

Bičių spiečių imitavimas sprendžiant optimizavimo uždavinius

Alfonsas Misevičius

Kauno technologijos universiteto
Multimedijos inžinerijos katedros
profesorius, daktaras
Kaunas University of Technology,
Department of Multimedia Engineering,
Professor, Doctor
Studentų g. 50–400a/416a, LT-51368 Kaunas
El. paštas: alfonsas.misevicius@ktu.lt

Vytautas Bukšnaitis

Kauno technologijos universiteto
Multimedijos inžinerijos katedros lektorius
Kaunas University of Technology,
Department of Multimedia Engineering, Lecturer
Studentų g. 50–400, LT-51368 Kaunas
El. paštas: vytautas.buksnaitis@ktu.lt

Jonas Blonskis

Kauno technologijos universiteto
Multimedijos inžinerijos katedros docentas, daktaras
Kaunas University of Technology,
Department of Multimedia Engineering, Assoc. Professor, Doctor
Studentų g. 50–400, LT-51368 Kaunas
El. paštas: jonas.blonskis@ktu.lt

Straipsnyje nagrinėjami klausimai, susiję su naujoviškų metodų taikymu sprendžiant optimizavimo uždavinius. Šiuo konkrečiu atveju diskutuojama apie bičių spiečių elgsenos imitavimą ir galimą jo taikymą kombinatorinio (diskretinio) tipo optimizavimo uždaviniams. Straipsnio pradžioje aptariami konceptualūs aspektai ir bendroji bičių spiečių imitavimo algoritmų idėja. Aprašoma bičių spiečiaus imitavimo algoritmo realizacija atskiram nagrinėjamam atvejui – kvadratinio paskirstymo uždaviniui, kuris yra vienas iš aktualių ir sudėtingų kombinatorinio optimizavimo uždavinių pavyzdžių. Straipsnyje pateikiami ir su realizuotu algoritmu atliktų eksperimentų rezultatai, kurie iliustruoja skirtingų veiksnių (parametru) įtaką gaunamų sprendinių kokybei ir patvirtina aukštą algoritmo efektyvumo lygį.

Įvadas

Sprendžiant sudėtingus optimizavimo uždavinius, susiduriama su vis naujais iššūkiais, kuriuos lemia tokie svarbūs veiksniai kaip išaugusi sprendžiamų uždavinių apimtis (didžiulis uždavinių kintamųjų skaičius), poreikis spręsti uždavinius, kai yra tam tikrų apribojimų (pvz., ribotiems laiko ištekliams), dinamiškai keičiasi aplinka, optimizavimo kriterijai. Papildomų sunkumų gali kelti tai, kad optimizuojamų kin-

tamųjų reikšmės gali būti su paklaidomis ar ne visai apibrėžtos; pagaliau, uždaviniai gali turėti ir kelis optimizavimo kriterijus, kuriuos reikia optimizuoti vienu metu (Zäpfel ir kt., 2010). Esant tokiam kontekstui, aktualu naudoti nestandartinius modernius optimizavimo metodus, kurie remtųsi ne pavienių sprendinių nagrinėjimu, kaip dažnai yra įprasta klasikinių, tradicinių algoritmų atveju, bet būtų grindžiami tam tikru manipuliavimu sprendinių rinkiniais, grupėmis. Ši metodų klasė šiuolaikinėje literatūroje yra

vadinama kolektyvinio intelekto (kolektyvinės intelektikos – KI) (angl. *swarm intelligence*) algoritmais (Bonabeau ir kt., 1999; Karaboga, Akay, 2009). Esminis šių algoritmų požymis ir yra grupinės elgsenos modeliavimas operuojant su sprendinių rinkiniais (vaizdžiai tariant, populiacijomis, būriais, kolonijomis, spiečiais) (Beasley, 2002). Tokiu būdu optimizavimo procesą galima asocijuoti su bendru grupės narių veikimu įsivaizduojamoje (sprendinių) paieškos erdvėje. Koordinuotai funkcionuojančius narius tikslinga traktuoti kaip agentus (sudėtingų struktūrų autonominius, tarpusavyje sąveikaujančius intelektualius komponentus), pabrėžiant tai, jog agentų visuma nėra tik mechaninė atskirų jos dalių suma, o yra (savi-)organizuota sistema, pasižyminti savitarpio sąveika, informacijos perdavimu, apdorojimu. Bendra kooperatyvinė veikla, dalijimasis informacija, protingas atliekamų darbų, funkcijų pasiskirstymas yra tie svarbiausi veiksniai, kurie leidžia siekti didesnio KI metodų efektyvumo.

Kolektyvinio intelekto metodų įvairovė yra gana plati, pradedant genetiniais algoritmais ir baigiant vadinamaisiais kultūriniais algoritmais (Zäpfel ir kt., 2010). Sprendinių populiacijų manipuliavimu ir kooperavimosi principu yra pagrįsti, pavyzdžiui, evoliuciniai algoritmai (angl. *evolutionary algorithms*) (Bäck ir kt., 1997), išsklaidytosios paieškos (angl. *scatter search*) algoritmai (Glover ir kt., 2002), skruzdėlių kolonijų elgsenos imitavimo algoritmai (angl. *ant colony optimization algorithms*) (Dorigo, Stützle, 2002), dalelių spiečių algoritmai (angl. *particle swarm optimization algorithms*) (Kennedy, Eberhart, 1995) ir pan. Viena iš naujesnių kolektyvinės intelektikos krypčių yra bičių spiečių (BS) (elgsenos) imitavimo algoritmai, arba tiesiog bičių spiečių algoritmai (Yonezawa, Kikuchi, 1995; Lučić, Teodorović, 2001; Karaboga, 2005; Karaboga, Basturk, 2007; Karaboga, Akay, 2009). Angliškas šių terminų atitikmuo yra *bee colony optimization algorithms*. (Dėl tikslumo reikėtų sakyti „dirbtinių bičių spiečių algoritmai“, bet trumpumo dėlei žodis „dirbtinių“ gali būti ir praleidžiamas.)

Konceptualūs bičių spiečių algoritmų aspektai

Bendrieji konceptualūs aspektai

Konceptualiai apibrėžiant bičių spiečių imitavimo algoritmus, visų pirma turėtų būti akcentuota, jog tai yra metaeuristinių optimizavimo metodų (algoritmų) pogrupis. Taigi, šie algoritmai negarantuoja, kad bus rasti optimalūs sprendiniai. Bet tokie algoritmai leidžia gerokai sumažinti skaičiavimų kiekį, o tai labai aktualu taupiau naudojant kompiuterinių skaičiavimų išteklius, sykiu ir techninius bei energinius išteklius. Kita vertus, gaunami sprendiniai, nors ir nebūdami optimalūs, dažnai yra pakankamai aukštos kokybės ir visiškai patenkina uždavinių sprendinių vartotojus (užsakovus). Taip pat pabrėžtina, jog bičių spiečių (kaip ir kitų kolektyvinę elgseną imituojančių) algoritmų efektyvumą daugiau ar mažiau priklauso nuo to, kokių konkrečiu būdu realizuojami atskiri algoritmo komponentai ir kaip lanksčiai, protingai bendrasis algoritmo principas adaptuojamas konkrečioms sprendžiamoms uždaviniams (Zäpfel ir kt., 2010).

Tokių algoritmų kaip BS (ir apskritai dauguma metaeuristinių metodų) konceptualus funkcionavimo principas yra labai universalus ir pasižymi palyginti dideliu nepriklausomumu. Tai reiškia, kad pats sprendimo būdas, koncepcija nepriklauso nuo sprendžiamų uždavinių – šie algoritmai gali būti taikomi iš esmės bet kokioms optimizuotinoms sistemoms, pradedant tolydziosiomis matematinėmis funkcijomis ir baigiant sudėtingomis save reguliuojančiomis struktūromis.

Biologinis kontekstas

Pagrindinė BS algoritmų idėja kilo stebint gyvų bičių spiečių (šeimų) elgesį. Suprantama, kad bičių veikla yra daugialypė ir, aišku, čia nesigilinsime į įvairias tos veiklos (fiziologines ir pan.) detales. Pagrindinis dėmesys skiriamas pačiai koncepcijai. BS algoritmų principo pagrindas ir yra tai, jog imituojama įprastinė bičių elgsena ir tos elgsenos dėsningumai ieškant ir renkant nektarą, kuris yra reikalingas,

apsirūpinant maisto (t. y. medaus) atsargomis bičių gyvenamoje vietoje (avilyje, lizde) (Lučić, Teodorović, 2001; Karaboga, 2005).

Natūralų bičių spiečių paprastai sudaro 50 000–100 000 bičių ir kiekviena atlieka tam tikras funkcijas. Dauguma jų ieško medaus gamybai reikalingo nektaro aplinkinėse avilio (lizdo) vietovėse. Būdingieji šių bičių elgsenos elementai (aspektai) yra tokie. Nektaro paieškos proceso pradinėje stadijoje atskiros bitės (vadinamieji žvalgai) išžvalgo, ištiria tam tikrą (pradinę) teritoriją. Grįžusios į avilį, šios bitės „informuoja“ kitas avilio bites apie aptikto maisto (nektaro) vietas, kiekį, kokybę. Informavimas (tais atvejais, kai yra rastos potencialiai geros nektaro vietos) vyksta tam tikru specifiniu būdu – šokant „ritualinius“ šokius atskiroje avilio „salėje“. Šokiuose yra užkoduota labai įvairi informacija apie nektarą. Taip bitės praneša apie surastą nektarą (pateikia „reklamą“) ir paskatina kitas bites skristi kartu. Bitės, iki tol buvusios tik „stebėtojai“, gali nuspręsti skristi ieškoti maisto, atsižvelgdamos į pateiktą informaciją. Taip tampama bitėmis darbininkais (sykiu ir kitų bičių verbuotojais — vadinamaisiais rekrutų verbuotojais). Galimi tokie bičių, atsakingų už maisto rinkimą, pagrindiniai elgesio scenarijai: a) jos gali atsakyti rinkti maistą esamoje teritorijoje (bitės, atsakiusios rinkti nektarą, pvz., dėl to, kad yra mažas nektaro kiekis arba nektaras yra jau pasibaigęs šioje vietovėje, gali (vėl) tapti žvalgais arba įgyti stebėtojų (vadinamųjų laisvųjų agentų) statusą); b) bitės gali toliau nešti nektarą iš esamos vietos, neinformuodamos apie tai kitų bičių; c) jos gali papildomai šokti šokius, t. y. pasikeisti informacija su kitomis bitėmis apie aptikto maisto išteklius, ir taip pakviesti daugiau padėjėjų į vietas, kuriose galbūt gausu maisto (kuo didesnis nektaro kiekis ir geresnė jo kokybė, tuo labiau rekrutų verbuotojai stengiasi patraukti į savo pusę kitas bites (laisvuosius agentus); tačiau bitės, suradusios ne tokias geras nektaro vietas, tikriausiai bus linkusios jų atsakyti ir bandys sekti paskui tas bites, kurios labiausiai reklamuoja savo vietas). Taigi, galima sakyti, kad nešančios nektarą bitės dalyvauja nepertraukiamame alternatyvų

rinkimosi ir sprendimų priėmimo procese. Toks iš daugybės atskirų sprendimų priėmimų susidedantis procesas ir apibrėžia bičių spiečiaus veiklos pobūdį apsirūpinant medumi.

Dirbtinių bičių spiečių algoritmai

Dirbtinių bičių spiečiai (DBS) yra idealizuoti gyvų bičių šeimų modeliai. Tačiau abiem atvejais jų paskirtis yra panaši – surasti geriausia (efektyviausia) galimą veiklos variantą, siekiant įgyvendinti tam tikrą tikslą. Kaip realybėje gyvos bitės skraidydamos atlieka tam tikrą vietų žvalgybą ir aptinka ten medaus šaltinius, taip algoritminės realizacijos atveju, jei taip galima sakyti, dirbtinės bitės yra atsakingos už efektyvų sprendinių erdvės išžvalgymą ir potencialiai gerų tos erdvės sričių radimą. Natūralūs bičių spiečiai siekia sunėsti kuo daugiau nektaro ir apsirūpinti kuo didesnėmis medaus atsargomis, o dirbtinių BS algoritmų tikslas yra optimizuoti duotą kriterijų, vykdant paiešką leidžiamoje sprendinių aibėje. Taigi, optimizavimo algoritmuose operuojama su gyvų bičių imitacijomis – tam tikromis virtualiomis bitėmis, kurios gali būti traktuojamos kaip abstraktūs (intelektualūs) agentai. Šie agentai yra susiejami su optimizavimo uždavinio sprendiniais (t. y. nepriklausomų dydžių – kintamųjų – reikšmių konfigūracijomis), taip pat su procedūromis, skirtomis tiems sprendiniams konstruoti arba pagerinti. Priminsime, kad optimizavimo uždavinį formaliai galima apibrėžti pora (S, f) ; čia S žymi sprendinių aibę, o f – skaliarinę tikslo funkciją (TF), kurios apibrėžimo sritis – S , o reikšmių aibė – realieji skaičiai. Tuomet išspręsti uždavinį (S, f) reiškia surasti sprendinį $\tilde{s} \in S$ ir tokį, kad $\tilde{s} \in \tilde{S} = \left\{ s^v \mid s^v = \arg \min_{s \in S} f(s) \right\}$; čia \tilde{s} yra vadinamas (globaliai) optimaliu sprendiniu. Tarsime, kad TF turi būti minimizuojama, kitaip tariant, turi būti randamas toks sprendinys, kuriam TF įgyja minimalią reikšmę. Be to, laikysime, jog sprendiniai yra diskretieji.

Bičių spiečiaus elgsenos imitavimo proceso pagrindinės formalios sudedamosios dalys konceptualiai aprašytos 1 pav. (taip pat žr.: Karaboga, Basturk, 2007).

Galimi įvairūs šios konceptualiosios schemos variantai ir realizacijos. Pavyzdžiui, vienos pirmųjų DBS algoritminių realizacijų buvo kreipiamos į matematinių multimodalinių funkcijų ekstremumą paiešką (Karaboga, 2005; Karaboga, Basturk, 2007). Kitos DBS algoritmų versijos dažnai taikytos ir sprendžiant kombinatorinius uždavinius; paprastai jos buvo naudojamos sprendiniams konstruoti, kai virtualiosios bitės yra susiejamos su atskirais sprendinių elementais (Lučić, Teodorović, 2001). Tai aktualu sudarant aukštos kokybės pradinis sprendinius, kurie papildomai optimizuojami iteraciniais algoritmais. Tačiau DBS algoritmai gali būti sėkmingai taikomi ir pačiuose iteraciniuose sprendinių pagerinimo procesuose. Šiuo atveju dirbtinės bitės atitinka ne atskirus sprendinių elementus, o visiškai suformuotus sprendinius (Sabat ir kt., 2010; Pan ir kt., 2011). Be to, virtualiosios bitės – kaip pakankamai intelektualizuoti agentai – gali būti asocijuojamos su tam tikromis sprendinių lokaliojo pagerinimo (arba lokaliosios paieškos, t. y. paieškos sprendinių kaimyninėse aplinkose) procedūromis (Davidović ir kt., 2011).

Bičių spiečiaus elgsenos imitavimo proceso sąvokų (žr. 1 pav.) interpretavimas jį atitin-

kančioje virtualių bičių spiečiaus algoritminėje realizacijoje, besiremiančioje iteraciniu sprendinių gerinimu, pateikiamas 1 lentelėje. Gali būti išskirti tokie trys pagrindiniai DBS bičių tipai (funkcinės grupės) (Karaboga, Basturk, 2007): bitės rekrutų verbuotojos (darbininkės), bitės laisvosios agentės (stebėtojos), bitės žvalgės. Veikdamos korporatyviai ir tinkamai pasidalydamos atliekamus darbus, šios virtualių bičių grupės efektyviai vykdo savireguliacinį paieškos procesą, kuriame nuosekliai gerinama sprendinių kokybė. Kaip matyti iš 1 pav. ir 1 lentelės, bičių grupės gali būti asocijuojamos su atitinkamomis fazėmis (etapais): rekrutų verbuotojų, laisvų agenčių ir žvalgių. Sudėtos kartu šios fazės sudaro vieną sutartinį DBS proceso ciklą. Atitinkamai parinktas ciklų skaičius paprastai apibrėžia DBS algoritmo baigimo sąlygą (vykdymo trukmę). Konkretus rekrutų verbuotojų, laisvųjų agentų, žvalgių pasiskirstymas priklauso nuo pasirinktos algoritminės realizacijos.

Pavyzdžiui, galima iš anksto nustatyti fiksuotą santykį $\frac{R}{B}$ ($0 < \frac{R}{B} \leq 1$); čia R žymi bičių rekrutų verbuotojų kiekį (laisvai pasirenkamą tarp 1 ir B), o B ($B \geq 1$) yra bendras bičių skaičius spie-

PRADŽIA

1: INICIALIZACIJA

2: BIČIŲ REKRUTŲ VERBUOTOJŲ FAZĖ (ETAPAS)

// šioje fazėje (etape) bitės rekrutų verbuotojai (darbininkai) renką maistą, t. y. nektarą, tam tikrose vietose

3: BIČIŲ LAISVŲJŲ AGENTŲ FAZĖ (ETAPAS)

// šioje fazėje (etape) bitės laisvieji agentai (stebėtojai) skrenda paskui bites rekrutų verbuotojus rinkti nektarą anksčiau

// bičių rekrutų verbuotojų atrastose (išžvalgytose) vietose

// (bitės laisvieji agentai pasirenka sektingas bites rekrutų verbuotojus, atsižvelgdamos į užkoduotą informaciją, pateiktą

// bičių-rekrutų verbuotojų ritualinių šokių metu)

4: BIČIŲ ŽVALGŲ FAZĖ (ETAPAS)

// šioje fazėje (etape) tam tikras bičių žvalgių skaičius siunčiamas ieškoti naujų nektaro šaltinių (vietų)

// (bitėmis žvalgiais tampama, kai pasibaigia maisto atsargos esamose vietose*;

// kitos bitės, pasibaigus nektarui, gali tapti laisvaisiais agentais (stebėtojais))

5: Įsimenamas geriausias (iki esamo momento) rastas maisto (nektaro) šaltinis

6: Jeigu nėra patenkinta baigimo sąlyga, tai grįžtama į 2 etapą tęsti proceso; priešingu atveju procesas baigiamas

// geriausias rastas maisto (nektaro) šaltinis yra šio proceso rezultatas

PABAIGA

* Bičių spiečiaus elgsenos imitavimo proceso algoritminėse realizacijose taip pat gali būti naudojamas specialus parametras, kuriuo galima reguliuoti maksimalų galimų skrydžių į tą pačią vietą (šaltinį) skaičių. Jei šis skaičius viršijamas, tai traktuojama, jog maisto (nektaro) atsargos šioje vietoje yra pasibaigusios; tuomet bitės arba turi skristi žvalgyti naujų vietų, arba tampa laisvaisiais agentais (stebėtojais).

1 p a v. *Bičių spiečiaus elgsenos imitavimo proceso konceptualusis aprašymas*

1 lentelė. Bičių spiečiaus elgsenos imitavimo procesas ir dirbtinių bičių spiečiaus algoritmas: sąvokų interpretavimas

| Terminai konceptualiaime aprašyme | Interpretavimas dirbtinių bičių spiečiaus (DBS) algoritme |
|--|---|
| Inicializacija | Nustatomi reikiami algoritmo parametrai (spiečiaus (sprendinių populiacijos) dydis (B), algoritmo ciklų skaičius/vykdyimo laikas ir pan.). Sukonstruojama pradinė sprendinių populiacija (spiečiaus analogas), sudaryta iš B sprendinių (vieną bitę atitinka vienas sprendinys, o bitės surasto maisto kiekis ir kokybė yra asocijuojama su atitinkamo sprendinio (s) įvertinimu – tikslo funkcijos (f) reikšme). Pradinės sprendinių populiacijos konstravimui gali būti naudojamos papildomos sprendinių lokaliajo pagerinimo (lokaliosios paieškos) procedūros. |
| Rekrutų verbuotojų fazė | Atliekamas populiacijos sprendinių gerinimas (tiksliau, bandoma pagerinti sprendinius), naudojant pasirinktą lokaliajo pagerinimo (lokaliosios paieškos) procedūrą. Jeigu po lokaliajo pagerinimo procedūros gaunamas geresnis sprendinys negu prieš tai buvęs, tai buvęs (senas) sprendinys pakeičiamas nauju (geresniu); priešingu atveju senasis sprendinys išlieka nepakitęs. |
| Laisvųjų agentų fazė | Virtualiosios bitės praneša apie „savo“ sprendinių kokybę, t. y. apskaičiuojamos TF reikšmės atitinkamiems sprendiniams. Kai yra įvertinti visi sprendiniai, atliekamas papildomas sprendinių optimizavimas atsižvelgiant į esamos duotu momentu populiacijos (spiečiaus) atskirų sprendinių kokybę. Kuo geresnė sprendinio kokybė, tuo didesnė tikimybė, jog virtuali bitė laisvasis agentas pasirinktą sprendinį, t. y. pasirinktasis sprendinys bus toliau gerinamas. Tikimybė yra proporcinga tokiam dydžiui: $\frac{f(s_i)}{\sum_{j=1}^B f(s_j)}$; čia s_p, s_j žymi atitinkamus populiacijos (spiečiaus) sprendinius; $f(s_i), f(s_j)$ yra TF reikšmės, apskaičiuotos sprendiniams s_p, s_j . |
| Žvalgų fazė | Jeigu populiacijos (spiečiaus) sprendinys negali būti daugiau pagerintas, atlikus tam tikrą pakartotinių pagerinimo bandymų skaičių, tai tokio sprendinio atsisakoma, t. y. jis yra pašalinamas iš populiacijos. Vietoje pašalintojo yra generuojamas/konstruojamas atsitiktiniu ar kitu būdu naujas sprendinys, kuris dalyvaus tolesniame procese kartu su kitais sprendiniais. |
| Įsimenamas geriausias rastas maisto (nektaro) šaltinis | Įsimenamas (įrašomas atskiroje atminties srityje) geriausias iki duotojo momento (ciklo) rastas sprendinys. (Minimizavimo atveju geriausias sprendinys yra tas, kuriam tikslo funkcija (f) įgyja mažiausią reikšmę.) Įsimenama tik tuomet, kai surandamas naujas geresnis sprendinys (t. y. sprendinys, geresnis už prieš tai buvusį geriausią). |
| Baigimo sąlyga | Kaip galima algoritmo baigimo sąlyga gali būti pasirinktas, pvz., tam tikras maksimalus algoritmo ciklų skaičius arba algoritmo vykdymo laiko limitas. |

čiuje (paprastai pasirenkama maždaug tarp 10 ir 100 – tai yra daug kartų mažiau, palyginti su realiais bičių spiečiais; žr. skyrelį „Biologinis kontekstas“). Kai dėl bičių žvalgų, dažnai algoritminėse realizacijose vadovujamasi taisykle, jog vieno ciklo metu tik ne daugiau kaip viena bitė gali vykdyti „žvalgyti“ naujų galimų sprendinių (t. y. gali būti naujai sukonstruojamas ne daugiau kaip vienas sprendinys). Pati bičių funkcinių grupių sudėtis nėra iš anksto griežtai fiksuojama; priklausymas vienai ar kitai grupei yra gana reliatyvus dalykas, nes bičių atliekamos funkcijos gali daug kartų pasikeisti proceso metu.

Kaip minėta, DBS algoritmuose gali būti naudojamos papildomos priemonės, t. y. procedūros sprendinių lokaliajam pagerinimui (lokaliajai paieškai). Taigi, dar daugiau nutolstama nuo gamtinio analogo, DBS algoritmai tampa dar labiau sintezuoti, hibridizuoti. Šiuo atveju galimas labai platus tokių pagerinimo procedūrų diapazonas, tam galima pritaikyti įvairios prigimties iteracinius euristinius algoritmus (pvz., godžiąją paiešką, tabu paiešką, atkaitinimo modeliavimą ir pan.). Toliau aprašoma tokio pobūdžio DBS algoritminė realizacija vienam iš kombinatorinio optimizavimo uždavinių – kvadratinio paskirstymo (KP) uždaviniui.

Dirbtinių bičių spiečiaus algoritmo realizacija kvadratinio paskirstymo uždaviniui

Kvadratinio paskirstymo uždavinys (angl. *quadratic assignment problem*) (Čela, 1998; Loiola ir kt., 2007) yra vienas iš sudėtingų kombinatorinio optimizavimo uždavinių. KP uždavinio atveju sprendinių aibės vaidmenį atlieka sveikųjų skaičių nuo 1 iki n visų galimų perstatymų aibė Π_n , o tikslo funkcija apibrėžiama taip:

$$z(p) = \sum_{i=1}^n \sum_{j=1}^n x_{ij} y_{p(i)p(j)}; \quad (1)$$

čia $X = (x_{ij})_{n \times n}$ ir $Y = (y_{kl})_{n \times n}$ yra vadinamosios kainų ir atstumų matricos su sveikosiomis reikšmėmis. Perstatymas $p = (p(1), p(2), \dots, p(n))$ gali būti traktuojamas kaip kurių nors n objektų išdėstymas, paskirstymas į tam tikras pozicijas; šiuo atveju $p(i) \in \{1, 2, \dots, n\}$ žymi pozicijos, į kurią paskirtas i -asis objektas, eilės numerį. KP uždavinys priklauso NP-sunkių kombinatorinio optimizavimo uždavinių klasei. Jis yra aktualus sprendžiant nemažai praktinių uždavinių; taip pat jis yra savotiška eksperimentavimo su optimizavimo algoritmais ir jų efektyvumo testavimo bazė.

DBS algoritmą KP uždaviniui programiškai realizavo A. Misevičius, naudodamas programavimo kalbą *Pascal*. Sudarytame algoritme, kurio detalizuotas aprašymas pateikiamas 2 pav., sprendiniai, t. y. perstatymai $(p^{(1)}, p^{(2)}, \dots, p^{(i)}, \dots, p^{(B)})$, yra asocijuojami su išvaizduojamomis, dirbtinėmis bitėmis (bičių skaičius yra B). Metafora „ i -oji bitė skrenda (arba yra siunčiama) į jos atitinkamą vietą (nektaro šaltinį)“ realizuoto algoritmo kontekste reiškia: i -asis populiacijos sprendinys ($p^{(i)}$) parenkamas lokalojo pagerinimo (lokaliosios paieškos) procedūrai *Lokaloji_Paieška()*, kuri bando tą sprendinį optimizuoti. Jeigu procedūra *Lokaloji_Paieška()* gauna geresnį sprendinį negu esamas sprendinys ($p^{(i)}$), tai tas sprendinys pakeičiamas nauju geresniu. Iš tiesų, lokalojo pagerinimo procedūros vaidmeniui yra panaudota patobulinta

tabu paieškos procedūra (angl. *enhanced tabu search*) (Misevičius, 2005).

Mūsų realizuotas algoritmas operuoja labai kompaktišku spiečiumi (sprendinių populiacija) (žr. sk. „Kompiuterinių eksperimentų rezultatai“). Visi pradinės populiacijos sprendiniai yra pagerinami algoritmo inicializacijos stadijoje (žr. 2 pav.). Paskui kiekvieno algoritmo ciklo metu atsitiktinai pasirenkamas vienas sprendinys, kuris pagerinamas naudojant procedūrą *Lokaloji_Paieška()* (tai atitinka rekrutų verbuotojų fazę). Atsitiktiniam išrinkimui pakanka generuoti tariamai atsitiktinį sveikąjį skaičių intervale $[1, B]$. Realizuojant laisvųjų agentų fazę ir norint labiau diversifikuoti proceso eigą, pasinaudojama kiek kitokia taisykle, negu nurodyta 1 lentelėje. Mūsų konkrečioje realizuotoje versijoje, išsirenkant sektiną rekrutų verbuotoją, iš pradžių atsitiktiniu būdu atrenkama (su galimais pasikartojimais) R sprendinių (čia R yra apriorinis parametras, reiškiantis rekrutų verbuotojų populiacijoje skaičių, $R = 1, 2, \dots, B$). Tuomet geriausias (tikslo funkcijos reikšmės (z) požiūriu) iš R atrinktų sprendinių ir yra aktualusis rekrutų verbuotojas, kuriuo turi būti sekama esamo ciklo metu.

Akivaizdu, jog gali būti pasirenkami įvairūs rekrutų verbuotojai, einant nuo vieno ciklo prie kito. Vis dėlto remiantis šia pateikta taisykle siekiama daugiau šansų suteikti tiems rekrutų verbuotojams, kurie asocijuojasi su geresniais sprendiniais. Kartu bandoma suderinti ir kombinuoti du esminius aspektus: indeterministinį (stochastinį) ir deterministinį. Atsitiktiniu būdu atrenkant (su galimais pasikartojimais) R sprendinių iš B , tikimybė, kad bus pasirinktas i -asis (pirmasis ($i = 1$), antrasis ($i = 2$) ir t. t.) geriausias populiacijos sprendinys, yra lygi

$$\left(1 - \frac{i-1}{B}\right)^R - \left(1 - \frac{i}{B}\right)^R$$

(Drezner, Misevičius, 2011). Antroje lentelėje parodytos šios tikimybės reikšmės, kai $B = 10$, $R = 1, 2, \dots, B$, $i = 1, 2, \dots, 10$. (Galima pastebėti tikimybių reikšmių mažėjimo tendenciją, mažėjant R ir esant nedidelėms eilės numerio i reikšmėms. Tačiau, kaip tai empiriškai veikia algoritmo rezultatų kokybę, galima nustatyti tik

```

procedure Bičių_Spiečiaus_Imitavimas_Kvadratinio_Paskirstymo_Uždaviniui;
//pradiniai duomenys:  $n$  – uždavinio apimtis (objektų skaičius),  $X, Y$  – kainų ir atstumų matricos,
//
//  $B$  – bičių skaičius (spiečiaus/sprendinių populiacijos dydis),  $R$  – rekrutų verbuotojų skaičius,  $C$  – ciklų skaičius,
//
//  $G$  – lokalsios paieškos gylis (paieškos iteracijų skaičius),  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  – lokalsios paieškos gylio reguliavimo koeficientai,
//
//  $ES$  – sprendinių populiacijos entropijos slenkstis
//rezultatai:  $p^*$  – geriausias rastas sprendinys
begin
//inicializacija
sukurti atsitiktiniu būdu pradinę sprendinių populiaciją  $P \subset \Pi_n$  ( $|P|=B$ );
optimizuoti kiekviena populiacijos  $P$  sprendinį  $p$ , panaudojant lokalojo pagerinimo procedūrą
Lokalioji_Paieška( $\alpha_1 G$ );
 $p^* := \arg \min_{p \in P} z(p)$ ; //  $p^*$  – žymi geriausią populiacijos narį (sprendinį)
for ciklas := 1 to  $C$  do begin
//rekrutų verbuotojų fazė
parinkti atsitiktiniu būdu sprendinį  $p \in P$  ir atlikti jo pagerinimo bandymą, panaudojant
lokalojo pagerinimo procedūrą Lokalioji_Paieška( $\alpha_2 G$ );
//laisvųjų agentų fazė
atrinkti atsitiktiniu būdu (su galimais pasikartojimais)  $R$  sprendinių iš  $B$ ;
išrinkti geriausią iš  $R$  atrinktų sprendinių ir atlikti jo pagerinimo bandymą, panaudojant
lokalojo pagerinimo procedūrą Lokalioji_Paieška( $\alpha_3 G$ );
if  $z(\arg \min_{p \in P} z(p)) < z(p^*)$  then  $p^* := \arg \min_{p \in P} z(p)$ ; //išimamas geriausias (iki šiol) surastas sprendinys
if Entropija( $P$ ) <  $ES$  then begin
//žvalgų fazė
(1) sukurti atsitiktiniu būdu naują sprendinių populiaciją  $P$ , perkeltiant į ją geriausią
buvusios populiacijos sprendinį;
(2) optimizuoti kiekviena naujos populiacijos sprendinį, panaudojant
lokalojo pagerinimo procedūrą Lokalioji_Paieška( $\alpha_4 G$ )
endif
endfor
end.

```

Pastabos. 1. Algoritmo variantai priklauso nuo parametro R ($R = 1, 2, \dots, B$), kuriuo apibrėžiamas pasiskirstymas tarp rekrutų verbuotojų ir kitų (likusių) bičių. 2. Sprendinių lokalojo pagerinimo procedūra (Lokalioji_Paieška(-)) naudojama šiais keturiais atvejais: pradinei pagerintai sprendinių populiacijai sudaryti; sprendiniams, susiejamiems su rekrutų verbuotojais (rekrutų verbuotojų sprendiniams), pagerinti; sprendiniams, susiejamiems su laisvaisiais agentais (laisvųjų agentų sprendiniams), pagerinti; naujiems sprendiniams, suformuotiems žvalgymo fazėje, pagerinti. Lokalsios paieškos gylis (intensyvumas), atsižvelgiant į konkretų atvejį, reguliuojamas parametru G ir koeficientais $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, kurie įgyja šias reikšmes: $\alpha_1 = 5, \alpha_2 = 1, \alpha_3 = 3, \alpha_4 = 5$. 3. Populiacijos (spiečiaus) entropija Entropija(P) apskaičiuojama pagal formulę (Taillard, 1995): $Entropija(P) = \sum_{i=1}^n \sum_{j=1}^n e_{ij} / n \log_2 n$, čia $e_{ij} = \begin{cases} 0, & \kappa_{ij} = 0 \\ -\frac{\kappa_{ij}}{|P|} \log_2 \frac{\kappa_{ij}}{|P|}, & \kappa_{ij} \neq 0; \end{cases}$ κ_{ij} žymi skaičių, nurodantį, kiek kartų esamoje sprendinių populiacijoje (spiečiuje) sprendinio, t. y. perstatymo p , elementas, kurio reikšmė lygi i , yra paskirtas į j -ąją perstatymo poziciją. 4. Entropijos slenkstis (ES) yra lygus 0,05.

2 pav. Bičių spiečiaus imitavimo algoritmo KP uždaviniui aprašymas

atlikus kompiuterinius eksperimentinius algoritmo tyrimus su skirtingomis parametro R reikšmėmis.)

Imituojant žvalgų fazę, taip pat nukrypstama nuo kanoninių reikalavimų. Mūsų atveju ši fazė aktyvuojama, kai sprendinių populiacijos entropija, kuri yra sprendinių diversifikacijos lygio matas, tampa mažesnė už duotą slenkstinę reikšmę (ES) (žr. 2 pav.). Tuomet yra konstruojami nauji populiacijos sprendiniai, kuriems kartu atliekamas ir galimas kokybės pagerinimas.

2 lentelė. i-ojo geriausiojo sprendinio išrinkimo tikimybės

| $R \setminus i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | 0,100 | 0,100 | 0,100 | 0,100 | 0,100 | 0,100 | 0,100 | 0,100 | 0,100 | 0,100 |
| 2 | 0,190 | 0,170 | 0,150 | 0,130 | 0,110 | 0,090 | 0,070 | 0,050 | 0,030 | 0,010 |
| 3 | 0,271 | 0,217 | 0,169 | 0,127 | 0,091 | 0,061 | 0,037 | 0,019 | 0,007 | 0,001 |
| 4 | 0,344 | 0,247 | 0,170 | 0,111 | 0,067 | 0,037 | 0,018 | 0,007 | 0,002 | 0,000 |
| 5 | 0,410 | 0,263 | 0,160 | 0,090 | 0,047 | 0,021 | 0,008 | 0,002 | 0,000 | 0,000 |
| 6 | 0,469 | 0,269 | 0,144 | 0,071 | 0,031 | 0,012 | 0,003 | 0,001 | 0,000 | 0,000 |
| 7 | 0,522 | 0,269 | 0,127 | 0,054 | 0,020 | 0,006 | 0,001 | 0,000 | 0,000 | 0,000 |
| 8 | 0,570 | 0,263 | 0,110 | 0,041 | 0,013 | 0,003 | 0,001 | 0,000 | 0,000 | 0,000 |
| 9 | 0,613 | 0,253 | 0,094 | 0,030 | 0,008 | 0,002 | 0,000 | 0,000 | 0,000 | 0,000 |
| 10 | 0,651 | 0,241 | 0,079 | 0,022 | 0,005 | 0,001 | 0,000 | 0,000 | 0,000 | 0,000 |

Kompiuterinių eksperimentų rezultatai

Atliekant kompiuterinius eksperimentus su realizuotu DBS algoritmu, buvo naudojami testiniai pavyzdžiai (gairės) iš viešos KP uždavinio testinių pavyzdžių duomenų bazės (bibliotekos) *QAPLIB* (Burkard ir kt., 1997). Eksperimentai buvo atliekami su dviejų tipų uždaviniais (testiniais pavyzdžiais): atsitiktinai (pagal tolygiojo pasiskirstymo dėsnį) sugeneruotais uždaviniais ir pseudoatsitiktiniu būdu suformuotais uždaviniais, kurie savo prigimtimi yra artimi „realaus pasaulio“ uždaviniais. Pirmojo tipo testinių pavyzdžių pavadinimai yra: *Tai20a*, *Tai25a* ir t. t. (trumpiau *Tai*a*), antrojo tipo pavyzdžių – *Tai20b*, *Tai25b* ir t. t. (trumpiau *Tai*b*).

Pagrindiniu algoritmo efektyvumo įvertinimo kriterijumi buvo pasirinktas gaunamų sprendinių kokybės, t. y. TF, vidutinis santykinis nuokrypis nuo galimai optimalios TF reikšmės (tiksliau, geriausios žinomos TF reikšmės – GŽTFR). Vidutinis santykinis nuokrypis $\bar{\delta}$ apskaičiuojamas formule

$$\bar{\delta} = 100(\bar{z} - \bar{z}) / \bar{z} [\%]; \quad (2)$$

čia \bar{z} yra TF reikšmių vidurkis, kuris apskaičiuojamas atlikus tam tikrą skaičių (10) algoritmo pakartotinių vykdymų; \bar{z} žymi geriausią žinomą TF reikšmę (šios reikšmės pateikiamos bibliotekoje *QAPLIB*). Buvo išbandyta dešimt algoritmo variantų ($R = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$), kai $B = |P| = 10$, $C = 20$, $G = 10$. Tuo pat metu mūsų sudarytas algoritmas buvo lygintas su pagerintu hibridiniu genetiniu algoritmu (PHGA) (angl. *improved hybrid genetic algorithm*) (Misevicius, 2004). Taip pat buvo atlikti papildomi eksperimentai siekiant nustatyti algoritmo valdymo parametrų B , C , G įtaką gaunamiems rezultatams. Papildomų bandymų metu buvo eksperimentuota su tokiais reikšmėmis: $B, G = 10, 15, 20, 25, 30, 35$; $C = 20, 25, 30, 35, 40, 45$ (buvo naudoti testiniai pavyzdžiai: *Tai100a*, *Tai100b*). Gautieji rezultatai pateikiami 3, 4 lentelėse ir 3 paveiksle.

Iš 3 ir 4 lentelių bei 3 pav., a, matyti, kad parametro R (rekrutų verbuotojų skaičiaus) efektas yra dvilypis: atsitiktinių testinių pavyzdžių (*Tai*a*) atveju šiek tiek geresni rezultatai pasiekiami esant didesnėms R reikšmėms ($R = 7, 8, 9, 10$), o realaus pasaulio testiniams pavyzdžiams

(*Tai*b*) kiek geresni rezultatai gaunami, kai R reikšmės yra mažesnės ($R = 1, 2, 3, 4, 5$). Taigi, atsitiktiniams uždaviniams yra svarbiau fokusuoti į geresnius, elitinius sprendinius aspektus, o realaus pasaulio uždaviniams – diversifikacijos, stochastiškumo veiksnys. Kai dėl parametrų B , C , G , galima pažymėti, jog tiek išaugęs spiečiaus dydis (B), tiek padidėjęs ciklų skaičius (C) tik menkai pakoreguoja algoritmo efektyvumą (gaunamų sprendinių kokybės ($\bar{\delta}$) atžvilgiu) (žr. 3 pav., b, c). Kita vertus, lokaliosios paieškos gylis (intensyvumas) (G) turi gana reikšmingą įtaką algoritmo gaunamiems rezultatams (žr. 3 pav., d).

Rezultatai patvirtina tai, jog gaunamų sprendinių kokybė priklauso nuo konkretaus uždavinio tipo ir specifinių ypatybių; vis dėlto abiejų skirtingų tirtų uždavinių tipų atveju atskiri DBS algoritmo variantai (būtent $R = 2, 3, 4, 5$) pasirodė esą, nors ir nedaug, geresni nei algoritmas PHGA, kuris yra vienas iš efektyviausių KP uždavinio sprendimo euristinių algoritmų. Penktoje lentelėje pateikiami daugiau euristinių algoritmų apimančio palyginimo rezultatai atskiriems KP uždavinio testiniams pavyzdžiams. Papildomai panaudoti šie jau anksčiau sukurti algoritmai: konstrukcinis algoritmas (KA), lokalioji paieška (LP), iteratyvioji lokalioji paieška (ILP), tabu paieška (TP), iteratyvioji tabu paieška (ITP), atkaitinimo modeliavimas (AM), skruzdėlių kolonijos imitavimas (SKI), genetinis algoritmas (GA), hibridinis genetinis algoritmas (HGA). Iš tiesų algoritmų KA, LP vykdymo laikas yra mažesnis, negu nurodytas, tačiau ir jų rezultatai yra daug blogesni negu DBS. Kitų algoritmų valdymo parametrai buvo parinkti taip, kad vykdymo laikas būtų panašus*. Taigi, išplėstinio palyginimo rezultatai patvirtina aukštą DBS algoritmo efektyvumo lygį.

* Konkrečios algoritmų parametrų reikšmės, kaip ir palyginime naudoti kiti algoritmai, čia nedetalizuojami dėl ribotos straipsnio apimties. (Reikėtų tik pabrėžti, jog pasiūlytas DBS algoritmas savo prigimtimi yra artimiausias hibridiniams genetiniams algoritmams. Ir vienu, ir kitu gali būti naudojami tie patys sprendinių lokaliojo pagerinimo algoritmai; valdymo parametrai taip pat yra panašūs, pvz., bičių spiečiaus dydis ir ciklų skaičius DBS algoritme ir atitinkamai populiacijos dydis ir generacijų skaičius genetiniame algoritme.)

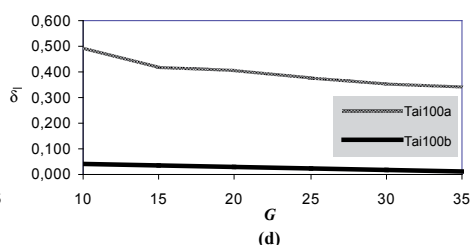
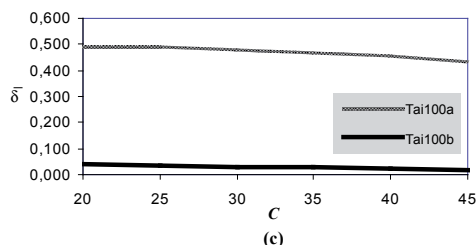
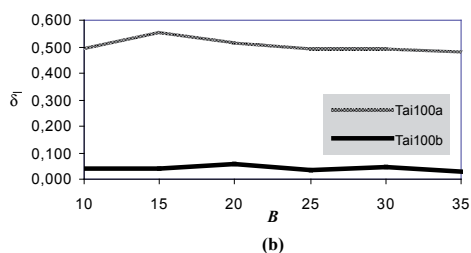
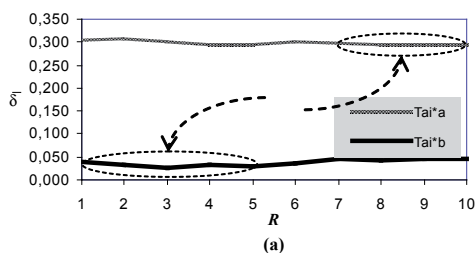
3 lentelė. DBS algoritmo variantų ir genetinio algoritmo (PHGA) palyginimo rezultatai (I)

| Testinis pavyzdys* | GŽTFR (\bar{z}) | $\bar{\delta}$ | | | | | | | | | | | Laikas (sek.)** | |
|--------------------|---------------------|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------------|----|
| | | R=1 | R=2 | R=3 | R=4 | R=5 | R=6 | R=7 | R=8 | R=9 | R=10 | PHGA | | |
| Tai20a | 703482 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 22 |
| Tai25a | 1167256 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,037 | 40 |
| Tai30a | 1818146 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 72 |
| Tai35a | 2422002 | 0,085 | 0,100 | 0,076 | 0,076 | 0,057 | 0,057 | 0,055 | 0,033 | 0,033 | 0,033 | 0,076 | 90 | |
| Tai40a | 3139370 | 0,292 | 0,293 | 0,300 | 0,331 | 0,322 | 0,337 | 0,314 | 0,270 | 0,270 | 0,270 | 0,340 | 100 | |
| Tai50a | 4938796 | 0,531 | 0,561 | 0,565 | 0,548 | 0,563 | 0,563 | 0,593 | 0,572 | 0,572 | 0,572 | 0,573 | 110 | |
| Tai60a | 7205962 | 0,585 | 0,624 | 0,604 | 0,598 | 0,575 | 0,566 | 0,566 | 0,607 | 0,614 | 0,614 | 0,601 | 120 | |
| Tai80a | 13499184 | 0,700 | 0,660 | 0,666 | 0,628 | 0,643 | 0,707 | 0,684 | 0,684 | 0,684 | 0,684 | 0,696 | 240 | |
| Tai100a | 21052466 | 0,538 | 0,529 | 0,489 | 0,472 | 0,479 | 0,487 | 0,474 | 0,471 | 0,486 | 0,490 | 0,497 | 480 | |
| Vidurkis: | | 0,303 | 0,307 | 0,300 | 0,295 | 0,293 | 0,302 | 0,298 | 0,293 | 0,295 | 0,296 | 0,313 | | |

* testinio pavyzdžio pavadinime esantis skaitmuo nurodo uždavinio apimtį (objektų skaičių); ** pateikiamas vidutinis vieno algoritmo pakartotinio vykdymo laikas sekundėmis (eksperimentams naudotas 3 GHz dažnio asmeninis kompiuteris)

4 lentelė. DBS algoritmo variantų ir genetinio algoritmo (PHGA) palyginimo rezultatai (II)

| Testinis pavyzdys | GŽTFR (\bar{z}) | $\bar{\delta}$ | | | | | | | | | | | Laikas (sek.) | |
|-------------------|---------------------|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|-----|
| | | R=1 | R=2 | R=3 | R=4 | R=5 | R=6 | R=7 | R=8 | R=9 | R=10 | PHGA | | |
| Tai20b | 122455319 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 1,6 |
| Tai25b | 344355646 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 2,1 |
| Tai30b | 637117113 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 5,0 |
| Tai35b | 283315445 | 0,053 | 0,031 | 0,012 | 0,026 | 0,026 | 0,026 | 0,026 | 0,026 | 0,012 | 0,019 | 0,015 | 8,6 | |
| Tai40b | 637250948 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 | 28 | |
| Tai50b | 458821517 | 0,030 | 0,018 | 0,006 | 0,007 | 0,004 | 0,005 | 0,005 | 0,010 | 0,010 | 0,011 | 0,020 | 45 | |
| Tai60b | 608215054 | 0,033 | 0,030 | 0,034 | 0,041 | 0,030 | 0,039 | 0,039 | 0,043 | 0,041 | 0,033 | 0,019 | 57 | |
| Tai80b | 818415043 | 0,044 | 0,033 | 0,035 | 0,029 | 0,030 | 0,091 | 0,143 | 0,147 | 0,148 | 0,142 | 0,089 | 70 | |
| Tai100b | 1185996137 | 0,032 | 0,033 | 0,027 | 0,023 | 0,046 | 0,050 | 0,040 | 0,041 | 0,049 | 0,049 | 0,055 | 95 | |
| Tai150b | 498896643 | 0,191 | 0,191 | 0,162 | 0,187 | 0,172 | 0,156 | 0,210 | 0,169 | 0,202 | 0,194 | 0,151 | 420 | |
| Vidurkis: | | 0,038 | 0,034 | 0,028 | 0,031 | 0,031 | 0,037 | 0,046 | 0,044 | 0,046 | 0,045 | 0,035 | | |



3 pav. Eksperimentų rezultatai, gauti sprendžiant KP uždavinį su skirtingomis DBS algoritmo parametru R, B, C, G reikšmėmis

5 lentelė. Skirtingų algoritmų palyginimo rezultatai

| Testinis pavyzdys* | GŽTFR (ž) | $\bar{\delta}$ | | | | | | | | | | | Laikas (sek.)** |
|--------------------|------------|----------------|-------|-------|-------|--------------|-------|-------|-------|-------|--------------|--------------|-----------------|
| | | KA | LP | ILP | TP | ITP | AM | SKI | GA | HGA | PHGA | DBS | |
| Tai50a | 4938796 | 2,568 | 1,763 | 0,892 | 0,683 | 0,560 | 1,424 | 0,687 | 1,562 | 0,579 | 0,573 | 0,563 | 110 |
| Tai50b | 458821517 | 3,433 | 0,316 | 0,150 | 0,068 | 0,034 | 0,239 | 0,055 | 0,890 | 0,021 | 0,020 | 0,006 | 45 |
| Tai60a | 7205962 | 2,852 | 1,846 | 1,003 | 0,897 | 0,595 | 1,396 | 0,864 | 1,443 | 0,615 | 0,601 | 0,575 | 120 |
| Tai60b | 608215054 | 3,571 | 0,400 | 0,070 | 0,069 | 0,085 | 0,309 | 0,065 | 0,934 | 0,020 | 0,019 | 0,034 | 57 |
| Tai80a | 13499184 | 2,730 | 1,956 | 1,153 | 0,816 | 0,613 | 1,672 | 0,902 | 1,648 | 0,684 | 0,696 | 0,643 | 240 |
| Tai80b | 818415043 | 4,519 | 0,337 | 0,234 | 0,254 | 0,167 | 0,401 | 0,216 | 0,998 | 0,066 | 0,089 | 0,035 | 70 |
| Tai100a | 21052466 | 2,567 | 1,829 | 1,221 | 0,778 | 0,465 | 1,105 | 0,871 | 1,517 | 0,533 | 0,497 | 0,479 | 480 |
| Tai100b | 1185996137 | 4,321 | 0,244 | 0,322 | 0,330 | 0,061 | 0,390 | 0,214 | 1,011 | 0,050 | 0,055 | 0,027 | 95 |
| Vidurkis: | | 3,320 | 1,086 | 0,631 | 0,487 | 0,323 | 0,867 | 0,484 | 1,250 | 0,321 | 0,319 | 0,295 | |

* testinio pavyzdžio pavadinime esantis skaitmuo nurodo uždavinio apimtį (objektų skaičių);

** pateikiamas maksimalus vieno algoritmo pakartotinio vykdymo laikas sekundėmis

Išvados

Šiame darbe nagrinėti klausimai, susiję su kolektyvinio intelekto naudojimo galimybėmis, būtent bičių spiečių veiksenos imitavimu sprendžiant (kombinatorinio) optimizavimo uždavinius. Akcentuota, jog grupinės intelektikos metodų (kokie yra dirbtinių bičių spiečių algoritmai) svarbiausi pozityvieji aspektai yra: kolektyvinė veikseną, dalijimasis informacija, darbų ir funkcijų pasidalijimas. Kolektyvinė veikseną (iš vienos pusės) ir kolektyvą sudarančių narių (agentų) autonimiškumas ir intelektualumas priimančiam individualius sprendimus konkrečiomis aplinkybėmis (iš kitos pusės) yra tie vienas kitą papildantys ir sustiprinantys veiksniai, kurie leidžia atskleisti papildomą, didesnę metaeuristinę optimizavimo metodų energiją.

LITERATŪRA

BÄCK, Thomas; FOGEL, David B.; MICHALEWICZ, Zbigniew (ed.) (1997). *Handbook of Evolutionary Computation*. Bristol–Philadelphia–New York: Institute of Physics Publishing, Oxford University Press. 1130 p.

BEASLEY, John E. (2002). Population heuristics. In P. M. Pardalos, M. G. C. Resende (eds). *Handbook of Applied Optimization*. Oxford–New York: Oxford University Press, p. 138–157.

BONABEAU, Eric; DORIGO, Marco; THERAULAZ, Guy (1999). *Swarm Intelligence. From Natural to Artificial Systems*. New York–Oxford: Oxford University Press. 320 p.

Yra pasiūlyta dirbtinio bičių spiečiaus algoritmo realizacija kvadratinio paskirstymo uždaviniui – vienam iš sudėtingų kombinatorinio optimizavimo uždavinių. Gauti preliminarūs šio algoritmo rezultatai atitinka išankstinius teorinius samprotavimus ir lūkesčius, kad potencialios natūralių bičių spiečių veiksenos imitavimo galimybės yra geros, ir patvirtina prielaidas apie didelį tokiu imitavimu besiremiančių DBS algoritmų efektyvumą.

DBS algoritmų efektyvumą būtų galima dar padidinti, tiksliau įvertinant sprendžiamo uždavinio specifiką, toliau tobulinant tiek pačią bičių spiečių imitavimo koncepciją, tiek sprendinių lokalojo pagerinimo procedūras, kurios ypač svarbios hibridiniuose DBS algoritmuose. Tokių hibridinių kolektyvinių sistemų elgesį imituojančių algoritmų sudarymas galėtų būti naujų tyrinėjimų tema.

BURKARD, Rainer; KARISCH, Stefan; RENDL, Franz (1997). QAPLIB – a quadratic assignment problem library. *Journal of Global Optimization*, vol. 10, p. 391–403. Žr. taip pat prieigą per internetą: <<http://www.seas.upenn.edu/qaplib/>>.

ÇELA, Eranda (1998). *The Quadratic Assignment Problem: Theory and Algorithms*. Dordrecht: Kluwer. 304 p.

DAVIDOVIĆ, Tatjana; RAMLJAK, Dušan; ŠELMIĆ, Milica; TEODOROVIĆ, Dušan (2011). Bee colony optimization for the p -center problem. *Computers & Operations Research*, vol. 38, p. 1367–1376.

DORIGO, Marco; STÜTZLE, Thomas (2002). The ant colony optimization metaheuristic: algorithms, applications, and advances. In F. Glover, G. Kochenberger (eds). *Handbook of Metaheuristics*. Norwell: Kluwer, p. 251–285.

DREZNER, Zvi; MISEVICIUS, Alfonsas (2011). *Enhancing the performance of hybrid genetic algorithms by differential improvement*. Working Paper (pateikta recenzuoti).

GLOVER, Fred; LAGUNA, Manuel; MARTI, Rafael (2002). Scatter search and path relinking: advances and applications. In F. Glover, G. Kochenberger (eds). *Handbook of Metaheuristics*. Norwell: Kluwer, p. 1–36.

KARABOGA, Dervis (2005). *An idea based on honey bee swarm for numerical optimization*. Technical Report-TR06, Erciyes University, Turkey.

KARABOGA, Dervis; AKAY, Bahriye (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, vol. 31, p. 61–85.

KARABOGA, Dervis; BASTURK, Bahriye, (2007). A Powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, vol. 39, p. 459–471.

KENNEDY, James; EBERHART, Russel (1995). *In Particle swarm optimization. Proceedings of the 4th IEEE International Conference on Neural Networks*, Perth, Australia, p. 1942–1948.

LOIOLA, Eliane M.; DE ABREU, Nair M.; BO-AVENTURA-NETTO, Paolo O.; HAHN, Peter; QUERIDO, Tania (2007). A survey for the quadratic assignment problem. *European Journal of Operational Research*, vol. 176, p. 657–690.

LUČIĆ, Panta; TEODOROVIĆ, Dušan (2001). Bee system: modeling combinatorial optimization

transportation engineering problems by swarm intelligence. In *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis* (Sao Miguel, Azores Islands), p. 441–445.

MISEVICIUS, Alfonsas (2004). An improved hybrid genetic algorithm: new results for the quadratic assignment problem. *Knowledge-Based Systems*, vol. 17, p. 65–73.

MISEVICIUS, Alfonsas (2005). A tabu search algorithm for the quadratic assignment problem. *Computational Optimization and Applications*, vol. 30, p. 95–111.

PAN, Quan-Ke; TASGETIREN, Fatih M.; SUGANTHAN, Ponnuthurai N.; CHUA, Tay Jin (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, vol. 181, p. 2455–2468.

SABAT, Samrat L.; UDGATA, Siba K.; ABRAHAM, Ajith (2010). Artificial bee colony algorithm for small signal model parameter extraction of MES-FET. *Engineering Applications of Artificial Intelligence*, vol. 23, p. 689–694.

TAILLARD, Éric D. (1995). Comparison of iterative searches for the quadratic assignment problem. *Location Science*, vol. 3, p. 87–105.

YONEZAWA, Yasuo; KIKUCHI, Takashi (1995). Ecological algorithm for optimal ordering used by collective honey bee behavior. In *Proceedings of the Seventh International Symposium on Micro Machine and Human Science* (Nagoja, Japan). New York: IEEE, p. 249–255.

ZÄPFEL, Günther; BRAUNE, Roland; BÖGL, Michael (2010). *Metaheuristic Search Concepts. A Tutorial with Applications to Production and Logistics*. Berlin–Heidelberg: Springer. 316 p.

BEE SWARM INTELLIGENCE IN (COMBINATORIAL) OPTIMIZATION

Alfonsas Misevičius, Jonas Blonskis, Vytautas Bukšnaitis

Summary

In this paper, we discuss some issues related to the innovative intelligent optimization methods. More precisely, we are concerned with the bee colony optimization approach, which is inspired by the behaviour of natural swarms of honey bees. Both the conceptual methodological facets of the swarm intelligence paradigm and the aspects of implementation of the artificial bee colony algorithms are considered. In particular, we introduce an imple-

mentation of the artificial bee colony optimization algorithm for the well-known combinatorial optimization problem of quadratic assignment (QAP). The results of computational experiments with different variants of the implemented algorithm are also presented and discussed. Based on the obtained results, it is concluded that the proposed algorithm may compete with other efficient heuristic techniques.