoritimų konvergavimas su f_8 tikslo funkcija



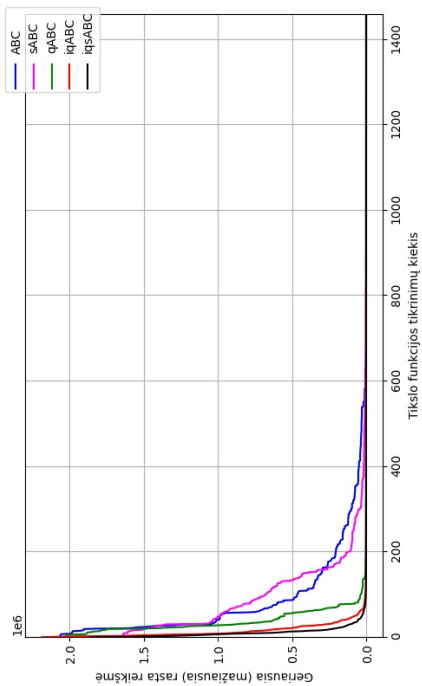
57 pav. Algoritimų konvergavimas su f_{10} tikslo funkcija



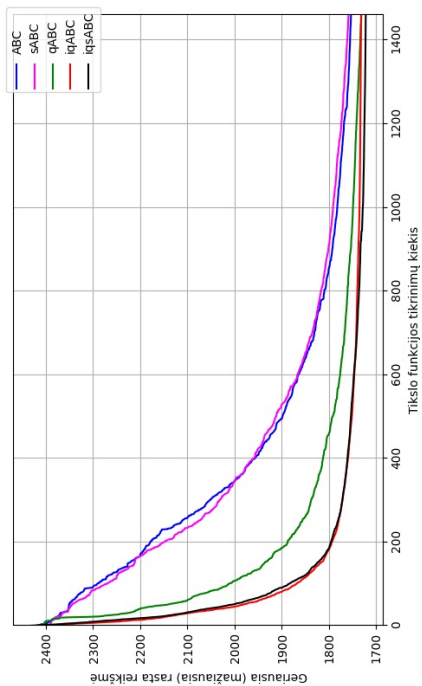
58 pav. Algoritmų konvergavimas su f_{11} tikslo funkcija



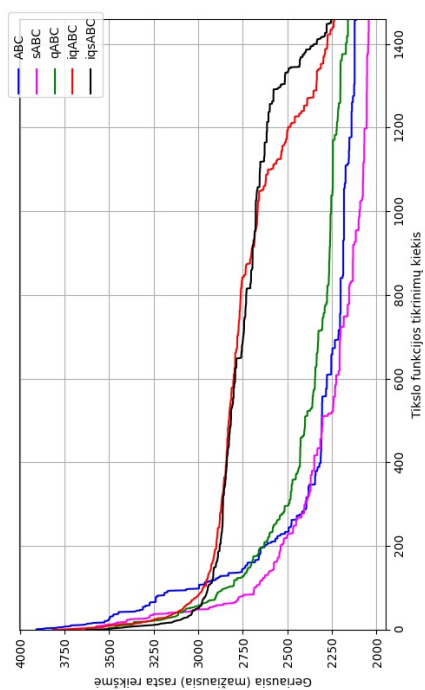
60 pav. Algoritmų konvergavimas su f_{13} tikslo funkcija



59 pav. Algoritmų konvergavimas su f_{12} tikslo funkcija



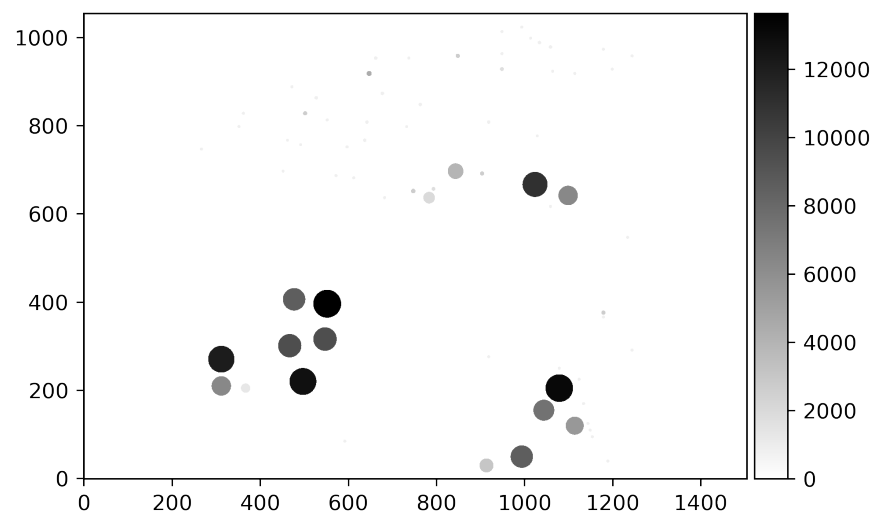
61 pav. Algoritmų konvergavimas su f_{14} tikslo funkcija



62 pav. Algoritmų konvergavimas su f_{15} tikslo funkcija

4 priedas

Landšafto tikslo funkcijos grafikai



63 pav. Tikslo funkcijos pritaikymo landšafto informacijai reikšmių atvaizdavimas

5 priedas

Pseudokodai

1 algoritmas: Stebinčios bitės iqsABC algoritmo žingsnio pseudokodas

```

1: for all  $i \leftarrow SN$  do
2:    $p(x_i) \leftarrow$  tikimybė iš (39)
3: end for
4:  $visosBites \leftarrow 1$ 
5:  $pasirinktaBite \leftarrow 1$ 
6: while  $visosBites < SN$  &  $visiIvertinimai \leq maxIvertinimai$  do
7:   if  $rand(0,1) \leq p_{santykinis}(x_{pasirinktaBite})$  then
8:     if  $bandymai_{stebinciu} \leq l_{santykinis}(x_{isimintas})$  then
9:        $i_{isimintas} \leftarrow$  geriausio sprendinio indeksas
10:       $x_{isimintas} \leftarrow$  geriausio sprendinio vektorius
11:       $v_{naujas} \leftarrow$  naujo sprendinio vektorius pagal (32)
12:      if  $fit(x_{isimintas}) \geq fit(v_{naujas})$  then
13:         $x_{isimintas} \leftarrow v_{naujas}$ 
14:         $bandymai(x_{isimintas}) \leftarrow 0$ 
15:      else
16:         $bandymai(x_{isimintas}) \leftarrow bandymai(x_{isimintas}) + 1$ 
17:         $bandymai_{stebinciu} \leftarrow bandymai_{stebinciu} + 1$ 
18:      end if
19:    else
20:       $v_{naujas} \leftarrow$  naujo sprendinio vektorius pagal(28)
21:      if  $fit(x_{pasirinktaBite}) \geq fit(v_{naujas})$  then
22:         $pasirinktaBite \leftarrow v_i$ 
23:         $bandymai(pasirinktaBite) \leftarrow 1$ 
24:      else
25:         $bandymai(pasirinktaBite) \leftarrow bandymai(pasirinktaBite) + 1$ 
26:      end if
27:    end if
28:     $visiIvertinimai \leftarrow visiIvertinimai + 1$ 
29:     $visosBites \leftarrow visosBites + 1$ 
30:  end if
31:   $pasirinktaBite \leftarrow pasirinktaBite + 1$ 
32:  if  $pasirinktaBite \geq SN$  then
33:     $pasirinktaBite \leftarrow 1$ 
34:  end if
35: end while

```

2 algoritmas: Dirbančios bitės iqsABC algoritmo žingsnio pseudokodas

```
1: for all  $i \leftarrow SN$  do
2:   if  $rand(0,1) \leq p(x_{pasirinktaBite})$  then
3:     if  $bandymai_{darbininkiu} \leq l_{santykiniis}(x_{isimintas})$  then
4:        $i_{isimintas} \leftarrow$  geriausio sprendinio indeksas
5:        $x_{isimintas} \leftarrow$  geriausio sprendinio vektorius
6:        $v_{naujas} \leftarrow$  naujo sprendinio vektorius pagal (33)
7:       if  $fit(x_{isimintas}) \geq fit(v_{naujas})$  then
8:          $x_{isimintas} \leftarrow v_{naujas}$ 
9:          $bandymai(x_{isimintas}) \leftarrow 0$ 
10:      else
11:         $bandymai(x_{isimintas}) \leftarrow bandymai(x_{isimintas}) + 1$ 
12:         $bandymai_{darbininkiu} \leftarrow bandymai_{darbininkiu} + 1$ 
13:      end if
14:    else
15:       $v_{naujas} \leftarrow$  naujo sprendinio vektorius pagal(28)
16:      if  $fit(x_{pasirinktaBite}) \geq fit(v_{naujas})$  then
17:         $pasirinktaBite \leftarrow v_i$ 
18:         $bandymai(pasirinktaBite) \leftarrow 1$ 
19:      else
20:         $bandymai(pasirinktaBite) \leftarrow bandymai(pasirinktaBite) + 1$ 
21:      end if
22:    end if
23:  end if
24:   $visiIvertinimai \leftarrow visiIvertinimai + 1$ 
25: end for
```
