Optimizavimo metodai

Užduoties „Optimizavimas be apribojimų“ ataskaita

Autorius: Vilius Minkevičius, Programų Sistemos 4 kursas 2 grupė

Darbo eiga

1. Optimizavimo užduoties formulavimas,
2. Gradientinio nusileidimo metodas,
3. Rezultatų atvaizdavimas,
4. Greičiausio nusileidimo metodas,
5. Deformuojamas simpleksas.

Užduoties formulavimas

Kintamieji: x – priekinės ir galinės sienų plotų suma, y – šoninių, z – viršutinės ir apatinės.

1. Vienetinio dėžės paviršiaus ploto reikalavimas: **x + y + z = 1**
2. Dėžės tūrio pakelto kvadratu funkcija: **$V^2(x, y, z) = \frac{1}{8}xyz$**

   Išvedimas: pastebėjau, kad įprastinės tūrio formulės kvadratą $a^2b^2c^2$ galima išreikšti sandauga (ab)(bc)(ac), o tai yra priešingų sienų plotų pusės. Tada atlikau sukeitimą ab = $\frac{x}{2}$, bc = $\frac{y}{2}$, ac = $\frac{z}{2}$.
3. Išreiškiu vieną kintamąjį per kitus: **z = 1 – x – y**
4. Tikslo funkcija: **f(x, y) = $-\frac{1}{8}xy(1 - x - y)$**

   Pasirinkau optimizuoti pagal tūrio kvadratą padaugintą iš -1. Pati mažiausia reikšmė – didžiausias tūris. Turint omeny, kad turime dėžės paviršiaus ploto reikalavimą, ši formuluotė turi apribojimų. Pirmas – plotas gali būti tik teigiamas. Antras – vienetinio pav. ploto dėžės sienų plotų suma lygi vienui. Optimizuojant laikomasi tik antrojo apribojimo, tačiau jis nemažina srities (vienetui gali būti lygi didelių teigiamų ir didelių neigiamų skaičių suma). Visgi optimizuojama be apribojimų ir išeinama iš realaus atvejo ribų.
5. Tikslo funkcijos gradiento funkcija: df(x, y) = ($\frac{2xy + y^2 - y}{8}$, $\frac{x^2 + 2xy - x}{8}$)

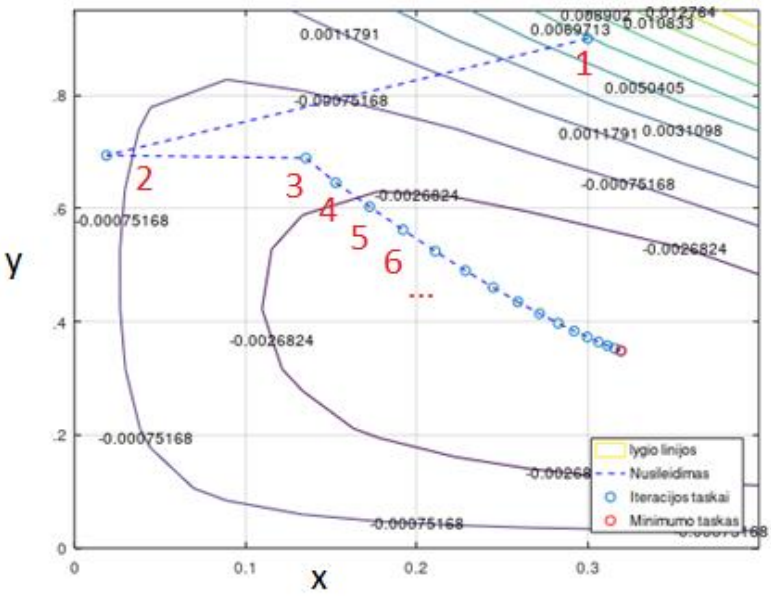
   Perrašiau tikslo funkciją f(x, y) = $\frac{1}{8}(-xy + x^2y + xy^2)$ **,** tada išvedžiau išvestines pagal x ir y.
6. Tikslo ir gradiento funkcijų reikšmės taškuose X0= (0,0), X1=(1,1) ir Xm= (0.3, 0.9):
   a. f(0, 0) = 0;
   b. f(1, 1) = 0.125;
   c. f(0.3, 0.9) = 0.00675;
   d. df(0, 0) = [0, 0];
   e. df(1, 1) = [0.25, 0.25];
   f. df(0.3, 0.9) = [0.05625, 0.04125]

Gradientinis nusileidimas

1. Programos pradžioje aprašiau ankstesniame skyriuje minėtus duomenis: tikslo funkciją, gradientą ir pradinį tašką. Tada aprašiau metodui būdingus parametrus: žingsnio koeficientą ir siekiamą tikslumą.
2. Toliau parašiau optimizavimo ciklą.
    a. Kiekvienoje iteracijoje apskaičiuojamas naujas taškas judant link minimumo. Poslinkis – anti-gradiento ir žingsnio sandauga. Bandymų metu pastebėjau, kad su žingsnio koeficientu = 5 metodas veikia patikimai ir pakankamai greitai.
    b. Skaičiavimai baigiami tada, kai einamos iteracijos gradiento kvadratinė norma mažesnė nei 0.001 ar kitas nurodytas dydis.
3. Rezultatas – gaunamos minimumo taško koordinatės su paklaida. Ji priklauso nuo skaičiavimo ciklo nutraukimo sąlygos – mažiausios leistinos gradiento normos reikšmės. Pastebėjau, kad nurodžius 0.001 kaip mažiausią normą, paklaida būna apie 0.01, tai yra – dešimt kartų didesnė.

Pradinis taškas	Minimumo taškas	Minimumo reikšmė	Tūris	Tikslumas	Žingsnio koeficientas	Iteracijų kiekis
(0.3, 0.9)	(0.33163, 0.33505)	-0.00460	0.0678	0.01	5	17
(0, 0)	(0, 0)	0	0	0.01	5	1
(1, 1)	(-∞,∞)	-∞	∞	0.01	5	14
(1, 1)	(0.338881, 0.338881)	-0.00462	0.06801	0.01	2	11

Rezultatų paaiškinimas. Pradėjus taške (0, 0) skaičiavimas atliko tik vieną iteraciją, nes gradientas lygus nuliui ir X0 sutapo su X1. Poslinkis lygus 0, skaičiavimas baigėsi. Skaičiavimas, pradėtas taške (1, 1) divergavo į neigiamą begalybę, nes lokalus minimumo taškas buvo peršoktas. Su mažesniu žingsniu skaičiavimas konvergavo tašką (0.33215 0.33215).

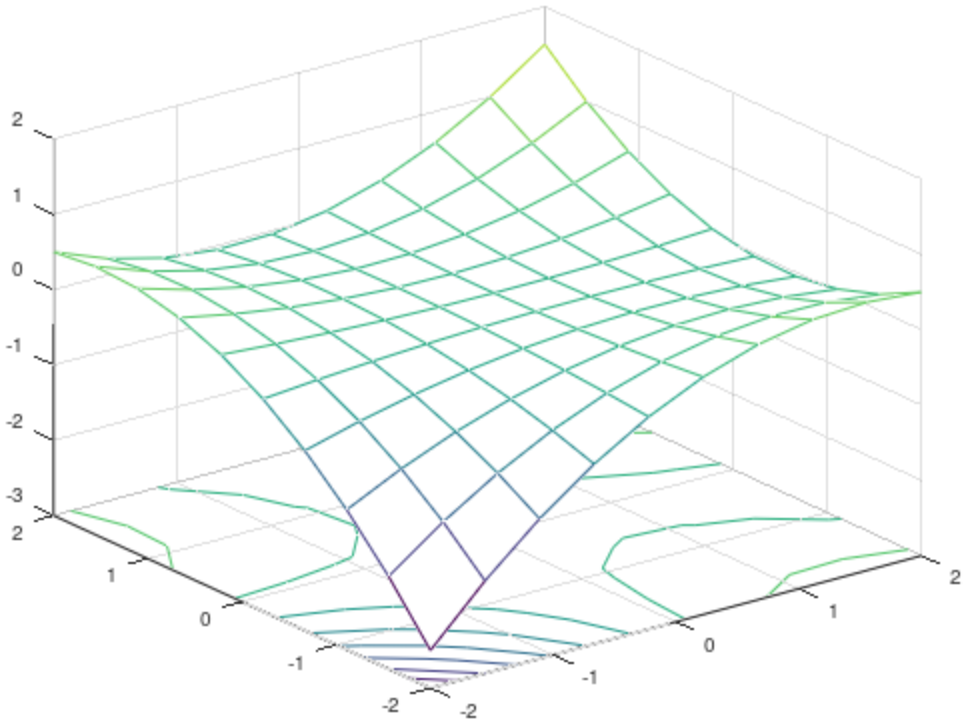


Pav. 1 sėkmingas skaičiavimas. Pradinis taškas (0.3, 0.9). Skaičiais pažymėti pirmų 6 iteracijų taškai.

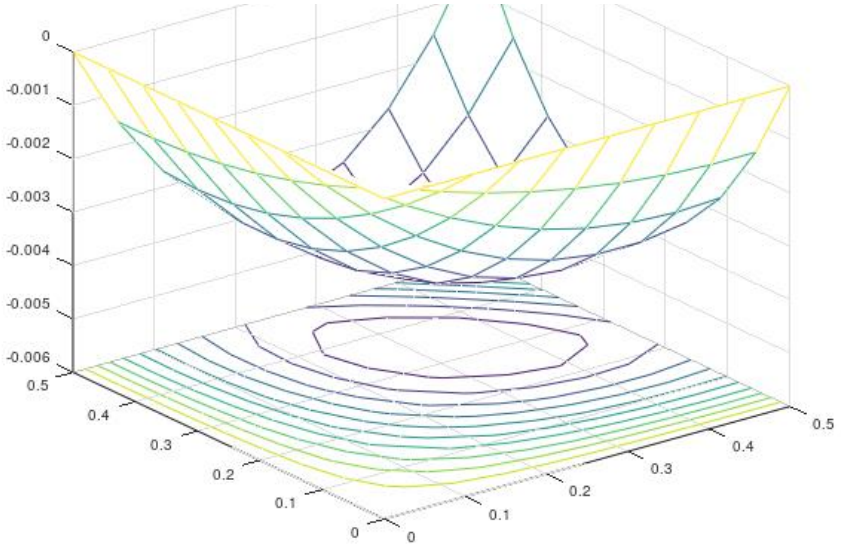
Atvaizdavimas

Skaičiavimo atvaizdavimui pasirinkta lygio linijų diagrama, kurioje šalia lygių yra sudėlioti punktyrais sujungti iteracijų taškai. Skaičiavimo paskutinis taškas – minimumas yra pažymėtas skirtinga spalva nei likę taškai. Taip pat pavyko nubraižyti tikslo funkcijos 3D diagramą.

Svarbu pabrėžti, kad tikslo funkcija nėra aprėžta iš apačios, todėl optimizavimo metodai gali diverguoti.



Pav. 2 Tikslo funkcija, kai x ir y priklauso tarp -2 ir 2. Matosi greitas funkcijos mažėjimas link neigiamos begalybės. Diagramos viduryje matyti nežymus įdubimas. Aukščio koordinatė – tikslo funkcijos reikšmė.



Pav. 3 - tikslo funkcija realių atvejų intervale: x,y tarp 0 ir 0.5. Matosi įdubimas, kurio minimumas minimumas arti taško (0.3, 0.3).

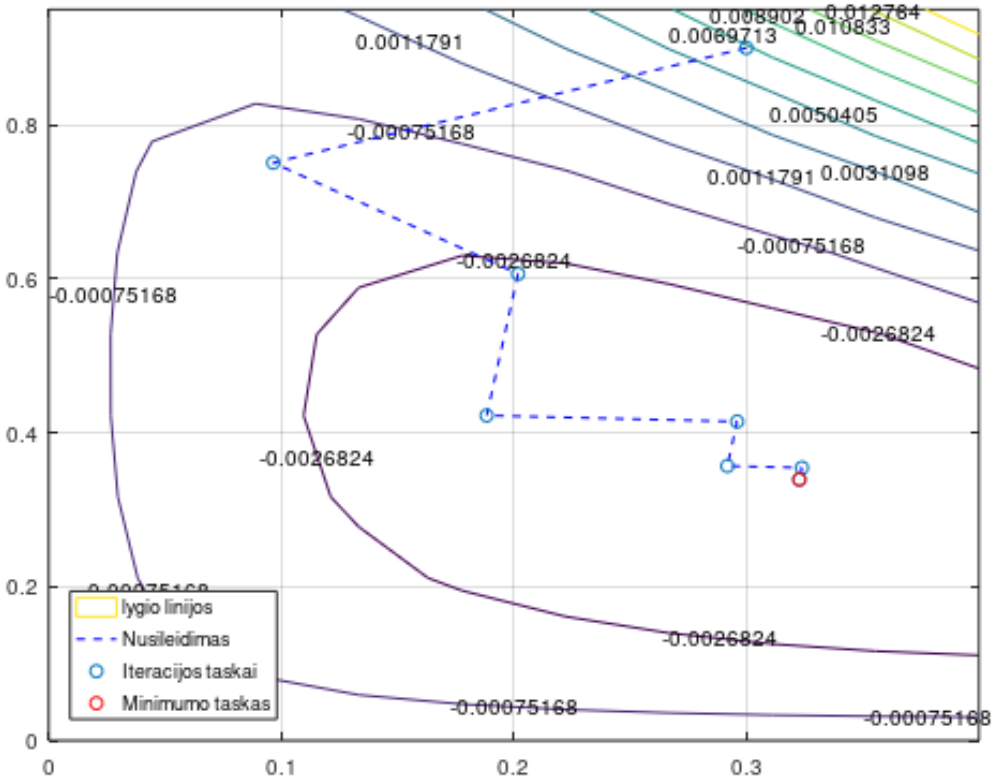
Greičiausiasis nusileidimas

1. Programos pradžioje apibrėžiau žingsnio koeficiento intervalą - gama (nuo 0 iki 20), kuriame bus ieškoma optimalaus žingsnio. Apskaičiuojamas optimalaus žingsnis su tikslumu 0.01.
2. Ciklo metu sprendžiamas vienmatis optimizavimo uždavinys siekiant rasti geriausią žingsnį. Tam naudojamas auksinio pjūvio metodas. Radus geriausią žingsnį nustatomas naujas iteracijos taškas.
3. Rezultatai:

Pradinis taškas	Minimumo taškas	Minimumo reikšmė	Gama intervalas	Tikslumas	Iteracijų kiekis
(0.3, 0.9)	(0.32277, 0.33919)	-0.004626	[0, 20]	0.001	7
(0, 0)	(0, 0)	0	[0, 20]	0.001	1
(1, 1)	(-∞,∞)	-∞	[0, 20]	0.001	14
(1, 1)	(0.33317, 0.33349)	-0.00463	[0, 5]	0.00001	37
(0.55, 0.55)	(0.33326, 0.33326)	-0.004630	[0, 20]	0.001	1

Rezultatai labai panašūs į ankstesnio metodo. Esminis skirtumas: tas pats tikslumas pasiektas su mažiau iteracijų. Paskutinis rezultatas parodo, kad įmanoma parinkti pradinį tašką taip, kad minimumas būtų rastas viena iteracija. Tam reikia, kad anti-gradientas būtų nukreiptas tiesiai į lokalųjį minimumą. Tada apskaičiuojamas optimalus žingsnis, su kuriuo nustatomas minimumas.

Svarbus šio metodo momentas: reikia pasirinkti intervalo galą žingsnio optimizavimo uždaviniui. Pasirinkus per didelį intervalą skaičiavimas gali diverguoti. Tas nutiko pasirinkus (1,1) kaip pradinį tašką.

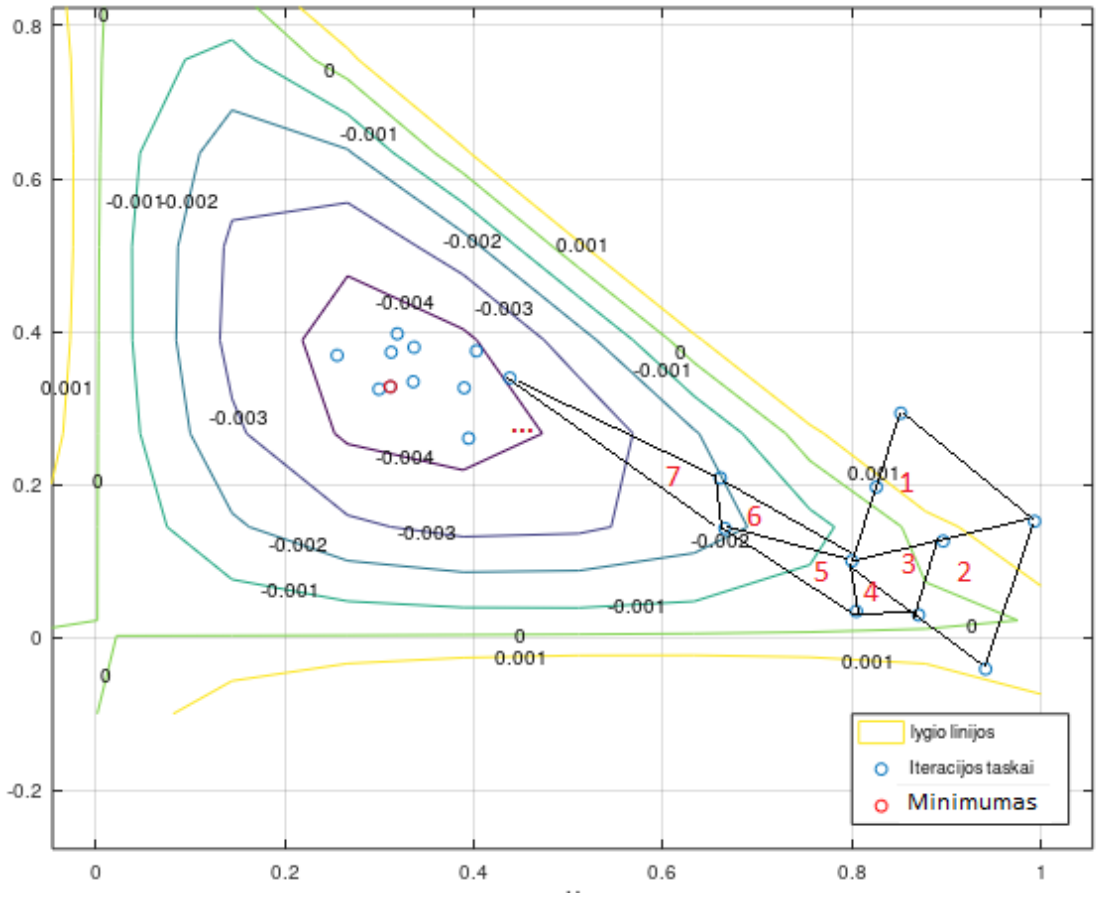


Pav. 4 sėkmingas skaičiavimas greičiausio nusileidimo metodu. Pradinis taškas (0.3. 0.9). Keliuose taškuose parinktas geras, bet ne optimalus žingsnio koeficientas, nes jis buvo už intervalo ribų.

Deformuojamas simpleksas

1. Pasirinktos standartinės vertės simplekso deformavimui: išplėtimo koeficientas 2, suspaudimo koeficientai 0.5 ir -0.5. Pradinio trikampio kraštinės ilgis 0.2.
2. Skaičiavimai vykdomi tol, kol trikampio kraštinė yra didesnė nei norimas tikslumas. Bandymų metu pastebėjau, kad gaunamas tikslumas yra labai arti nurodyto mažiausio kraštinės ilgio. Sumažinus kraštinės ilgį dešimt kartų, paklaida tiek pat kartų sumažėja.
3. Rezultatu laikomas paskutinės iteracijos sudaryto trikampio geriausias taškas.
4. Įgyvendinant šį optimizavimo metodą susidurta su nesėkmingais skaičiavimais ir begaliniais ciklais. Sumažinus kraštinės ilgį sumažėja tokių atvejų. Rezultatai:

Pradinis taškas	Minimumo taškas	Minimumo reikšmė	Pradinis trikampio ilgis	Tikslumas	Iteracijų kiekis
(0.3, 0.9)	-	-	0.2	0.001	Begalinis ciklas
(0.3, 0.9)	(0.33336, 0.33177)	-0.004629	0.1	0.001	24
(0, 0)	(0.33218, 0.33521)	-0.004629	0.2	0.001	24
(1, 1)	-	-	0.2	0.001	Begalinis ciklas
(0.8, 0.1)	(0.33330, 0.33323)	-0.00463	0.2	0.001	28



Pav. 5 - pavaizduotas optimizavimas deformuojamu simpleksu. Pradinis taškas (0.8, 0.1). Raudonai sužymėta dalis iteracijose sudarytų trikampių.

Metodų palyginimas

Pateikiu lentelę, kiek iteracijų prireikia nagrinėtiems metodams siekiant atsakymo su 0.001 paklaida. Pradinis taškas yra pagal studento numerį (0.3, 0.9).

Pavadinimas	Minimumo taškas	Iteracijų kiekis
Gradientinis nusileidimas	(0.3319, 0.3347)	17
Greičiausias nusileidimas	(0.3327, 0.3346)	7
Deformuojamas simpleksas	(0.3333, 0.3317)	24

Su greičiausio nusileidimo metodu atliekama mažiau iteracijų nei su gradietiniu siekiant to paties tikslumo. Tačiau kiekvienoje iteracijoje sprendžiamas vienmatis optimizavimo uždavinys, dėl ko pagreitėjimas gali būti nežymus. Deformuojamo simplekso skaičiavimams reikia panašaus kiekio iteracijų kaip gradientinio nusileidimo, tačiau skaičiuojant naudojama tikslo funkcija, o ne jos gradientas.

Išvados

1. Nagrinėtiems metodams reikia nurodyti kintamuosius, kurie nulemia paieškos greitį. Pasirinkus per didelę reikšmę metodas gali tapti nepatikimas. Prastas pasirinkimas padaro didžiausią neigiamą poveikį optimizuojant gradientiniu nusileidimu.
2. Suformuluotas uždavinys taip, kad tikslo funkcija nėra iš apačios aprėžta ir dėl to dalis skaičiavimų diverguoja. Tenka nurodyti mažesnes vertes metodo kintamiesiems.
3. Skirtingi metodai skirtingose vietose būna sėkmingi, nesėkmingi. Nusileidimo metodai taške (0, 0) sustoja, o (1, 1) greitai randa minimumą, o deformuojamo simplekso skaičiavimai – baigiasi priešingai.

Priedas – kodo fragmentai

```
while norm(df(Xi(1), Xi(2)), 2) > tikslumas
  ankstesnis_X = Xi;
  Xi = Xi - zingsnio_koef * df(Xi(1), Xi(2));
  taskai = [taskai; Xi];
  Iteraciju_kiekis += 1;
endwhile
```

Pav. 6 Gradientinio nusileidimo metodo pagrindinis ciklas. df - tikslo funkcijos gradientas. Xi - aproksimuojamas minimumas.

```
while norm(df(Xi(1), Xi(2)), 2) > tikslumas #Iteraciju_kiekis < 60
  ankstesnis_X = Xi;
  grad = df(Xi(1), Xi(2));
  fg = @(g) fl(Xi(1) - g * grad(1), Xi(2) - g * grad(2));
  zingsnio_koef = optimizavimas_aukso_pjuviais(fg, g0, gn, g_tikslumas);
  Xi = Xi - zingsnio_koef * df(Xi(1), Xi(2));
  taskai = [taskai; Xi];
  Iteraciju_kiekis += 1;
endwhile
```

Pav. 7 Greičiausiojo nusileidimo ciklas. g0 ir gn - žingsnio koeficiento intervalo galai.

```
# Nustatomas deformacijos koeficientas
if FXnaujas > trikampis(1, 1) && FXnaujas < trikampis(2, 1) # Nauja
  O = 1;
elseif FXnaujas < trikampis(1, 1) # Naujas taskas itin geras
  O = ispletimo_koef;
elseif FXnaujas > trikampis(3, 1) # Naujas taskas yra prasciausias
  O = niu_suspaudimas;
  zingsnis_sekmingas = false;
elseif FXnaujas > trikampis(2, 1) && FXnaujas < trikampis(3, 1) # N
  O = beta_suspaudimas;
endif
```

Pav. 8 Deformuojamų simpleksų metodo ištrauka - deformacijos koeficiento nustatymo logika.

```
f = @(x, y) - (1 / 8) * x * y * (1 - x - y);
x=linspace(-2,2,10);
y=linspace(-2,2,10);
[xx,yy]=meshgrid(x,y);
reiksmes = arrayfun(f, xx, yy);
meshc(xx,yy,reiksmes);
```

Pav. 9 Tikslo funkcijos pavaizdavimas 3D grafiku su meshc funkcija. f - tikslo funkcija. Xx – x reikšmių matrica, yy – y reikšmių.