VILNIAUS UNIVERSITETAS

NERIJUS GALIAUSKAS

OPTIMIZAVIMO ALGORITMAI DAUGIAMATĖMS SKALĖMS SU MIESTO KVARTALO ATSTUMAIS IR JŲ LYGIAGRETINIMAS

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

Vilnius, 2015

Disertacija rengta 2010-2014 metais Vilniaus universiteto Matematikos ir informatikos institute

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

Disertacija ginama Vilniaus universiteto Informatikos inžinerijos mokslo krypties taryboje:

Pirmininkas:
prof. dr. Albertas Čaplinskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Nariai:
prof. dr. Romas Baronas (Vilniaus universitetas, fiziniai mokslai, informatika – 09 P),
prof. dr. Wojciech Froelich (Silezijos universitetas, technologijos mokslai, informatikos inžinerija – 07 T),
prof. habil. dr. Kazys Kazlauskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T),
prof. habil. dr. Rimantas Šeinauskas (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

Disertacija ginama viešame Informatikos inžinerijos mokslo krypties tarybos posėdyje 2015 m. rugsėjo mėn. 30 d. 9:30 val. Vilniaus universiteto Matematikos ir informatikos institute, 203 auditorijoje.

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

Disertacijos santrauka išsiuntinėta 2015 m. rugpjūčio mėn. 28 d.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje ir interneto svetainėje adresu: http://www.vu.lt/naujienos/ivykiu-kalendorius.

VILNIUS UNIVERSITY

NERIJUS GALIAUSKAS

OPTIMIZATION ALGORITHMS FOR MULTIDIMENSIONAL SCALING WITH CITY-BLOCK DISTANCES AND THEIR PARALLELIZATION

Summary of Doctoral Dissertation
Technological Sciences, Informatics Engineering (07 T)

Vilnius, 2015

The dissertation was prepared during the period 2010–2014 at the Institute of Mathematics and Informatics of Vilnius University

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

The dissertation is defended at the Council of the Scientific Field of Informatics Engineering of Vilnius University:

Chairman:
Prof. Dr. Albertas Čaplinskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Members:
Prof. Dr. Romas Baronas (Vilnius University, Physical Sciences, Informatics – 09P),
Prof. Dr. Wojciech Froelich (University of Silesia, Technological Sciences, Informatics Engineering – 07 T),
Prof. Dr. Kazys Kazlauskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T),
Prof. Dr. Rimantas Šeinauskas (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07 T).

The dissertation is defended at the public meeting of the Council of the Scientific Field of Informatics Engineering on the 30th of September, 2015, at 9:30 a.m., at the Institute of Mathematics and Informatics of Vilnius University, in 203 room.

Address: Akademijos str. 4, LT-08663 Vilnius, Lithuania.

The summary of the dissertation was distributed on the 28th of August, 2015.

The dissertation is available for review at Vilnius University Library and on this website: http://www.vu.lt/naujienos/ivykiu-kalendorius.

Turinys

Įvadas

Šiame skyriuje suformuluota tiriamoji problema, pagrįstas jos aktualumas, apibrėžti tyrimo tikslas ir uždaviniai. Be to, įvardytas mokslinis tyrimo naujumas ir pateikti ginamieji disertacijos teiginiai. Tyrimo rezultatai pristatyti devyniose mokslinėse konferencijose ir publikuoti keturiose publikacijose. Santraukoje pateikiamas publikacijų sąrašas. Be to, skyriaus gale pateikiami keli faktai apie disertacijos autorių.

Tiriamoji problema ir jos aktualumas

Tiriamoji problema susijusi su daugiamačių elementų vizualizavimu. Šioje disertacijoje daugiamačiu elementu vadinamas realusis vektorius, turintis keturias ar daugiau komponenčių. Šių vektorių vaizdavimas mažos dimensijos, t. y., vienmatėje, dvimatėje arba trimatėje, Dekarto koordinačių sistemoje yra vadinamas daugiamačių elementų vizualizavimu. Aibė taškų, vaizduojančių daugiamačius elementus mažos dimensijos Dekarto koordinačių sistemoje, vadinama daugiamačių elementų vaizdu. Atkreiptinas dėmesys, kad daugiamačių elementų vaizdas – mažos dimensijos realiosios vektorinės erdvės poaibis. Bet koks daugiamačių elementų vaizdas gali suteikti naudingos informacijos apie šiuos elementus. Pavyzdžiui, vaizdas gali padėti atskleisti ryšius tarp daugiamačių elementų, t. y. parodyti, kurie elementai yra susiję labiau, o kurie mažiau ir pan.

Esti įvairių metodų daugiamačių elementų vaizdams sudaryti. Daugiamatės skalės (angl. *Multidimensional Scaling*; toliau – DS) – vienas iš jų. Psichometrika, rinkos analizė, farmakologija – tik kelios iš sričių, kuriose taikomos DS. Kai taikomos DS, daugiamačių elementų vaizdas yra sudaromas konstruojant ir minimizuojant tam tikrą funkciją. Norint sukonstruoti funkciją, reikia 1) apibrėžti skirtingumus tarp kiekvienos daugiamačių elementų poros ir 2) pasirinkti pageidaujamą atstumo funkciją, apibrėžtą mažos dimensijos realiojoje vektorinėje erdvėje. DS metodu sudarytas vaizdas pasižymi šia ypatybe: pageidaujami atstumai tarp vaizdo taškų yra panašūs arba lygūs į apibrėžtus skirtingumus tarp atitinkamų daugiamačių elementų.

Tų pačių daugiamačių elementų vaizdai, sudaryti DS metodu, dažniausiai yra skirtingi, jeigu pasirinktos atstumo funkcijos yra skirtingos. Tarkime, kad taikant DS metodą sudaryti du tam tikrų daugiamačių elementų vaizdai: pirmasis vaizdas sudarytas su miesto kvartalo atstumo funkcija (dar žinoma, kaip pirmosios eilės Minkowski, L1 atstumo funkcija), o antrasis – su bet kuria kita atstumo funkcija. Labai tikėtina, kad, remiantis DS metodo taikymo rezultatais, pirmasis vaizdas suteiks daugiau naudingos informacijos apie pavaizduotus daugiamačius elementus negu antrasis. Deja, minimizuojama funkcija su miesto kvartalo atstumais tampa funkcija su modulio (absoliučiojo dydžio) elementais. Šios funkcijos minimizavimas – gana sudėtingas uždavinys, nes ji gali turėti ne vieną minimumo tašką ir gali būti nediferencijuojama kuriame nors iš jų. Taigi minimizavimo uždavinys, kylantis DS su miesto kvartalo atstumais, yra pagrindinė šioje disertacijoje tiriama problema.

Tyrimo tikslas ir uždaviniai

Tiriamoji problema, t. y. minimizavimo uždavinys, kylantis DS su miesto kvartalo atstumais, yra ekvivalenti tam tikram optimizavimo uždaviniui. Optimizavimo uždavinį, ekvivalentų tiriamajai problemai, toliau vadinkime uždaviniu PDS. Algoritmų uždaviniui PDS spręsti šiuo metu nėra. Tad pagrindinis tyrimo tikslas – pateikti kelis algoritmus uždaviniui PDS spręsti. Siekiant įgyvendinti užsibrėžtą tikslą, išspręsti šie uždaviniai:

(1) Ištirti pagrindines tiriamosios problemos tikslo funkcijos ypatybes ir apžvelgti egzistuojančius algoritmus tiriamajai problemai spręsti.
(2) Sukonstruoti keletą nuosekliųjų ir lygiagrečiųjų algoritmų uždaviniui PDS spręsti.
(3) Realizuoti sukonstruotus algoritmus ir ištirti juos skaitiškai, naudojant tam tikrus daugiamačių elementų rinkinius.
(4) Palyginti skaitinių tyrimų rezultatus su rezultatais, gautais taikant kitus algoritmus.

Mokslinis tyrimo naujumas

(1) Įrodytas tiriamosios problemos ir uždavinio PDS ekvivalentumas.
(2) Sudarytas algoritmas, grąžinantis lokalųjį uždavinio PDS sprendinį.
(3) Parodyta, kad uždavinys PDS gali būti performuluotas į dviejų lygmenų optimizavimo uždavinį.
(4) Sudarytas algoritmas, grąžinantis globalųjį uždavinio PDS sprendinį.
(5) Sudarytas lygiagretusis algoritmas, grąžinantis globalųjį uždavinio PDS sprendinį.

Ginamieji teiginiai

(1) Tiriamoji problema yra ekvivalenti optimizavimo uždaviniui su iškiląja kvadratine tikslo funkcija ir tiesiniais bei papildomumo (netiesiniais) apribojimais.
(2) Algoritmas, pagrįstas aktyviosios aibės metodu, grąžina lokalųjį uždavinio PDS sprendinį.
(3) Uždavinys PDS yra dviejų lygmenų optimizavimo uždavinys su iškiluoju kvadratinio programavimo uždaviniu apatiniame lygmenyje ir kombinatorinio optimizavimo uždaviniu – viršutiniame.
(4) Algoritmas, pagrįstas šakų ir rėžių metodu, grąžina globalųjį uždavinio PDS sprendinį.

Rezultatų aprobavimas

Tyrimo rezultatai pristatyti devyniose mokslinėse konferencijose (keturios iš jų – tarptautinės) ir publikuoti šiose publikacijose:

(1) Roger Fletcher, Nerijus Galiauskas, Julius Žilinskas. Quadratic programming with complementarity constraints for multidimensional scaling with city-block distances. *Baltic Journal of Modern Computing*, 2(4):248–259, 2014. ISSN 2255-8942.

(2) Nerijus Galiauskas, Julius Žilinskas. On multidimensional scaling with city-block distances. In *Lecture Notes in Computer Science*, volume 8426, p. 82–87. Springer, 2014. ISSN 0302-9743.
(3) Nerijus Galiauskas, Julius Žilinskas. Parallel branch and bound for multidimensional scaling with L1 distances formulated as quadratic programming with complementarity constraints. In *3PGCIC 2013: 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Compiégne, France, October 28–30, 2013*, p. 509–512. IEEE, 2013. ISBN 978-0-7695-5094-7.
(4) Nerijus Galiauskas, Julius Žilinskas. Kvadratinio programavimo uždaviniai. *Jaunųjų mokslininkų darbai*, 33(4):115–118, 2011. ISSN 1648-8776.

Apie autorių

Nerijus Galiauskas gimė 1985 m. vasario 22 d. Vilniuje. Baigė Vilniaus Mykolo Biržiškos gimnaziją (2004 m.). Vilniaus universitete įgijo matematikos bakalauro (2008 m.) ir matematikos magistro (2010 m.) laipsnius. Tame pačiame universitete tęsė doktorantūros studijas (2010–2014 m.). Matematinis optimizavimas ir matematinio optimizavimo uždavinių sprendimas, taikant našių skaičiavimų sistemas, – pagrindinės jo tyrimų sritys. Studijų metu Nerijus dirbo informacinių technologijų mokytoju (2008–2010 m., Vilniaus Taikos progimnazija) ir jaunesniuoju mokslo darbuotoju dviejuose moksliniuose projektuose (2012–2014 m., Vilniaus universitetas).

1. Pagrindinės sąvokos ir susiję darbai

Šiame skyriuje yra apibrėžtos optimizavimo uždavinio ir kitos susijusios sąvokos, taip pat aprašyti keli optimizavimo metodai. Skyriaus gale suformuluota tiriamoji problema bei pristatyti du žinomi algoritmai šiai problemai spręsti.

1.1. Matematinis optimizavimas

Šiame skyrelyje apibrėžtos šios sąvokos: optimizavimo uždavinys, ekvivalentumas tarp dviejų optimizavimo uždavinių ir dviejų lygmenų optimizavimo uždavinys. Taip pat aprašyti šie optimizavimo metodai: aktyviosios aibės metodas iškilojo kvadratinio programavimo uždaviniams ir šakų ir rėžių metodas. Be to, pateikta ir viena iš lygiagrečiųjų šakų ir rėžių metodų klasifikacijų.

1.1.1. Optimizavimo uždavinys

Apibrėžkime matematinio optimizavimo uždavinį [**8, 32, 34, 35, 38**]. Tarkime, kad $q \in \mathbb{N}$, r_e, $r \in \mathbb{N} \cup \{0\}$ $(r_e \leq r)$ ir $\mathbb{E} = \{1, \ldots, r_e\}$, $\mathbb{I} = \{r_e + 1, \ldots, r\}$. Tegu $c_i : \mathbb{R}^q \to \mathbb{R}$, $i \in \mathbb{E} \cup \mathbb{I}$. Sakykime, kad $\mathbb{U} \subseteq \mathbb{R}^q$ $(\mathbb{U} \neq \emptyset)$ ir

$$\mathbb{U} = \{u \in \mathbb{R}^q : \begin{array}{l} c_i(u) = 0,\ i \in \mathbb{E}, \\ c_i(u) \geq 0,\ i \in \mathbb{I}\}. \end{array} \tag{1}$$

Tegu $f : \mathbb{U} \to \mathbb{R}$. Jeigu $u^* \in \mathbb{U}$ ir

$$f(u^*) \leq f(u) \tag{2}$$

su visais $u \in \mathbb{U}$, taško u^* paieška aibėje $\mathbb{U}$ yra vadinama optimizavimo uždaviniu. Jis kartais vadinamas ir matematinio programavimo uždaviniu. Atkreiptinas dėmesys, kad taškas u^*, tenkinantis nelygybę (2), aibėje $\mathbb{U}$ gali ir neegzistuoti. Šio taško egzistavimas aibėje $\mathbb{U}$ priklauso nuo aibės $\mathbb{U}$ ir funkcijos f ypatybių. Šiame darbe nagrinėjami tokie optimizavimo uždaviniai, kurie visada turi sprendinį, t. y. taškas u^*, tenkinantis nelygybę (2), aibėje $\mathbb{U}$ visada egzistuoja.

Aibė $\mathbb{U}$, apibrėžta lygybe (1), vadinama leistinąja aibe, o funkcija f – tikslo funkcija. Funkcijos c_i, $i \in \mathbb{E} \cup \mathbb{I}$, yra vadinamos apribojimų funkcijomis arba tiesiog apribojimais. Apribojimai c_i, $i \in \mathbb{E}$, vadinami lygybiniais apribojimais, o c_i, $i \in \mathbb{I}$, – nelygybiniais. Lygybinių ir nelygybinių apribojimų, aktyvių taške $u \in \mathbb{U}$, indeksų aibė yra vadinama aktyviąja aibe taške u ir dažnai žymima simboliu $\mathbb{A}(u)$, t. y.:

$$\mathbb{A}(u) = \{i \in \mathbb{E} \cup \mathbb{I} : c_i(u) = 0\}.$$

Lygybiniai apribojimai c_i, $i \in \mathbb{E}$, yra vadinami papildomumo apribojimais, jeigu egzistuoja tokios funkcijos c_i', $c_i'' : \mathbb{R}^q \to \mathbb{R}$, kad $c_i(u) = c_i'(u)c_i''(u)$, $i \in \mathbb{E}$, $u \in \mathbb{U}$. Tegu $\mathbb{S} \subseteq \mathbb{E} \cup \mathbb{I}$. Apribojimai c_i, $i \in \mathbb{S}$, yra vadinami tiesiniais apribojimais, jeigu egzistuoja tokie $c_i \in \mathbb{R}^q$ ir $d_i \in \mathbb{R}$, kad $c_i(u) = c_i^T u - d_i$, $i \in \mathbb{S}$, $u \in \mathbb{U}$. Jeigu $c_i \in \mathbb{R}^q$, $i \in \mathbb{S}$, yra tiesinių apribojimų gradientai, tai šių gradientų matrica $(c_{i_1} \, \dots \, c_{i_{|\mathbb{S}|}}) \in \mathbb{R}^{q \times |\mathbb{S}|}$, $i_j \in \mathbb{S}$, $1 \le j \le |\mathbb{S}|$, šiame darbe dažnai yra žymima simboliu $c_{\mathbb{S}}$. Atitinkamai, vektorius $(d_{i_1}, \dots, d_{i_{|\mathbb{S}|}})^T \in \mathbb{R}^q$, $i_j \in \mathbb{S}$, $1 \le j \le |\mathbb{S}|$ žymimas simboliu $d_{\mathbb{S}}$.

Taškas $u^* \in \mathbb{U}$ yra vadinamas funkcijos f globalaus minimumo tašku aibėje $\mathbb{U}$, jeigu tenkina nelygybę (2) su visais $u \in \mathbb{U}$. Sutrumpintai dažnai jį įvardijame tiesiog „minimumo tašku“. Visų funkcijos f minimumo taškų aibėje $\mathbb{U}$ aibė dažnai žymima simboliais $\mathbb{U}_f^*$ arba $\operatorname{argmin}\{f(u) : u \in \mathbb{U}\}$.

Tegu $u' \in \mathbb{U}$ ir $\mathbb{U}_\epsilon(u') = \{u \in \mathbb{U} : d(u, u') < \epsilon\}$, kai $\epsilon > 0$ ir $d : \mathbb{U} \times \mathbb{U} \to \mathbb{R}$ yra kokia nors metrika, apibrėžta aibėje $\mathbb{U}$. Taškas $u^* \in \mathbb{U}$ yra vadinamas funkcijos f lokalaus minimumo tašku aibėje $\mathbb{U}$, jeigu egzistuoja toks taškas $u' \in \mathbb{U}$ ir skaičius $\epsilon > 0$, kad u^* yra funkcijos f globalaus minimumo taškas aibėje $\mathbb{U}_\epsilon(u')$, t. y. $u^* \in \operatorname{argmin}\{f(u) : u \in \mathbb{U}_\epsilon(u')\}$.

Funkcijos f minimumo taško aibėje $\mathbb{U}$ paieškos uždavinys, t. y. optimizavimo uždavinys, dažnai yra žymimas taip:

$$\underset{u \in \mathbb{U}}{\text{minimize}} \quad f(u). \tag{3}$$

Priklausomai nuo tikslo funkcijos f ir leistinosios aibės $\mathbb{U}$ ypatybių, optimizavimo uždavinys (3) gali priklausyti vienai ar kitai optimizavimo uždavinių klasei. Pavyzdžiui, optimizavimo uždavinys (3) yra vadinamas:

(1) Kombinatorinio optimizavimo uždaviniu, jeigu leistinoji aibė $\mathbb{U}$ yra baigtinė arba skaiti aibė.

(2) Kvadratinio programavimo uždaviniu, jeigu

$$\mathbb{U} = \{u \in \mathbb{R}^q : c_i^T u = d_i, \; i \in \mathbb{E}, \\ c_i^T u \ge d_i, \; i \in \mathbb{I}\}$$

ir

$$f(u) = 0.5 u^T A u + b^T u.$$

Čia $c_i \in \mathbb{R}^q$, $d_i \in \mathbb{R}$, $i \in \mathbb{E} \cup \mathbb{I}$, ir $A \in \mathbb{R}^{q\times q}$ ($A = A^T$, $A \neq 0$), $b \in \mathbb{R}^q$. Šiuo atveju funkcija f yra vadinama kvadratine funkcija. Jeigu $\mathbb{I} = \emptyset$, uždavinys (3) yra vadinamas kvadratinio programavimo uždaviniu su lygybiniais apribojimais.

1.1.2. Optimizavimo uždavinių ekvivalentumas

Įveskime ekvivalentumo tarp dviejų optimizavimo uždavinių sąvoką [**5, 23**]. Tarkime, kad q', $q'' \in \mathbb{N}$ ir $\mathbb{U}' \subseteq \mathbb{R}^{q'}$, $\mathbb{U}'' \subseteq \mathbb{R}^{q''}$ ($\mathbb{U}' \neq \emptyset$, $\mathbb{U}'' \neq \emptyset$). Tegu $f' : \mathbb{U}' \to \mathbb{R}$ ir $f'' : \mathbb{U}'' \to \mathbb{R}$. Sakome, kad du optimizavimo uždaviniai

$$\begin{array}{cc} \text{minimize} & f'(u') \\ u' \in \mathbb{U}' & \end{array} \quad \text{ir} \quad \begin{array}{cc} \text{minimize} & f''(u'') \\ u'' \in \mathbb{U}'' & \end{array}$$

yra ekvivalentūs tik tada, kai egzistuoja tokia bijekcija $h : \mathbb{U}' \to \mathbb{U}''$, kad

$$u'^* \in \mathbb{U}'^*_{f'} \text{ tada ir tik tada, kai } h(u'^*) \in \mathbb{U}''^*_{f''}.$$

Ekvivalenčių optimizavimo uždavinių pavyzdžių galima rasti čia: [**5, 14, 18, 39**].

1.1.3. Dviejų lygmenų optimizavimo uždavinys

Apibrėžkime dviejų lygmenų optimizavimo uždavinį [**10, 13, 15, 41**]. Tarkime, kad q', $q'' \in \mathbb{N}$ ir $\mathbb{U}' \subseteq \mathbb{R}^{q'}$, $\mathbb{U}'' \subseteq \mathbb{R}^{q''}$ ($\mathbb{U}' \neq \emptyset$, $\mathbb{U}'' \neq \emptyset$). Tegu r', $r'' \in \mathbb{N} \cup \{0\}$ ir f', c'_i, f'', $c''_j : \mathbb{U}' \times \mathbb{U}'' \to \mathbb{R}$, $1 \leq i \leq r'$, $1 \leq j \leq r''$. Optimizavimo uždavinys

$$\begin{array}{cl} \text{minimize} & f'(u', u''^*) \\ u' \in \mathbb{U}',\ u''^* \in \mathbb{U}''^*_{f''}(u') & \\ \text{subject to} & c'_i(u', u''^*) \geq 0,\ 1 \leq i \leq r', \end{array} \tag{4}$$

kuriame $\mathbb{U}''^*_{f''}(u') \subseteq \mathbb{U}''$ žymi visų funkcijos f'' minimumo taškų aibėje $\mathbb{U}''(u') = \{u'' \in \mathbb{U}'' : c''_i(u', u'') \geq 0,\ 1 \leq i \leq r''\}$ aibę, t. y. uždavinio

$$\begin{array}{cl} \text{minimize} & f''(u', u'') \\ u'' \in \mathbb{U}'' & \\ \text{subject to} & c''_i(u', u'') \geq 0,\ 1 \leq i \leq r'', \end{array} \tag{5}$$

sprendinių aibę, vadinamas dviejų lygmenų optimizavimo uždaviniu. Tad ir optimizavimo uždaviniai (4) ir (5) yra vadinami viršutinio ir apatinio lygmens uždaviniais. Savo ruožtu kintamieji u' ir u'', tikslo funkcijos f' ir f'' ir nelygybiniai apribojimai c'_i, $1 \leq i \leq r'$, ir c''_j, $1 \leq j \leq r''$, vadinami viršutinio ir apatinio lygmens kintamaisiais, tikslo funkcijomis ir nelygybiniais apribojimais.

1.1.4. Aktyviosios aibės metodas

Tarkime, kad $q \in \mathbb{N}$, r_e, $r \in \mathbb{N} \cup \{0\}$ ($r_e \leq q$, $r_e \leq r$) ir $\mathbb{E} = \{1, \ldots, r_e\}$, $\mathbb{I} = \{r_e + 1, \ldots, r\}$. Tegu $c_i \in \mathbb{R}^q$, $d_i \in \mathbb{R}$, $i \in \mathbb{E} \cup \mathbb{I}$, ir $\mathbb{U} \subseteq \mathbb{R}^q$ yra tokia aibė, kad

$$\mathbb{U} = \{u \in \mathbb{R}^q : \begin{array}{l} c_i^T u = d_i,\ i \in \mathbb{E}, \\ c_i^T u \geq d_i,\ i \in \mathbb{I}\}. \end{array}$$

Tarkime, kad vektoriai c_i, $i \in \mathbb{E}$, – tiesiškai nepriklausomi, t. y. $\operatorname{rank}(c_\mathbb{E}) = |\mathbb{E}|$. Taip pat tarkime, kad $A \in \mathbb{R}^{q\times q}$ ($A = A^T$, $A \neq 0$) ir $b \in \mathbb{R}^q$. Tegu $f : \mathbb{U} \to \mathbb{R}$ yra kvadratinė funkcija, t. y.

$$f(u) = 0.5u^T Au + b^T u.$$

Tarkime, kad funkcija f yra iškiloji [**4**]. Nagrinėkime iškilojo kvadratinio programavimo uždavinį

$$\underset{u \in \mathbb{U}}{\text{minimize}} \quad f(u). \tag{6}$$

Tarkime, kad $u^* \in \mathbb{U}$ yra funkcijos f minimumo taškas aibėje $\mathbb{U}$, $\mathbb{A}(u^*) = \{i \in \mathbb{E} \cup \mathbb{I} : c_i^T u^* = d_i\}$ ir kad vektoriai c_i, $i \in \mathbb{A}(u^*)$, yra tiesiškai nepriklausomi. Šiuo atveju, remiantis Karush-Kuhn-Tucker (KKT) sąlygomis uždaviniui (6) [**1, 5, 34**] ir aktyviosios aibės taške u^* struktūra, esti tokie Lagrange daugikliai $\lambda_i^* \in \mathbb{R}$, $i \in \mathbb{E} \cup \mathbb{I}$, kad

$$\begin{aligned} Au^* + b - \textstyle\sum_{i \in \mathbb{A}(u^*)} \lambda_i^* c_i &= 0, \\ c_i^T u^* &= d_i, \ i \in \mathbb{A}(u^*), \\ c_i^T u^* &\geq d_i, \ i \in \mathbb{I} \setminus \mathbb{A}(u^*), \\ \lambda_i^* &\geq 0, \ i \in \mathbb{I} \cap \mathbb{A}(u^*). \end{aligned} \tag{7}$$

Taigi norint rasti funkcijos f minimumo tašką aibėje $\mathbb{U}$, pakanka sudaryti ir išspręsti tokį iškilojo kvadratinio programavimo uždavinį su lygybiniais apribojimais [**34**]:

$$\begin{array}{ll} \underset{u \in \mathbb{R}^q}{\text{minimize}} & f(u) \\ \text{subject to} & c_i^T u = d_i, \ i \in \mathbb{A}(u^*). \end{array} \tag{8}$$

Uždavinio (8) sprendimas yra ekvivalentus tam tikros tiesinių lygčių sistemos, turinčios $q + |\mathbb{A}(u^*)|$ lygčių, sprendimui. Deja, nieko nežinant apie minimumo tašką u^*, sunku ką nors pasakyti ir apie aibės $\mathbb{A}(u^*)$ struktūrą. Vis dėlto aibės $\mathbb{E}$ ir $\mathbb{I}$ yra baigtinės. Vadinasi, skirtingų aktyviųjų aibių aibės $\mathbb{R}^q$ taškuose skaičius taip pat baigtinis. Tegu $\mathbb{A}$ žymi aibę tokių aktyviųjų aibių $\mathbb{A}(u) = \{i \in \mathbb{E} \cup \mathbb{I} : c_i^T u = d_i\}$, $u \in \mathbb{R}^q$, kad c_i, $i \in \mathbb{A}(u)$, yra tiesiškai nepriklausomi vektoriai, t. y.

$$\mathbb{A} = \{\mathbb{A}(u) : \ \ u \in \mathbb{U} \text{ ir } \operatorname{rank}(c_{\mathbb{A}(u)}) = |\mathbb{A}(u)|\}.$$

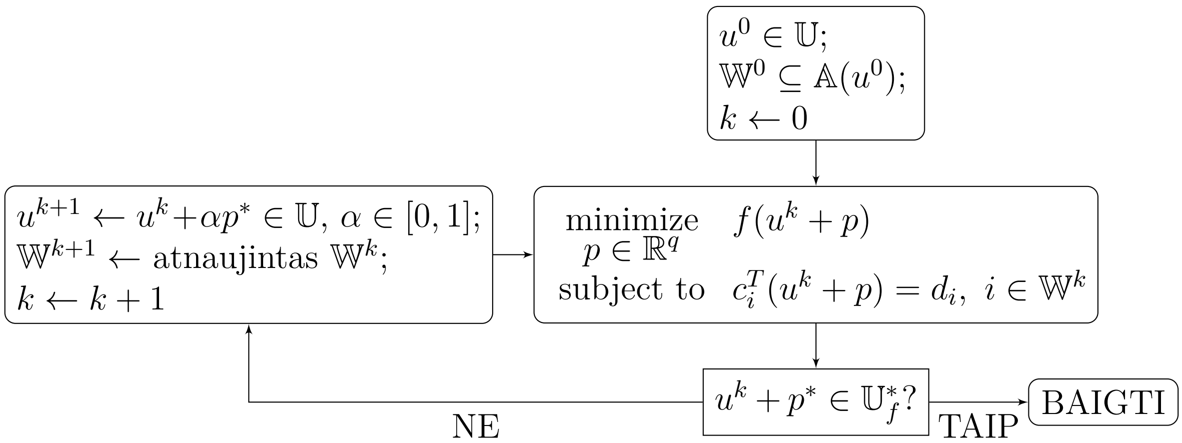
Atkreipkime dėmesį, kad $|\mathbb{A}| \leq 2^{|\mathbb{I}|}$. Taigi galime rasti aibę $\mathbb{A}(u^*)$ ir minimumo tašką u^* taikydami šią procedūrą:

1. Pasirenkame bet kurią aibės $\mathbb{A}$ aktyviąją aibę, tarkime $\mathbb{W} \in \mathbb{A}$, ir pašaliname ją iš aibės $\mathbb{A}$ ($\mathbb{A} \leftarrow \mathbb{A} \setminus \{\mathbb{W}\}$).
2. Randame funkcijos f minimumo tašką aibėje $\{u \in \mathbb{R}^q : c_i^T u = d_i, \ i \in \mathbb{W}\}$. Tegu $u^*_{\mathbb{W}} \in \mathbb{R}^q$ yra minimumo taškas.
3. Patikriname, ar taškas $u^*_{\mathbb{W}}$ ir atitinkami $\lambda_i^* \in \mathbb{R}$, $i \in \mathbb{W}$, tenkina visas (7) sąlygas:
   (a) Jeigu taip, $u^* = u^*_{\mathbb{W}}$ yra funkcijos f minimumo taškas aibėje $\mathbb{U}$ ir $\mathbb{A}(u^*) = \mathbb{W}$.
   (b) Jeigu ne, kartojame pirmąjį žingsnį.

Aktyviosios aibės metodas yra labai panašus į jau aprašytą procedūrą [**17, 20, 34**]. Šio metodo veikimo schema pateikiama 1 pav. Reikėtų paaiškinti šią schemą.

Kai taikomas aktyviosios aibės metodas uždaviniui (6) spręsti, iš pradžių parenkamas pradinis leistinasis taškas $u^0 \in \mathbb{U}$ ir pradinė aktyvioji aibė $\mathbb{W}^0 \subseteq \mathbb{A}(u^0)$. Paskui randamas funkcijos $f(u^k + p)$ minimumo taškas aibėje $\{p \in \mathbb{R}^q : c_i^T(u^k + p) - d_i = c_i^T p = 0, \ i \in \mathbb{W}^k\}$, tarkime p^*. Minimumo taškas p^* dažnai vadinamas žingsnio kryptimi. Tegu $\mathbb{U}_f^*$ žymi visų funkcijos f minimumo taškų aibėje $\mathbb{U}$ aibę. Jeigu taškas

$u^k + p^* \in \mathbb{R}^q$ ir atitinkami Lagrange daugikliai $\lambda_i^* \in \mathbb{R}$, $i \in \mathbb{W}^k$, tenkina sąlygas (7), tai $u^k + p^* = u^k \in \mathbb{U}_f^*$, priešingu atveju – $u^k + p^* \notin \mathbb{U}_f^*$. Jeigu $u^k + p^* \notin \mathbb{U}_f^*$, sukonstruojamas kitas leistinasis taškas $u^{k+1} \in \mathbb{U}$ toks, kad $f(u^{k+1}) \le f(u^k)$. Taip pat sudaroma kita leistinoji aibė $\mathbb{W}^{k+1}$: iš aibės $\mathbb{W}^k$ atimamas arba pridedamas atitinkamo apribojimo indeksas. Kartais $\mathbb{W}^{k+1} = \mathbb{W}^k$. Šis procesas tęsiamas, kol $u^K + p^* = u^K$ tampa funkcijos f minimumo tašku aibėje $\mathbb{U}$ (čia $K \in \mathbb{N}$ žymi iteracijų, reikalingų minimumo taškui rasti, skaičių).



1 PAV. Aktyviosios aibės metodo veikimo schema.

1.1.5. Šakų ir rėžių metodas

Taikant šakų ir rėžių metodą kuriam nors optimizavimo uždaviniui spręsti, 1) uždavinio leistinoji aibė dalijama į mažesnius poaibius, kol dalijimas nebeįmanomas arba beprasmis; ir 2) kiekviename iš leistinųjų poaibių yra vertinama mažiausia tikslo funkcijos reikšmė [**6, 7, 9**].

Dalijimas į poaibius yra neįmanomas, jeigu poaibis turi tik vieną tašką. Jeigu tikslo funkcijos reikšmė šiame taške yra geresnė (mažesnė) už tuo metu geriausią tikslo funkcijos reikšmę, taškas yra laikomas geriausiu tikslo funkcijos minimumo tašku visoje leistinojoje aibėje. Dalijimas į poaibius yra beprasmis, jeigu poaibyje tikrai nėra sprendžiamo uždavinio sprendinio. Leistinosios aibės dalijimas į poaibius dažnai vadinamas šakojimu. Kai sprendžiamas optimizavimo uždavinys, leistinieji poaibiai yra saugomi tam tikroje aibėje. Šią aibę pavadinkime paieškos aibe. Su paieškos aibės elementais atliekamos šios operacijos: parinkimo, pridėjimo, šalinimo. Šakų ir rėžių metodas baigia darbą, kai paieškos aibė tampa tuščia. Paskutinis geriausias tikslo funkcijos minimumo taškas visoje leistinojoje aibėje yra laikomas sprendžiamo uždavinio sprendiniu.

Jeigu sprendžiamas optimizavimo uždavinys turi sprendinį, kiekviename leistinajame poaibyje šio uždavinio tikslo funkcijos mažiausia reikšmė yra apibrėžta. Vadinasi, kiekviename leistinajame poaibyje esti po apatinį mažiausios tikslo funkcijos reikšmės rėžį. Apatinio rėžio skaičiavimas suvokiamas kaip mažiausios tikslo funkcijos reikšmės vertinimas. Be to, šie rėžiai padeda nustatyti, kada konkrečiame leistinajame poaibyje tikrai nėra sprendžiamo uždavinio sprendinio.

1.1.6. Lygiagretieji šakų ir rėžių metodai

Bet kokio uždavinio sprendimo algoritmas gali būti nuoseklus arba lygiagretus [**24, 27, 31**]. Tarkime, kad algoritmas yra baigtinė žingsnių seka tam tikram tikslui pasiekti. Sakome, kad algoritmas yra nuoseklusis, jeigu visi sekos žingsniai vykdomi nuosekliai, t. y. vienas po kito. Jeigu du ar daugiau sekos žingsnių vykdomi tuo pačiu metu, t. y. lygiagrečiai, sakome, kad algoritmas – lygiagretusis. Tarkime, kad $\mathcal{A}$ – nuoseklusis algoritmas kuriam nors optimizavimo uždaviniui spręsti, ir tarkime, kad jis pagrįstas šakų ir rėžių metodu. Yra sudarytų įvairių strategijų, leidžiančių algoritmą $\mathcal{A}$ vykdyti lygiagrečiai [**2, 12, 19, 26**]. Literatūroje, galima rasti įvairių strategijų klasifikacijų [**42**]. Tegu $\mathcal{A}_P$ yra lygiagrečioji nuoseklaus algoritmo $\mathcal{A}$ versija. Pagal klasifikaciją, pateiktą straipsnyje [**19**], sakome, kad lygiagretusis algoritmas $\mathcal{A}_P$ turi:

1-ąjį lygiagretumo laipsnį, jeigu lygiagrečiai vykdomos tokios operacijos, kaip šakojimas, rėžių skaičiavimas, leistinųjų poaibių šalinimo analizė ir pan. Tokių operacijų vykdymas lygiagrečiai paspartina viso algoritmo vykdymą. Lygiagretusis algoritmas, pristatytas šiame darbe, turi pirmąjį lygiagretumo laipsnį.

2-ąjį lygiagretumo laipsnį, jeigu paieškos aibė yra konstruojama lygiagrečiai. Šį lygiagretumo laipsnį turinčių algoritmų pavyzdžių galima rasti čia: [**30, 36**].

3-ąjį lygiagretumo laipsnį, jeigu kelios paieškos aibės yra konstruojamos lygiagrečiai. Pavyzdžiai: [**25, 33**].

Aišku, kad lygiagretusis algoritmas gali turėti kelis lygiagretumo laipsnius. Tokių algoritmų pavyzdžių galima rasti čia: [**33, 36**].

1.2. Daugiamatės skalės

Tarkime, kad turime n $(n > 1)$ daugiamačių elementų, t. y. n realiųjų vektorių, turinčių k $(k > 3)$ komponenčių. Taikydami DS metodą su miesto kvartalo atstumais, sudarykime turimų elementų vaizdą m-atėje $(m \in \{1, 2, 3\})$ Dekarto koordinačių sistemoje [**3, 11, 16, 29, 40**].

Tegu δ_{ij} žymi skirtingumą (arba panašumą) tarp daugiamačių elementų i ir j, $1 \leq i, j \leq n$. Nagrinėkime atvejį, kai $\delta_{ij} \in \mathbb{R}$ ir $\delta_{ij} = \delta_{ji} > 0$, $\delta_{ii} = 0$, $1 \leq i, j \leq n$. Skirtingumai tarp daugiamačių elementų apibrėžia simetrinę matricą, jos dydis yra n. Ši matrica dažnai yra vadinama skirtingumų matrica ir žymima simboliu Δ^n. Daugiamačiai elementai šiuo atveju suvokiami kaip realūs vektoriai. Taigi skirtingumus tarp jų galima apibrėžti, pavyzdžiui, Minkowski atstumu tarp kiekvienos daugiamačių elementų poros [**43**].

Tarkime, kad $x_i = (x_{1i}, \ldots, x_{mi})^T \in \mathbb{R}^m$, $1 \leq i \leq n$, ir $x = (x_1^T, \ldots, x_n^T)^T \in \mathbb{R}^{mn}$, t. y. $x = (x_{11}, \ldots, x_{m1}, x_{12}, \ldots, x_{m2}, \ldots, x_{mn})^T$. Tegu $f_1 : \mathbb{R}^{mn} \to \mathbb{R}$ žymi funkciją tokią, kad

$$f_1(x) = \sum_{i<j}^{n} \left(\sum_{k=1}^{m} |x_{ki} - x_{kj}| - \delta_{ij} \right)^2. \tag{9}$$

Funkcija (9) yra vadinama Stress funkcija su miesto kvartalo atstumais [**28**]. Tegu $x^* = (x_1^{*T}, \ldots, x_n^{*T})^T \in \mathbb{R}^{mn}$ yra funkcijos f_1 globalaus minimumo taškas aibėje $\mathbb{R}^{mn}$, t. y. $f_1(x^*) \leq f_1(x)$ su visais $x \in \mathbb{R}^{mn}$. Tada aibė taškų $\{x_1^*, \ldots, x_n^*\} \subset \mathbb{R}^m$

yra suvokiama kaip n daugiamačių elementų vaizdas m-atėje Dekarto koordinačių sistemoje, gautas DS metodu su miesto kvartalo atstumais. Atkreipkime dėmesį, kad funkcija f_1 turi realaus skaičiaus modulio elementų. Šios funkcijos minimizavimo uždavinys

$$\underset{x \in \mathbb{R}^{mn}}{\text{minimize}} \quad f_1(x) \tag{10}$$

yra gana sudėtingas: funkcija f_1 gali turėti ne vieną minimumo tašką [**3**], be to, gali būti nediferencijuojama kuriame nors iš minimumo taškų [**44**]. Tad minimizavimo uždavinys (10) yra pagrindinė šioje disertacijoje tiriama problema.

Optimizavimo uždaviniui (10) spręsti yra sudarytas ne vienas algoritmas, jų apžvalga pristatyta straipsniuose [**22, 46**]. Atkreipkime dėmesį į du algoritmus: algoritmą SMOOTH [**21, 22**] ir algoritmą, aprašytą straipsnyje [**45**]. Pastarasis vadinamas algoritmu BB2009. Algoritmas SMOOTH šiuo metu yra vienas iš geriausių algoritmų funkcijos f_1 lokaliam minimumo taškui aibėje $\mathbb{R}^{mn}$ rasti. O štai algoritmas BB2009 šiuo metu yra vienintelis, pagrįstas šakų ir rėžių metodu ir skirtas uždaviniui (10) spręsti.

2. Algoritmai DS su miesto kvartalo atstumais

Tiriamoji problema, t. y. optimizavimo uždavinys, kylantis DS su miesto kvartalo atstumais, yra ekvivalentus tam tikram optimizavimo uždaviniui. Optimizavimo uždavinys, ekvivalentus tiriamajai problemai, darbe vadinamas uždaviniu PDS. Šiame skyriuje, iš pradžių yra pristatytos pagrindinės Stress funkcijos su miesto kvartalo atstumais ypatybės. Paskui yra suformuluotas uždavinys PDS ir įrodytas ekvivalentumas tarp tiriamosios problemos ir uždavinio PDS. Galiausiai pristatyti trys algoritmai uždaviniui PDS spręsti.

2.1. Stress funkcijos su miesto kvartalo atstumais ypatybės

Parodykime, kad Stress funkcija su miesto kvartalo atstumais f_1 yra invariantinė perkėlimo, sukimo ir atspindėjimo atžvilgiu ir nediferencijuojama kai kuriuose iš apibrėžimo srities taškų. Tegu $L \in \mathbb{R}^{l_1 \times l_2}$ ir

$$\operatorname{diag}_k(L) = \operatorname{diag}(\underbrace{L, \dots, L}_{k}) = \begin{pmatrix} L & & \\ & \ddots & \\ & & L \end{pmatrix} \in \mathbb{R}^{l_1 k \times l_2 k} \tag{11}$$

žymi tam tikrą diagonalinę matricą.

Jeigu $c \in \mathbb{R}$, tai $|(x_{ki} - c) - (x_{kj} - c)| = |x_{ki} - x_{kj}|$ su visais $x \in \mathbb{R}^{mn}$. Vadinasi,

$$f_1(x) = f_1(x + v)$$

su visais $x \in \mathbb{R}^{mn}$ ir $v = (c_1, \dots, c_m, \dots, c_1, \dots, c_m)^T \in \mathbb{R}^{mn}$. Dėl šios ypatybės funkcija f_1 yra vadinama invariantine perkėlimo atžvilgiu. Dėl šios ypatybės funkcijos f_1 globalaus (ar lokalaus) minimumo taško galima ieškoti aibės $\mathbb{R}^{mn}$ poaibyje, o ne visoje aibėje $\mathbb{R}^{mn}$. Pavyzdžiui, poaibyje $\mathbb{X}' = \{x \in \mathbb{R}^{mn} : \sum_{i=1}^{n} x_{ki} = 0,\ 1 \le k \le m\} \subset \mathbb{R}^{mn}$. Be to, jeigu $x = (x_1^T, \ldots, x_n^T)^T \in \mathbb{X}'$, tai aibės $\{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ taškai yra išsidėstę aplink m-atės Dekarto koordinačių sistemos centrą.

Jeigu $p = (p_1, \ldots, p_m)^T \in \mathbb{P}(m)$, kai $\mathbb{P}(m)$ žymi visų aibės $\{1, \ldots, m\}$ kėlinių aibę, tai $\sum_{k=1}^{m} |x_{ki} - x_{kj}| = \sum_{k=1}^{m} |x_{p_k i} - x_{p_k j}|$ su visais $x \in \mathbb{R}^{mn}$. Vadinasi,

$$f_1(x) = f_1(\Pi(p)x) \tag{12}$$

su visais $x \in \mathbb{R}^{mn}$ ir $p \in \mathbb{P}(m)$, kai $\Pi(p) = \mathrm{diag}_n(I(p)) \in \{0,1\}^{mn \times mn}$, $I(p) = (e_{p_1} \ldots e_{p_m}) \in \{0,1\}^{m \times m}$ ir $e_{p_k} \in \{0,1\}^m$, $\|e_{p_k}\| = 1$, $e_{p_k p_k} = 1$, $1 \le k \le m$. Dėl šios ypatybės funkcija f_1 vadinama invariantine sukimo atžvilgiu.

Atkreipkime dėmesį, kad $|x_{ki} - x_{kj}| = |(-x_{ki}) - (-x_{kj})| = |x_{kj} - x_{ki}|$ su visais $x \in \mathbb{R}^{mn}$. Vadinasi,

$$f_1(x) = f_1(P(u)x)$$

su visais $x \in \mathbb{R}^{mn}$, kai $P(u) = \mathrm{diag}_n(T(u)) \in \{-1,0,1\}^{mn \times mn}$, $T(u) = \mathrm{diag}(u_1, \ldots, u_m) \in \{-1,0,1\}^{m \times m}$ ir $u \in \{-1,1\}^m$. Dėl šios ypatybės funkcija f_1 vadinama invariantine atspindėjimo atžvilgiu.

Nesunku patikrinti, kad

$$\frac{\partial f_1}{\partial x_{qr}}(x) = 2 \sum_{i<j}^{n} \left[\left(\sum_{k=1}^{m} |x_{ki} - x_{kj}| - \delta_{ij} \right) \cdot \frac{\partial |x_{qi} - x_{qj}|}{\partial x_{qr}} \right],$$

kai

$$\frac{\partial |x_{qi} - x_{qj}|}{\partial x_{qr}} = \frac{(x_{qi} - x_{qj})}{|x_{qi} - x_{qj}|} \cdot \frac{\partial (x_{qi} - x_{qj})}{\partial x_{qr}} = \begin{cases} \mathrm{sign}(x_{qi} - x_{qj}), & \text{jeigu } i = r, \\ -\mathrm{sign}(x_{qi} - x_{qj}), & \text{jeigu } j = r, \\ 0, & \text{kitu atveju,} \end{cases}$$

su visais $1 \le q \le m$, $1 \le r \le n$. Vadinasi, Stress funkcija su miesto kvartalo atstumais nėra diferencijuojama taške $x \in \mathbb{R}^{mn}$, kai $x_{qi} = x_{qj}$ su tam tikrais $q \in \{1, \ldots, m\}$ ir $i, j \in \{1, \ldots, n\}$ $(i < j)$. Taigi funkcija f_1 nėra diferencijuojama visoje apibrėžimo srityje $\mathbb{R}^{mn}$. Be to, ji gali būti nediferencijuojama net lokalaus minimumo taške [**16**].

2.2. Ekvivalentus optimizavimo uždavinys

Tegu $m \in \{1,2,3\}$, $n \in \mathbb{N}$ $(n > 1)$ ir $\delta_{ij} \in \mathbb{R}$ $(\delta_{ij} = \delta_{ji} > 0,\ \delta_{ii} = 0)$, $1 \le i, j \le n$. Tarkime, kad $x = (x_{11}, \ldots, x_{m1}, x_{12}, \ldots, x_{m2}, \ldots, x_{mn})^T \in \mathbb{R}^{mn}$ ir $d^{\pm} = (d_{112}^{+}, d_{112}^{-}, \ldots, d_{m12}^{+}, d_{m12}^{-}, \ldots, d_{m(n-1)n}^{+}, d_{m(n-1)n}^{-})^T \in \mathbb{R}^{mn(n-1)}$. Tegu $\mathbb{Y}$ yra toks aibės $\mathbb{R}^{mn^2}$ poaibis, kad,

$$\mathbb{Y} = \{y = (x^T, {d^{\pm}}^T)^T \in \mathbb{R}^{mn^2} : \begin{array}{l} x_{ki} - x_{kj} = d_{kij}^{+} - d_{kij}^{-}, \\ d_{kij}^{+} d_{kij}^{-} = 0, \\ d_{kij}^{+}, d_{kij}^{-} \ge 0, \\ 1 \le i < j \le n, 1 \le k \le m\}, \end{array} \tag{13}$$

ir $g : \mathbb{Y} \to \mathbb{R}$ yra tokia funkcija, kad

$$g(y) = \sum_{i<j}^{n} \left(\sum_{k=1}^{m} (d_{kij}^{+} + d_{kij}^{-}) - \delta_{ij} \right)^2. \tag{14}$$

Parodykime, kad optimizavimo uždavinys (10) yra ekvivalentus optimizavimo uždaviniui

$$\underset{y \in \mathbb{Y}}{\text{minimize}} \quad g(y). \tag{15}$$

Teorema 1. *Optimizavimo uždaviniai (10) ir (15) yra ekvivalentūs uždaviniai.*

Įrodymas. Tegu $\mathbb{X} = \mathbb{R}^{mn}$ žymi uždavinio (10) leistinąją sritį. Funkcijos f_1 visų minimumo taškų aibėje $\mathbb{X}$ aibę pažymėkime simboliu $\mathbb{X}^*_{f_1}$, t. y., jeigu $x \in \mathbb{X}^*_{f_1}$, tai $f_1(x^*) \leq f_1(x)$ su visais $x \in \mathbb{X}$. Tad tegu ir $\mathbb{Y}^*_g$ žymi visų funkcijos g minimumo taškų aibėje $\mathbb{Y}$ aibę. Parodykime, kad egzistuoja tokia bijekcija (injekcija ir surjekcija) $h : \mathbb{X} \to \mathbb{Y}$, kad

$$x^* \in \mathbb{X}^*_{f_1} \text{ tada ir tik tada, kai } h(x^*) = y^* \in \mathbb{Y}^*_g. \tag{16}$$

Nagrinėkime funkciją

$$h(x) = \begin{pmatrix} x \\ z^{\pm}(x) \end{pmatrix},$$

kai $z^{\pm}(x) = (z^+_{112}(x), z^-_{112}(x), \ldots, z^+_{m12}(x), z^-_{m12}(x), \ldots, z^+_{m(n-1)n}(x), z^-_{m(n-1)n}(x))^T \in \mathbb{R}^{mn(n-1)}$ ir

$$z^+_{kij}(x) = \begin{cases} x_{ki} - x_{kj}, & \text{jeigu } x_{ki} > x_{kj}, \\ 0, & \text{kitu atveju,} \end{cases}$$
$$z^-_{kij}(x) = \begin{cases} -(x_{ki} - x_{kj}), & \text{jeigu } x_{ki} < x_{kj}, \\ 0, & \text{kitu atveju.} \end{cases}$$

Atkreipkime dėmesį, kad

$$\begin{aligned} &z^+_{kij}(x) - z^-_{kij}(x) = x_{ki} - x_{kj}, \\ &z^+_{kij}(x) z^-_{kij}(x) = 0, \\ &z^+_{kij}(x), z^-_{kij}(x) \geq 0, \end{aligned}$$

su visais $1 \leq i < j \leq n$, $1 \leq k \leq m$, tik kai $x \in \mathbb{X}$. Taigi $h(x) \in \mathbb{Y}$ su visais $x \in \mathbb{X}$.

Iš funkcijos h apibrėžimo aišku, kad funkcija h yra injekcija, t. y., jeigu $h(x^1) = h(x^2)$, tai $x^1 = x^2$ su visais x^1, $x^2 \in \mathbb{X}$.

Funkcija h yra surjekcija, t. y. kiekvienam $y = (x^T, {d^{\pm}}^T)^T \in \mathbb{Y}$ egzistuoja tam tikras $x \in \mathbb{X}$ toks, kad $y = h(x)$. Iš tikrųjų tegu $(x^T, {d^{\pm}}^T)^T \in \mathbb{Y}$. Parodykime, kad $d^{\pm} = z^{\pm}(x)$. Remiantis aibės $\mathbb{Y}$ apibrėžimu, nesunku patikrinti, kad

$$\begin{aligned} &\text{jeigu } d^+_{kij} = 0, \text{ tai } d^-_{kij} = -(x_{ki} - x_{kj}) \geq 0 \text{ ir} \\ &\text{jeigu } d^-_{kij} = 0, \text{ tai } d^+_{kij} = x_{ki} - x_{kj} \geq 0. \end{aligned}$$

Tarkime, kad $z^+_{kij}(x) = 0$. Tada $x_{ki} \leq x_{kj}$ ir $z^-_{kij}(x) = -(x_{ki} - x_{kj}) \geq 0$. Jeigu $z^-_{kij}(x) = 0$, tai $x_{ki} \geq x_{kj}$ ir $z^+_{kij}(x) = x_{ki} - x_{kj} \geq 0$. Taigi $d^{\pm} = z^{\pm}(x)$, tik kai $(x^T, {d^{\pm}}^T)^T = y \in \mathbb{Y}$. Iš čia akivaizdu, kad kiekvienam $y \in \mathbb{Y}$ egzistuoja tam tikras $x \in \mathbb{X}$ toks, kad $y = h(x)$. Taigi funkcija h yra surjekcija.

Nesunku patikrinti, kad jeigu $(x^T, {d^{\pm}}^T)^T \in \mathbb{Y}$, tai $d^+_{kij} + d^-_{kij} = |x_{ki} - x_{kj}|$ su visais $1 \leq i < j \leq n$, $1 \leq k \leq m$. Tad aišku, kad

$$f_1(x) = g(y), \text{ kai tik } y = (x^T, {d^{\pm}}^T)^T \in \mathbb{Y}. \tag{17}$$

Vadinasi, $x^* \in \mathbb{X}^*_{f_1}$, t. y. $f_1(x^*) \leq f_1(x)$ su visais $x \in \mathbb{X}$, tik tada, kai $g(h(x^*)) \leq g(h(x))$ su visais $h(x) \in \mathbb{Y}$, t. y. $h(x^*) = y^* \in \mathbb{Y}^*_g$. Taigi bijekcija h tenkina formulę (16). Taip įrodoma, kad uždaviniai (10) ir (15) yra ekvivalentūs optimizavimo uždaviniai. $\square$

Funkcija f_1 yra invariantinė perkėlimo atžvilgiu. Kadangi $f_1(x) = g(y)$, tik kai $y = (x^T, d^{\pm T})^T \in \mathbb{Y}$, funkcija g taip pat yra invariantinė perkėlimo atžvilgiu. Tegu

$$\mathbb{Y}' = \mathbb{Y} \cap \{y \in (x^T, d^{\pm T})^T \in \mathbb{R}^{mn^2} : \sum_{i=1}^{n} x_{ki} = 0,\ 1 \leq k \leq m\}.$$

Toliau nagrinėkime optimizavimo uždavinį

$$\begin{array}{ll} \text{minimize} & g(y) \\ y \in \mathbb{Y}' & \end{array} \tag{18}$$

Optimizavimo uždavinys (18) čia vadinamas uždaviniu PDS.

Uždavinio PDS tikslo funkcija g – kvadratinė funkcija, o leistinoji aibė $\mathbb{Y}'$ apibrėžta tiesiniais bei papildomumo apribojimais. Užrašykime uždavinį (18) matricos forma. Tarkime, kad $y = (x^T,\ d^{\pm T})^T = (y_1, \ldots,\ y_{mn},\ y_{mn+1}, \ldots,\ y_{mn^2})^T \in \mathbb{R}^{mn^2}$, $s = mn(n-1)/2$ ir $\mathrm{diag}_k(L) \in \mathbb{R}^{l_1 k \times l_2 k}$, kai $L \in \mathbb{R}^{l_1 \times l_2}$, yra diagonalinė matrica, apibrėžta lygybe (11). Tegu $I^l \in \{0,1\}^{l \times l}$ žymi tam tikrą vienetinę matricą, o $O^{l_1 \times l_2} \in \{0\}^{l_1 \times l_2}$ – nulinę. Tada optimizavimo uždavinys (18) gali būti užrašytas tokia matricos forma:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} y^T A y - b^T y + \text{const.} \\ y \in \mathbb{R}^{mn^2} & \\ \text{subject to} & c_i^T y = 0,\ i \in \mathbb{E} = \{1, \ldots, m+s\}, \\ & c_i^T y \geq 0,\ i \in \mathbb{I} = \{m+s+1, \ldots, m+3s\}, \\ & y_{(mn+i)} y_{(mn+1+i)} = 0,\ i = 1, 3, 5, \ldots, 2s-1. \end{array} \tag{19}$$

kai:

$$\begin{aligned} A &= \begin{pmatrix} O^{mn \times mn} & O^{mn \times 2s} \\ O^{2s \times mn} & \mathrm{diag}_{n(n-1)/2}(E) \end{pmatrix} \in \{0,1\}^{mn^2 \times mn^2}, \\ b &= (O^{1 \times mn}, \Delta_{12}^T, \Delta_{13}^T, \ldots, \Delta_{(n-1)n}^T)^T \in \mathbb{R}^{mn^2}, \\ c_{\mathbb{E}} &= \begin{pmatrix} C^1 & C^2 \\ O^{2s \times m} & \mathrm{diag}_s \left(\begin{smallmatrix} -1 \\ 1 \end{smallmatrix} \right) \end{pmatrix} \in \{-1,0,1\}^{mn^2 \times m+s}, \\ c_{\mathbb{I}} &= \begin{pmatrix} O^{mn \times 2s} \\ I^{2s} \end{pmatrix} \in \{0,1\}^{mn^2 \times 2s}. \end{aligned}$$

Čia $E \in \{1\}^{2m \times 2m}$, $\Delta_{ij} = (\delta_{ij}, \ldots, \delta_{ij})^T \in \mathbb{R}^{2m}$, $1 \le i < j \le n$, $C^1 \in \{0,1\}^{mn \times m}$, $C^2 \in \{-1,0,1\}^{mn \times s}$ ir

$$C^1 = \begin{pmatrix} I^m \\ \vdots \\ I^m \end{pmatrix},$$

$$C^2 = \begin{pmatrix} I^m & I^m & \ldots & I^m & & & & & & \\ -I^m & & & & I^m & I^m & \ldots & I^m & & \\ & -I^m & & & -I^m & & & & & \\ & & & & & -I^m & & & & \\ & & \ddots & & & & \ddots & & \ldots & \\ & & & & & & & & & I^m \\ & & & -I^m & & & & -I^m & & -I^m \end{pmatrix}.$$

Atkreipkime dėmesį, kad matricos A determinantas lygus 0, t. y. $\det(A) = 0$. Be to, nesunku patikrinti, kad $A = ((2m)^{-1/2}A)^T\ ((2m)^{-1/2}A)$. Taigi $A \succeq 0$, t. y. kvadratinės funkcijos g Hessian matrica A, apibrėžta neneigiamai. Vadinasi, funkcija g – negriežtai iškila kvadratinė funkcija.

2.3. Algoritmas, pagrįstas aktyviosios aibės metodu

Šiame skyrelyje pristatytas algoritmas uždaviniui (19) spręsti. Jis pagrįstas aktyviosios aibės metodu (žr. skyrelį 1.1.4) ir grąžina funkcijos g lokalųjį minimumo tašką aibėje $\mathbb{Y}'$.

Prisiminkime, kad taikant aktyviosios aibės metodą iškilojo kvadratinio programavimo uždaviniui spręsti sukonstruojama baigtinė leistinųjų taškų seka, konverguojanti į uždavinio sprendinį. Atkreipkime dėmesį, kad iškilojo kvadratinio programavimo uždavinio leistinoji aibė yra apibrėžta tik tiesiniais apribojimais. Tačiau uždavinio (19) leistinoji aibė yra apibrėžta tiek tiesiniais, tiek papildomumo (netiesiniais) apribojimais. Taigi negalime tiesiogiai taikyti aktyviosios aibės metodo uždaviniui (19) spręsti. Vis dėlto šis metodas gali būti pritaikytas uždaviniui (19), jei modifikuojamos aktyviųjų aibių $\mathbb{W}^{k+1}$ sudarymo taisyklės. Tiksliau tariant, kai sudaroma aktyvioji aibė $\mathbb{W}^{k+1}$ iš aibės $\mathbb{W}^k$, uždraudžiame aibėje $\mathbb{W}^k$ pašalinti kai kurių nelygybinių apribojimų indeksus. Taip kiekvienas konverguojančios sekos elementas yra išsaugomas leistinojoje aibėje, t. y. priklauso aibei $\mathbb{Y}'$. Deja, dėl šios modifikacijos algoritmas grąžina funkcijos g lokalųjį minimumo tašką aibėje $\mathbb{Y}'$.

Algoritmas, pagrįstas aktyviosios aibės metodu ir skirtas uždaviniui (19) spręsti, detaliai aprašytas algoritme 1. Šį algoritmą pavadinome „Modified Active-Set“ (MAS) algoritmu.

2.4. Algoritmas, pagrįstas šakų ir rėžių metodu

Šiame skyrelyje pristatytas kitas algoritmas uždaviniui (19) spręsti. Jis pagrįstas šakų ir rėžių metodu (žr. skyrelį 1.1.5) ir grąžina funkcijos g globalaus minimumo tašką aibėje $\mathbb{Y}'$. Optimizavimo uždavinys (19) gali būti suformuluotas kaip dviejų lygmenų optimizavimo uždavinys: su iškilojo kvadratinio programavimo uždaviniu apatiniame lygmenyje ir kombinatorinio optimizavimo uždaviniu – viršutiniame. Pritaikėme šakų ir rėžių metodą viršutinio lygmens kombinatorinio optimizavimo

Algoritmas 1 Modified Active-Set (MAS)

Duota: $m \in \{1,2,3\}$, $n \in \mathbb{N}$ $(n > 1)$, $\delta_{ij} \in \mathbb{R}$ $(\delta_{ij} = \delta_{ji} > 0$, $\delta_{ii} = 0)$, $1 \le i, j \le n$
Rasti: y^* – funkcijos g minimumo taškas aibėje $\mathbb{Y}'$
$y^0 \in \mathbb{Y}'$ – bet koks leistinasis taškas
$\mathbb{W}^0 \leftarrow \mathbb{E}$
for all $i \in \{1, 3, \ldots, 2s-1\}$ **do**
  $j \leftarrow 0$
  if $y^0_{(mn+1+i)} = 0$ **then**
    $j \leftarrow 1$
    if $y^0_{(mn+i)} = 0$ **then**
      $j \leftarrow$ atsitiktinis skaičius iš aibės $\{0, 1\}$
  $\mathbb{W}^0 \leftarrow \mathbb{W}^0 \cup \{m + s + i + j\}$
$stop \leftarrow 0$; $k \leftarrow 0$
while $stop \neq 1$ **do**
  $\begin{pmatrix} p^k \\ \lambda^k \end{pmatrix} \in \left\{ \begin{pmatrix} p \\ \lambda \end{pmatrix} \in \mathbb{R}^{mn^2+|\mathbb{W}^k|} : \begin{pmatrix} -A & c_{\mathbb{W}^k} \\ c^T_{\mathbb{W}^k} & 0 \end{pmatrix} \begin{pmatrix} p \\ \lambda \end{pmatrix} = \begin{pmatrix} Ay^k - b \\ 0 \end{pmatrix} \right\}$
  if $p^k \neq 0$ **then**
    $\tilde{\mathbb{I}} \leftarrow \{i \in \mathbb{I} : i \notin \mathbb{W}^k \text{ ir } c_i^T p^k < 0\}$
    $\alpha^k \leftarrow \min\{1, \min\{-(c_i^T y^k)/(c_i^T p^k) : i \in \tilde{\mathbb{I}}\}\}$
    if $\alpha^k = -(c_{i^*}^T y^k)/(c_{i^*}^T p^k) \le 1$ su tam tikru $i^* \in \tilde{\mathbb{I}}$ **then**
      $\mathbb{W}^{k+1} \leftarrow \mathbb{W}^k \cup \{i^*\}$
    $y^{k+1} \leftarrow y^k + \alpha^k p^k$
  else
    $\lambda \leftarrow 0 \in \mathbb{R}^{m+3s}$; $j \leftarrow 0$
    for all $i \in \mathbb{W}^k$ **do**
      $j \leftarrow j + 1$; $\lambda_i \leftarrow \lambda_j^k$
    $\tilde{\mathbb{I}} \leftarrow \{i \in \mathbb{W}^k \cap \mathbb{I} : \lambda_i < 0\}$
    $next \leftarrow 1$
    while $next = 1$ **do**
      if $\tilde{\mathbb{I}} \neq \emptyset$ **then**
        $i^* \leftarrow \operatorname{argmin}\{\lambda_i : i \in \tilde{\mathbb{I}}\}$
        $j \leftarrow 1$
        if $(i^* - (m + s))\%2 = 0$ **then**
          $j \leftarrow -1$
        if $(i^* + j) \in \mathbb{W}^k$ **then**
          $\mathbb{W}^{k+1} \leftarrow \mathbb{W}^k \setminus \{i^*\}$; $next \leftarrow 0$
        else
          $\tilde{\mathbb{I}} \leftarrow \tilde{\mathbb{I}} \setminus \{i^*\}$
      else
        $y^* \leftarrow y^k$; $next \leftarrow 0$; $stop \leftarrow 1$
  $k \leftarrow k + 1$

uždaviniui. Tad šiame skyrelyje uždavinys (19) suformuluotas kaip dviejų lygmenų optimizavimo uždavinys, apibrėžtos šakų ir rėžių metodo operacijos viršutinio lygmens uždaviniui ir pristatytas pats algoritmas.

2.4.1. Dviejų lygmenų optimizavimas

Parodykime, kad leistinoji aibė $\mathbb{Y}'$ gali būti padalyta į baigtinį poaibių skaičių. Tarkime, kad $t : \mathbb{Y}' \to \{0,1\}^{2s}$ yra tokia 0-1 vektorinė funkcija, kad:

$$t_i(y) = \begin{cases} 1, & \text{if } y_{mn+i} > 0, \\ 0, & \text{if } y_{mn+i} = 0, \end{cases}$$

$1 \leq i \leq 2s$. Tegu $t(\mathbb{Y}') = \{t(y) : y \in \mathbb{Y}'\}$ žymi funkcijos t reikšmių sritį. Kadangi $y_{(mn+i)}y_{(mn+1+i)} = 0$, tai $0 \leq t_i(y) + t_{i+1}(y) \leq 1$, $i = 1, 3, 5, \ldots, 2s-1$. Taigi funkcijos t reikšmių sritis gali būti apibrėžta ir taip:

$$t(\mathbb{Y}') = \{z \in \{0,1\}^{2s} : z_i + z_{i+1} \leq 1,\ i = 1, 3, 5, \ldots, 2s-1\}.$$

Aišku, kad aibė $t(\mathbb{Y}')$ yra baigtinė. Prisiminkime apribojimų indeksų aibių $\mathbb{E}$ ir $\mathbb{I}$ apibrėžimus (žr. (19)) ir tegu $\mathbb{I}_{eq}(z) \subset \mathbb{I}$ yra tokia aibė, kad

$$\mathbb{I}_{eq}(z) = \{i \in \mathbb{I} : z_{i-(m+s)} = 0\}, \tag{22}$$

kai $z \in t(\mathbb{Y}')$. Tada $\mathbb{Y}' = \bigcup_{z \in t(\mathbb{Y}')} \mathbb{Y}'(z)$, kai

$$\begin{aligned} \mathbb{Y}'(z) = \ & \{y \in \mathbb{Y}' : c_i^T y = 0,\ i \in \mathbb{I}_{eq}(z)\} = \\ & \{y \in \mathbb{R}^{mn^2} : c^T_{\mathbb{E} \cup \mathbb{I}_{eq}(z)} y = 0,\ c^T_{\mathbb{I} \setminus \mathbb{I}_{eq}(z)} y \geq 0\}. \end{aligned}$$

Taigi uždavinio (19) leistinoji aibė gali būti padalyta į baigtinį poaibių skaičių. Be to, pakanka nagrinėti tokius $z \in t(\mathbb{Y}')$, kurie tenkina lygybes $z_i + z_{i+1} = 1$, $i = 1, 3, 5, \ldots, 2s-1$. Iš tikrųjų, tarkime, kad

$$\begin{aligned} z^1 &= (z_1^1, \ldots, z^1_{(i-1)}, 1, 0, z^1_{(i+2)}, \ldots, z^1_{(2s)})^T, \\ z^2 &= (z_1^2, \ldots, z^2_{(i-1)}, 0, 0, z^2_{(i+2)}, \ldots, z^2_{(2s)})^T \text{ ir} \\ z^3 &= (z_1^3, \ldots, z^3_{(i-1)}, 0, 1, z^3_{(i+2)}, \ldots, z^3_{(2s)})^T \in t(\mathbb{Y}'), \end{aligned}$$

kai $z_j^1 = z_j^2 = z_j^3$, $j \in \{1, \ldots, 2s\} \setminus \{i, i+1\}$. Iš aibės $\mathbb{Y}'(z)$ apibrėžimo išplaukia, kad $\mathbb{Y}'(z^2) \subset \mathbb{Y}'(z^1)$ ir $\mathbb{Y}'(z^2) \subset \mathbb{Y}'(z^3)$. Taigi leistinoji aibė $\mathbb{Y}'$ gali būti apibrėžta ir taip: $\mathbb{Y}' = \bigcup_{z \in \mathbb{Z}} \mathbb{Y}'(z)$, kai

$$\begin{aligned} \mathbb{Z} = \ & \{z \in \{0,1\}^{2s} : z_i + z_{i+1} = 1,\ i = 1, 3, 5, \ldots, 2s-1\} = \\ & \{z \in \{0,1\}^{2s} : z_{2i-1} + z_{2i} = 1,\ 1 \leq i \leq s\}. \end{aligned}$$

Taip pat atkreipkime dėmesį, kad jeigu $N = s/m = n(n-1)/2$ ir $y = (x^T, d^{\pm T})^T \in \mathbb{Y}'(z)$, $z \in \mathbb{Z}$, tai:

$$\begin{aligned} \textstyle\sum_{i=1}^n x_{ki} = 0, &\quad x_{ki} - x_{kj} = d^+_{kij} - d^-_{kij}, \\ (1 - z_{2k-1}) d^+_{k12} = 0, &\quad (1 - z_{2k}) d^-_{k12} = 0, \\ (1 - z_{2(m+k)-1}) d^+_{k13} = 0, &\quad (1 - z_{2(m+k)}) d^-_{k13} = 0, \\ &\quad \ldots \\ (1 - z_{2Nk-1}) d^+_{k(n-1)n} = 0, &\quad (1 - z_{2Nk}) d^-_{k(n-1)n} = 0, \\ d^+_{kij} \geq 0, &\quad d^-_{kij} \geq 0, \end{aligned} \tag{23}$$

su visais $1 \le i < j \le n$, $1 \le k \le m$.

Taigi uždavinys (19) gali būti suformuluotas kaip dviejų lygmenų optimizavimo uždavinys:

$$\begin{array}{cc} \text{minimize} & g(y^*), \\ z \in \mathbb{Z},\ y^* \in \mathbb{Y}_g'^*(z) & \end{array} \tag{24}$$

kai $\mathbb{Y}_g'^*(z) \subset \mathbb{Y}'(z)$ žymi visų funkcijos g minimumo taškų aibėje $\mathbb{Y}'(z)$ aibę, t. y. uždavinio

$$\begin{array}{cc} \text{minimize} & g(y). \\ y \in \mathbb{Y}'(z) & \end{array} \tag{25}$$

sprendinių aibę. Viršutinio lygmens uždavinys (24) yra kombinatorinio optimizavimo uždavinys, o apatinio lygmens uždavinys (25) – negriežtai iškilojo kvadratinio programavimo uždavinys. Atkreipkime dėmesį, kad z_i, $z_{i+1} \in \{0,1\}$, kai $z_i + z_{i+1} = 1$, tik tada, kai $(z_i, z_{i+1}) \in \{(0,1),\ (1,0)\}$. Vadinasi, apatinio lygmens uždavinių skaičius lygus $|\mathbb{Z}| = 2^s$.

2.4.2. Operacijos

Pritaikėme šakų ir rėžių metodą viršutinio lygmens kombinatoriniam optimizavimo uždaviniui (24). Reikia parodyti, kaip šios operacijos realizuojamos: 1) viršutinio lygmens leistinosios aibės $\mathbb{Z}$ skaidymas, 2) mažiausios viršutinio lygmens tikslo funkcijos reikšmės įvertinimas kiekviename leistinajame poaibyje ir 3) nereikalingų leistinųjų poaibių šalinimas.

Tegu $\mathbb{V} \subset \{0,1\}^{2s}$ yra tokia aibė, kad $\mathbb{V} = \mathbb{V}(0) \cup \mathbb{V}(1) \cup \mathbb{V}(2) \cup \ldots \cup \mathbb{V}(s)$ ir

$$\begin{aligned} \mathbb{V}(k) = \{v \in \{0,1\}^{2s} :\ & v_{2i-1} + v_{2i} = 1,\ 1 \le i \le k, \\ & v_{2i-1} + v_{2i} = 2,\ k+1 \le i \le s\}, \end{aligned}$$

su visais $0 \le k \le s$. Tada viršutinio lygmens leistinoji aibė $\mathbb{Z}$ skaidoma į šiuos poaibius:

$$\begin{aligned} \mathbb{Z}(v) = \{z \in \{0,1\}^{2s} :\ & z_i = v_i,\ 1 \le i \le 2k, \\ & z_{2i-1} + z_{2i} = 1,\ k+1 \le i \le s\}, \end{aligned} \tag{26}$$

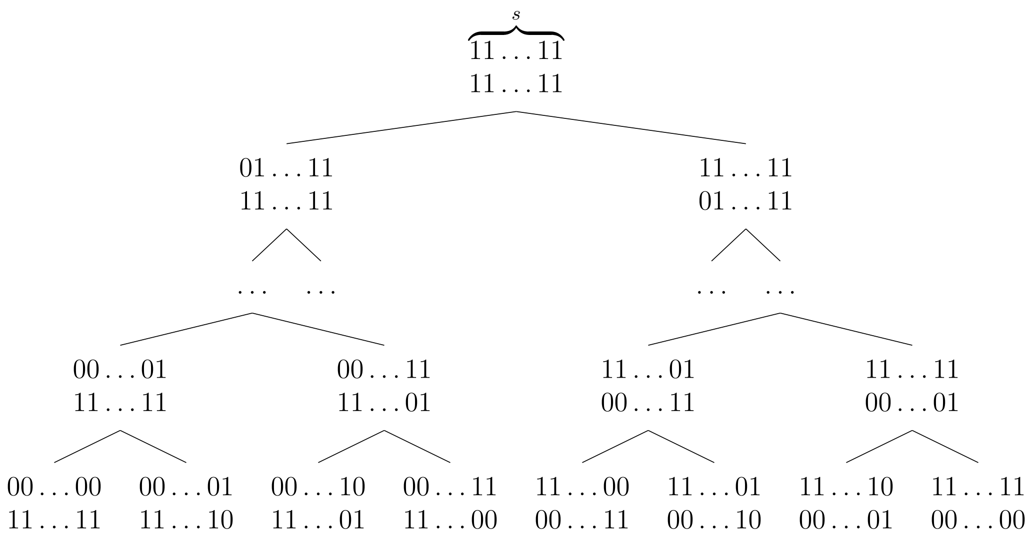
kai $v \in \mathbb{V}(k)$, $k \in \{0, 1, \ldots, s\}$. Nesunku patikrinti, kad

(1) $|\mathbb{V}(k)| = 2^k$.
(2) $|\mathbb{V}| = \sum_{k=0}^{s} |\mathbb{V}(k)| = \sum_{k=0}^{s} 2^k = 2^{s+1} - 1$.
(3) $|\mathbb{Z}(v)| = 2^{s-k}$, $v \in \mathbb{V}(k)$.
(4) $\mathbb{V}(s) = \mathbb{Z}(v) = \mathbb{Z}$, $v \in \mathbb{V}(0)$.
(5) $\bigcup_{i=1}^{2^k} \mathbb{Z}(v_i) = \mathbb{Z}$, $v_i \in \mathbb{V}(k)$, $1 \le i \le 2^k$.
(6) $\bigcap_{i=1}^{2^k} \mathbb{Z}(v_i) = \emptyset$, $v_i \in \mathbb{V}(k)$, $1 \le i \le 2^k$.

Aibė $\mathbb{V}$ gali būti pavaizduota kaip medis. Kiekviena medžio viršūnė atitinka tam tikrą aibės $\mathbb{V}$ elementą, kuris savo ruožtu atitinka tam tikrą aibės $\mathbb{Z}$ poaibį. Šis medis vadinamas paieškos medžiu. Paieškos medis, sukonstruotas uždaviniui (24) vaizduojamas pav. 2. Kiekviena viršūnė

$$\begin{array}{l} \overbrace{v_1 \ldots v_{2k-1} 1 \ldots 11}^{s} \\ \underbrace{v_2 \ldots v_{2k}}_{k}\ \ 1 \ldots 11 \end{array}$$

atitinka tam tikrą vektorių $v \in \mathbb{V}(k)$, $k \in \{0, 1, \ldots, s\}$, kuris savo ruožtu atitinka leistinąjį poabį $\mathbb{Z}(v) \subseteq \mathbb{Z}$.



2 Pav. Paieškos medis, sudarytas uždaviniui (24).

Mažiausia tikslo funkcijos g reikšmė aibėje $\mathbb{Z}(v)$ yra lygi $g(y^*)$, kai

$$y^* \in \operatorname{argmin}\{g(y) : y \in \bigcup_{z \in \mathbb{Z}(v)} \mathbb{Y}'(z)\}.$$

Atkreipkime dėmesį, kad $s = |\mathbb{I}_{eq}(z)| \geq |\mathbb{I}_{eq}(v)| = k$, kai tik $z \in \mathbb{Z}(v)$ ir $v \in \mathbb{V}(k)$, $k \in \{0, 1, \ldots, s\}$ (čia aibė $\mathbb{I}_{eq}$ yra apibrėžta lygybe (22)). Taigi aibė $\mathbb{Y}'(v)$ nėra mažesnė už aibę $\bigcup_{z \in \mathbb{Z}(v)} \mathbb{Y}'(z)$, t. y. $\bigcup_{z \in \mathbb{Z}(v)} \mathbb{Y}'(z) \subseteq \mathbb{Y}'(v)$ su visais $v \in \mathbb{V}$. Vadinasi, $g(y^{**}) \leq g(y^*)$, kai

$$y^{**} \in \operatorname{argmin}\{g(y) : y \in \mathbb{Y}'(v)\}.$$

Tikslo funkcijos reikšmė taške y^{**}, t. y. reikšmė $g(y^{**})$, naudojama kaip apatinis mažiausios tikslo funkcijos g reikšmės leistinajame poaibyje $\mathbb{Z}(v)$, $v \in \mathbb{V}$, rėžis.

Atkreipkime dėmesį, kad neprivalome spręsti apatinio lygmens uždavinių (25) su kiekvienu $z \in \mathbb{Z}$. Kai kuriuos $z \in \mathbb{Z}$ galima pašalinti. Apibrėžkime tris aibės $\mathbb{Z}$ elementų šalinimo taisykles.

Taisyklė I. Pakanka nagrinėti tokius $z \in \mathbb{Z}$, kurie priklauso poaibiui $\mathbb{Z}(v)$, $v \in \mathbb{V}(m)$. Ši taisyklė pagrįsta tuo, kad funkcija f_1 yra invariantinė atspindėjimo atžvilgiu.

Taisyklė II. Tegu $\mu : \mathbb{N}^3 \to -\mathbb{N} \cup \mathbb{N}$ yra tokia funkcija, kad:

$$\mu(k, i, j) = 2(k + m(j - i + (2n - i)(i - 1)/2 - 1)) - 1. \tag{27}$$

Atkreipkime dėmesį, kad

$$\begin{array}{ll}
\mu(1,1,2) = 1, & \ldots,\ \mu(m,1,2) = 2m - 1, \\
\mu(1,1,3) = 2(m+1) - 1, & \ldots,\ \mu(m,1,3) = 2(2m) - 1, \\
 & \ldots \\
\mu(1,n-1,n) = 2(s - (m-1)) - 1, & \ldots,\ \mu(m,n-1,n) = 2s - 1.
\end{array}$$

Tada pakanka nagrinėti tokius $z \in \mathbb{Z}$, kurie tenkina šias sąlygas:

$$(z_{\mu(k,i,l)}, z_{\mu(k,l,j)}, z_{\mu(k,i,j)}) \in \begin{array}{l} \{(0,0,0),(0,1,0),(0,1,1), \\ (1,0,0),(1,0,1),(1,1,1)\} \end{array} \tag{28}$$

su visais $k \in \{1,\dots,m\}$, i, l, $j \in \{1,\dots,n\}$ $(i < l < j)$.

Taisyklė III. Prisiminkime, kad $N = n(n-1)/2$. Kiekvienam $z \in \mathbb{Z}$ apibrėžkime tokį vektorių $w(z) = (w_1(z),\dots,w_m(z))^T \in \{0,1,\dots,2^N-1\}^m$, kad

$$w_k(z) = \sum_{i=1}^{N} 2^{N-i} z_{2(k+(i-1)m)-1},$$

$1 \le k \le m$. Vektorius $w(z)$ yra unikalus kiekvienam $z \in \mathbb{Z}$, t. y., jeigu z^1, $z^2 \in \mathbb{Z}$ $(z^1 \neq z^2)$, tai $w(z^1) \neq w(z^2)$. Taigi pakanka nagrinėti tokius $z \in \mathbb{Z}$, kurie tenkina šias nelygybes:

$$w_1(z) \ge \ldots \ge w_m(z). \tag{29}$$

Ši taisyklė yra pagrįsta tuo, kad funkcija f_1 yra invariantinė sukimo atžvilgiu.

2.4.3. Algoritmas

Algoritmas, grąžinantis funkcijos g globalaus minimumo tašką aibėje $\mathbb{Y}' = \bigcup_{z\in\mathbb{Z}} \mathbb{Y}'(z)$, aprašytas algoritme 2. Šis algoritmas yra pagrįstas šakų ir rėžių metodu, tad pavadinome jį „Branch-and-Bound“ (BB) algoritmu.

2.5. Algoritmas, pagrįstas lygiagrečiuoju šakų ir rėžių metodu

Šiame skyrelyje pristatytas trečiasis algoritmas uždaviniui (19) spręsti. Šis algoritmas yra lygiagrečioji algoritmo BB (žr. algoritmas 2) versija ir turi 1-ąjį lygiagretumo laipsnį (žr. skyrelį 1.1.6). Algoritmas grąžina funkcijos g globalaus minimumo tašką aibėje $\mathbb{Y}'$.

Algoritmas 2 Branch-and-Bound (BB)

Duota: $m \in \{1,2,3\}$, $n \in \mathbb{N}$ $(n > 1)$, $\delta_{ij} \in \mathbb{R}$ $(\delta_{ij} = \delta_{ji} > 0,\ \delta_{ii} = 0)$, $1 \le i,j \le n$
Rasti: y^* – funkcijos g minimumo taškas aibėje $\mathbb{Y}'$

$v \in \mathbb{V}(m)$
$V \leftarrow \{\mathbb{Z}(v)\}$
$g^* \leftarrow +\infty$
while $V \neq \emptyset$ **do**
    $\mathbb{Z}(\nu) \in V$; $V \leftarrow V \setminus \{\mathbb{Z}(\nu)\}$
    $i^* \leftarrow \sum_{i=1}^{s} (\nu_{2i-1} + \nu_{2i})$
    for $i = 1, 2$ **do**
        $\nu^i \leftarrow \nu$; $\nu^i_{(2i^*+i)} \leftarrow 0$
        Iš aibės $\mathbb{Z}(\nu^i)$ pašaliname visu nereikalingus elementus
        if $\mathbb{Z}(\nu^i) \neq \emptyset$ **then**
            $y^{**} \in \operatorname{argmin}\{g(y) : y \in \mathbb{Y}'(\nu^i)\}$
            if $g(y^{**}) < g^*$ **then**
                if $y^{**} \in \mathbb{Y}'$ **then**
                    $y^* \leftarrow y^{**}$; $g^* \leftarrow g(y^{**})$
                else
                    $V \leftarrow V \cup \{\mathbb{Z}(\nu^i)\}$

Tarkime, kad koks nors kompiuteris gali atlikti $P = 2^l + 1$ ($l = 0, 1, 2, \ldots$) procesų tuo pačiu metu, t. y. turi P skaičiavimo vienetų (branduolių). Šiame kompiuteryje vykdant pateiktąjį lygiagretųjį algoritmą, iš pradžių aibė $\mathbb{Z}(v)$, $v \in \mathbb{V}(m)$, padalijama į $P - 1$ nesikertančių poaibių $\mathbb{Z}(v^i)$, kai $v^i \in \mathbb{V}(m + l)$, $1 \leq i \leq P - 1$, ir $v_j^1 = \ldots = v_j^{P-1}$, $1 \leq j \leq 2m$. Paskui, i-asis procesas taikydamas algoritmą 2 randa funkcijos g globalaus minimumo tašką aibėje $\bigcup_{z \in \mathbb{Z}(v^i)} \mathbb{Y}'(z)$ ir rastą tašką persiunčia tam tikram procesui, kuris vadinamas procesu-šeimininku, $1 \leq i \leq P - 1$. Galiausiai procesas-šeimininkas palygina gautuosius $P - 1$ minimumo taškus ir grąžina geriausią iš jų. Taigi toks procesų skaičius užtikrina vienodą aibės $\mathbb{Z}(v)$, $v \in \mathbb{V}(m)$, padalijimą ($|\mathbb{Z}(v^i)| = 2^{s-m-l}$ su visais $1 \leq i \leq P - 1$) ir leidžia realizuoti šeimininko-darbininko komunikacijos modelį tarp procesų [**37**].

Lygiagretusis algoritmas aprašytas algoritme 3. Jis pagrįstas lygiagrečiuoju šakų ir rėžių metodu, tad pavadinome jį „Parallel Branch-and-Bound“ (PBB) algoritmu.

3. Skaitinis algoritmų tyrimas

Siekdami atskleisti pagrindines siūlomų algoritmų ypatybes, juos realizavome ir atlikome tam tikrus skaitinius tyrimus. Visi algoritmai realizuoti Fortran 95 programavimo kalba, o skaitiniai tyrimai atlikti su tam tikromis skirtingumo matricomis. Šiame skyriuje pristatyti skaitinių tyrimų rezultatai.

3.1. Algoritmas MAS

Suformuluokime algoritmo MAS (žr. algoritmas 1) skaitinio tyrimo tikslą. Tarkime, kad $m \in \{1, 2, 3\}$, $n \in \mathbb{N}$ ($n > 1$) ir $\Delta^n = (\delta_{ij}) \in \mathbb{R}^{n \times n}$ ($\delta_{ij} = \delta_{ji} > 0$, $\delta_{ii} = 0$, $1 \leq i, j \leq n$). Algoritmo MAS skaitinio tyrimo rezultatai palyginti su atitinkamais algoritmo SMOOTH rezultatais (žr. skyrelį 1.2). Tad kad būtų paprasčiau, algoritmus SMOOTH ir MAS toliau žymėkime simboliais $\mathcal{S}$ ir $\mathcal{M}$ atitinkamai. Prisiminkime, kad kiekvienas funkcijos f_1 lokalaus arba globalaus minimumo taškas aibėje $\mathbb{R}^{mn}$ apibrėžia tam tikrą n daugiamačių elementų vaizdą m-atėje Dekarto koordinačių sistemoje (žr. skyrelį 1.2). Tegu $E^*_{\mathcal{A}}$ žymi vaizdo, apibrėžto funkcijos f_1 minimumo taške x^* ir gauto taikant algoritmą $\mathcal{A} \in \{\mathcal{S}, \mathcal{M}\}$, tokią santykinę paklaidą [**16**]:

$$E^*_{\mathcal{A}} = \sqrt{\frac{f_1(x^*)}{\sum_{i<j}^{n} \delta_{ij}^2}}. \tag{30}$$

Kuo paklaida $E^*_{\mathcal{A}}$ artimesnė skaičiui 0, tuo vaizdas yra geresnis. Tegu

$$E_{\mathcal{A}} = \{E^{*\,1}_{\mathcal{A}}, \ldots, E^{*\,30}_{\mathcal{A}}\}$$

žymi multiaibę 30-ies santykinių paklaidų, apskaičiuotų vaizduose, gautuose taikant algoritmą $\mathcal{A} \in \{\mathcal{S}, \mathcal{M}\}$. Tarkime, kad $E^{*\,i}_{\mathcal{S}}$ – mažiausia santykinė paklaida, gauta algoritmą $\mathcal{S}$ vykdant 10 kartų, t. y.

$$E^{*\,i}_{\mathcal{S}} = \min\{E^{*\,1i}_{\mathcal{S}}, \ldots, E^{*\,10i}_{\mathcal{S}}\},$$

Algoritmas 3 Parallel Branch-and-Bound (PBB)

Duota: $m \in \{1,2,3\}$, $n \in \mathbb{N}$ $(n > 1)$, $\delta_{ij} \in \mathbb{R}$ $(\delta_{ij} = \delta_{ji} > 0$, $\delta_{ii} = 0)$, $1 \le i,j \le n$
Rasti: y^* – funkcijos g minimumo taškas aibėje $\mathbb{Y}'$

GAUTI P, R
$g^* \leftarrow +\infty$
if $R = 0$ **then**
$W \leftarrow 1 \in \{0,1\}^{P-1}$
while $\sum_{i=1}^{P-1} W_i > 0$ **do**
if YRA PRANEŠIMAS IŠ PROCESO i **then**
GAUTI $[Y^*, N]$
if $N = 0$ **then**
$W_i \leftarrow 0$
if $g(Y^*) < g^*$ **then**
$y^* \leftarrow Y^*$; $g^* \leftarrow g(Y^*)$
for $j = 1, P-1$ **do**
if $W_j > 0$ **then**
SIŲSTI $[y^*]$ PROCESUI j
else
$v \in \mathbb{V}(m)$; $l \leftarrow \log(P-1)$; $j \leftarrow R - 1$
for $i = m+1, m+l$ **do**
if $\mathrm{mod}(j,2) = 1$ **then**
$v_{2i-1} \leftarrow 0$
else
$v_{2i} \leftarrow 0$
$j \leftarrow \mathrm{div}(i,2)$
$V \leftarrow \{\mathbb{Z}(v)\}$
while $V \ne \emptyset$ **do**
$\mathbb{Z}(\nu) \in V$; $V \leftarrow V \setminus \{\mathbb{Z}(\nu)\}$; $i^* \leftarrow \sum_{i=1}^{s} (\nu_{2i-1} + \nu_{2i})$
for $i = 1, 2$ **do**
$\nu^i \leftarrow \nu$; $\nu^i_{(2i^*+i)} \leftarrow 0$
Iš aibės $\mathbb{Z}(\nu^i)$ pašalinti visus nereikalingus elementus
if $\mathbb{Z}(\nu^i) \ne \emptyset$ **then**
$y^{**} \in \mathrm{argmin}\{g(y) : y \in \mathbb{Y}'(\nu^i)\}$
if YRA PRANEŠIMAS IŠ PROCESO 0 **then**
GAUTI $[y^*]$
if $g(y^*) < g(y^{**})$ **then**
$Y^* \leftarrow y^*$; $g^* \leftarrow g(y^*)$
if $g(y^{**}) < g^*$ **then**
if $y^{**} \in \mathbb{Y}'$ **then**
$Y^* \leftarrow y^{**}$; $g^* \leftarrow g(y^{**})$
$N \leftarrow 1$; SIŲSTI $[Y^*, N]$ PROCESUI 0
else
$V \leftarrow V \cup \{\mathbb{Z}(\nu^i)\}$
$N \leftarrow 0$; SIŲSTI $[Y^*, N]$ PROCESUI 0

$1 \le i \le 30$. Tegu $t^i_{\mathcal{S}}$ yra laikas (sekundėmis), reikalingas rasti i-ajam multiaibės $E_{\mathcal{S}}$ elementui. Tarkime, kad ${E^*_{\mathcal{M}}}^i$ yra mažiausia santykinė paklaida, gauta algoritmą

$\mathcal{M}$ vykdant mažiausiai $t_{\mathcal{S}}^i$ sekundžių, $1 \le i \le 30$. Tegu $t_{\mathcal{M}}^i$ yra laikas (sekundėmis) ir $K_{\mathcal{M}}^i$ yra funkcijos f_1 lokaliųjų minimumo taškų aibėje $\mathbb{R}^{mn}$ skaičius, reikalingas i-ajam multiaibės $E_{\mathcal{M}}$ elementui rasti. Taigi

$$E_{\mathcal{M}}^{*\,i} = \min\{E_{\mathcal{M}}^{*\,1i}, \ldots, E_{\mathcal{M}}^{*\,K_{\mathcal{M}}^i i}\} \text{ ir } t_{\mathcal{M}}^i = t_{\mathcal{S}}^i + \epsilon,$$

kai $\epsilon \ge 0$, $1 \le i \le 30$. Atkreipkime dėmesį, kad $K_{\mathcal{S}}^i = 10$, $1 \le i \le 30$. Pagrindinis skaitinio tyrimo tikslas buvo palyginti kai kurias multiaibių $E_{\mathcal{S}}$ ir $E_{\mathcal{M}}$ ypatybes, kai taikomos įvairios skirtingumo matricos ir skirtingos m reikšmės. Ypač domino šios ypatybės: mažiausia reikšmė ($\min E_{\mathcal{A}}$), didžiausia reikšmė ($\max E_{\mathcal{A}}$), aritmetinis vidurkis ($\operatorname{mean} E_{\mathcal{A}}$) ir standartinis nuokrypis ($\operatorname{std} E_{\mathcal{A}}$), kai $\mathcal{A} \in \{\mathcal{S}, \mathcal{M}\}$.

Skaitinis tyrimas atliktas kompiuteriu su 1) Intel(R) Core(TM)2 Duo procesoriumi, jo taktinis dažnis – 2.40 GHz, ir 2) Xubuntu 14.04 (64-bit, branduolio versija: 3.13.0-24-generic) operacine sistema. Kiekvienas multiaibės $E_{\mathcal{S}}$ elementas gautas su programa „SMOOTH“ [**22**] (programos versija: 0.4, autorius: Patrick Groenen, 2003). Skaitinio tyrimo rezultatai pateikiami lentelėje 1.

Lentelėje 1 simbolis $t_{\mathcal{A}}$ žymi vidutinį laiką (sekundėmis), o $K_{\mathcal{A}}$ – vidutinį funkcijos f_1 lokalių minimumo taškų skaičių aibėje $\mathbb{R}^{mn}$, reikalingą vienam multiaibės $E_{\mathcal{A}}$ elementui rasti, t. y. $t_{\mathcal{A}} = \sum_{i=1}^{30} t_{\mathcal{A}}^i/30$ ir $K_{\mathcal{A}} = \sum_{i=1}^{30} K_{\mathcal{A}}^i/30$, $\mathcal{A} \in \{\mathcal{S}, \mathcal{M}\}$. Atkreipkime dėmesį į multiaibių $E_{\mathcal{S}}$ ir $E_{\mathcal{M}}$ standartinių nuokrypių reikšmes (lentelės 1 stulpelis nr. 6). Jos yra beveik vienodos, kai pasirinktos skirtingumo matricos Δ_{cube}, Δ_{regs} ir Δ_{simp}. Tačiau $\operatorname{std} E_{\mathcal{S}}$ ir $\operatorname{std} E_{\mathcal{M}}$ reikšmės yra skirtingos su kitomis skirtingumo matricomis. Tai rodo, kad algoritmas $\mathcal{M}$ yra jautresnis duomenims parinkti nei algoritmas $\mathcal{S}$.

Didelis skirtumas tarp reikšmių $\min E_{\mathcal{A}}$ ir $\max E_{\mathcal{A}}$ rodo, kad algoritmo $\mathcal{A}$ jautrumas pradiniam taškui parinkti yra didelis. Remiantis lentelės 1 duomenimis, galima daryti išvadą, kad algoritmas $\mathcal{M}$ yra jautresnis pradiniam taškui parinkti nei algoritmas $\mathcal{S}$.

Atkreipkime dėmesį, kad $K_{\mathcal{M}} > K_{\mathcal{S}}$ beveik visais atvejais, išskyrus atvejį $\Delta_{\text{ruusk}}^{20}$. Tai rodo, kad algoritmas $\mathcal{M}$ per apytiksliai tą patį laiką sugeneruoja daugiau funkcijos f_1 lokalių minimumo taškų nei algoritmas $\mathcal{S}$.

3.2. Algoritmas BB

Algoritmo BB (žr. algoritmas 2) skaitinio tyrimo rezultatai palyginti su atitinkamais algoritmo BB2009 rezultatais (žr. skyrelį 1.2). Kad būtų papraščiau, algoritmus BB2009 ir BB pažymėkime simboliais $\mathcal{B}_{09}$ ir $\mathcal{B}_{14}$ atitinkamai. Vienas iš daugiausia laiko užimančių algoritmo $\mathcal{B}_{14}$ žingsnių yra tam tikro iškilojo kvadratinio programavimo uždavinio sprendimas (žr. algoritmo 2 eilutę nr. 11). Panašus žingsnis yra atliekamas ir su algoritmu $\mathcal{B}_{09}$. Tarkime, kad algoritmas $\mathcal{A} \in \{\mathcal{B}_{09}, \mathcal{B}_{14}\}$ šį daugiausia laiko užimantį žingsnį atlieka $K_{\mathcal{A}}$ kartų. Pagrindinis skaitinio tyrimo tikslas buvo palyginti skaičių $K_{\mathcal{B}_{09}}$ ir $K_{\mathcal{B}_{14}}$ reikšmes, kai taikomos įvairios skirtingumo matricos ir skirtingos m reikšmės.

Skaitinis tyrimas atliktas kompiuteriu su 1) Intel(R) Pentium(R) 4 CPU 3.00GHz procesoriumi, jo taktinis dažnis – 2.80 GHz, ir 2) CentOS 6.5 (64-bit) operacine sistema. Tyrimo rezultatai pateikiami lentelėje 2. Simbolis E^* žymi vaizdo, apibrėžto funkcijos f_1 minimumo taške x^*, santykinę paklaidą, kai taikomi algoritmai $\mathcal{B}_{14}$ ir

1 LENTELĖ. Algoritmo $\mathcal{M}$ skaitinio tyrimo rezultatai ir atitinkami rezultatai, gauti taikant algoritmą $\mathcal{S}$.

Δ^n	m	$\min E_{\mathcal{A}}$	$\max E_{\mathcal{A}}$	$\mathrm{mean}\, E_{\mathcal{A}}$	$\mathrm{std}\, E_{\mathcal{A}}$	$t_{\mathcal{A}}$	$K_{\mathcal{A}}$	$\mathcal{A}$
Δ^4_{cube}	3	0.0001	0.0012	0.0009	0.0002	0.519	10	$\mathcal{S}$
		0.0000	0.0000	0.0000	0.0000	0.519	975	$\mathcal{M}$
Δ^8_{cube}	3	0.0012	0.0013	0.0013	0.0000	1.449	10	$\mathcal{S}$
		0.0000	0.0000	0.0000	0.0000	1.463	62	$\mathcal{M}$
Δ^7_{regs}	3	0.0945	0.0945	0.0945	0.0000	1.566	10	$\mathcal{S}$
		0.0945	0.0945	0.0945	0.0000	1.571	134	$\mathcal{M}$
Δ^9_{regs}	2	0.2991	0.2991	0.2991	0.0000	1.382	10	$\mathcal{S}$
		0.2991	0.3031	0.2996	0.0011	1.387	184	$\mathcal{M}$
$\Delta^{13}_{\text{regs}}$	1	0.5311	0.5311	0.5311	0.0000	0.857	10	$\mathcal{S}$
		0.5311	0.5311	0.5311	0.0000	0.858	379	$\mathcal{M}$
Δ^7_{simp}	3	0.0015	0.0016	0.0016	0.0000	1.673	10	$\mathcal{S}$
		0.0000	0.0000	0.0000	0.0000	1.679	150	$\mathcal{M}$
Δ^9_{simp}	2	0.2759	0.2759	0.2759	0.0000	0.972	10	$\mathcal{S}$
		0.2759	0.2808	0.2760	0.0009	0.977	88	$\mathcal{M}$
$\Delta^{13}_{\text{simp}}$	1	0.5279	0.5281	0.5279	0.0000	1.203	10	$\mathcal{S}$
		0.5279	0.5279	0.5279	0.0000	1.205	314	$\mathcal{M}$
Δ^9_{hwa}	1	0.0109	0.0109	0.0109	0.0000	0.074	10	$\mathcal{S}$
		0.0107	0.0107	0.0107	0.0000	0.075	55	$\mathcal{M}$
	2	0.0108	0.0110	0.0110	0.0001	0.655	10	$\mathcal{S}$
		0.0000	0.0027	0.0001	0.0005	0.713	40	$\mathcal{M}$
Δ^{12}_{hwa}	1	0.1790	0.1790	0.1790	0.0000	0.211	10	$\mathcal{S}$
		0.1790	0.1871	0.1821	0.0023	0.214	48	$\mathcal{M}$
Δ^8_{ruusk}	1	0.2975	0.2975	0.2975	0.0000	0.201	10	$\mathcal{S}$
		0.2975	0.3292	0.3112	0.0087	0.201	480	$\mathcal{M}$
	2	0.1096	0.1096	0.1096	0.0000	1.130	10	$\mathcal{S}$
		0.1097	0.1306	0.1198	0.0055	1.133	172	$\mathcal{M}$
	3	0.0189	0.0254	0.0214	0.0018	2.386	10	$\mathcal{S}$
		0.0188	0.0411	0.0289	0.0066	2.402	86	$\mathcal{M}$
$\Delta^{20}_{\text{ruusk}}$	2	0.0524	0.0555	0.0546	0.0010	4.074	10	$\mathcal{S}$
		0.0523	0.1322	0.0850	0.0232	4.962	3	$\mathcal{M}$
$\Delta^{12}_{\text{uhlen}}$	1	0.2112	0.2112	0.2112	0.0000	0.199	10	$\mathcal{S}$
		0.2112	0.2251	0.2151	0.0036	0.201	52	$\mathcal{M}$
	2	0.0825	0.0909	0.0874	0.0023	2.407	10	$\mathcal{S}$
		0.0840	0.1248	0.1033	0.0105	2.440	37	$\mathcal{M}$
$\Delta^{10}_{\text{cola}}$	1	0.3645	0.3645	0.3645	0.0000	0.318	10	$\mathcal{S}$
		0.3656	0.3959	0.3762	0.0075	0.319	414	$\mathcal{M}$
	2	0.1679	0.1694	0.1694	0.0003	1.692	10	$\mathcal{S}$
		0.1729	0.2089	0.1837	0.0078	1.702	100	$\mathcal{M}$

$\mathcal{B}_{09}$ (žr. (30)). Algoritmai $\mathcal{B}_{14}$ ir $\mathcal{B}_{09}$ grąžina globalųjį funkcijos f_1 minimumo tašką aibėje $\mathbb{R}^{mn}$, todėl santykinės paklaidos šiame taške sutampa. Atkreipkime dėmesį, kad beveik visais atvejais, kai $m = 1$, skaičių $K_{\mathcal{B}_{14}}$ reikšmės yra didesnės negu skaičių

2 LENTELĖ. Algoritmo $\mathcal{B}_{14}$ skaitinio tyrimo rezultatai ir atitinkami rezultatai, gauti taikant algoritmą $\mathcal{B}_{09}$.

Δ^n	m	E^*	$\mathcal{B}_{09}$	$\mathcal{B}_{14}$
Δ^4_{cube}	1	0.4082	14	18
	2	0.0000	73	12
	3	0.0000	353	6
Δ^8_{cube}	1	0.4787	11,260	10,948
	2	0.2245	2,157,090	205,032
	3	0.0000	35,216,122	355,611
Δ^4_{regs}	1	0.4082	14	22
	2	0.0000	63	32
	3	0.0000	133	38
Δ^5_{regs}	1	0.4472	73	142
	2	0.1907	1,322	800
	3	0.0000	23,017	256
Δ^6_{regs}	1	0.4714	432	868
	2	0.2309	27,255	21,393
	3	0.0000	335,771	55,606
Δ^7_{regs}	1	0.4880	2,951	6,478
	2	0.2621	1,655,631	1,020,040
	3	0.0945	92,710,201	20,115,704
Δ^4_{simp}	1	0.3651	14	21
	2	0.0000	73	13
	3	0.0000	313	12
Δ^5_{simp}	1	0.4140	73	112
	2	0.0000	662	66
	3	0.0000	9,837	39
Δ^6_{simp}	1	0.4554	432	702
	2	0.1869	16,076	15,632
	3	0.0000	578,691	1,185
Δ^7_{simp}	1	0.4745	2,951	4,890
	2	0.2247	422,940	585,436
	3	0.0000	20,674,563	168,547
Δ^9_{hwa}	1	0.0107	2,217	1,591
	2	0.0000	2,344,833	151,835
Δ^{12}_{hwa}	1	0.1790	71,748	32,629
Δ^8_{ruusk}	1	0.2975	665	1,590
	2	0.1096	82,617	374,190
$\Delta^{12}_{\text{uhlen}}$	1	0.2112	36,559	236,168
$\Delta^{10}_{\text{cola}}$	1	0.3642	60,077	63,744

$K_{\mathcal{B}_{09}}$ (išskyrus šiuos atvejus: Δ^8_{cube}, Δ^9_{hwa} ir Δ^{12}_{hwa}). Kai $m = 2$ ir $m = 3$, beveik visos skaičių $K_{\mathcal{B}_{14}}$ reikšmės yra mažesnės negu skaičių $K_{\mathcal{B}_{14}}$ (išskyrus šiuos atvejus: Δ^7_{simp}, $m = 2$, ir Δ^8_{ruusk}, $m = 2$). Viena iš pagrindinių priežasčių, kodėl esti toks skirtumas tarp reikšmių $K_{\mathcal{B}_{09}}$ ir $K_{\mathcal{B}_{14}}$, – skirtingos paieškos medžių struktūros. Algoritmas $\mathcal{B}_{14}$

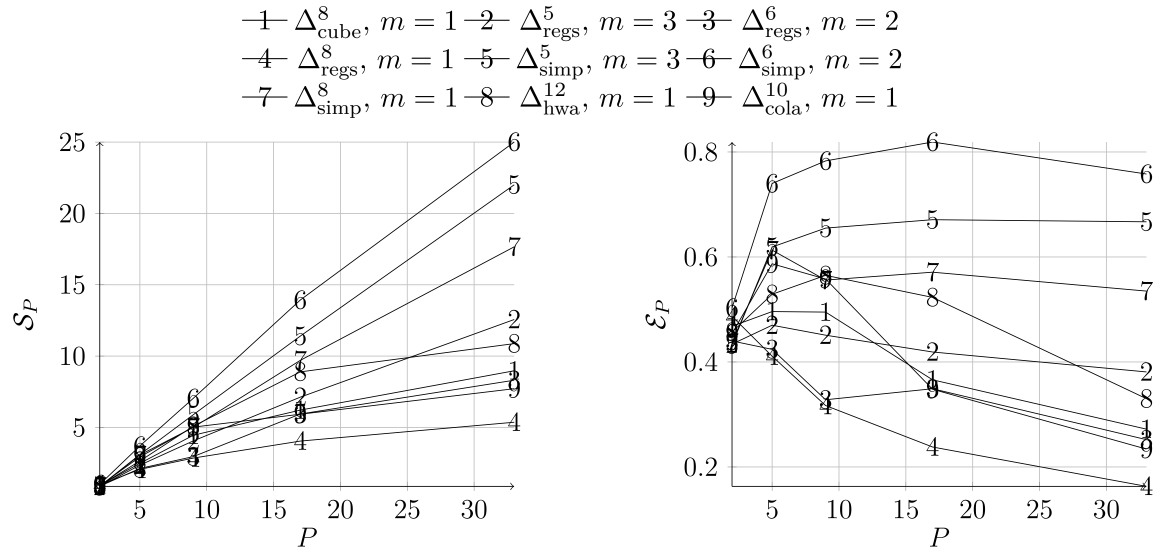
viršutinio lygmens leistinąją aibę $\mathbb{Z}$ (tiksliau, jos poaibį $\mathbb{Z}(v) \subset \mathbb{Z}$, $v \in \mathbb{V}(m)$) dalija į du nesikertančius poaibius. O algoritmas $\mathcal{B}_{09}$ atitinkamą viršutinio lygmens leistinąją aibę gali padalyti ir į daugiau kaip du poaibius.

3.3. Algoritmas PBB

Algoritmas PBB yra lygiagretusis algoritmas (žr. algoritmas 3). Pagrindinis skaitinio tyrimo tikslas buvo įvertinti lygiagretaus algoritmo PBB spartinimą $\mathcal{S}_P$ ir efektyvumą $\mathcal{E}_P$, kai taikomos įvairios skirtingumo matricos ir skirtingos m ir P (skaičiavimo vienetų) reikšmės.

Skaitinis tyrimas atliktas kompiuterių klasteryje. Kiekvienas klasterio kompiuteris turėjo 4 branduolių Intel(R) Core(TM) i5-760 procesorių su 2.8 GHz taktiniu dažniu. Klasteryje įdiegta CentOS 6.3 (32-bit) operacinė sistema. Lygiagretusis algoritmas buvo vykdomas naudojant 2, 5, 9, 17 ir 33 skaičiavimo vienetus (branduolius). Algoritmo PBB spartinimas ir efektyvumas vaizduojami pav. 3.

Geriausios spartinimo $\mathcal{S}_P$ reikšmės yra įgyjamos su skirtingumo matricomis Δ_{simp}, o prasčiausios – su $\Delta^{8}_{\mathrm{regs}}$ ir $m = 1$. Deja, visais atvejais tiesinis spartinimas nėra pasiekiamas. Geriausios efektyvumo $\mathcal{E}_P$ reikšmės yra įgyjamos taip pat su skirtingumo matricomis Δ_{simp}, o praščiausios – vėlgi su $\Delta^{8}_{\mathrm{regs}}$ ir $m = 1$. Visais atvejais efektyvumas mažėja didėjant skaičiavimo vienetų skaičiui. Šie rezultatai rodo, kad lygiagretaus algoritmo PBB spartinimas ir efektyvumas priklauso nuo pradinių duomenų. Be to, statinis darbų paskirstymas procesams-darbininkams yra viena iš pagrindinių priežasčių, kodėl nepasiektas tiesinis pagreitėjimas ir mažėja efektyvumas. Iš tikrųjų, procesas-darbininkas atlikęs savo darbą, nebegauna naujų užduočių.



(A) Spartinimas.

(B) Efektyvumas.

3 Pav. Lygiagretaus algoritmo PBB spartinimas ir efektyvumas.

4. Išvados

(1) Optimizavimo uždavinio, kylančio daugiamatėse skalėse (DS) su miesto kvartalo atstumais, tikslo funkcija yra funkcija su modulio (absoliučiojo dydžio) elementais. Tokia funkcija gali būti nediferencijuojama kuriame nors iš minimumo taškų. O siūlomo uždavinio PDS tikslo funkcija yra iškiloji kvadratinė funkcija, neturinti modulio elementų.

(2) Optimizavimo uždavinio, kylančio DS su miesto kvartalo atstumais, tikslo funkcija turi mn kintamųjų. O uždavinio PDS tikslo funkcija – mn^2 kintamųjų. Taigi PDS uždavinys turi n kartų daugiau kintamųjų negu optimizavimo uždavinys, kylantis DS su miesto kvartalo atstumais.

(3) Uždavinio PDS leistinoji aibė apibrėžta tiek tiesiniais, tiek papildomumo apribojimais. Papildomumo apribojimai yra neiškilieji kvadratiniai apribojimai. O uždavinio PDS tikslo funkcija – iškiloji kvadratinė. Vadinasi, uždavinys PDS yra iškilojo kvadratinio programavimo uždavinys su neiškilaisiais kvadratiniais apribojimais. Šis uždavinys priklauso NP-hard sudėtingumo klasei.

(4) Uždavinys PDS gali būti suformuluotas kaip dviejų lygmenų optimizavimo uždavinys. Pirmieji $m+s$ apatinio lygmens leistinieji apribojimai nepriklauso nuo viršutinio lygmens kintamojo. Taip pat apatinio lygmens tikslo funkcija nepriklauso nuo viršutinio lygmens kintamojo. Taigi sprendžiant apatinio lygmens problemas dalis skaičiavimų gali būti atlikti iš anksto.

(5) Optimizavimo uždavinys, kylantis daugiamatėse skalėse (DS) su miesto kvartalo atstumais, gali būti suformuluotas kaip dviejų lygmenų optimizavimo uždavinys. Uždavinio PDS ir pradinio uždavinio dviejų lygmenų formuluotės yra skirtingos. Tiksliau tariant, uždavinio PDS dviejų lygmenų formuluotėje yra $2^{mn(n-1)/2}$ apatinio lygmens uždavinių. O štai pradinio uždavinio dviejų lygmenų formuluotėje – $(n!)^m$ apatinio lygmens uždavinių.

(6) Pasiūlytas algoritmas, pagrįstas aktyviosios aibės metodu (algoritmas MAS), yra jautresnis pradiniam taškui parinkti negu algoritmas SMOOTH, šiuo metu pats geriausias lokalios paieškos algoritmas DS su miesto kvartalo atstumais. Tačiau algoritmas MAS yra žymiai greitesnis: per apytiksliai tą patį laiką MAS sugeneruoja iki 97 kartų daugiau lokaliųjų minimumo taškų negu algoritmas SMOOTH.

(7) Pasiūlytas algoritmas, pagrįstas šakų ir rėžių metodu (algoritmas BB), žymiai sumažina (iki 9×10^{18} kartų) spręstinų apatinio lygmens uždavinių skaičių. Be to, šis algoritmas išsprendžia iki 488 kartų mažiau apatinio lygmens uždavinių nei literatūroje aprašytas panašus algoritmas.

(8) Pasiūlytas lygiagretusis algoritmas, pagrįstas lygiagrečiuoju šakų ir rėžių metodu (algoritmas PBB), yra iki 25 kartų greitesnis, kai naudojami 33 skaičiavimo vienetai, negu jo nuoseklioji versija. Vidutinis lygiagretumo efektyvumas yra apie 0,4.

Literatūra

[1] Mordecai Avriel. *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

[2] David A. Bader, William E. Hart, and Cynthia A. Phillips. Parallel algorithm design for branch and bound. In *Tutorials on Emerging Methodologies and Applications in Operations Research*, pages 5–1. Springer, 2005.

[3] Ingwer Borg and Patrick J. F. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

[4] Jonathan M. Borwein and Adrian S. Lewis. *Convex analysis and nonlinear optimization: theory and examples*, volume 3. Springer Science & Business Media, 2010.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] Gilles Brassard and Paul Bratley. *Fundamentals of algorithmics*. Prentice-Hall, Inc., 1996.

[7] Michael J. Brusco and Stephanie Stahl. *Branch-and-bound applications in combinatorial data analysis*. Springer, 2006.

[8] Edwin K. P. Chong and Stanislaw H. Zak. *An introduction to optimization*. John Wiley & Sons, 2013.

[9] Jens Clausen. Branch and bound algorithms – principles and examples. *Department of Computer Science, University of Copenhagen*, pages 1–30, 1999.

[10] Benoít Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

[11] Michael A. A. Cox and Trevor F. Cox. Multidimensional scaling. In *Handbook of Data Visualization*, Springer Handbooks of Computational Statistics, pages 315–347. Springer Berlin Heidelberg, 2008.

[12] Teodor Gabriel Crainic, Bertrand Le Cun, and Catherine Roucairol. Chapter 1. parallel branch-and-bound algorithms. *Parallel combinatorial optimization*, 1:1–28, 2006.

[13] Etienne De Klerk. *Aspects of semidefinite programming: interior point algorithms and selected applications*. Springer Science & Business Media, 2002.

[14] Caterina De Simone. The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79(1):71–75, 1990.

[15] Stephan Dempe, Vyatcheslav V. Kalashnikov, and Nataliya Kalashnykova. Optimality conditions for bilevel programming problems. In *Optimization with Multivalued Mappings*, pages 3–28. Springer, 2006.

[16] Gintautas Dzemyda, Olga Kurasova, and Julius Žilinskas. *Multidimensional Data Visualization: Methods and Applications*, volume 75 of *Springer Optimization and Its Applications*. Springer, 2012.

[17] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[18] Masao Fukushima. Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical programming*, 53(1-3):99–110, 1992.

[19] Bernard Gendron and Teodor Gabriel Crainic. Parallel branch-and-branch algorithms: Survey and synthesis. *Operations research*, 42(6):1042–1066, 1994.

[20] Philip E. Gill and Elizabeth Wong. Methods for convex and general quadratic programming. *Mathematical Programming Computation*, pages 1–42, 2014.

[21] Patrick J. F. Groenen, Willem J. Heiser, and Jacqueline J. Meulman. City-block scaling: smoothing strategies for avoiding local minima. In *Classification, Data Analysis, and Data Highways*, pages 46–53. Springer, 1998.

[22] Patrick J. F. Groenen, Willem J. Heiser, and Jacqueline J. Meulman. Global optimization in least-squares multidimensional scaling by distance smoothing. *Journal of classification*, 16(2):225–254, 1999.

[23] Toshihide Ibaraki. Representation theorems for equivalent optimization problems. *Information and Control*, 21(5):397–435, 1972.

[24] Joseph Jada. *An introduction to parallel algorithms.* Addison Wesley, 1992.

[25] Virendra K. Janakiram, Edward F. Gehringer, Dharma P. Agrawal, and Ravi Mehrotra. A randomized parallel branch-and-bound algorithm. *International Journal of Parallel Programming*, 17(3):277–301, 1988.

[26] J. M. Jansen and F. W. Sijstermans. Parallel branch-and-bound algorithms. *Future Generation Computer Systems*, 4(4):271–279, 1989.

[27] David B. Kirk and W. Hwu Wen-mei. *Programming massively parallel processors: a hands-on approach.* Newnes, 2012.

[28] Joseph B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[29] Joseph B. Kruskal and Myron Wish. *Multidimensional scaling.* Sage, 1978.

[30] Gautham K. Kudva and Joseph F. Pekny. A distributed exact algorithm for the multiple resource constrained sequencing problem. *Annals of Operations Research*, 42(1):25–54, 1993.

[31] Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. *Introduction to Parallel Computing: Design and Analysis of Algorithms.* Benjamin/Cummings Publishing Company Redwood City, CA, 1994.

[32] Leon S. Lasdon. *Optimization theory for large systems.* Courier Corporation, 2013.

[33] Donald L. Miller and Joseph F. Pekny. The role of performance metrics for parallel mathematical programming algorithms. *ORSA Journal on Computing*, 5(1):26–28, 1993.

[34] Jorge Nocedal and Stephen J Wright. *Numerical Optimization.* Springer, 2006.

[35] Pablo Pedregal. *Introduction to optimization.* Springer Science & Business Media, 2003.

[36] Joseph F. Pekny and Donald L. Miller. A parallel branch and bound algorithm for solving large asymmetric traveling salesman problems. *Mathematical programming*, 55(1-3):17–33, 1992.

[37] Gary Shao, Francine Berman, and Rich Wolski. Master/slave computing on the grid. In *Heterogeneous Computing Workshop, 2000.(HCW 2000) Proceedings. 9th*, pages 3–16. IEEE, 2000.

[38] Hamdy A. Taha. *Operations research: an introduction.* Pearson/Prentice Hall, 2007.

[39] Fabio Tardella. On the equivalence between some discrete and continuous optimization problems. *Annals of Operations Research*, 25(1):291–300, 1990.

[40] Warren S Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[41] Luis N. Vicente and Paul H. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5(3):291–306, 1994.

[42] Julius Žilinskas. *Black box global optimization: covering methods and their parallelization.* PhD thesis, Kaunas University of Technology, June 2002.

[43] Rui Xu and Don Wunsch. *Clustering*, volume 10. Wiley-IEEE Press, 2008.

[44] Antanas Žilinskas and Julius Žilinskas. Two level minimization in multidimensional scaling. *Journal of Global Optimization*, 38(4):581–596, 2007.

[45] Antanas Žilinskas and Julius Žilinskas. Branch and bound algorithm for multidimensional scaling with city-block metric. *Journal of Global Optimization*, 43(2-3):357–372, 2009.

[46] Antanas Žilinskas and Julius Žilinskas. Optimization-based visualization. In *Encyclopedia of Optimization*, pages 2785–2791. Springer, 2009.

Summary

In this chapter, the research problem and its relevance are presented. The purpose, tasks and novelty of the research are given here, too. Also, we present here propositions for defense. The results of the research were approved in a few scientific conferences and publications. A list of the publications is given here, too. At the end of the chapter, a few facts about the author of the dissertation are presented.

The research problem and its relevance

The research problem is related to the problem of a visualization of multidimensional elements. A multidimensional element is perceived here as a vector in a real vector space of four dimensions or higher. A representation of such vectors in a lower-dimensional – i.e., one-, two-, or three-dimensional – Cartesian coordinate system is perceived here as a visualization of multidimensional elements. The set of points, which represent the multidimensional elements in the lower-dimensional Cartesian coordinate system, is called here the image of multidimensional elements. Note that an image of multidimensional elements is a subset of a lower-dimensional real vector space. Any image of multidimensional elements may give us some important information about them. For example, it may help us to find out if the elements have any regularities, outliers, or clusters [**16**].

There are a number of techniques that can be used to visualize multidimensional elements. One of them is called multidimensional scaling (MDS). MDS is used in areas such as psychometrics, market analysis, pharmacology, and others. In applying MDS, an image of multidimensional elements is obtained by constructing and minimizing a certain function. In order to construct the function, 1) dissimilarities between every pair of multidimensional elements have to be defined, and 2) a desired distance function, defined on a lower-dimensional real vector space, has to be selected. The obtained image has the following feature: the desired distances between points of the image are similar to, or even equal to, the defined dissimilarities.

Images of the same multidimensional elements are usually different if they are obtained by using MDS with different distance functions. Various applications of MDS show that images obtained by using MDS with the city-block distance function (also known as Minkowski distance of order one, L1 distance function) may give us more information about the multidimensional elements than by using other distance functions. However, if city-block distances are selected, the function to be minimized contains a set of absolute value (modulus) terms. In this case, minimization of such a function becomes a complicated optimization problem: the function may have more than one local minimizer and may even be non-differentiable at a minimizer. Thus, the research problem considered here is the minimization problem arising in MDS with city-block distances.

The purpose and tasks of the research

The research problem – i.e., the minimization problem arising in MDS with city-block distances – is equivalent to a certain optimization problem. Let us name the optimization problem, equivalent to the research problem, as problem CMDS. Currently there are no specific algorithms for the problem CMDS. Thus, the main purpose of this research is to offer several algorithms for the problem CMDS. In order to achieve this goal, the following tasks should be performed:

(1) To explore the main features of the objective function of the research problem and to become familiar with algorithms for the research problem.
(2) To construct a few sequential and parallel algorithms for the problem CMDS.
(3) To implement the algorithms and to perform a numerical investigation of them by using particular sets of multidimensional elements.
(4) To compare the results of the investigation with corresponding results obtained by using other algorithms.

The novelty of the research

(1) The equivalence between the research problem and the problem CMDS is proven.
(2) An algorithm is proposed that returns a local minimizer of the objective function of the problem CMDS.
(3) A two-level formulation of the problem CMDS is constructed.
(4) An algorithm is proposed that returns a global minimizer of the objective function of the problem CMDS.
(5) A parallel algorithm is proposed that returns a global minimizer of the objective function of the problem CMDS.

Propositions for defense

(1) The research problem is equivalent to an optimization problem with a convex quadratic objective function, and linear and complementarity (non-linear) constraints.
(2) The algorithm, based on the active-set method, returns a local minimizer of the objective function of the problem CMDS.
(3) The problem CMDS is a two-level optimization problem with a convex quadratic programming problem at the lower-level, and a combinatorial optimization problem at the upper-level.
(4) The algorithm, based on the branch-and-bound method, returns a global minimizer of the objective function of the problem CMDS.

Approbation of results

Results of the research were presented at 9 scientific conferences (4 of them were international conferences) and were published in the following scientific papers:

(1) Roger Fletcher, Nerijus Galiauskas, Julius Žilinskas. Quadratic programming with complementarity constraints for multidimensional scaling with city-block distances. *Baltic Journal of Modern Computing*, 2(4):248-259, 2014. ISSN 2255-8942.

(2) Nerijus Galiauskas, Julius Žilinskas. On multidimensional scaling with city-block distances. In *Lecture Notes in Computer Science*, volume 8426, pages 82-87. Springer, 2014. ISSN 0302-9743.
(3) Nerijus Galiauskas, Julius Žilinskas. Parallel branch and bound for multidimensional scaling with L1 distances formulated as quadratic programming with complementarity constraints. In *3PGCIC 2013: 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Compiégne, France, October 28-30, 2013*, pages 509-512. IEEE, 2013. ISBN 978-0-7695-5094-7.
(4) Nerijus Galiauskas, Julius Žilinskas. Quadratic programming problems. *Journal of Young Scientists*, 33(4):115-118, 2011. ISSN 1648-8776. [In Lithuanian].

About the author

Nerijus Galiauskas was born on February 22, 1985, in Vilnius, Lithuania. In 2004, he graduated from Vilnius Mykolas Birziska Gymnasium. He received his bachelor's and master's degrees in mathematics from Vilnius University in 2008 and 2010, respectively. During the period from 2010 to 2014 he was a PhD student at the same university. The mathematical optimization and parallel computing in optimization are the main topics of his research. At the time of studies, Nerijus was a computer science teacher at Vilnius Taikos Progymnasium (2008–2010) and a junior researcher of two scientific projects at Vilnius University (2012–2014).

Nerijus Galiauskas

OPTIMIZAVIMO ALGORITMAI DAUGIAMATĖMS SKALĖMS SU MIESTO KVARTALO ATSTUMAIS IR JŲ LYGIAGRETINIMAS

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

Redaktorė Jorūnė Rimeisytė

Nerijus Galiauskas

OPTIMIZATION ALGORITHMS FOR MULTIDIMENSIONAL SCALING WITH CITY-BLOCK DISTANCES AND THEIR PARALLELIZATION

Summary of Doctoral Dissertation
Technological Sciences, Informatics Engineering (07 T)

Editor Alex Sullivan