VILNIAUS UNIVERSITETAS

Linas
STRIPINIS

Globaliojo optimizavimo algoritmų, nereikalaujančių išvestinių informacijos, kūrimas, tobulinimas ir realizacija

DAKTARO DISERTACIJOS SANTRAUKA

Gamtos mokslai,
Informatika (N 009)

Vilnius, 2021

Disertacija rengta 2016– 2020 metais Vilniaus universitete.

Mokslinis vadovas:
prof. dr. Remigijus Paulavičius (Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Gynimo taryba:
Pirmininkė – **prof. dr. Olga Kurasova** (Vilniaus universitetas, gamtos mokslai, informatika – N 009).
Nariai:
prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, gamtos mokslai, informatika – N 009),
doc. dr. Algirdas Lančinskas (Vilniaus Universitetas, gamtos mokslai, informatika – N 009),
prof. dr. Dmitrij Šešok (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – T 007),
prof. habil. dr. Anatoly Zhigljavsky (Kardifo universitetas, gamtos mokslai, informatika – N 009).

Disertacija ginama viešame 2021 m. vasario mėn. 25 d. 13 val. Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų instituto 203 auditorijoje.
Adresas: Akademijos g. 4, LT-08412, Vilnius, Lietuva
Disertacijos santrauka išsiuntinėta 2021 m. sausio mėn. 25 d.
Disertaciją galima peržiūrėti Vilniaus univeristeto bibliotekoje ir VU interneto svetainėje adresu:
https://www.vu.lt/naujienos/ivykiu-kalendorius

VILNIUS UNIVERSITY

Linas
STRIPINIS

Improvement, development and implementation of derivative-free global optimization algorithms

SUMMARY OF DOCTORAL DISSERTATION

Natural Sciences,
Informatics (N 009)

Vilnius, 2021

This dissertation was written between 2016 and 2020 at Vilnius University.

Academic supervisor:
Prof. Dr. Remigijus Paulavičius (Vilnius University, Natural Sciences, Informatics – N 009).

Defence Panel:
Chairman – Prof. Dr. Olga Kurasova (Vilnius University, Natural Sciences, Informatics – N 009).
Members:
Prof. Habil. Dr. Gintautas Dzemyda (Vilnius University, Natural Sciences, Informatics – N 009),
Assoc. Prof. Dr. Algirdas Lančinskas (Vilnius University, Natural Sciences, Informatics – N 009),
Prof. Dr. Dmitrij Šešok (Vilnius Gediminas Technical University, Technological Sciences,, Informatics Engineering – T 007),
Prof. Habil. Dr. Anatoly Zhigljavsky (Cardiff University, Natural Sciences, Informatics – N 009).

The dissertation shall be defended at a public meeting of the Dissertation Defence Panel at 1 p. m. on the 25th of February, 2021 in Room 203 of the Institute of Data Science and Digital Technologies of Vilnius University. Address: Akademijos street 4, LT-08412, Vilnius, Lithuania.
The summary of the doctoral dissertation was distributed on the 25th of January, 2021. The text of this dissertation can be accessed at the library of Vilnius University, as well as on the website of Vilnius University: www.vu.lt/lt/naujienos/ivykiu-kalendorius

1. Įvadas

Dėl paprastumo ir efektyvumo, išvestinių informacijos nereikalaujantis globalios paieškos DIRECT (DIviding RECTangles) algoritmas sulaukė daug optimizavimo bendruomenės dėmesio, buvo pasiūlytos įvairios naujos algoritmo idėjos ir modifikacijos. Tačiau daugelio DIRECT tipo algoritmų efektyvumas blogėja sprendžiant daugybę lokalių minimumų turinčias problemas arba kai norima rasti kur kas tikslesnę sprendinio reikšmę. Siekiant sumažinti algoritmo trūkumus šis darbas pateikia naują potencialiai optimalių hiper-stačiakampių atrankos schemą DIRECT tipo algoritmuose. Išsamus eksperimentinis tyrimas atskleidė sukurto algoritmo potencialą, ypač sprendžiant sudėtingesnius optimizavimo uždavinius.

Tačiau originalus DIRECT algoritmas sprendžia problemas tik su intervaliniais ribojimais ir dėl šios priežasties algoritmo pritaikomumas yra labai ribotas, kai praktinės optimizavimo problemos dažnai yra apribotos įvairaus tipo ribojimų funkcijomis. Pirmųjų algoritmo modifikacijų rezultatai nebuvo konkurencingi, palyginus su kitais išvestinių nereikalaujančiais optimizavimo metodais. Tik pastaraisiais metais buvo pasiūlyta keletas perspektyvių DIRECT tipo versijų uždaviniams su ribojimais spręsti. Šiame darbe pateikiami du skirtingi metodai uždaviniams su ribojimais spręsti, o vieną iš jų galima pritaikyti sprendžiant problemas su paslėptais apribojimais. Išsamūs eksperimentiniai tyrimai atskleidė siūlomų algoritmų potencialumą ir efektyvumų, palyginus su kitais egzistuojančiais DIRECT tipo algoritmais sprendžiant globaliojo optimizavimo problemas su įvairaus tipo ribojimais, įskaitant svarbias praktines problemas.

Šiuolaikinės problemos dažnai negali būti efektyviai išspręstos naudojant ir pačius greičiausius nuoseklius kompiuterius. Todėl lygiagretinimo technologijos galėtų išspręsti kylančią problemą. Tačiau iteracinė `DIRECT` tipų metodų prigimtis riboja algoritmo galimybes efektyviam lygiagretinimui, ir yra žinomos tik kelios lygiagrečios `DIRECT` versijos. Mūsų žiniomis, visos esamos lygiagrečios `DIRECT` tipo versijos yra skirtos globaliojo optimizavimo problemoms su intervaliniais ribojimais spręsti. Kadangi sukurta potencialiai optimali hiper-stačiakampių schema, atlieka atranką du kartus per iteraciją ir pasirenkamas didesnis skaičius dalinamų sričių, algoritmai sukurti šiame darbe atrodo perspektyvesni lygiagretinti, palyginus su pirmine `DIRECT` algoritmo versija. Šiame darbe pristatomi pirmieji lygiagretūs `DIRECT` tipo algoritmai, skirti globaliojo optimizavimo problemoms su įvairaus tipo apribojimais spręsti.

2. Tyrimo sritis ir problemos aktualumas

Išvestinių informacijos nereikalaujantis `DIRECT` algoritmas [21] yra gerai žinomas ir plačiai naudojamas globaliojo optimizavimo problemoms spręsti. Be to, neseniai atliktas tyrimas [46] parodė, kad išvestinių nereikalaujančių algoritmų klasėje `DIRECT` yra vienas efektyviausių algoritmų. Algoritmas yra modifikacija klasikinio Lipšico optimizavimo [31, 32, 44, 45, 49, 53] modifikacija, kai informacija apie Lipšico konstantą nėra reikalinga. Dėl šios savybės `DIRECT` tipo metodai yra plačiai naudojami sprendžiant įvairias realaus pasaulio optimizavimo problemas [1, 2, 4, 6, 7, 11, 26, 35, 38].

Nepaisant to, atlikti tyrimai atskleidė, kad algoritmas turi du trūkumus [20, 34, 35, 59]. Pirmasis – lėtas globalaus minimumo srities radimas sprendžiant problemas, kurios turi daugybę lokalių sprendinių, antrasis – lėtas konvergavimas į globalųjį sprendinį dideliu tikslumu. Kitas svarbus trūkumas yra algoritmo ribotumas sprendžiant problemas tik su intervaliniais ribojimais. DIRECT algoritmo pritaikymas uždaviniams su kitokio tipo ribojimais yra sparčiai vystoma sritis ir egzistuoja tik kelios algoritmo modifikacijos. Todėl pagrindiniai šio darbo tikslai yra sukurti algoritmus, kurie sustiprintų DIRECT trūkumus ir gebėtų spręsti problemas, su įvairaus tipo ribojimais.

Lipšico globaliojo optimizavimo metodai atlieka daug skaičiavimų, todėl paprastas būdas juos pagreitinti yra jų lygiagretinimas [38, 17, 18, 30, 32]. Tačiau iteracinė DIRECT tipo algoritmų prigimtis riboja efektyvaus lygiagretinimo galimybes [10, 15, 13, 16, 14, 41, 58]. Sukurtos tik kelios lygiagrečios DIRECT tipo algoritmų versijos, daugiausia tos pačios tyrėjų grupės. Buvusių versijų autoriai daugiausia sutelkė dėmesį į lygiagretaus efektyvumo gerinimą kurdami daugiau skaičiavimų kiekvienoje iteracijoje, naudodami „aggressive“ DIRECT versiją, kuri sušvelnina atrankos kriterijus, kiekvienoje iteracijoje pasirenkant ir dalijant kiekvieno skirtingo skersmens geriausią hiper-stačiakampį [1]. Optimizavimo požiūriu šis metodas nėra patrauklus, nes algoritmas tyrinėja daugybę „nereikalingų“ (ne potencialiai optimalių) hiper-stačiakampių [9, 10]. Kiek yra žinoma, visos esamos lygiagrečios DIRECT tipo algoritmų versijos yra skirtos tik globaliojo optimizavimo problemoms su stačiakampio sritimis spręsti. Kita šiame darbe tiriama problema yra lygiagrečių DIRECT tipo algoritmų kūrimas globaliojo optimizavimo problemoms su įvairiais ribojimais spręsti.

3. Tyrimo objektas

Šios disertacijos darbo objektas yra nuoseklūs ir lygiagretūs `DIRECT` tipo algoritmai skirti globaliojo optimizavimo problemoms su bendraisiais apribojimais ir jų pritaikymas praktinėms optimizavimo problemoms spręsti.

4. Darbo tikslai ir uždaviniai

Disertacijoje išsikelti darbo tikslai:

1. Padidinti pažangiausių `DIRECT` tipo globaliojo optimizavimo algoritmų efektyvumą sprendžiant optimizavimo problemas su daugybe lokalių minimumų ir tada, kai reikia didelio tikslumo sprendimo.
2. Išplėsti `DIRECT` tipo globaliojo optimizavimo algoritmus problemoms su bendraisiais ir paslėptaisiais apribojimais spręsti.
3. Sukurti efektyvius išvestinių nereikalaujančius atvirojo kodo algoritmus, atsižvelgiant į algoritminius patobulinimus: efektyvias duomenų struktūras ir lygiagretinimo metodus.

Siekiant įgyvendinti išsikeltus tikslus užsibrėžti šie uždaviniai:

1. Įvertinti esamų pažangiausių `DIRECT` tipo visuotinio optimizavimo algoritmų veikimą ir nustatyti jų trūkumus.
2. Tobulinti esamus ir kurti naujus algoritmus, atsižvelgiant į nustatytus trūkumus.
3. Sukurti bendrųjų apribojimų valdymo strategiją `DIRECT` algoritminėje sistemoje.
4. Sukurti pagalbinėmis funkcijomis pagrįstą `DIRECT` tipo algoritmą optimizavimo problemoms su paslėptais apribojimais spręsti.

5. Realizuoti efektyvias nuoseklias ir lygiagrečias siūlomų algoritmų versijas ir palyginti jų veikimą su kitais egzistuojančiais metodais.
6. Efektyviai spręsti sudėtingas praktines (potencialiai “juodosios dėžės ”) optimizavimo problemas naudojant įdiegtus ir atvirai prieinamus įrankius.

5. Tyrimo metodai

Analizuojant gautus mokslinius rezultatus globaliojo optimizavimo ir lygiagretinimo srityse buvo naudota algoritmų teorija, konvergavimo analizė, lygiagrečiųjų skaičiavimų teorija, informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai. Pasitelkus eksperimentinio tyrimo metodiką atlikta statistinė tyrimų rezultatų analizė, taip pat algoritmų efektyvumo vertinimui pasitelkti duomenų [28] ir našumo [8] profiliai.

6. Mokslinis darbo naujumas

Pagrindiniai mokslinio darbo naujumai yra šie:

1. Remiantis teoriniais ir eksperimentiniais tyrimais, buvo nustatyta, kad originalaus pasiūlyto `DIRECT` algoritmo efektyvumas blogėja sprendžiant optimizavimo problemas, kurios turi daugybę lokalių minimumų ir tada kai reikia didelio sprendinio tikslumo. Norint pašalinti anksčiau minėtus trūkumus, buvo sukurta nauja potencialiai optimalių hiper-stačiakampių atrankos strategija, o siūlomai schemai nereikia jokių papildomų parametrų ar lokalios paieškos metodų. Pasiūlyta modifikacija davė ženkliai geresnių

rezultatus nei originali `DIRECT` versija sprendžiant daug sudėtingesnius uždavinius: aukštesnės dimensijos; ir problemas su daugybę lokalių optimumų. Taip pat pasiūlytas algoritmas gali pagerinti galutinio sprendimo kokybę atlikdamas mažiau funkcijų įvertinimų.

2. Originalus `DIRECT` algoritmas yra deterministinis metodas skirtas globaliojo optimizavimo uždaviniams su intervaliniais ribojimais spręsti. Šiame darbe sukurtos algoritmo modifikacijos:
   2.1 Pasiūlyta algoritmo versija skirta netiesinio programavimo optimalių sprendinių paieškai. Algoritmas veikia dviem fazėmis, kai pirmoje fazėje algoritmas, naudojantis apribojimų funkcijų informaciją, bando surasti taškus, kurie tenkina apribojimų funkcijas, o antrojoje – naudojantis pagalbinių funkcijų metodu, kuris sujungia tikslo ir apribojimo funkcijų informaciją, siekia rasti globalųjį sprendinį.
   2.2 Pasiūlyta algoritmo versija skirta spręsti problemas su paslėptais ribojimais. Algoritmas veikia dviem fazėmis: pirmojoje fazėje algoritmas tolygiai dalindamas hiper-stačiakampį bando identifikuoti nepaslėptas optimizavimo sritis, o antrojoje – naudojasi baudos funkcijos metodu, kuris įvertina būtinus baudos parametrus srityse, kuriose tikslo funkcija negali būti vertinama.
3. Sukurti algoritmai buvo realizuoti naudojant efektyvesnes duomenų saugojimo ir organizavimo struktūras. Be to, šiame darbe pirmą kartą pristatyta lygiagreti `DIRECT` algoritmo versija, skirta spręsti globaliojo optimizavimo problemas su

įvairaus tipo ribojimais.

4. Surinkta `DIRECTlib` – testinių ir praktinių uždavinių internetinė biblioteka su įvairaus tipo ribojimais, skirta `DIRECT` tipo optimizavimo algoritmams testuoti ir lyginti. Linas Stripinis, and Remigijus Paulavičius. (2020). `DIRECTlib` - a library of global optimization problems for `DIRECT`-type methods (Version 1.2) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.1403547.
   URL:
   `https://zenodo.org/record/3948890#.XyVOk5dR2Uk`

Sukurti algoritmai buvo realizuoti programine įranga „`MatLab`“:

- Globaliojo optimizavimo problemoms spręsti, kai leistina sritis yra stačiakampio sritis. Realizuota nuosekli ir lygiagreti `DIRECT-GL` versija, taip pat tarpinės `DIRECT-G` ir `DIRECT-L` versijų realizacijos;
  URL: `https://github.com/LinasStripinis/DIRECT-GL`
- Netiesinio programavimo problemoms spręsti, realizuota nuosekli ir lygiagreti `DIRECT-GLce` versija, taip pat keturios tarpinės `DIRECT-GLc` ir `DIRECT-GLce-min` versijų realizacijos;
  URL: `https://github.com/LinasStripinis/DIRECT-GL`
- Problemoms su paslėptais ribojimais spręsti, realizuota nuosekli `DIRECT-GLh` versija;
  URL: `https://github.com/LinasStripinis/DIRECT-GL`

7. Dalyvavimas mokslinėse programose

Linas Stripinis dalyvavo Lietuvos mokslo tarybos finansuojamoje programoje „Dviejų lygmenų optimizavimo algoritmų kūrimas ir taikymai“ (2017 - 2020). (Nr. P-MIP-17-60)

8. Ginamieji teiginiai

1. Naujas dviejų žingsnių išrinkimo strategija pagrįstas algoritmas (`DIRECT-GL`) veikia geriausiai tarp `DIRECT` tipo metodų sprendžiant sudėtingiausias neišgaubtas problemas iš `DIRECTlib` ir greičiausiai konverguoja į sprendinį dideliu tikslumu.
2. Naujas pagalbine funkcija grįstas algoritmas (`DIRECT-GLce`) problemoms su bendraisiais apribojimais yra vidutiniškai efektyviausias tarp `DIRECT` tipo metodų sprendžiant sudėtingiausias `DIRECTlib` problemas: didelės dimensijos; apribotas netiesiniais ribojimais; ir lygybiniais ribojimais.
3. Naujas pagalbine funkcija grįstas algoritmas praktiškai orientuotoms problemoms su paslėptais apribojimais (`DIRECT-GLh`) yra efektyviausias `DIRECT` tipo metodas, atsižvelgiant į funkcijų vertinimų skaičių ir greitį, sprendžiant problemas iš `DIRECTlib`.
4. Sukurtuose algoritmuose taikomos dinaminės duomenų struktūros sumažina skaičiavimų skaičių ir supaprastina potencialiai optimalių hiper-stačiakampių išrinkimo žingsnį; tokiu būdu realizuotų algoritmų greitis paspartėja daugiau nei du kartus.
5. Taikoma šeimininko ir darbuotojų lygiagreti schema, sukurta pirmajame lygiagrečiame *SPMD* algoritme problemoms su bendraisiais apribojimais, geba išsaugoti determinizmą, leidžia pasiekti gerą lygiagretų efektyvumą sprendžiant pasirinktas problemas.

9. Darbo rezultatų aprobavimas

Pagrindiniai tyrimo rezultatai pristatyti tarptautinėse konferencijose. Rezultatai pristatyti šių tarptautinių konferencijų plenarinėse sesijose:

1. Paulavičius, R., Stripinis, L. and **Žilinskas**, J. `DIRECT`-type algorithms for constrained global optimization, EUROPT 2017: 15th EUROPT Workshop on Advances in Continuous Optimization, July 12-14, 2017. Montreal, Canada (Plenary Session)
2. **Stripinis, L.**, Žilinskas, J. and Paulavičius, R. Improved `DIRECT`-type algorithm for constrained global optimization problems, EUROPT 2018: 16th EUROPT Workshop on Advances in Continuous Optimization, July 12-13, 2018. Almeria, Spain (Plenary Session)
3. **Stripinis, L.** Paulavičius, R., and Žilinskas, J. Importance of optimization techniques for the social sciences, The International EURO mini Conference Modelling and Simulation of Social-Behavioural Phenomena in Creative Societies, September 18–20, 2019. Vilnius, Lithuania (Plenary Session)

Rezultatai pristatyti šių tarptautinių konferencijų stendinėse sesijose:

4. **Stripinis, L.** and Paulavičius, R. Improved `DIRECT`-type algorithms for generally constrained global optimization problems, 9th International workshop on Data Analysis Methods for Software Systems (DAMSS), November 30 - December 2, 2017. Druskininkai, Lithuania (Poster Session)
5. **Stripinis, L.** and Paulavičius, R. Improved `DIRECT`-type algorithms for generally constrained global optimization

problems, 10th International workshop on Data Analysis Methods for Software Systems (DAMSS), November 29 – December 1, 2018. Druskininkai (Poster Session)

10. Disertacijos struktūra

Disertaciją sudaro įvadas, penki skyriai, literatūros sąrašas ir publikacijų sąrašas. Bendra disertacijos apimtis – 154 p., įskaitant 27 paveikslus ir 17 lentelių. Disertacijoje cituojami 125 informacijos šaltiniai.

Pirmame skyriuje aprašomas tyrimo sritis ir problemos aktualumas, pateikiami darbo tikslai ir uždaviniai, tyrimo objektai, aprašomi tyrimo metodai ir tyrimo aprobavimas. Antrame skyriuje pateikiami teoriniai `DIRECT` tipo algoritmų pagrindai. Trečiame skyrius pristatoma `DIRECT` algoritmo modifikacija uždaviniams su intervaliniais ribojimais spręsti. Ketvirtame skyriuje `DIRECT-GL` algoritmas pritaikomas uždaviniams su įvairaus tipo ribojimais spręsti. Penktame skyriuje pristatoma `DIRECT-GL` algoritmo modifikacija problemoms su paslėptais apribojimais spręsti. Šeštame skyriuje aptariami būdai paspartinti `DIRECT` tipo algoritmus ir pristatomos pirmos lygiagrečios `DIRECT` tipo algoritmų versijos problemos su įvairaus tipo ribojimais. Pagrindiniai disertacijos rezultatai aptariami paskutiniuose skyriuose.

11. `DIRECT` algoritmas

Šioje dalyje analizuojama deterministinių išvestinių nereikalaujančių globaliojo optimizavimo `DIRECT` tipo algoritmų mokslinė literatūra: optimizavimo problema, egzistuojančios modifikacijos, algoritmo pritaikomumas uždaviniams su ribojimais ir žinomi algoritmo trūkumai.

DIRECT algoritmas yra deterministinis metodas skirtas globaliojo optimizavimo uždaviniams spręsti tik su intervaliniais apribojimais:

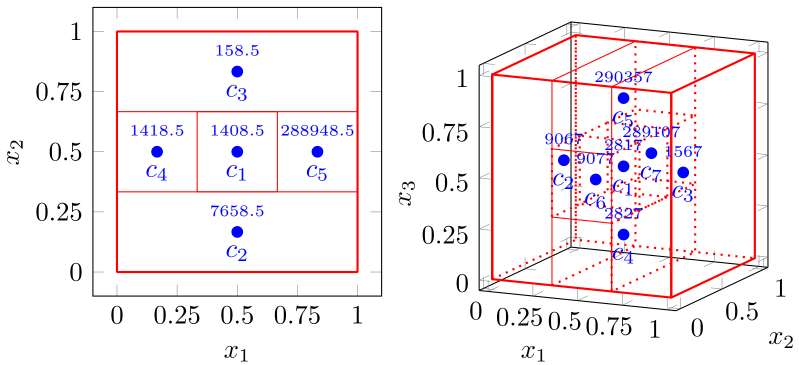
$$\min_{\mathbf{x} \in D} \quad f(\mathbf{x}) \tag{1}$$

kur $f : \mathbb{R}^n \to \mathbb{R}$ žymi tikslo funkciją, o leistina sritis yra stačiakampis $D = [\mathbf{a}, \mathbf{b}] = \{\mathbf{x} \in \mathbb{R}^n : a^j \leq x^j \leq b^j, j = 1, \ldots, n\}$. Be to daroma prielaida, kad tikslo funkcija $f(\mathbf{x})$ yra tolydi Lipšico funkcija, bet gali būti netiesinė, nediferencijuojama, neišgaubta ir su daugybe lokalių sprendinių. Pirminio žingsnio metu DIRECT algoritmas normalizuoja leistinąją sritį D į vienetinį hiper-stačiakampį $\bar{D}$ ir grįžta į originalią erdvę D tik vertinant tikslo funkciją. Nepriklausomai nuo kintamųjų skaičiaus, pirmasis tikslo funkcijos vertinimas atliekamas vienetinio hiper-stačiakampio $\bar{D}$ centro taške $\mathbf{c}_1 = (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$. Tada pradinis hiper-stačiakampis pagal kiekvieną ilgiausią kraštinę yra padalijamas į tris dalis, kaip parodyta 1 paveikslėlyje. Apskaičiuojami nauji taškai ir tikslo funkcija įvertinama $\mid 1/3I^n \pm \mathbf{c}_1 \mid$, kur I^n yra vienetinė matrica ir $j = 2...2n$. Toliau $\bar{D}$ yra padalijamas į $2n + 1$ mažesnius, nepersidengiančius hiper-stačiakampius taip, kad geriausios tikslo funkcijos reikšmės būtų priskirtos didžiausiuose hiper-stačiakampiuose.

Esminis DIRECT žingsnis yra potencialiai optimalių hiper-stačiakampių (POH) išrinkimas. k-oje iteracijoje hiper-stačiakampių aibė apibrėžiamas kaip:

$$\mathcal{H}^k = \{\bar{D}_i^k : i \in \mathbb{I}^k\},$$

kur $\bar{D}_i^k = [\mathbf{a}_i, \mathbf{b}_i] = \{\mathbf{c} \in \mathbb{R}^n : 0 \leq a_i^j \leq c^j \leq b_i^j \leq 1, j = 1, \ldots, n, \forall i \in \mathbb{I}^k\}$ ir $\mathbb{I}^k$ yra visų hiper-stačiakampių $\mathcal{H}^k$ indeksų aibė k-toje iteracijoje. Kitos iteracijos hiper-stačiakampių aibė $\mathcal{H}^{k+1}$ yra



Paveikslėlis 1: Vizualus hiper-stačiakampio dalinimo ir vertinimo strategija DIRECT algoritme sprendžiant *Rosenbrock* uždavinį, kai $n = 2, 3$.

gaunama išdalinus potencialiai optimalius hiper-stačiakampius iš $\mathcal{H}^k$ aibės.

Pirmoje iteracijoje algoritmas turi tik vieną kandidatą potencialiai optimalių hiper-stačiakampių atrankoje, kuris atitinka visą $\bar{D}$. Potencialiai optimalių hiper-stačiakampių reikalavimas oficialiai išdėstytas 1 apibrėžime.

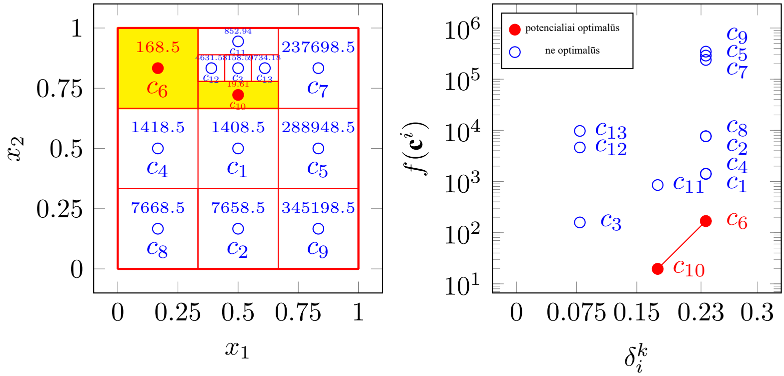
Apibrėžimas 1 *Tegul* $\mathbf{c}_i$ *žymi centro tašką ir* δ_i *skersmenį hiper-stačiakampio* $\bar{D}_i$. *Tegul* $\varepsilon > 0$ *būna teigiama konstanta ir* f_{min} *yra geriausia kol kas turima tikslo funkcijos reikšmė. Hiper-stačiakampis* $\bar{D}_j, j \in \mathbb{I}^k$ *yra potenciliai optimalus jeigu egzistuoja tokia (Lipšico) konstanta* $\tilde{L} > 0$*, kad:*

$$f(\mathbf{c}_j) - \tilde{L}\delta_j \leq f(\mathbf{c}_i) - \tilde{L}\delta_i, \quad \forall i \in \mathbb{I}^k, \tag{2}$$

$$f(\mathbf{c}_j) - \tilde{L}\delta_j \leq f_{min} - \varepsilon|f_{min}|, \tag{3}$$

kur yra hiper-stačiakampio skersmuo yra apskaičiuojamas taip:

$$\delta_i = \frac{1}{2}\|\mathbf{b} - \mathbf{a}\|_2. \tag{4}$$



Paveikslėlis 2: Atrinktų potencialiai optimalių hiper-stačiakampių vizualizavimas atliekant ketvirtą DIRECT algoritmo iteraciją, sprendžiant dviejų kintamųjų *Rosenbrock* uždavinį.

Parametras ε naudojamas siekiant išvengti intensyvesnio lokalių sričių tyrinėjimo. Originalaus algoritmo autoriai [21] rekomendavo naudoti ε nuo 10^{-3} iki 10^{-7}. Geometrinis atrankos procedūros aiškinimas pateiktas 2 paveikslėlyje.

Nuo tada kai originalaus DIRECT buvo paskelbtas, pasiūlyta daugybė algoritmo modifikacijų. Remiantis optimizavimo uždavinių klasifikacija, DIRECT tipo algoritmus galima suskirstyti į keturias pagrindines grupes, kaip parodyta 1 lentelėje.

12. Nauja schema potencialiai optimaliems hiper-stačiakampiams identifikuoti

Kituose keturiuose disertacijos skyriuse pristatomi naujai sukurti algoritmai. Pirmas iš algoritmų DIRECT-GL buvo sukurtas atsižvelgiant į originalaus DIRECT trūkumus. Pasiūlytas algoritmas sprendžia problemas kai leistinoji sritis yra stačiakampis. Vėliau naudojant baudos funkcijas, DIRECT-GL buvo pritaikytas

Lentelė 1: DIRECT algoritmų klasifikacija.

Problemos tipas	Algoritmo pavadinimas	Šaltinis
Stačiakampių sritims	DIRECT	[9]
	glbSolve	[3]
	DIRECT-l	[11]
	DISIMPL-C, DISIMPL-V	[37]
	PLOR	[27]
	Aggressive DIRECT	[1]
	BIRECT	[33]
Tikslo funckija yra apribota tiesiškai	Lc-DISIMPL, Lv-DISIMPL	[39]
Netiesinis programavimas	DIRECT-L1	[9]
	Filter DIRECT	[5]
	eDIRECTc	[25]
	EPGO	[43]
	DF-EPGO	[42]
Problemoms su paslėptais ribojimais	DIRECT-NAS	[10]
	DIRECT-sub	[29]
	DIRECT-Barrier	[10]

uždaviniams su įvairaus tipo ribojimais spręsti. Galiausiai šioje disertacijoje pristatoma pirmoji lygiagreti DIRECT tipo algoritmo versija uždaviniams su ribojimais spręsti.

Šiame skyriuje pateiktas naujas būdas identifikuoti potencialiai optimalių hiper-stačiakampių rinkinį. Naudojant dviejų fazių strategiją, padidinamas hiper-stačiakampių rinkinys pridedant daugiau vidutinio dydžio hiper-stačiakampių su geriausiomis tikslo funkcijos reikšmėmis ir papildomai arčiausiai dabartinio minimumo taško. Pirmajame žingsnyje algoritmas veikia globaliau (palyginti su atrankos procedūra, naudojama DIRECT), o antrasis žingsnis užtikrina greitesnį ir išsamesnį turimo minimumo srities egzaminavimą.

Tegul $\mathbb{L}^k$ yra visų skirtingų hiper-stačiakampių diametrų δ^k indeksų aibė k-toje iteracijoje $\mathcal{H}^k$. Mažiausia reikšmė $l^k_{\min} \in \mathbb{L}^k$ atitinka hiper-stačiakampių grupę, kurio diametras $\delta^k_{\min}$ yra mažiausias. Didžiausia reikšmė $l^k_{\max} \in \mathbb{L}^k$ atitinka hiper-stačiakampių grupė, kurio diametras $\delta^k_{\max}$ yra didžiausias, t. y. $l^k_{\max} = \max\{\mathbb{L}^k\} < \infty$. Galiausiai, tegul $l^k_i \in \mathbb{L}^k$ yra hiper-stačiakampių grupės $\bar{D}^k_i$ indeksas su skersmeniu δ^k_i. Toliau apibrėžimai 2 ir 3 pateikia potencialiai optimalių hiper-stačiakampių atranką k-toje iteracijoje.

Apibrėžimas 2 *(Patobulinti globalią paiešką `DIRECT-G`)*

- ***1 žingsnis***. *Rasti indeksą $j \in \mathbb{I}^k$ ir jam atitinkantį hiper-stačiakampį $\bar{D}^k_i$, kad:*

$$D^k_j = \arg\max_j \{l^k_j : j = \arg\min_{i \in \mathbb{I}_k :\, l^k_{\min} \leq l^k_i \leq l^k_{\max}} \{f(\mathbf{c}_i)\}\}. \qquad (5)$$

- ***2 žingsnis***. *Nustatyti $l^k_{\min} = l^k_j + 1$. Jeigu $l^k_j \leq l^k_{\max}$ grįžti į **1 žingsnį**; kitu atveju nutraukti ciklą.*

Pirmajame žingsnyje identifikuotas hiper-stačiakampis, kuriame yra mažiausia tikslo funkcijos reikšmė $\mathbf{x}_{\min}$. Jei tokių reikšmių yra kelios, pirmenybė teikiama hiper-stačiakampiui su didžiausiu indeksu.

Apibrėžimas 3 *(Patobulinti lokalią paiešką `DIRECT-L`)*

- ***1 žingsnis***. *Kiekvienoje k-toje iteracijoje įvertinti euklidinį atstumą nuo turimo $\mathbf{x}_{\min}$ iki visų kitų turimų taškų:*

$$d(\mathbf{x}_{\min}, \mathbf{c}_i) = \sqrt{\sum_{j=1}^{n} (x^j_{\min} - c^j_i)^2} \qquad (6)$$

- ***2 žingsnis*** *.Pritaikyti procedūrą aprašytą 2 apibrėžime, pakeičiant tikslo funkcijos reikšmes euklidiniu atstumu* $d(\mathbf{x}_{\min}, \mathbf{c}_i)$.

Šia strategija padidiname potencialiai optimalių hiper-stačiakampių, esančių arti dabartinio minimalaus taško $\mathbf{x}_{\min}$ skaičių.

Pagrindinė siūlomo `DIRECT-GL` algoritmo savybė yra ta, kad kiekvieną iteraciją potencialiai optimalių hiper-stačiakampių atranka yra atliekama du kartus. Pirmiausia surandama aibė $\mathbb{G}$ (naudojant 2 apibrėžimą – `DIRECT-G`) po to surandama aibė $\mathbb{L}$ (naudojant 3 apibrėžime – `DIRECT-L`) ir galiausiai siekiant pašalinti besidubliuojančius hiper-stačiakampius randama unikali abiejų aibių $\mathbb{S} = \mathbb{G} \cup \mathbb{L}$ sąjunga.

Problemų tik su intervaliniais ribojimais sprendimo rezultatai

Kadangi visi globalūs sprendiniai f^* yra žinomi visiems testiniams uždaviniams, algoritmai buvo stabdomi, kai buvo rastas toks taškas $\mathbf{x}$, kad paklaida procentais

$$pe = 100\% \times \begin{cases} \frac{f(\mathbf{x}) - f^*}{|f^*|}, & f^* \neq 0, \\ f(\mathbf{x}), & f^* = 0, \end{cases} \tag{7}$$

yra mažesnė už ε_{pe}, arba kai funkcijų vertinimų skaičius viršija nustatytą ribą 1×10^6. Šiame tyrime buvo nagrinėjamos dvi skirtingos ε_{pe} reikšmės: 10^{-2}, 10^{-8}.

Apibendrinti eksperimentų rezultatai sprendžiant 59 testinius uždavinius iš `DIRECTlib`, pateikti 2 lentelėje. Originalus `DIRECT` parodė geresnius rezultatus sprendžiant lengvesnius uždavinius su vienu optimaliu sprendiniu. Tačiau visais kitais atvejais, sukurtas `DIRECT-GL` algoritmas davė kur kas geresnių rezultatų nei `DIRECT`, ypač kai sprendinį norima rasti didesniu tikslumu.

Lentelė 2: Vidurkiai funkcijos vertinimų sprendžiant testinius uždavinius iš DIRECTlib, skirtingais tikslumais

Algoritmas	ε_{pe}	Vidurkis			Mediana	# neišspręsta
		bendras	$n \leq 4$	$n \geq 5$		
DIRECT-GL	10^{-2}	**111,904**	**10,472**	**197,487**	5,225	**3/59**
	10^{-8}	**192,039**	**56,815**	**306,134**	**10,239**	**5/59**
DIRECT-G	10^{-2}	223,363	57,276	363,499	7,493	11/59
	10^{-8}	288,128	99,574	447,221	22,833	14/59
DIRECT-L	10^{-2}	244,429	39,509	417,331	7,921	13/59
	10^{-8}	291,075	78,251	470,644	17,253	15/59
DIRECT	10^{-2}	273,752	76,257	440,388	7,449	14/59
	10^{-8}	613,762	492,316	716,233	$> 10^6$	33/59
BIRECT	10^{-2}	189,092	38,606	316,065	**2,370**	10/59
	10^{-8}	701,818	553,215	827,201	$> 10^6$	39/59
DIRECT-l	10^{-2}	377,794	75,400	632,939	8,589	21/59
	10^{-8}	648,630	509,209	766,266	$> 10^6$	36/59
PLOR	10^{-2}	395,030	156,400	596,375	7,015	23/59
	10^{-8}	630,087	560,916	688,450	$> 10^6$	35/59
DISIMPL-V	10^{-2}	393,794	38,470	693,600	14,671	23/59
	10^{-8}	705,544	494,555	883,567	$> 10^6$	41/59
DISIMPL-C	10^{-2}	570,515	128,505	943,461	$> 10^6$	33/59
	10^{-8}	648,187	297,306	944,243	$> 10^6$	37/59

13. Pagalbinė funkcija netiesinio programavimo problemoms

Šiame skyriuje pristatytas DIRECT-GLce algoritmas netiesinio programavimo problemoms spręsti:

$$
\begin{aligned}
&\min_{\mathbf{x} \in D} f(\mathbf{x}) \\
&\text{s.t. } \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \\
&\quad\;\; \mathbf{h}(\mathbf{x}) = \mathbf{0},
\end{aligned}
\tag{8}
$$

kur $f : \mathbb{R}^n \to \mathbb{R}, \mathbf{g} : \mathbb{R}^n \to \mathbb{R}^u, \mathbf{h} : \mathbb{R}^n \to \mathbb{R}^v$ žymi tikslo ir apribojimų funkcijas, o leistina sritis kuri tenkina visas apribojimų funkcijas apibrėžtas kaip $D^{\text{feas}} = D \cap \Omega$, kur $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \text{ ir } \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$.

Algoritmas naudoja naują baudos funkcijos metodą, kuris apjungia tikslo ir apribojimų funkcijų informaciją ir nereikalauja jokių baudos parametrų iš vartotojo. `DIRECT-GLce` algoritmas veikia dviem fazėmis: pirmoje fazėje algoritmas bando identifikuoti sritis tenkinančias ribojimų funkcijas, antroje – fazėje siekia rasti geresnį sprendinį.

13.1 Situacijos, kai pradiniai taškai netenkina apribojimų

Atlikus eksperimentinę analizę esamų algoritmų netiesinio programavimo problemoms spręsti paaiškėjo, kad egzistuoja aibės problemų kai pradiniai taškai hiper-stačiakampyje netenkina apribojimų funkcijų, ir bent vieno tokio taško suradimas gali būti labai brangi procedūra. Tokioms situacijoms naudojame papildomą schemą `DIRECT-GLce` algoritme, kuri minimizuoja ne tikslo funkcijos reikšmes, o apribojimų funkcijų reikšmių sumą:

$$\min_{\mathbf{x} \in D} \varphi(\mathbf{x}), \tag{9}$$

kur

$$\varphi(\mathbf{x}) = \sum_{i=1}^{u} \max\{g_i(\mathbf{x}), 0\} + \sum_{i=1}^{v} |h_i(\mathbf{x})|, \tag{10}$$

kol taškas tenkinantis ribojimus $\mathbf{x} \in D_{\varepsilon_\varphi}^{\text{feas}}$ bus surastas, kur

$$D_{\varepsilon_\varphi}^{\text{feas}} = \{\mathbf{x} : 0 \leq \varphi(\mathbf{x}) \leq \varepsilon_\varphi, \mathbf{x} \in D\}. \tag{11}$$

ε_φ yra leistina apribojimų funkcijų paklaida.

13.2 Nauja baudos funkcija

Šioje disertacijoje pristatomos dvi naujos baudos funkcijos, kurios nereikalauja jokių įvesties parametrų. Pirmoji iš jų yra (8) problemos transformacija į (12) ir realizuota `DIRECT-GLc` algoritme:

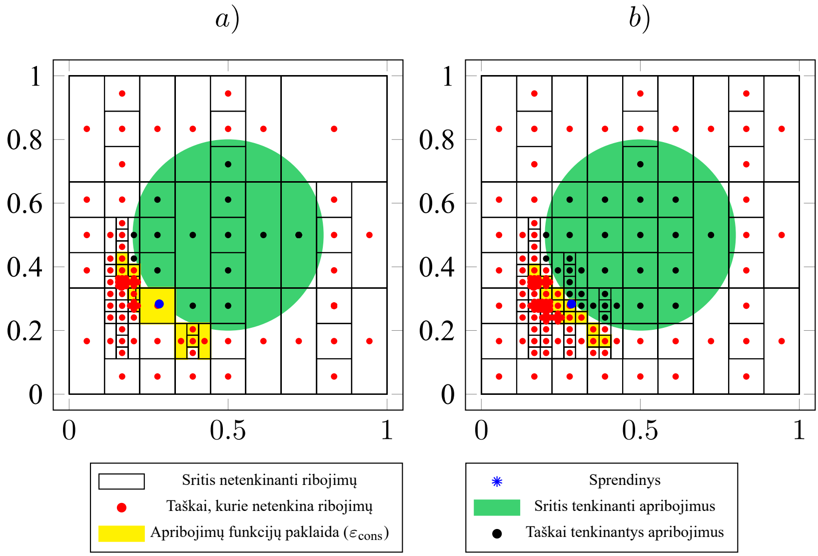
$$\min_{\mathbf{x}\in D} f(\mathbf{x}) + \xi(\mathbf{x}, f_{\min}^{\text{feas}}),$$
$$\xi(\mathbf{x}, f_{\min}^{\text{feas}}) = \begin{cases} 0, & \text{jei} \mathbf{x} \in D_{\varepsilon_\varphi}^{\text{feas}} \\ \varphi(\mathbf{x}) + \Delta, & \text{kitu atveju,} \end{cases} \tag{12}$$

kur $\Delta = |f(\mathbf{x}) - f_{\min}^{\text{feas}}|$ yra lygus skirtumui tarp geriausios iki šiol rastos tikslo funkcijos reikšmės $f_{\min}^{\text{feas}}$ ir tikslo funkcijos reikšmės apribojimus netenkinančiame taške $\mathbf{x} \notin D_{\varepsilon_\varphi}^{\text{feas}}$. Pagrindinis parametro Δ tikslas yra užkirsti kelią `DIRECT-GLce` algoritmo konvergavimui į sritis netenkinančias apribojimų funkcijų, baudžiant tikslo funkcijų reikšmes reikiamais dydžiais. Kai geresnė tikslo funkcijos $f_{\min}^{\text{feas}}$ reikšmė yra randama, pagalbinė funkcija $\xi(\mathbf{x}, f_{\min}^{\text{feas}})$ yra perskaičiuojama.

Kadangi optimizavimo pradžioje skirtumas tarp $f_{\min}^{\text{feas}}$ ir globaliojo sprendimo f^* gali būti didelis, dėl to $\xi(\mathbf{x}, f_{\min}^{\text{feas}})$ reikšmė gali būti per daug padidinta. Į tai atsižvelgus buvo pasiūlyta alternatyvi baudos funkcija, atlikus nedidelį pakeitimą (13) baudos funkcijoje:

$$\min_{\mathbf{x}\in D} f(\mathbf{x}) + \tilde{\xi}(\mathbf{x}, f_{\min}^{\text{feas}}),$$
$$\tilde{\xi}(\mathbf{x}, f_{\min}^{\text{feas}}) = \begin{cases} 0, & \text{jei} \mathbf{x} \in D_{\varepsilon_\varphi}^{\text{feas}} \\ 0, & \text{jei} \mathbf{x} \in D_{\varepsilon_{\text{cons}}}^{\text{inf}} \\ \varphi(\mathbf{x}) + \Delta, & \text{kitu atveju,} \end{cases} \tag{13}$$

kur $D_{\varepsilon_{\text{cons}}}^{\text{inf}} = \{\mathbf{x} : f(\mathbf{x}) \leq f_{\min}^{\text{feas}}, \varepsilon_\varphi < \varphi(\mathbf{x}) \leq \varepsilon_{\text{cons}}, \mathbf{x} \in D\}$ ir $\varepsilon_{\text{cons}}$ yra nedidelė apribojimų funkcijų paklaida, kuri skirta algoritmui palengvinti tyrinėjimą aplink ribojimų ribas. Įdiegus šią modifikaciją, naujasis DIRECT-GLce algoritmas atrenka ir dalina daugiau hiper-stačiakampių, kurių centriniai taškai yra arti apribojimo ribų. Geometrinė $\varepsilon_{\text{cons}}$ parametro iliustracija pateikta 3 paveikslėlyje.



Paveikslėlis 3: DIRECT-GLce algoritmo geometrinė interpretacija sprendžiant *T1* $(n = 2)$ uždavinį a) šešta iteracija, b) aštunta iteracija.

Kaip ir kiti šiame darbe sukurti algoritmai, DIRECT-GLce buvo realizuotas „MatLab“ programavimo kalboje. Siekiant pagreitinti brangių testinių problemų globalią sprendimo paiešką, kitų DIRECT tipo metodų autoriai į algoritmą inkorporuoja lokalią paieškos procedūrą. Remiantis ta pačia idėja, DIRECT-GLce algoritmas taip pat buvo modifikuotas ir „MatLab“ lokalios paieškos procedūra fmincon buvo įdiegta į pasiūlyto DIRECT-GLce-min algoritmo versiją.

Lyginamoji analizė su DIRECT-L1 ir L-DISIMPL algoritmais sprendžiant 60 testinių uždavinių su įvairiais ribojimais pateikta 3 lentelėje. Sustojimos sąlygos yra tos pačios, kaip ir ankstesnėje analizėje, tik viena ε_{pe} reikšmė 10^{-2} buvo nagrinėjama.

Lentelė 3: Skaičius funkcijos vertinimų sprendžiant testinius uždavinius iš DIRECTlib

Algoritmas	Vidurkis					Mediana	# neiš-
	bendras	$n \leq 3$	$n \geq 4$	T rib.	NT rib.		spręsta
DIRECT-GLc	$180,141$	$1,916$	$358,245$	$85,240$	$296,132$	$4,391$	$9/60$
DIRECT-GLce	$147,307$	$4,628$	$289,442$	$95,496$	$210,632$	$10,226$	$5/60$
DIRECT-GLce-min	$\mathbf{24,304}$	**127**	$\mathbf{48,485}$	$\mathbf{1,900}$	$\mathbf{51,688}$	**144**	**1/60**
DIRECT-L1 ($r = 10$)	$566,523$	$485,916$	$629,993$	$475,675$	$677,559$	$> 10^6$	$33/60$
DIRECT-L1 ($r = 10^2$)	$551,262$	$348,098$	$732,695$	$481,975$	$635,945$	$> 10^6$	$31/60$
DIRECT-L1 ($r = 10^3$)	$554,459$	$367,544$	$720,292$	$468,701$	$659,275$	$> 10^6$	$31/60$
Lc-DISIMPL	–	–	–	$156,467$	–		$5/33$
Lv-DISIMPL	–	–	–	$91,237$	–		$3/33$

r. – Baudos parametrai apribojimų funkcijoms

Pirmiausia nesunku pastebėti, kad hibridinė DIRECT-GLce-min versija yra efektyviausia iš visų nagrinėjamų algoritmų. Palyginus likusius algoritmus akivaizdu, kad sprendžiant problemas kai $n \leq 3$, funkcijų vertinimų skaičius dažniausiai būna mažesnis, naudojant DIRECT-GLc algoritmą. Naudojant $\varepsilon_{\text{cons}}$ parametrą DIRECT-GLce algoritme reikia daugiau funkcijų vertinimų, sprendžiant paprastesnes problemas, palyginti su kitais algoritmais, tačiau sprendžiant sudėtingesnes problemas DIRECT-GLce yra kur kas perspektyvesnis.

Lentelė 4: Skaičius funkcijos vertinimų sprendžiant testinius uždavinius iš `DIRECTlib`

Algoritmas	Vidurkis			Mediana	# neišspręsta
	bendras	$n \leq 3$	$n \geq 4$		
`DIRECT-GLc`	$43,065$	$4,775$	$89,864$	$1,432$	$3/20$
`DIRECT-GLce`	$50,124$	$5,354$	$104,843$	$2,851$	$3/20$
`DIRECT-GLce-min`	$\mathbf{21,405}$	$2,415$	$\mathbf{44,616}$	$\mathbf{173}$	$\mathbf{1/20}$
`Filter DIRECT`	$151,131$	$\mathbf{1,984}$	$499,140$	$1,692$	$6/20$
`EPGO`	$41,629$	$6,087$	$85,069$	$13,637$	$11/20$
`DF-EPGO`	–	–	–	–	$10/20$

Eksperimentiniai rezultatai palyginus pasiūlytus algoritmus su `Filter DIRECT`, `EPGO` ir `DF-EPGO`, pateikti 4 lentelėje. Palyginus bendrą vidurkį, sukurti algoritmai apytiksliai reikalauja tris-kart mažiau tikslo funkcijų vertinimų nei `Filter DIRECT`. Tačiau sprendžiant paprastesnes testines problemas, t. y. tikslo funkcijas ($n \leq 3$) `Filter DIRECT`, yra labai perspektyvi alternatyva. Visiškai skirtingi rezultatai matomi sprendžiant sunkesnes problemas, t. y. tikslo funkcijas ($n \geq 4$), kai sukurti algoritmai davė ženkliai geresnių rezultatų.

14. Pagalbinė funkcija problemoms su paslėptais ribojimais spręsti

Šiame skyriuje pristatytas `DIRECT-GLh` algoritmas problemoms su paslėptais ribojimais spręsti:

$$\min_{\mathbf{x} \in D^{\text{feas}}} \quad f(\mathbf{x}), \tag{14}$$

kur $D^{\text{feas}} = D \cap D^{\text{hidden}}$ ir D^{hidden} nėra pateiktas analitiškai, t. y.

problema turi nežinomų paslėptų apribojimų. Spręsdami problemą (14), tik įvertinę tikslo funkciją tam tikrame taške, algoritmai gali nuspręsti ar jis priklauso D^{feas}. Todėl dvi pagrindinės kylančios problemos `DIRECT` tipo algoritmuose yra šios:

1. aptikti bent vieną tašką, kad $\mathbf{x} \in D^{\text{feas}}$;
2. rasti tinkamą baudos funkciją hiper-stačiakampiams, kurie $\mathbf{x} \in D^{\text{hidden}}$.

14.1 Situacijos, kai pradiniai taškai netenkina apribojimų

Ankstesni tyrimai parodė, kad rasti bent vieną tašką, kai $\mathbf{x} \in D^{\text{feas}}$, sprendžiant optimizavimo problemas su įvairaus tipo apribojimais dažnai gali būti labai brangu. Esant problemų su paslėptais apribojimais, tai gali būti dar sudėtingesnis uždavinys, nes nėra jokios informacijos apie apribojimus. Norint išspręsti šią situaciją pasiūlytasis `DIRECT-GLh` atlieka tolygų $\bar{D}$ dalijimą taip, kaip nurodyta 4 apibrėžime.

Apibrėžimas 4 *Nustatyti* $\mathbb{I}^k = \{1\}$*, kur* $k = 1$ *ir atlikti tokius žingsnius:*

- ***1 žingsnis****. Rasti indeksą* $j \in \mathbb{I}^k$ *ir jam atitinkamą hiper-stačiakampį* $\bar{D}_j$*, kad*

$$\bar{D}_j = \arg\max_{j} \left\{ j = \arg\max_{i \in \mathbb{I}^k} \{\delta_i\} \right\}. \tag{15}$$

- ***2 žingsnis****. Padalyti hiper-stačiakampį* $\bar{D}_j$ *ir patikrinti ar bent vienas iš naujų hiper-stačiakampių centro taškų* $\mathbf{x} \in D^{\text{feas}}$*. Taip pat, nustatyti* $k = k + 1$ *ir atnaujinti* $\mathbb{I}^k$ *bei naujų hiper-stačiakampių* i *skersmenis* δ_{j_i}*.*
- ***3 žingsnis****. Jei* $D^{\text{feas}} = \emptyset$ ***kartoti*** *nuo* ***1 žingsnio****; kitu atveju* ***stabdyti****.*

Pirmojo žingsnio metu pasirenkamas didžiausias hiper-stačiakampis. Jei yra keli hiper-stačiakampiai su tokiu pačiu maksimaliu skersmeniu, pirmenybė teikiama hiper-stačiakampiui, kurio indekso reikšmė yra didžiausia. Antrame žingsnyje pasirinktas hiper-stačiakampis yra padalijamas ir patikrinama, ar nors vienas taškas neturi paslėpto apribojimo $\mathbf{x} \in D^{\text{feas}}$. Procesas tęsiamas, kol randamas bent vienas neapribotas taškas, t. y. $D^{\text{feas}} \neq \emptyset$. Po to dedamos visos pastangos sprendimui tobulinti.

14.2 Baudos funkcija problemoms su paslėptais ribojimais spręsti

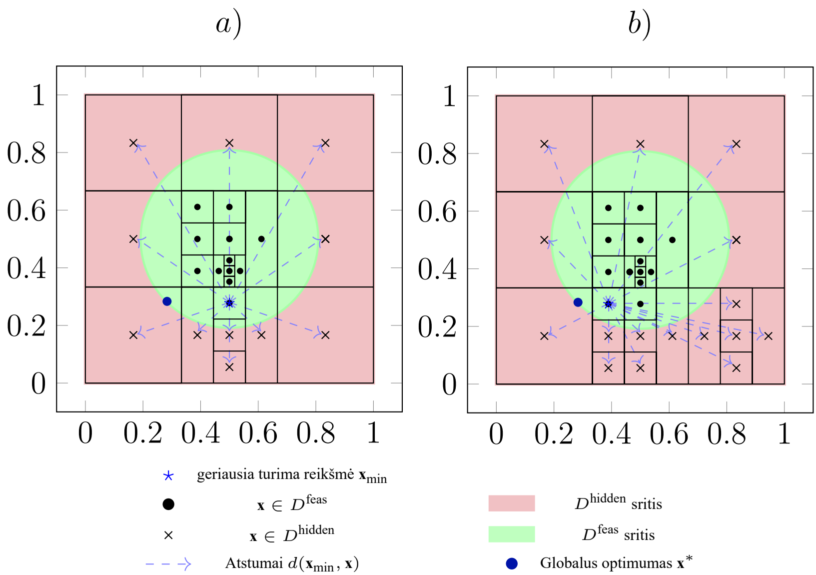
Kitame žingsnyje (1) problema transformuojama į (14):

$$\min_{\mathbf{x}\in D} \phi(\mathbf{x}, \mathbf{x}_{\min}, f_{\min}),$$
$$\phi(\mathbf{x}, \mathbf{x}_{\min}, f_{\min}) = \begin{cases} f(\mathbf{x}), & \text{jei } \mathbf{x} \in D^{\text{feas}} \\ f_{\min} + d(\mathbf{x}_{\min}, \mathbf{x}), & \text{kitu atveju,} \end{cases} \tag{16}$$

kur $f_{\min}$ turima geriausia tikslo funkcijos reikšmė ir $d(\mathbf{x}^{\min}, \mathbf{x})$ yra euklidinis atstumas (6) nuo geriausio turimo taško ($\mathbf{x}^{\min}$) iki taško ($\mathbf{x}$), kuris yra apribotas $\mathbf{x} \in D^{\text{hidden}}$.

Kai $\mathbf{x} \in D^{\text{feas}}$, tikslo funkcija yra įvertinama $f(\mathbf{x})$, kitu atveju $f_{\min} + d(\mathbf{x}_{\min}, \mathbf{x})$ yra priskiriama hiper-stačiakampiui, kurio centro taškas yra apribotas. Geometrinė strategijos interpretacija pavaizduota 4 paveikslėlyje. Naudojant šią strategiją taškai, kurie $\mathbf{x} \in D^{\text{hidden}}$ yra baudžiami, priklausomai nuo to, kaip toli nutolę nuo geriausio turimo taško $\mathbf{x}_{\min}$. Tokiu būdu konvergavimas į sritis tenkinančias ribojimus yra garantuotas, net kai susiduriama su sudėtingomis apribojimų formomis.

Kiekvieną kartą, kai randama geresnė reikšmė $f_{\min}$, visi hiper-stačiakampiai, kurių centro taškai yra apriboti $\mathbf{x} \in D^{\text{hidden}}$, yra perskaičiuojami. Kita vertus, kiekvienoje iteracijoje atstumo funkcija $d(\mathbf{x}_{\min}, \mathbf{x})$ yra perskaičiuojama potencialiai optimalių hiper-stačiakampių atrankoje `DIRECT-GL` algoritme, tokiu atveju pagalbinės funkcijos atnaujinimas $\phi(\mathbf{x}, \mathbf{x}_{\min}, f_{\min})$ nereikalauja daug papildomų skaičiavimų.



Paveikslėlis 4: `DIRECT-GLh` algoritmo geometrinė interpretacija sprendžiant *T1* uždavinį, kai $n = 2$ a) trečia iteracija, b) ketvirta iteracija.

Didžiausia formuluotės (14) problema yra ta, kad tikslo funkcijos ir atstumų reikšmės gali labai skirtis. Siekdami pašalinti galimą reikšmių dominavimą, jos yra normalizuojamos:

$$
\begin{aligned}
\tilde{f}(\mathbf{x}) &= \begin{cases} 1, & \text{if} f_{\max} == f_{\min} \\ \dfrac{f(\mathbf{x}) - f_{\min}}{f_{\max} - f_{\min}}, & \text{kitu atveju.} \end{cases} \\
\tilde{d}(\mathbf{x}_{\min}, \mathbf{x}) &= \frac{d(\mathbf{x}_{\min}, \mathbf{x}) - d_{\min}}{d_{\max} - d_{\min}},
\end{aligned} \tag{17}
$$

kur $d_{\max}$ $\bar{D}$ įstrižainės ilgis is $d_{\min}$ yra lygus 0. Tai prideda šiek tiek papildomų skaičiavimų, nes normalizuotos tikslo ir atstumo funkcijos turėtų būti atnaujinamos kiekvieną kartą, kai randamas hiper-stačiakampis su nauja didžiausia/mažiausia tikslo funkcijos reikšme. Atsižvelgus į (14) ir (17) gaunama:

$$
\begin{aligned}
&\min_{\mathbf{x} \in D} \tilde{\phi}(\mathbf{x}, \mathbf{x}_{\min}, \tilde{f}_{\min}), \\
&\tilde{\phi}(\mathbf{x}, \mathbf{x}_{\min}, \tilde{f}_{\min}) = \begin{cases} \tilde{f}(\mathbf{x}), & \text{if } \mathbf{x} \in D^{\text{feas}}, \\ \tilde{f}_{\min} + \tilde{d}(\mathbf{x}_{\min}, \mathbf{x}), & \text{kitu atveju.} \end{cases}
\end{aligned} \tag{18}
$$

14.3 Eksperimentinė analizė

Šiame skyriuje pateiki svarbiausi sukurto algoritmo rezultatai naudojant 67 testinius uždavinius iš `DIRECTlib` [54]. Tiriami algoritmai buvo sustabdyti, kai procentinė paklaida (7) yra mažesnė už $\varepsilon_{\text{pe}} = 10^{-2}$ arba kai funkcijų vertinimų skaičius viršija nustatytą ribą 10^6. Be to, maksimalus vienos testo problemos sprendimo laikas buvo apribotas iki šešių valandų. Visi skaičiavimai buvo atlikti naudojant 6 branduolių kompiuterį su 8-osios kartos Intel R Core ^{TM} $i7 - 8750H$ $@2.20GHz$ procesoriumi, 16 GB RAM ir „`MatLab`“ R2020b.

Eksperimentų rezultatai 5 lentelėje rodo, kad pasiūlytasis `DIRECT-GLh` algoritmas yra efektyviausias iš visų šios klasės `DIRECT` tipo algoritmų. Didžiausias `DIRECT-GLh` algoritmo privalumas yra greitis, bendras vidutinis visų išbandytų problemų sprendimo laikas, veikia maždaug $97,7$ karto greičiau, palyginti su antru geriausiu algoritmu – `DIRECT-NAS`. Be to, $62/67$ testo uždaviniai buvo išspręsti greičiau naudojant sukurtą `DIRECT-GLh` algoritmą.

Lentelė 5: Skaičius funkcijos vertinimų ir laikas (sekundėmis) sprendžiant testinius uždavinius iš `DIRECTlib`

Algoritmas	Kriterijus	Vidurkis					Mediana	# neiš-
		bendras	$n \leq 3$	$n \geq 4$	T. rib.	NT. rib.		spręsta
`DIRECT-GLh`	Laikas(s)	**14.74**	**1.02**	**31.67**	**19.68**	**9.66**	**1.02**	**5/67**
	f_{eval}	**99, 472**	**3, 330**	**218, 047**	**102, 587**	**96, 262**	**3, 330**	
`DIRECT-NAS`	Laikas(s)	$5,220.33$	81.73	$11,559.16$	$4,057.97$	$6,417.90$	9.69	$15/67$
	f_{eval}	$237,875$	$4,450$	$525,766$	$190,888$	$286,286$	$5,099$	
`DIRECT-Barrier`	Laikas(s)	602.64	886.48	252.57	246.74	969.32	91.98	$35/67$
	f_{eval}	$550,537$	$382,419$	$757,882$	$549,374$	$551,735$	$> 10^6$	
`DIRECT-sub` ($\gamma = 2$)	Laikas(s)	413.34	667.73	99.59	100.73	735.42	54.27	$43/67$
	f_{eval}	$653,228$	$497,573$	$845,201$	$721,758$	$582,621$	$> 10^6$	
`DIRECT-sub` ($\gamma = 5$)	Laikas(s)	581.86	746.07	379.33	207.45	967.62	83.95	$36/67$
	f_{eval}	$562,997$	$393,538$	$757,432$	$523,720$	$590,223$	$> 10^6$	

T. rib. – Tiesiniai ribojimai

NT. rib. – Netiesiniai ribojimai

15. `DIRECT-GLce` lygiagretinimo schema

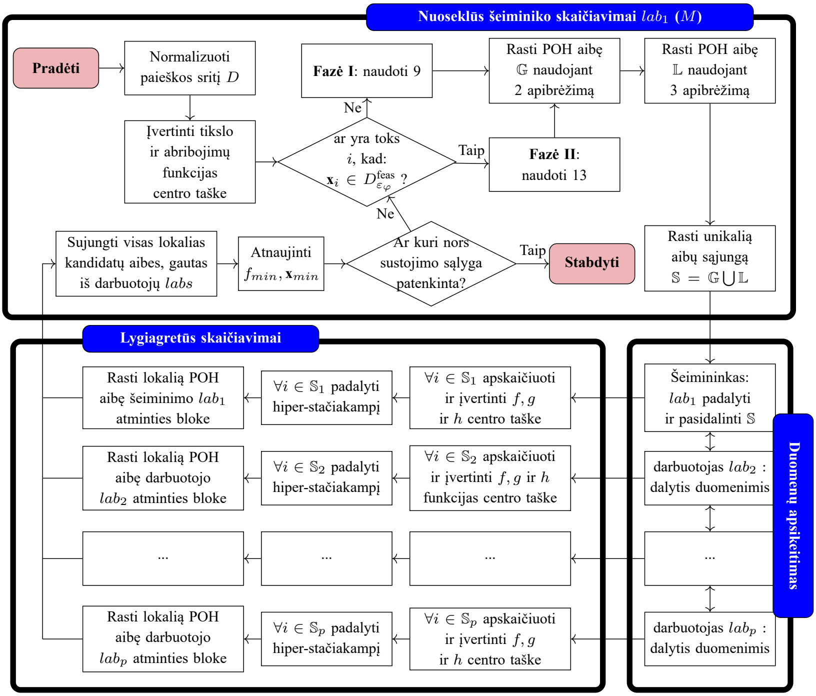
Disertacijoje pristatomos trys lygiagrečios `DIRECT-GLce` algoritmo versijos „`MatLab`“ programinės įrangos aplinkoje. *Parallel Computing Toolbox* pateikia keletą skirtingų lygiagrečių programavimo paradigmų, pavyzdžiui, lygiagretų `for` – ciklą arba `SPMD` (kiekvienas procesorius vykdo tą pačią programą, tačiau su skirtingais duomenimis).

Dvi lygiagrečios `DIRECT-GLce` algoritmo versijos, realizuotos atitinkamai naudojant lygiagretų `for` – ciklą (`pD-GLce-parfor`) ir

SPMD (pD-GLce-spmd). Trečioji lygiagreti versija (pD-ACe-spmd) skiriasi nuo pD-GLce-spmd potencialiai optimalių hiper-stačiakampių išrinkimo žingsnyje, kur pD-GLce-spmd yra naudojamas agresyvus variantas [1].

Šiame skyriuje pateikiama efektyviausia iš pasiūlytų lygiagrečių DIRECT-GLce algoritmo versijų (pD-GLce-spmd). pD-GLce-spmd realizuota naudojant paskirstytosios atminties programavimo alternatyvą „MatLab“ programinės įrangos aplinkoje (*SPMD*). Šeimininko ir darbininko paradigma naudojama pasiūlytoje lygiagrečioje schemoje, kur šeimininkas irgi veikia kaip darbininkas. Kiekvienas darbuotojas saugo duomenis savo vietinėje atmintyje, o duomenimis keičiamasi naudojantis signalų perdavimu [24]. Lygiagretaus algoritmo schema pateikta 5 paveikslėlyje. Šeimininkas žymimas M, kuris nusprendžia, kurie hiper-stačiakampiai bus atrinkti ir paskirstyti tarp darbuotojų. Be to, šeimininkas yra atsakingas už algoritmo sustabdymą.

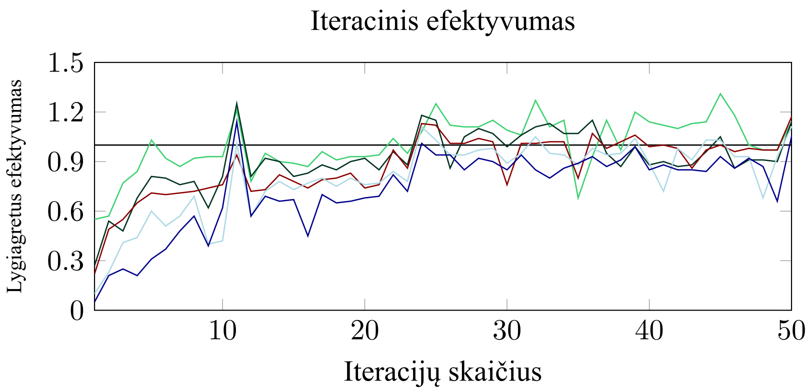
Šeimininkas taip pat atlieka darbo balansavimą, paskirstydamas pasirinktus hiper-stačiakampius. Kai darbuotojai $W_i, i = 1, \dots, p-1$ gauna užduotis iš šeimininko, kiekvienas iš jų atlieka tolimesnius veiksmus su hiper-stačiakampiu, o tada atlieka lokalią potencialiai optimalių hiper-stačiakampių atranką. Gauta aibė išsiunčiama šeimininkui, kad atliktų globalią atranką, ir sustoja, kol bus gautos tolesnės instrukcijos. Jei kuri nors iš sustojimo sąlygų yra patenkinama, visi darbuotojai gauna pranešimą, kad šeimininkas tapo neaktyvus, o darbuotojai nutraukia veiklą be papildomų pranešimų.

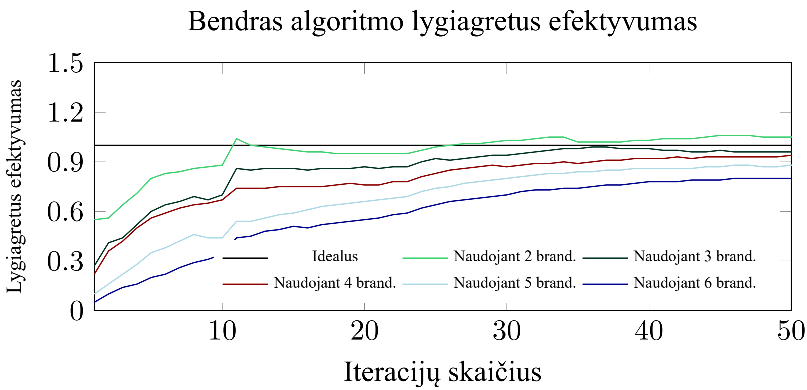


Paveikslėlis 5: pD-GLce-spmd lygiagretinimo tinklo schema.

15.1 Lygiagretaus pD-GLce-spmd algoritmo rezultatai

6 paveikslėlyje pateikta, kaip pD-GLce-spmd efektyvumas padidėja kartu su iteracijos skaičiumi sprendžiant *Michalewicz* testinį uždavinį. Pradiniuose etapuose efektyvumas yra mažesnis didinant branduolių skaičių, tačiau ilgoje iteracinėje sekoje efektyvumas artėja prie idealaus.





Paveikslėlis 6: `pD-GLce-spmd` algoritmo efektyvumas sprendžiant *Michalewicz*($n = 25$) testinį uždavinį, pirmose 50 iteracijose.

16. Bendros išvados

1. Sukurtas naujas, potencialiai optimalių hiper-stačiakampių identifikavimo schema pagrįstas `DIRECT-GL` algoritmas, o taikant siūlomą metodą buvo efektyviai pašalinti du gerai žinomi `DIRECT` tipo algoritmų trūkumai. Išsamūs eksperimentiniai tyrimai parodė sukurto `DIRECT-GL` algoritmo galimybes, palyginti su kitais esamais `DIRECT` tipo algoritmais:

1.1 Bendru atveju, norint pasiekti $\varepsilon_{\mathrm{pe}} = 10^{-2}$ globaliojo sprendinio tikslumą, `DIRECT-GL` maždaug reikia 59% mažiau tikslo funkcijų vertinimų, palyginus su antruoju geriausiu `BIRECT` algoritmu.

1.2 Kai reikia tikslesnio sprendinio ($\varepsilon_{\mathrm{pe}} = 10^{-8}$), sukurtas algoritmas reikalauja apie 72% mažiau funkcijų vertinimų, palyginti su antruoju geriausiu `PLOR` algoritmu.

1.3 Bendru atveju, `DIRECT-GL` išsprendžia apie 32% daugiau testinių problemų, palyginti su antruoju geriausiu šios klasės algoritmu `DIRECT Version 4.0`.

2. Toliau buvo sukurtas naujas pagalbinėmis funkcijomis pagrįstas `DIRECT-GLce` algoritmas, skirtas globaliojo optimizavimo problemoms su bendraisiais apribojimais spręsti. Testo ir praktinėms inžinerijos problemoms spręsti `DIRECT-GLce` našumas yra vienas iš geriausių ir vidutiniškai yra geriausias tarp visų pažangiausių `DIRECT` tipo algoritmų:

2.1 Sukurtas `DIRECT-GLc` algoritmas turi daugiausiai geriausių rezultatų ir apie 55% problemų išsprendė efektyviausiai.

2.2 Bendru atveju `DIRECT-GLce` išsprendė apie 43% daugiau testinių problemų, palyginus su antru geriausiu algoritmu `DIRECT-L1`.

2.3 Hibridizuotas `DIRECT-GLce-min` yra efektyviausias algoritmas tarp visų tirtų `DIRECT` tipo metodų ir bendru atveju išsprendė daugiausiai testinių problemų (85/88), iš jų (62/88) su didžiausiu efektyvumu.

3. Sukurtas naujas pagalbinės funkcijos pagrįstas algoritmas `DIRECT-GLh`, skirtas spręsti globaliojo optimizavimo problemas su paslėptais apribojimais. Eksperimentiniai tyrimai

atskleidė sukurto algoritmo potencialą, palyginus su visais esamais `DIRECT` tipo algoritmais šioje klasėje:

3.1 Sukurtas `DIRECT-GLh` algoritmas išsprendė apie 52% testinių uždavinių naudojant mažiausią tikslo funkcijų vertinimų skaičių, palyginus su visais kitais tirtais `DIRECT` tipo metodais.

3.2 Bendru atveju, `DIRECT-GLh` reikėjo apie 42% mažiau tikslo funkcijų vertinimų, palyginti su antru geriausiu `DIRECT-NAS` algoritmu.

4. Sukurtos dinamiškos duomenų struktūros, skirtos geresniam duomenų saugojimui ir organizavimui, padidino sukurtų algoritmų greitį maždaug apie 62%.
5. Pasiūlytas `pD-GLce-spmd` yra pirmasis deterministinis lygiagretus `DIRECT` tipo algoritmas, skirtas globaliojo optimizavimo problemoms su bendraisiais apribojimais spręsti, ir pasiekė gerą lygiagretų efektyvumą naudojant kelių branduolių kompiuterį.

Publikacijų sąrašas

Straipsniai recenzuojamuose užsienio leidiniuose:

1. Stripinis, L., Paulavičius, R., and Žilinskas, J. Improved scheme for selection of potentially optimal hyper-rectangles in DIRECT // Optimization letters. Heidelberg : Springer. ISSN 1862-4472. eISSN 1862-4480. 2018, vol. 12, no 7, p. 1699-1712. DOI: 10.1007/s11590-017-1228-4.
2. Stripinis, L., Paulavičius, R., and Žilinskas, J. Penalty functions and two step selection procedure based DIRECT-type algorithm for constrained global optimization // Structural and multidisciplinary optimization. Heidelberg : Springer-Verlag. ISSN 1615-147X. eISSN 1615-1488. 2019, vol. 59, no 1, p. 2155-2175. DOI: 10.1007/s00158-018-2181-2.
3. Stripinis, L., Casado L. G., Žilinskas, J. and Paulavičius, R. On MATLAB experience in accelerating DIRECT-GLce algorithm for constrained global optimization through dynamic data structures and parallelization // Applied Mathematics and Computation. ISSN: 0096-3003. 2021, vol. 390, p. 1-17. DOI: 10.1016/j. amc.2020.125596
4. Stripinis, L. and Paulavičius, R. A modified DIRECT-GL algorithm for global optimization with hidden constraints // Optimization letters. (2020) 1–12 submitted.

Trumpos žinios apie disertantą

Linas Stripinis gimė 1989 m. Lapkričio 22 d. Skuode. 2009 m. baigė Ylakių gimnaziją. Lietuvos edukologijos universitete įgijo matematikos, informatikos bakalauro laipsnį ir mokytojo kvalifikaciją (2014 m.) ir matematikos magistro laipsnį (2016 m.). 2016–2020 m. Jis buvo Vilniaus universiteto doktorantas. Nuo 2017 m. dirba Vilniaus Universitete jaunesniuoju mokslo darbuotoju.

Literatūros sąrašas

[1] C. A. Baker, L. T. Watson, B. Grossman, W. H. Mason, and R. T. Haftka. Parallel global aircraft configuration design space exploration. In A. Tentner, editor, *High Performance Computing Symposium 2000*, pages 54–66. Soc. for Computer Simulation Internat, 2000.

[2] M. C. Bartholomew-Biggs, S. C. Parkhurst, and S. P. Wilson. Using DIRECT to solve an aircraft routing problem. *Computational Optimization and Applications*, 21(3):311–323, 2002.

[3] Mattias Björkman and Kenneth Holmström. Global optimization using the DIRECT algorithm in Matlab. *Advanced Modeling and Optimization*, 1(2):17–37, 1999.

[4] R. G. Carter, J. M. Gablonsky, A. Patrick, C. T. Kelley, and O. J. Eslinger. Algorithms for noisy problems in gas transmission pipeline optimization. *Optimization and Engineering*, 2(2):139–157, 2001.

[5] M. F. P. Costa, A. M. A. C. Rocha, and E. M. G. P. Fernandes. Filter-based direct method for constrained global optimization. *Journal of Global Optimization*, 71(3):517–536, 2018.

[6] Steven E. Cox, Raphael T. Haftka, Chuck A. Baker, Bernard Grossman, William H. Mason, and Layne T. Watson. A comparison of global optimization methods for the design of a high-speed civil transport. *Journal of Global Optimization*, 21(4):415–432, 2001.

[7] D. Di Serafino, G. Liuzzi, V. Piccialli, F. Riccio, and G. Toraldo. A modified DIviding RECTangles algorithm for a problem in astrophysics. *Journal of Optimization Theory and Applications*, 151(1):175–190, 2011.

[8] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

[9] D. E. Finkel. MATLAB source code for DIRECT. `http://www4.ncsu.edu/~ctk/Finkel_Direct/`, 2004. Online; accessed: 2017-03-22.

[10] J. M. Gablonsky. *Modifications of the DIRECT Algorithm*. PhD thesis, North Carolina State University, 2001.

[11] J. M. Gablonsky and C. T. Kelley. A locally-biased form of the DIRECT algorithm. *Journal of Global Optimization*, 21(1):27–37, 2001.

[12] Vladimir A Grishagin, Yaroslav D Sergeyev, and Roman G Strongin. Parallel characteristical algorithms for solving

problems of global optimization. *Journal of Global Optimization*, 10(2):185–206, 1997.

[13] J. He, A. Verstak, L. T. Watson, and M. Sosonkina. Design and implementation of a massively parallel version of direct. *Computational Optimization and Applications*, 2008.

[14] J. He, L. T. Watson, and M. Sosonkina. Algorithm 897: VTDIRECT95: Serial and Parallel Codes for the Global Optimization Algorithm DIRECT. *ACM Transactions on Mathematical Software*, 2010.

[15] Jian He, Alex Verstak, Masha Sosonkina, and Layne T Watson. Performance modeling and analysis of a massively parallel DIRECT–Part 2. *The International Journal of High Performance Computing Applications*, 23(1):29–41, 2009.

[16] Jian He, Alex Verstak, Layne T Watson, and Masha Sosonkina. Performance modeling and analysis of a massively parallel DIRECT–part 1. *The International Journal of High Performance Computing Applications*, 23(1):14–28, 2009.

[17] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Nonconvex Optimization and Its Application. Kluwer Academic Publishers, 1995.

[18] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Berlin, 1996.

[19] H John. Adaptation in natural and artificial systems. *The University of Michigan Press, Ann Arbor*, 1975.

[20] D. R. Jones. The DIRECT global optimization algorithm. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *The Encyclopedia of Optimization*, pages 431–440. Kluwer Academic Publishers, Dordrect, 2001.

[21] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79(1):157–181, 1993.

[22] J. Kennedy and R. Eberhart. Particle swarm optimization. *In Proceedings of the IEEE international conference on neural networks IV*, 1995.

[23] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 1983.

[24] Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.

[25] H. Liu, S. Xu, X. Chen, X. Wang, and Q. Ma. Constrained global optimization via a direct-type constraint-handling technique and an adaptive metamodeling strategy. *Structural and Multidisciplinary Optimization*, 55(1):155–177, 2017.

[26] G. Liuzzi, S. Lucidi, and V. Piccialli. A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems. *Computational Optimization and Applications*, 45:353–375, 2010.

[27] Jonas Mockus, Remigijus Paulavičius, Dainius Rusakevičius, Dmitrij Šešok, and Julius Žilinskas. Application of Reduced-set

Pareto-Lipschitzian Optimization to truss optimization. *Journal of Global Optimization*, 67(1-2):425–450, 2017.

[28] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[29] J. Na, Y. Lim, and C. Han. A modified DIRECT algorithm for hidden constraints in an LNG process optimization. *Energy*, page 488–500, 2017.

[30] P. M. Pardalos and H. E. Romeijn, editors. *Handbook of Global Optimization*, volume 2. Kluwer Academic Publishers, Dordrecht, 2002.

[31] R. Paulavičius and J. Žilinskas. Analysis of different norms and corresponding Lipschitz constants for global optimization. *Technological and Economic Development of Economy*, 36(4):383–387, 2006.

[32] R. Paulavičius and J. Žilinskas. Analysis of different norms and corresponding Lipschitz constants for global optimization in multidimensional case. *Information Technology and Control*, 36(4):383–387, 2007.

[33] Remigijus Paulavičius, Lakhdar Chiter, and Julius Žilinskas. Global optimization based on bisection of rectangles, function values at diagonals, and a set of Lipschitz constants. *Journal of Global Optimization*, 71(1):5–20, 2018.

[34] Remigijus Paulavičius, Ya. D. Sergeyev, Dmitri E. Kvasov, and Julius Žilinskas. Globally-biased DISIMPL algorithm for

expensive global optimization. *Journal of Global Optimization*, 59(2-3):545–567, 2014.

[35] Remigijus Paulavičius, Ya. D. Sergeyev, Dmitri E. Kvasov, and Julius Žilinskas. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. *Expert Systems with Applications*, 144:11305, 2020.

[36] Remigijus Paulavičius and Julius Žilinskas. Parallel branch and bound algorithm with combination of Lipschitz bounds over multidimensional simplices for multicore computers. In *Parallel Scientific Computing and Optimization*, volume 15, pages 93–102. Springer New York, 2009.

[37] Remigijus Paulavičius and Julius Žilinskas. Simplicial Lipschitz optimization without the Lipschitz constant. *Journal of Global Optimization*, 59(1):23–40, 2013.

[38] Remigijus Paulavičius and Julius Žilinskas. *Simplicial Global Optimization*. SpringerBriefs in Optimization. Springer New York, New York, NY, 2014.

[39] Remigijus Paulavičius and Julius Žilinskas. Advantages of simplicial partitioning for Lipschitz optimization problems with linear constraints. *Optimization Letters*, 10(2):237–246, 2016.

[40] Remigijus Paulavičius, Julius Žilinskas, and Andreas Grothey. Parallel branch and bound for global optimization with combination of Lipschitz bounds. *Optimization Methods and Software*, 26(3):487–498, 2011.

[41] Remigijus Paulavičius, Julius Žilinskas, Juan F.R. Herrera, and Leocadio G. Casado. A Parallel DISIMPL for Pile Placement

Optimization in Grillage-Type Foundations. In *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 525–530. IEEE, 2013.

[42] G. Di Pillo, G. Liuzzi, S. Lucidi, V. Piccialli, and F. Rinaldi. A DIRECT-type approach for derivative-free constrained global optimization. *Computational Optimization and Applications*, 65(2):361–397, 2016.

[43] G. Di Pillo, S. Lucidi, and F. Rinaldi. An approach to constrained global optimization based on exact penalty functions. *Journal of Optimization Theory and Applications*, 54(2):251–260, 2010.

[44] János D Pintér. *Global optimization in action: continuous and Lipschitz optimization: algorithms, implementations and applications*, volume 6 of *Nonconvex Optimization and Its Applications*. Springer US, 1996.

[45] S. A. Piyavskii. An algorithm for finding the absolute minimum of a function. *Theory of Optimal Solutions*, 2:13–24, 1967. in Russian.

[46] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2012.

[47] Ya D Sergeev and RG Strongin. A global minimization algorithm with parallel iterations. *USSR Computational Mathematics and Mathematical Physics*, 29(2):7–15, 1989.

[48] Ya D Sergeyev and VA Grishagin. A parallel method for

finding the global minimum of univariate functions. *Journal of optimization theory and applications*, 80(3):513–536, 1994.

[49] Ya. D. Sergeyev and D. E. Kvasov. Lipschitz global optimization. In J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh, and J. C. Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science (in 8 volumes)*, volume 4, pages 2812–2828. John Wiley & Sons, New York, 2011.

[50] Ya. D. Sergeyev, R. G. Strongin, and D. Lera. *Introduction to Global Optimization Exploiting Space-Filling Curves*. SpringerBriefs in Optimization. Springer, New York, 2013.

[51] YaD Sergeyev and VA Grishagin. Sequential and parallel global optimization algorithms. *Optim. Methods Softw*, 3:111–124, 1994.

[52] Yaroslav D Sergeyev and Dmitri E Kvasov. *Deterministic Global Optimization: An Introduction to the Diagonal Approach*. SpringerBriefs in Optimization. Springer, 2017.

[53] B. O. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9:379–388, 1972.

[54] Linas Stripinis and Remigijus Paulavičius. DIRECTLib – a library of global optimization problems for DIRECT-type methods, v1.2, 2020.

[55] R. G. Strongin and Ya. D. Sergeyev. *Global Optimization with Non-Convex Constraints: Sequential and Parallel Algorithms*. Kluwer Academic Publishers, Dordrecht, 2000.

[56] Roman G Strongin and Yaroslav D Sergeyev. Global multidimensional optimization on parallel computer. *Parallel Computing*, 18(11):1259–1273, 1992.

[57] Roman G Strongin and Yaroslav D Sergeyev. Global optimization: fractal approach and non-redundant parallelism. *Journal of Global Optimization*, 27(1):25–50, 2003.

[58] L. T. Watson and C. A. Baker. A fully-distributed parallel global search algorithm. engineering computations. *Engineering Computations*, 18(1/2):155–169, 2001.

[59] A. Žilinskas. On strong homogeneity of two global optimization algorithms based on statistical models of multimodal objective functions. *Applied Mathematics and Computation*, 218(16):8131–8136, 2012.

Introduction

Due to its simplicity and efficiency, the derivative-free global-search `DIRECT` (DIviding RECTangles) algorithm has received much consideration from the optimization community, and various novel ideas and extensions have been proposed. However, the efficiency of many `DIRECT`-type algorithms solving multimodal problems and also in cases when a solution with high accuracy is required has decreased. This thesis presents the new scheme for selecting potentially optimal hyper-rectangles in the `DIRECT` framework, which addresses both its weaknesses. An extensive experimental investigation revealed the potential and competitiveness of the added enhancements in our recent proposals, especially for more challenging multi-modal optimization problems.

Unfortunately, the original `DIRECT` algorithm addresses optimization problems only with bounds on the variables, and due to it, the application of the algorithm is limited, as various applied optimization problems often hold other types of constraints. The initial `DIRECT` extensions for problems with general constraints were not competitive, compared with other derivative-free global optimization methods. Only in recent years were a few promising `DIRECT`-type modifications were proposed. In this thesis, two different constraint handling techniques are presented, and one of these strategies can even be applied to solve problems with hidden constraints. The proposed algorithms effectively explore hyper-rectangles with infeasible midpoints close to the boundaries of feasibility and may cover feasible regions. An extensive experimental investigation revealed the potential of the proposed approaches compared with other existing `DIRECT`-type algorithms for constrained global optimization problems, including important engineering issues.

Contemporary problems often can not be solved with algorithms reasonably fast using a single core on the fastest computers. However, most DIRECT-type algorithms present challenges for efficient parallel implementation, and only a few parallel versions of DIRECT are known. To the best of our knowledge, all the existing parallel DIRECT-type versions are focused on box-constrained global optimization problems. Since the newly proposed selection scheme per iteration selects a larger number of subdividing regions, the algorithms developed in this thesis look more promising for parallelization than DIRECT. Therefore, the first parallel DIRECT-type algorithms for constrained global optimization problems are also introduced in this thesis.

Research Context And Motivation

The DIRECT algorithm [21] is well-known and broadly used for derivative-free global optimization problems. This algorithm is an extension to classical Lipschitz optimization [31, 32, 44, 45, 49, 53], where the Lipschitz constant is not assumed to be known. This feature made DIRECT-type approaches attractive for various real-world optimization problems [1, 2, 4, 6, 7, 11, 26, 35, 38]. Furthermore, the recent extensive study [46] on the large collection of optimization test problems showed that, on average, the performance of DIRECT-type algorithms is one of the best amongst all the tested state-of-the-art derivative-free global optimization approaches. The DIRECT-type algorithms outperformed algorithms belonging to other well-known optimization such classes, as Genetic [19], Simulated annealing [23], and Particle swarm optimization [22].

Nevertheless, earlier research has revealed that the DIRECT algorithm has two weaknesses [20, 34, 35, 59]. The first is delaying the discovery of the global minimum and the second is the slow fine-tuning of the solution with high accuracy, mainly in optimization problems with many local minima. Another disadvantage is that the originally published algorithm solves only box-constrained global optimization problems, and it does not naturally address other types of constraints. A less studied field in `DIRECT` is applying the algorithm for problems with general or even more complex hidden constraints. Therefore, the main research objectives of this thesis are to develop algorithms that can address its weaknesses and solve problems with various types of constraints within the `DIRECT` framework.

Lipschitz-type global optimization methods are computationally intensive [17, 18, 30, 32, 38, 50, 52] and therefore a natural way to speed it up is to parallelize them [12, 36, 40, 47, 48, 51, 56, 55, 57]. However, the iterative nature of partitioning-based `DIRECT`-type algorithms limit possibilities for effective parallelism [10, 13, 14, 15, 16, 41, 58]. Only a few parallel versions of `DIRECT`-type algorithms, mainly by the same group of researchers have been developed. Most of them focused on improving parallel efficiency by creating more computations in each iteration using an "aggressive" version of `DIRECT`, which relaxes the selection criteria by picking and subdividing a hyper-rectangle of every diameter in every iteration [1]. From the optimization point of view, this approach is not appealing because it examines many "unnecessary" (non-potentially optimal) hyper-rectangles [9, 10]. Moreover, all the existing parallel `DIRECT`-type versions are focused on box-constrained global optimization problems to the best of our knowledge. Therefore, another important issue investigated in this thesis is the development of efficient parallel `DIRECT`-type algorithms for generally constrained global optimization problems.

The Object of the Thesis

The object of the thesis is sequential and parallel `DIRECT`-type algorithms for global optimization problems with general constraints and their applicability for practical optimization problems.

The Aims and Tasks of the Research

The aims of the thesis are:

1. To increase efficiency of state-of-the-art `DIRECT`-type global optimization algorithms on optimization problems with many local minima, and where the solution with high accuracy is needed;
2. To extend `DIRECT`-type algorithms on global optimization problems with general and hidden constraints;
3. To develop efficient open-source derivative-free algorithms by taking into account algorithmic improvements, efficient data structures, and parallelization techniques.

In order to achieve these aims, the following research tasks must be accomplished:

1. To evaluate the performance of the existing state-of-the-art `DIRECT`-type global optimization algorithms and determine their weaknesses;
2. To improve existing and develop new algorithms considering identified drawbacks;
3. To develop a general constraint-handling strategy in the `DIRECT` algorithmic framework;

4. To develop an auxiliary functions-based `DIRECT`-type algorithm for optimization problems with hidden constraints;
5. To implement efficient sequential and parallel versions of the proposed algorithms, and compare their performance to other related approaches;
6. To efficiently solve challenging practical (potentially black-box) optimization problems using implemented and openly accessible tools.

Research Methodology

To analyse the received scientific results in global optimization and parallelization fields, algorithmic theory, convergence analysis, parallel computing theory, information retrieval, organization, analysis, comparative analysis, and generalization methods have been used. For the interpretation of the experimental investigation, statistical analysis and performance profiles [8] were applied to evaluate the algorithms' efficiency.

Scientific Novelty of the Work

1. Based on theoretical and experimental research, it was revealed that the efficiency of the original `DIRECT` algorithm deteriorates on optimization problems with many local minima and in cases where the solution with high accuracy is needed. To overcome these shortcomings, a new strategy for selecting potentially optimal hyper-rectangles was developed, and the proposed scheme does not require any extra parameters or local search subroutines (`DIRECT-GL`, `DIRECT-G` and `DIRECT-L`). This modification significantly outperforms the original `DIRECT`

version, solving much more complicated multi-modal and high dimension optimization problems, and can increase the accuracy of the final solution using fewer function evaluations.

2. The original `DIRECT` algorithm addresses optimization problems only with bounds on the variables. In this thesis the following algorithmic extensions for problems with other types of constraints were developed:
   2.1. New approaches (`DIRECT-GLc`, `DIRECT-GLce` and `DIRECT-GLce-min`) for general-constrained global optimization problems were proposed. The algorithm works in two phases. During the first phase, the algorithm handles infeasible initial points using constraint function information. The second phase uses an auxiliary function approach that combines information on the objective and constraint functions and seeks to find a feasible global solution.
   2.2. An approach (`DIRECT-GLh`) for problems with hidden constraints (can also be used for problems with general constraints) were proposed. The algorithm works in two phases. During the first phase, the algorithm handles infeasible initial points uniformly dividing the hyper-rectangle. The second phase uses an auxiliary function approach that estimates the necessary penalty values for the infeasible points and seeks to find a feasible global solution.
3. Instead of traditionally used static data structures, the developed algorithms were implemented using more effective dynamic data structures. Moreover, for the first time, parallel `DIRECT`-type implementations (`pD-GLce-parfor`,

`pD-GLce-spmd` and `pD-ACe-spmd`) for problems with general constraints were introduced.

4. `DIRECTlib` - a library of box and generally constrained test and practical engineering global optimization problems for benchmarking of various `DIRECT`-type algorithms was created: Stripinis, L. and Paulavičius, R. DIRECTLib – a library of global optimization problems for DIRECT-type methods, v1.2 (2020). 10.5281/zenodo.3948890; URL: `https://zenodo.org/record/3948890#.XyVOk5dR2Uk`

The developed algorithms were implemented in the `MatLab` software:

- For box-constrained global optimization problems, two implementations of sequential and parallel `DIRECT-GL` versions, also four intermediate `DIRECT-G` and `DIRECT-L` versions;
  URL: `https://github.com/LinasStripinis/DIRECT-GL`
- For general-constrained global optimization problems, two implementations of the sequential and parallel `DIRECT-GLce` versions, also two intermediate `DIRECT-GLc` and `DIRECT-GLce-min` versions;
  URL 1: `https://github.com/LinasStripinis/DIRECT-GL`
  URL 2: `https://github.com/blockchain-group/pDIRECT-GLce`
- For global optimization problems with hidden constraints, implementation of sequential `DIRECT-GLh` version;
  URL: `https://github.com/LinasStripinis/DIRECT-GL`

Participation In Scientific Programs

The author participated in the research project "Development and Applications of Bilevel Optimization Algorithms" funded by the Lithuanian Research Council (2017 - 2020). (No. P-MIP-17-60)

Defended Statements

1. A new two-step selection strategy-based algorithm (`DIRECT-GL`) performs the best among `DIRECT`-type algorithms for the most challenging non-convex problems from `DIRECTlib`, and has the fastest fine-tuning solution to a high accuracy.
2. A new auxiliary function-based algorithm (`DIRECT-GLce`) for problems with general constraints is on average the most efficient `DIRECT`-type method solving the most complex classes of optimization problems from `DIRECTlib`: high dimensional; non-linear constrained; and equality constrained.
3. A new auxiliary function-based algorithm for practically oriented (potentially black-box) problems with hidden constrains (`DIRECT-GLh`) is the most efficient `DIRECT`-type solver with respect to the number of function evaluations and the execution time solving optimization problems from `DIRECTlib`.
4. Created dynamic data structures in the developed algorithms reduce the number of necessary calculations and simplify the selection of potential optimal hyper-rectangles; this way it increases the speed of implemented algorithms more than twice.
5. Created the master-slave paradigm based SPMD-type parallel algorithm for problems with general constraints enables

preserving the determinism, and a good parallel efficiency on multi-core computing systems was achieved.

General Conclusions

1. First, a new selection scheme for the selection of potentially optimal hyper-rectangles based algorithm `DIRECT-GL` has been developed. Using the proposed approach two well-known weaknesses of `DIRECT`-type algorithms were effectively addressed. Extensive experimental studies have shown the potential of the developed `DIRECT-GL` algorithm compared to other existing `DIRECT`-type algorithms:
   - 1.1 Overall, `DIRECT-GL` requires around 59% less function evaluations to reach $\varepsilon_{\text{pe}} = 10^{-2}$ precision from the global solution compared with the second best `BIRECT` algorithm.
   - 1.2 Where a more accurate solution is needed ($\varepsilon_{\text{pe}} = 10^{-8}$), the developed algorithm in the overall required $\sim$ 72% less function evaluations compared with the second best `PLOR` algorithm.
   - 1.3 Overall, `DIRECT-GL` solved $\sim$ 32% more box-constrained test problems compared with the second-best algorithm, `DIRECT Version 4.0`, in this class.
2. Next, a new auxiliary function-based algorithm `DIRECT-GLce` has been developed to address global optimization problems with general constraints in the `DIRECT`-framework. For the considered test and practical engineering problems, `DIRECT-GLce` performance is for all problems among the best ones, and on average, is the best one among all state-of-the-art `DIRECT`-type algorithms:

2.1 Developed `DIRECT-GLc` algorithm has the most wins, and solved $\sim$ 55% of the problems with the highest efficiency.
2.2 Overall, `DIRECT-GLce` solved $\sim$ 43% more test problems compared with the second-best algorithm `DIRECT-L1`.
2.3 Hybridized `DIRECT-GLce-min` is the most effective algorithm among all pure and hybridized `DIRECT`-type methods, and overall, solved the largest number of tested problems $(85/88)$, out of them $(62/88)$ with the highest efficiency.

3. A new auxiliary function-based algorithm `DIRECT-GLh` has been developed to address global optimization problems with hidden constraints. Experimental studies revealed the potential of developed algorithm compared with all existing `DIRECT`-type algorithms in this class:
   3.1 Developed `DIRECT-GLh` algorithm has the most wins, and solved $\sim$ 52% of the test problems with the lowest number of function evaluations, among all other `DIRECT`-type approaches.
   3.2 Overall, `DIRECT-GLh` required $\sim$ 42% less function evaluations compared with the second best `DIRECT-NAS` algorithm.
4. Created dynamic data structures for a better data storage and organization, increased the developed algorithms' speed by approximately $\sim$ 62%.
5. The proposed `pD-GLce-spmd` is the first deterministic parallel `DIRECT`-type algorithm for general global optimization problems, and achieves a good parallel efficiency on a multi-core infrastructure.

Structure of the Dissertation

The dissertation consists of an introduction, five chapters, bibliography and the publications list. The total scope of the dissertation is 154 pages, including 27 figures and 17 tables. The dissertation was based on 125 literature sources.

First chapter describes the research context, presents the problem statement, discusses the motivation, aims, and objectives of the research states, research questions, describes research methods, and approbation of the research. Second chapter presents theoretical backgrounds of the `DIRECT`-type algorithms. Third chapter develops an extension of the `DIRECT` algorithm for box-constrained global optimization. Fourth chapter develops an extension of the `DIRECT-GL` algorithm for general-constrained global optimization. Fifth chapter develops an extension of the `DIRECT-GL` algorithm for problems with hidden constraints. Last chapter discusses ways to speed up the `DIRECT`-type algorithms and develop the first parallel `DIRECT`-type algorithm for problems with general constraints. And finally thesis is concludes with the main results of the research.

About the Author

Linas Stripinis was born on the 22nd of November 1989 in Skuodas. In 2009, he graduated from Ylakiai gymnasium. He received a Bachelor's degree in mathematics, informatics, teacher qualification (in 2014 m.), and a Master's degree in mathematics (in 2016 m.) at Lithuanian University of Educational Sciences. From 2016 to 2020, he was a Ph.D. student at Vilnius University. Since 2017 have been working at Vilnius University as a Junior Researcher.

Linas Stripinis

GLOBALIOJO OPTIMIZAVIMO ALGORITMŲ,
NEREIKALAUJANčIŲ IŠVESTINIŲ INFORMACIJOS,
KŪRIMAS, TOBULINIMAS IR REALIZACIJA

Daktaro disertacijos santrauka
Gamtos mokslai
Informatika (N 009)
Redaktorė Asta Markevičienė

Linas Stripinis

IMPROVEMENT, DEVELOPMENT AND IMPLEMENTATION
OF DERIVATIVE-FREE GLOBAL OPTIMIZATION
ALGORITHMS

Doctoral Dissertacion
Natural Sciences
Informatics (N 009)
Editor Zuzana Šiušaitė

UŽRAŠAMS

Vilniaus universiteto leidykla
Saulėtekio al. 9, LT-10222 Vilnius
El. p. info@leidykla.vu.lt,
www.leidykla.vu.lt
Tiražas 30 egz.