VYTAUTO DIDŽIOJO UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Kęstutis Žilinskas

Stochastinio tiesinio programavimo Monte Karlo metodu tyrimas

Daktaro disertacija

Fiziniai mokslai (P 000)
informatika (09 P)
Informatika, sistemų teorija (P 175)

Vilnius 2007

Disertacija rengta 2005-2007 metais Matematikos ir informatikos institute

Mokslinis vadovas:

Prof. habil. dr. Leonidas Sakalauskas (Matematikos ir informatikos institutas, fiziniai mokslai, informatika – 09 P).

PADĖKA

Disertacijos autorius dėkoja moksliniam vadovui prof. habil. dr. Leonidui Sakalauskui, už per doktorantūros studijas suteiktas nuoširdžias, labai atsakingas bei vertingas mokslines konsultacijas, už patarimus ir pasiūlymus bei nuolatinį skatinimą tobulėti, Matematikos ir informatikos instituto direktoriui prof. habil. dr. Gintautui Dzemydai už visapusišką paramą studijuojant doktorantūroje, MII duomenų analizės skyriaus operacijų tyrimo sektoriaus darbuotojams bei doktorantams už bendradarbiavimą, pagalbą ir supratimą, disertacijos recenzentams prof. habil. dr. Antanui Žilinskui bei dr. Sauliui Minkevičiui, atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų patarimų bei kritinių pastabų, Šiaulių universiteto matematikos ir informatikos fakulteto dekanui doc. dr. Dariui Šiaučiūnui ir Informatikos katedros vedėjai doc. dr. Sigitai Turskienei bei visiems katedros darbuotojams, padėjusiems atspausdinti šią disertaciją bei visiems draugams ir artimiesiems už jų paramą, kantrybę, supratingumą ir meilę.

Kęstutis Žilinskas

Turinys

Žymenys ir santrumpos

b. t. – beveik tikrai

P(A) – įvykio A tikimybė;

$\Re^n$ – *n*-matė Euklido erdvė;

(Ω, Σ, P_x) – tikimybinė erdvė;

P_x – erdvės tikimybinis matas, kurio tankio funkcija $p : \Re_n \times \Omega \to \Re_+$;

$\xi \in \Omega$ – elementarusis įvykis tikimybinėje erdvėje (Ω, Σ, P_x);

$\|x\|$ – *n*-mačio vektoriaus euklidinė norma, $\|x\| = \sqrt{x_1^2 + ... + x_n^2}$;

$\|x - y\|$ – atstumas tarp dviejų taškų metrinėje erdvėje;

I_n – *n*-tos eilės vienetinė matrica;

$\pi_D(x)$ – vektoriaus *x* projekciją į sritį *D*;

$f(x, \xi)$*f(x)* – n-matė funkcija $f : \Re_n \times \Omega \to \Re_+$;

$\nabla_x f(x)$ – funkcijos *f(x)* gradientas;

X~ $N(0,1)$ – atsitiktinis dydis, pasiskirstęs pagal standartinį normalinį (Gauso)dėsnį;

X~ $N(\mu, \sigma^2)$ – atsitiktinis dydis, pasiskirstęs pagal normalinį (Gauso) dėsnį, kurio vidurkis μ ir dispersija σ^2;

X~ $N(0, E_n)$ – atsitiktinis *n*-matis vektorius, kurio komponentės yra pasiskirsčiusios pagal standartinį normalinį dėsnį;

X~ $N(\mu, \Sigma)$ – atsitiktinis *n*-matis vektorius, kurio komponentės yra pasiskirsčiusios pagal daugiamatį Gauso dėsnį, μ – vidurkis, Σ – kovariacijų matrica;

$p(\cdot)$ – tankio funkcija;

$Gauss()$ – atsitiktinio dydžio, pasiskirsčiusio pagal standartinį normalinį dėsnį, skaičiavimo funkcija;

$Fish(\gamma, n, m)$ – Fišerio skirstinio γ - kvantilis su *n* ir *m* laisvės laipsnių skaičiumi;

[·] – skaičiaus sveikoji dalis;

iter – algoritmo kartojimų skaičius;

N – algoritmo žingsnių skaičius;

$N_0, N_{\min}, N_{\max}$ – pradinis, mažiausias ir didžiausias Monte Karlo imties ilgis;

n, n_1, n_2 – funkcijos kintamųjų skaičius;

m, m_1, m_2 – leistinosios srities ribojimų skaičius;

δ – funkcijos reikšmės tikslumas;

$F = \min_{x \in D} f(x)$ – tikslo funkcijos minimali reikšmė;

$\tilde{F}(x)$ – tikslo funkcijos reikšmės įvertis;

$\tilde{g}(x)$ – gradiento įvertis;

$x^t = \left(x_1^t, x_2^t, \ldots, x_{n_1}^t\right)$ – sprendinys (taškas) t-oje iteracijoje;

$x_1, x_2, \ldots, x_n$ – vektoriaus x komponentės;

$f\left(x, y^t\right)$ – tikslo funkcijos reikšmė *t*-ajame Monte Karlo sekos žingsnyje;

$g\left(x, y^t\right)$ – gradiento reikšmė *t*-ajame Monte Karlo sekos žingsnyje;

N^t – Monte Karlo imties ilgis *t*-oje iteracijoje;

$f\left(x, y^i\right)$ – tikslo funkcijos reikšmė *i*-tajame Monte Karlo sekos žingsnyje;

$g\left(x, y^i\right)$ – gradiento reikšmė *i*-tajame Monte Karlo sekos žingsnyje;

$\tilde{F}\left(x^t\right)$ – tikslo funkcijos reikšmės įvertis *t*-oje iteracijoje;

$\tilde{g}\left(x^t\right)$ – gradiento įvertis *t*-oje iteracijoje;

$\tilde{D}^2\left(x^t\right)$ – Monte Karlo imties dispersija *t*-oje iteracijoje;

$Z\left(x^t\right)$ – kovariacijų matrica *t*-oje iteracijoje;

$g = \left(g_1, g_2, \ldots, g_n\right)$ – gradientas;

$V(x)$ – gradiento leistinųjų krypčių aibę taške x;

$V_\varepsilon(x)$ – gradiento ε-leistinųjų krypčių aibę taške x;

$\tilde{g}_\varepsilon^t = \tilde{g}\left(x^t\right)_{V_\varepsilon\left(x^t\right)}$ – gradiento projekcija ε-leistinųjų krypčių aibėje t-oje iteracijoje;

T_t^2 – Hotelingo T^2-statistika *t*-oje iteracijoje;

$\rho_{x^t}(g)$ – gradientinės paieškos žingsnio ilgis *t*-oje iteracijoje;

$\tilde{g}_\varepsilon^t = \tilde{g}\left(x^t\right)_{V_\varepsilon\left(x^t\right)}$ – gradiento projekcija ε - leistinųjų krypčių aibėje t-oje iteracijoje;

$x^* = (x_1, x_2, \ldots, x_n)$ – optimalus sprendinys (taškas);

$F(x^*)$ – tikslo funkcijos reikšmė optimaliame taške;

c, b, q, μ, σ – dviejų etapų uždavinio parametrų vektoriai;

A, W, T – dviejų etapų uždavinio parametrų matricos;

D – dviejų etapų uždavinio ribojimų sritis.

STP – stochastinis tiesinis programavimas.

SNP –stochastinis netiesinis programavimas.

1 skyrius. Įvadas

1.1. Tyrimų sritis

Sprendžiant išteklių bei finansų planavimo, darbų kalendorinio paskirstymo ir vadybos uždavinius, dažnai susiduriama su problemomis: parametrai gali būti nedeterminuoti ir susiję su įvairaus pobūdžio neapibrėžtimi. Ši neapibrėžtis dažniausiai yra aprašoma statistiniais tikimybiniais metodais, o minėti uždaviniai sprendžiami stochastinio tiesinio arba netiesinio programavimo metodais. Dviejų arba kelių etapų stochastinio tiesinio programavimo uždaviniai yra klasikinio tiesinio programavimo apibendrinimas, kai uždavinio parametrai yra atsitiktiniai kintamieji. Šio darbo tyrimų sritis yra stochastinio netiesinio programavimo bei Monte Karlo metodo tyrimas ir taikymas dviejų etapų stochastinio tiesinio programavimo algoritmams sudaryti.

1.2. Problemos aktualumas

Stochastinis programavimas, kaip tiesinio programavimo apibendrinimas, atsirado XX a. šeštajame dešimtmetyje dėl techninio, ekonominio ir finansinio planavimo poreikių. Galima paminėti aktualias stochastinio programavimo taikymo sritis:

- elektros energijos gamyba;
- produkcijos gamyba ir transportavimas;
- personalo valdymas;
- logistika;
- investicijų valdymas;
- objektų išdėstymas;
- mechanizmų stabilumas;
- biologinių sistemų analizė ir kt.

Klasikiniuose tiesinio programavimo algoritmuose nebuvo atsižvelgiama į daugelio planavimo parametrų neapibrėžtumą. Sąvoka „stochastinis programavimas“ atsirado kartu su netiesinio programavimo sąvoka, kai matematinio programavimo pradininkai Dancigas, Čarnesas ir Kuperis pradėjo analizuoti tiesinio programavimo uždavinius su atsitiktiniais koeficientais. Atsitiktinių parametrų naudojimas tiesinio programavimo modeliuose veda prie sudėtingų netiesinių ekstreminių uždavinių, kurie dažniausiai negali būti sprendžiami tiesioginiais tiesinio ir netiesinio programavimo metodais.

Pagrindinės stochastinio programavimo uždavinių sprendimo problemos: sudėtingas tikslo funkcijos tikslių reikšmių apskaičiavimas bei gautojo taško priklausomumo leistinajai sričiai patikrinimas. Stochastiniame programavime reikia rasti uždavinio optimalų sprendinį, kai uždavinio tikslo funkcijos tikslią reikšmę optimaliame taške apskaičiuoti beveik neįmanoma ir kai yra ribojimai, kurių patikrinti taip pat beveik neįmanoma. Sprendžiant tiesinio programavimo uždavinius, sprendimo eiga turi būti pasirinkta dar nežinant kai kurių parametrų reikšmių, kurios yra tikslinamos vėlesniame sprendimo etape. Šios parametrų reikšmės gali turėti įtakos sprendimo eigos pasirinkimui, ar net sukurti skirtingą sprendimo eigą.

Stochastinio tiesinio programavimo STP uždavinių sprendimas duotu tikslumu yra aktuali ir dar nepakankamai išnagrinėta teorinė ir praktinė problema.

1.3. Tyrimų objektas

Disertacijos tyrimų objektas yra tiesioginis ir dualusis stochastinio tiesinio programavimo uždaviniai, dviejų etapų stochastinio tiesinio programavimo uždavinys, Monte Karlo metodo taikymas stochastiniam diferencijavimui ir optimizavimui, Monte Karlo įverčių taikymas statistinėms hipotezėms tikrinti.

1.4. Tyrimų tikslas ir uždaviniai

Bendru atveju STP uždaviniai formuluojami kaip stochastinio dinaminio programavimo uždaviniai. Šių uždavinių sprendimas suvedamas į determinuoto didelio matavimo tiesinio programavimo uždavinio, gauto diskretizuojant atsitiktinius parametrus, sprendimą. Tokiu būdu gauto uždavinio ribojimų matrica pasižymi tam tikra struktūra, kuria pasinaudojama kuriant įvairius dekompozicijos algoritmus STP uždaviniams spręsti. Atskiras STP atvejis yra dviejų etapų stochastinio programavimo uždavinys, į kurį gali būti suvedami daugelis taikomųjų uždavinių, kai neapibrėžtis aprašoma tolydiniu tikimybiniu dėsniu. Dekompozicijos metodų, paremtų tolydžiojo tikimybinio dėsnio diskretizavimu, taikymas šiam uždaviniui spręsti susijęs su skaičiavimo problemomis, saugant kompiuterio atmintyje didelius duomenų masyvus, bei gauto sprendinio tikslumo įvertinimu.

Kita kryptis, leidžianti kurti praktinius stochastinio programavimo algoritmus, yra stochastinio netiesinio programavimo ir Monte Karlo metodų taikymas.

Darbo tikslas yra atlikti žinomų stochastinio tiesinio programavimo metodų analitinį tyrimą, ištirti stochastinio gradiento įvertinimo bei projektavimo metodus ir sudaryti iteracinį algoritmą bei programinę įrangą stochastinio tiesinio programavimo uždaviniui spręsti Monte Karlo metodu duotu tikslumu.

Siekiant šio tikslo, darbe sprendžiami tokie uždaviniai:

- ištirti stochastinio netiesinio gradientinio optimizavimo metodus;
- sudaryti ir ištirti stochastinio diferencijavimo ir stochastinio gradiento įvertinimo metodus;
- sudaryti ir ištirti stochastinio gradiento ε-projektavimo metodus;
- atlikti kompiuterinį eksperimentą įvairiems stochastinio diferencijavimo metodams palyginti;
- sudaryti stochastinio tiesinio programavimo ε-leistinųjų krypčių algoritmą Monte Karlo metodu;
- sudaryti algoritmą Monte Karlo imčių tūriui parinkti;
- sudaryti algoritmą sprendinio optimalumo hipotezei patikrinti statistiniais kriterijais;
- sudaryti metodiką stochastinio tiesinio programavimo algoritmų efektyvumui tirti kompiuterinio modeliavimo būdu;
- kompiuterinio modeliavimo būdu ištirti sudaryto algoritmo konvergavimą, sprendžiant standartinės testinių pavyzdžių duomenų bazės uždavinius;
- palyginti sudaryto algoritmo efektyvumą su Dancigo-Vulfo ir Benderso dekompozicijos metodais.

1.5. Mokslinis naujumas

Darbe gauti tokie nauji rezultatai:

1. Sudaryti ir ištirti stochastinio diferencijavimo algoritmai:
    - tiesioginio diferencijavimo (dualaus uždavinio sprendimo),
    - baigtinių skirtumų,
    - modeliuojamojo pokyčio,
    - tikėtinumo santykio.
2. Sudarytas stochastinio gradiento ε-projektavimo algoritmas.

3. Pasiūlytas dviejų etapų stochastinio tiesinio programavimo Monte Karlo metodas bei sudarytas jį realizuojantis iteracinis algoritmas.

4. Sudaryta ir ištirta Monte Karlo imčių tūrio parinkimo taisyklė, garantuojanti konvergavimą bei racionaliai panaudojanti skaičiuojamuosius išteklius.

5. Sudarytas algoritmas optimalumo hipotezei patikrinti, taikant statistinius kriterijus.

6. Sprendžiant testinius uždavinius parodyta, jog sudarytas algoritmas leidžia surasti STP uždavinių sprendinius reikiamu tikslumu panaudojant priimtinus kompiuterinio laiko ir atminties išteklius.

1.6. Praktinė darbo reikšmė

Darbe gauti tokie praktiniai rezultatai:

1. Sukurti algoritmai ir programinių modulių biblioteka Delphi, FreePascal, MathCad, C++ priemonėmis, realizuojanti stochastinio diferencijavimo algoritmą Monte Karlo metodu.
2. Sukurti algoritmai ir programinių modulių biblioteka Delphi, FreePascal, MathCad, C++ priemonėmis, realizuojanti dviejų etapų stochastinio tiesinio programavimo algoritmą Monte Karlo metodu.
3. Sudarytas algoritmas yra pritaikytas personalo organizavimo uždaviniui spręsti.
4. Sudarytas algoritmas yra pritaikytas investavimo ir elektros energijos gavybos uždaviniui spręsti.

1.7. Darbo rezultatų aprobavimas

Tyrimų rezultatai buvo pristatyti ir aptarti:

- 2005 m. sausio 26-27 d. KTU konferencijoje “Informacinės technologijos – 2005” skaitytas pranešimas “Stochastinio tiesinio optimizavimo algoritmas Monte Karlo imčių serijomis.”
- 2005 m. balandžio 7 d. KTU konferencijoje “Matematika ir matematinis modeliavimas” skaitytas pranešimas “Stochastinio tiesinio optimizavimo Monte Karlo imčių serijomis algoritmas.”
- 2005 m. rugsėjo 15-17 d. 12-oje tarptautinėje mokslinėje kompiuterininkų konferencijoje “Kompiuterininkų dienos – 2005” skaitytas pranešimas “Monte Karlo metodo taikymas netiesiniam stochastiniam programavimui”.

- 2006 m. vasario 17-18 d. jaunųjų Lietuvos ir Britanijos mokslininkų bei doktorantų seminare "Optimalus Projektavimas" (INYS "Optimal Process Design") skaitytas pranešimas "Application of the Monte-Caro Method to Stochastic Linear Programming".
- 2006 m. gegužės 26 d. Lietuvos jaunųjų mokslininkų konferencijoje "Operacijų tyrimai ir taikymai" skaitytas pranešimas "Stochastinio tiesinio programavimo statistinių stabdymo kriterijų tyrimas".
- 2007 m. gegužės 18 d. Lietuvos jaunųjų mokslininkų konferencijoje "Operacijų tyrimai ir taikymai" skaitytas pranešimas "Stochastic Linear Programming by Decomposition Method".

1.8. Darbo rezultatų publikavimas

Tyrimo rezultatai publikuoti šiuose moksliniuose leidiniuose:

Užsienio mokslo leidiniuose, įtrauktuose į Mokslinės informacijos instituto pagrindinių žurnalų sąrašą:

1. Sakalauskas L., Žilinskas K. Application of statistical criteria to optimality testing in Stochastic Programming. *Technological and economic development of economy*. ISSN 1392-8619. Vilnius: Technika, 2006, XII tomas, Nr.4, p. 314-320. (Business source complete nuo 2005).

Tarptautinių konferencijų darbuose, įtrauktuose į Mokslinės informacijos instituto sąrašą:

2. Sakalauskas L., Žilinskas K. Application of the Monte-Carlo Method to Stochastic Linear Programming. *Series on Computers and Operations Research*. ISBN 981-256-909-X. New Jersey, London: World Scientific, 2006, Vol. 7, Computer Aided Methods in Optimal Design and Operations, p. 39-48. (ISI Proceedings).

Recenzuojamuose Lietuvos ir užsienio leidiniuose:

3. Sakalauskas L., Žilinskas K. Monte Karlo metodo taikymas netiesiniam stochastiniam programavimui. *Informacijos mokslai*. ISSN 1392-0561. Vilnius: Vilniaus universiteto leidykla, 2005, 34 tomas, p. 326-330.
4. Sakalauskas L., Žilinskas K. Stochastinio tiesinio optimizavimo Monte Karlo imčių serijomis algoritmas. *Matematika ir matematinis modeliavimas*. ISSN 1822-2757. Kaunas: Technologija, 2005(1), p. 20-25.

Konferencijų pranešimų medžiagoje:

5. Sakalauskas L., Žilinskas K. „Stochastinio tiesinio optimizavimo algoritmas Monte Karlo imčių serijomis“. „*Informacinės technologijos – 2005. Konferencijos pranešimų medžiaga*“, Kaunas, Technologija, 2005, T. 2, 425-431. (ISBN 9955-09-789-2).

1.9. Disertacijos struktūra

Disertaciją sudaro 5 skyriai, literatūros sąrašas ir priedai.

1-asis skyrius yra įvadinis.

2-ame skyriuje aprašomas stochastinio tiesinio programavimo algoritmų analitinis tyrimas.

3-ame skyriuje aprašomas dviejų etapų stochastinio tiesinio programavimo metodas.

4-ame skyriuje pristatomas dviejų etapų stochastinio programavimo algoritmų tyrimas.

5-ame skyriuje pateikiamos išvados.

Prieduose pateikti:

1) Dancigo-Vulfo dekompozicijos metodo taikymo pavyzdys;
2) Benderso dekompozicijos metodo taikymo pavyzdys;
3) Duomenų bazės uždaviniai;
4) AMPL taikymas;
5) Neraiškių sistemų matematinis programavimas;
6) Klasikiniai tiesinio programavimo metodai;
7) Relaksacijos metodas;
8) Dekompozicijos metodas atskiriant kintamuosius.

2 skyrius. Stochastinio tiesinio programavimo algoritmų analitinis tyrimas

2.1. Stochastinio tiesinio programavimo uždavinio formuluotė

Daugelyje verslo ir technikos sričių iškyla uždavinių, susijusių su įvairaus pobūdžio duomenų neapibrėžtimi, kuriems spręsti taikomi stochastinio programavimo metodai. Stochastinis tiesinis programavimas (STP) – tai tiesinis programavimas, kuriame kai kurie duomenys yra tikimybinio pobūdžio. Jei deterministiniame tiesiniame programavime duomenys yra fiksuoti skaičiai, tai stochastiniame tiesiniame programavime šie duomenys nėra žinomi, bet gali būti žinomas jų tikimybinis skirstinys ar pasiskirstymo funkcija.

Bendru atveju STP uždaviniai formuluojami kaip stochastinio dinaminio programavimo uždaviniai. Šių uždavinių pritaikymo sritys yra apžvelgiamos 2.4 skyrelyje. Vienas plačiausiai taikomų ir dažniausiai sprendžiamų STP uždavinių yra dviejų etapų stochastinio tiesinio programavimo uždavinys. Jei duomenų neapibrėžtis šiame uždavinyje yra aprašoma tolydžiuoju tikimybiniu dėsniu, tai jis gali būti suformuluotas kaip netiesinio programavimo su tiesiniais ribojimais uždavinys. Šiame skyriuje aptarsime tiesinio ir netiesinio stochastinio programavimo uždavinių formuluotes bei apžvelgsime metodus, dažniausiai taikomus jiems spręsti.

2.1.1. Stochastinio programavimo uždavinys

Stochastinio programavimo uždaviniai gali būti interpretuojami kaip netiesinio programavimo apibendrinimas. Bendru atveju matematinio (tiesinio arba netiesinio) programavimo uždavinys gali būti formuluojamas tokiu būdu:

minimizuoti

$$f_0(x),$$

kai tenkinamos sąlygos

$$f_i(x) \le 0, \quad i = 1, 2, \dots, m,$$
$$x \in X \subset \Re^n$$

čia $f_0 : \Re^n \to \Re$ yra tikslo funkcija, $f_i : \Re^n \to \Re,\ i = 0, 1, \dots, n$ – yra ribojimų funkcijos.

Jei matematinio programavimo uždavinys yra tiesinis, tikslo funkciją yra nusako vektorius c, o ribojimus – matrica A. Turime tiesinio programavimo uždavinį:

minimizuoti

$$f(x)=\sum_{i=1}^{n} c_i x_i \ ,$$

kai tenkinamos sąlygos

$$\sum_{j=1}^{n} a_{ij} x_j - b_i \le 0, \ i=1,2,\dots,m;$$
$$x_i \ge 0, \ i=1,2,\dots,n.$$

Tegul STP uždavinio sprendinys priklauso nuo tam tikros tikimybinės erdvės (įvykių erdvės, atsitiktinių parametrų erdvės) (Ω, F, P) elemento ω. Jeigu šio elemento reikšmė yra žinoma ir fiksuota, galima formuluoti parametrinio deterministinio programavimo uždavinį

minimizuoti

$$f_0(x,\omega), \tag{2.1}$$

kai tenkinamos sąlygos

$$\begin{aligned} &f_i(x,\omega) \le 0, \ \ i=1,2,\dots,m, \\ &x \in X \subset \Re^n \end{aligned} , \tag{2.2}$$

Analogiškai formuluojamas ir stochastinio tiesinio programavimo uždavinys:

minimizuoti

$$f(x)=\sum_{i=1}^{n} c_i(\omega) x_i \ , \tag{2.3}$$

kai tenkinamos sąlygos

$$A(\omega)x - b(\omega) \le 0,$$
$$x_i \ge 0, \ i=1,2,\dots,n.$$

kur *A* yra atsitiktinė $m \times n$ matrica, *b* – atsitiktinis m-matis vektorius.

Uždavinio formuluotė priklauso nuo to, ar sprendimo eigoje galima patikslinti atsitiktinių parametrų reikšmes tam tikrų stebėjimais ar bandymais. Galimi dviejų tipų optimizavimo uždaviniai: operatyvinio ir perspektyvinio.

Operatyvinio optimizavimo uždavinio sprendinys gaunamas po tam tikro stebėjimo (eksperimento), priklauso nuo šio stebėjimo rezultato ir yra atsitiktinis vektorius. Tokie uždaviniai iškyla operatyviniame ekonominiame planavime, medicininėje diagnostikoje ir pan. Sprendžiant operatyvinio stochastinio programavimo uždavinius, sprendinys nustatomas realiuoju laiku, todėl jam rasti pakanka pritaikyti įprastus optimizavimo metodus.

Perspektyvinio stochastinio programavimo uždaviniuose sprendinys x randamas neatliekant tarpinio atsitiktinių parametrų stebėjimo. Šiuo atveju sprendinys yra nustatomas sprendžiant matematinio programavimo uždavinį (2.1)-(2.2) su tikimybinio pobūdžio tikslo funkcija ir ribojimais.

Tokie uždaviniai iškyla skaičiuojant valdomų objektų optimalias trajektorijas, projektuojant sistemas, kai jos parametrai gali būti fiksuoti. Perspektyvinio stochastinio programavimo uždavinio sprendinys yra optimalus visos atsitiktinių parametrų visumos požiūriu, tačiau tokiu būdu gautas sprendinys nėra optimalus atskirai paimtai atsitiktinio parametro reikšmei.

2.1.2. Dviejų etapų stochastinio programavimo uždavinys

Labai dažnai operatyvinio planavimo uždaviniuose gauti sprendiniai turi būti koreguojami jų realizavimo eigoje. Dažniausiai tai atsitinka, kai neįmanoma tiksliai ir vienareikšmiškai įvertinti atskirus faktorius. Pavyzdžiui, negalint tiksliai prognozuoti poreikių, dažnai įvyksta gamybos perprodukcija arba neįvykdomi sutartiniai įsipareigojimai. Todėl iškyla uždaviniai rasti – gamybos planus, kurie įvertina neapibrėžtį ir kurių korekcijai reikėtų minimalių išlaidų. Dviejų etapų stochastinio programavimo uždavinys leidžia realizuoti tokių planų radimą.

Jei išlaidų matricos A koeficientai yra atsitiktiniai, t. y. priklauso nuo atsitiktinio parametro ω, bei vektorius b taip pat yra atsitiktinis, tai bendru atveju koks bebūtų sprendinys x, jis negali tenkinti ribojimų sistemos

$$A(\omega)x = b(\omega) \tag{2.4}$$

nes kai kuriems i galios nelygybė $\sum_{j=1}^{n} a_{ij}(\omega)x_j > b_i(\omega)$, kai kuriems kitiems i galios nelygybė $\sum_{j=1}^{n} a_{ij}(\omega)x_j < b_i(\omega)$. Tokiems neatitikimams išvengti konstruojamas antrojo etapo vektorius $y = (y_1, y_2, \ldots, y_r)$ su ribojimų matricomis $W(\omega)$, $T(\omega)$ bei laisvųjų narių vektoriumi $h(\omega) = (h_1(\omega), h_2(\omega), \ldots, h_r(\omega))$ ir nagrinėjama sistema

$$\begin{cases} W(\omega)y + T(\omega)x = h(\omega), \\ x \geq 0, \quad y \geq 0, \end{cases}, \tag{2.5}$$

kuri yra dviejų etapų stochastinio programavimo uždavinio ribojimų sistema. Vektorius x randamas prieš tai, kai sužinoma konkreti atsitiktinio parametro ω reikšmė. Po to, kai ši reikšmė sužinoma, aukščiau minėti neatitikimai likviduojami randant vektorių y. Plano x realizavimo išlaidos $\sum_{j=1}^{n} c_j x_j$, antrojo etapo išlaidos $\sum_{i=1}^{r} q_i y_i$. Antrojo etapo vektorius randamas iš išlaidų minimumo sąlygos (2.5) duotiesiems x ir ω. Pažymėjus šį vektorių $y(x,\omega)$, bendrosios išlaidos plano x realizavimui lygios

$$F(x) = \sum_{j=1}^{n} c_j x_j + E\left(\sum_{i=1}^{r} q_i y_i(x,\omega) \right), \tag{2.6}$$

kai $x \geq 0$.

Šiame darbe nagrinėjamas dviejų etapų stochastinio tiesinio programavimo uždavinys (Shapiro & Homem De-Mello, (1998), Sakalauskas & Žilinskas, (2005b)):

$$F(x) = c \cdot x + E\left\{ \min_y [q \cdot y \mid W \cdot y + T \cdot x = h,\ y \in R_+^m] \right\} \to \min,$$

$$\text{kai} \qquad Ax = b, \quad x \in \Re_+^n,$$

čia vektorius h ir matricos W, T gali būti atsitiktiniai.

Pointegrinė funkcija uždavinio išraiškoje gaunama sprendžiant tiesinio programavimo uždavinį

$$f(x,\xi) = \min_{\substack{Wy+Tx=h \\ y \in R_+^m}} q \cdot y, \tag{2.7}$$

kurio leistinųjų reikšmių sritis $D = \left\{ x \mid A \cdot x = b,\ x \in R_+^n \right\}$ ir matas P_x nusakomas h, W ir T pasiskirstymu.

2.1.3. Daugelio etapų stochastinio tiesinio programavimo uždavinys

Dviejų etapų stochastinio tiesinio programavimo uždaviniuose visos parametrų neapibrėžčių problemos išsprendžiamos antrajame uždavinio sprendimo etape. Dažnai uždavinių atsitiktinių kintamųjų reikšmės stebimos ilgą laiko tarpą, kurį sudaro keletas etapų, kai sprendimai priimami pasibaigus kiekvienam etapui. Pavyzdžiui, gamybos ir įrengimų valdymo uždaviniuose išankstiniai poreikiai nuolat koreguojami firmų užsakymais, todėl produkcijos planavimo sprendimai kiekviename etape priimami atsižvelgiant į konkrečius žinomus poreikius tame etape bei į numatomus ateities tiksliai neapibrėžtus poreikius. Bendru atveju ilgalaikis gamybos planavimas yra daugelio etapų sprendimų priėmimo procesas, kai resursų ištekliai bei produkcijos poreikiai nuolat koreguojami proceso eigoje ir kurio tikslo funkcijos reikšmė įvertinama viso proceso pabaigoje. Tuomet iškyla daugelio etapų stochastinio tiesinio programavimo uždavinių poreikis.

Pagrindinė daugelio etapų uždavinių ypatybė yra nuolatinis atsitiktinių reiškinių kitimas proceso metu. Todėl priimamas sprendimas gali labai pasikeisti priklausomai nuo ankstesniuose etapuose priimtų sprendimų. Pavyzdžiui, planuojant hidroelektrinės elektros energijos gavybą, potvynio atveju pagaminamas elektros energijos kiekis turi būti pakoreguotas visuose energijos gavybos etapuose. Sprendimai, priimti viename etape, gali turėti didelę įtaką vėlesniems sprendimams. Hidroelektrinės atveju bet kuriame etape priimtas planavimo sprendimas gali būti labai skirtingas dėl ilgesnio laikotarpio atmosferinių sąlygų, vieni sprendimai priimami sausros metu, kiti – ilgalaikio lietaus metu.

Tegul *T* yra uždavinio etapų skaičius, x_t - sprendimas, priimamas *t*-ame etape, *A*, *c*, *b* apibrėžti.

Galima suformuluoti daugelio etapų stochastinio tiesinio programavimo uždavinį:

$$F(x) = c_1 x_1 + E\sum_{t=2}^{T} c_t(\xi_t)x_t \to \min_x, \tag{2.8}$$

kai

$$\begin{aligned} &A_{11} && = b_1, \\ &A_{t1}(\xi_t)x_1 + \sum_{\tau=2}^{t} A_{t\tau}(\xi_t)x_\tau = b_t(\xi_t), && b.t.\ t = 2,3,\ldots,T, \\ &x_1 \geq 0, \quad x_t \geq 0, && b.t.\ t = 2,3,\ldots,T. \end{aligned} \tag{2.9}$$

arba

$$\begin{aligned} &F(x) = E\left\{\sum_{t=1}^{T} c_t(\xi_t)x_t\right\} \to \min, \\ &kai \\ &\sum_{\tau=1}^{t} A_{t\tau}(\xi_t)x_t = b_t(\xi_t), \quad b.t.\ t = 1,2,\ldots,T, \\ &x_t \geq 0, \quad b.t.\ t = 1,2,\ldots,T, \\ &x_t \in X_t \quad \forall t. \end{aligned} \tag{2.10}$$

2.2. Stochastinio netiesinio programavimo uždavinys

Stochastinio netiesinio programavimo uždaviniai iškyla, kai netiesinio programavimo uždavinio sprendinys x priklauso nuo tam tikrų atsitiktinių parametrų $\omega \in \Omega$, kurie yra elementarūs įvykiai tam tikros tikimybinėje erdvėje (Ω, Σ, P), o funkcijos, nusakančios tikslo funkciją bei ribojimus, yra netiesinės ir tenkina tam tikras integruotinumo, diferencijuotinumo ir iškilumo sąlygas. Matas P yra absoliučiai tolydus, apibrėžiamas įvedant tankio funkciją, pvz. $p : \Re^n \times \Omega \to \Re_+$.

2.2.1. Bendroji stochastinio netiesinio programavimo uždavinio formuluotė

Sprendžiant optimizavimo uždavinį perspektyvinio planavimo metu sprendinys yra fiksuotas visos atsitiktinių įvykių aibės požiūriu. Jis gali būti gautas sprendžiant matematinio programavimo uždavinį (2.1) su tikimybinio pobūdžio ribojimais (Sakalauskas, Žilinskas, (2005a)):

minimizuoti

$$F_0(x), \tag{2.11}$$

kai tenkinamos sąlygos

$$\begin{aligned} &F_i(x) \le 0, \quad i = 1, 2, \dots, m, \\ &x \in X \subset \Re^n, \end{aligned} \tag{2.12}$$

čia tikslo ir ribojimų funkcijos yra tikimybinio pobūdžio. Šios funkcijos gali būti apibrėžiamos dviem būdais. Galima tikslo arba ribojimo funkcijas paimti lygiomis atsitiktinių funkcijų tikėtinoms reikšmėms.

$$F_0(x) = Ef_0(x, \omega) = f_0(x, \omega) P(d\omega),$$

$$F_i(x) = Ef_i(x, \omega) = f_i(x, \omega) P(d\omega) \le 0,\ i = 1, 2, \dots, m,$$

Kitu atveju tikslo funkciją galima interpretuoti kaip tikimybę viršyti tam tikrą reikšmę,

$$G_0(x) = P\{f_0(x, \omega) \ge a\},$$

o ribojimai gali būti išpildomi tik su tam tikra tikimybe.

$$G_i(x) = P\{f_i(x, \omega) \le 0\} - p_i \ge 0,\ i = 1, 2, \dots, m, \tag{2.13}$$

kur $a, p_i,\ i = 1, 2, \dots, m$ – tam tikri skaičiai.

Dažnai stochastiniame programavime siekiama pirmiausia rasti funkcijas $\widetilde{F}_i(x)$, o po to taikyti netiesinio programavimo metodus. Pavyzdžiui, pradinis uždavinys tam tikru tikslumu pakeičiamas determinuotu uždavinio ekvivalentu, pavyzdžiui, vietoje funkcijų $\widetilde{F}_i(x)$ naudojamos funkcijos $\overline{F}_i(x) = f_i(x, \overline{\omega})$, kur $\overline{\omega}$ yra parametro ω vidurkis.

Tokie metodai vadinami netiesioginiais stochastinio programavimo metodais.

Uždavinių sprendimo metodai, pagrįsti prielaida, kad funkcijų $f_i(x, \omega),\ i = 1, 2, \dots, m$ reikšmes galima apskaičiuoti kiekvienai parametru x arba w reikšmei, vadinami tiesioginiais stochastinio programavimo metodais.

2.2.2 Stochastinio netiesinio programavimo uždavinių pavyzdžiai

Panagrinėsime stochastinės nelygybės sprendimą:

$$hx \le 1, \tag{2.14}$$

kur h – atsitiktinis binarinis dydis, įgyjantis dvi reikšmes ± 1 su vienoda tikimybe $\frac{1}{2}$. Deterministiniu atveju, kai h nėra atsitiktinis dydis, šios nelygybės sprendinys yra pustiesė. Stochastinės nelygybės sprendinys yra toks skaičius x, kuris maksimizuoja reikšmę $G(x) = P\{hx \le 1\}$, t. y. stochastinės nelygybės sprendinio radimas tolygus trūkios funkcijos *G(x)* maksimumo radimui.

Sudėtingesnis pavyzdys – resursų planavimo uždavinys. Tegul sandėlyje telpa a vienetų tam tikro produkto, kurio atsargą reikia sukaupti. Planavimo momentu nėra tikslios informacijos apie šio produkto poreikį, bet yra pagrindas įvertinti poreikį, kaip atsitiktinį dydį ω, kurio pasiskirstymo funkcija $H(z) = P\{\omega < z\}$. Tegul planuojama atsarga sudaro x vienetų. Sudaroma išlaidų funkcija:

$$f(x,\omega) = \begin{cases} \alpha(x-\omega), & x \ge \omega, \\ \beta(\xi-\omega), & x < \omega, \end{cases} \tag{2.15}$$

kur α, β yra atitinkamai saugojimo kaštai bei deficito likvidavimo išlaidos. Ši funkcija nėra diferencijuojama taške $x = \omega$. Tuomet resursų planavimo uždavinys – rasti mažiausias resursų aptarnavimo išlaidas arba rasti išlaidų funkcijų vidurkio minimumą:

$$F(x) = Ef(x,\omega) = \alpha\int_0^x (x-z)\,dH(z) + \beta\int_x^\infty (z-x)\,dH(z), \tag{2.16}$$

kai $0 \le x \le a$. Funkcija *F(x)* taip pat nediferencijuojama.

2.3. Netiesioginiai stochastinio programavimo metodai

Netiesioginiai stochastinio programavimo metodai yra pagrįsti stochastinio uždavinio suvedimu arba pakeitimu determinuotu analogu – tiesinio arba netiesinio programavimo uždaviniu, kurio sprendimo metodai yra žinomi.

2.3.1. Baigtinė atsitiktinių įvykių erdvė

Jei atsitiktinių parametrų stebėjimo (eksperimento) metu nustatoma, kad įvykių erdvė baigtinė, tai dažnai galima nustatyti determinuotą sprendinio radimo taisyklę $x = x(\omega)$ iš funkcijos $f_0(x,\omega)$ minimumo, esant sąlygoms

$$f_i(x,\omega) \le 0, \quad i = 1,2,\dots,m \,,$$

$$x \in X \,.$$

Dažnai ši taisyklė yra visiškai priimtina tikslumo požiūriu ir tik lieka klausimas, kaip esant naujai parametro ω reikšmei rasti sprendinį x, nesprendžiant uždavinį iš naujo, o tik pataisant ankstesnį sprendinį, gautą senajai parametro ω reikšmei.

Tačiau ne visuomet pavyksta šitaip išspręsti uždavinį, net ir tuomet, kai jis suvedamas į tiesinio programavimo uždavinį. Tuomet naudojamasi sprendinio parametrizavimo idėja, t. y. sprendinys $x(\omega)$ ieškomas iš anksto numatytos funkcijos, nusakomos vektoriumi $y = (y_1, y_2, \dots, y_r)$, pavidalu: $x(\xi) = x(y,\omega) = (x_1(y,\omega), x_2(y,\omega), \dots, x_n(y,\omega))$. Šio vektoriaus reikšmė randama pasinaudojant informacija apie atsitiktinio parametro ω skirstinį $P(d\omega)$ iki ω stebėjimo. Taip operatyvinis programavimo uždavinys parametrizavimo pagalba pakeičiamas perspektyviniu programavimo uždaviniu.

Pavyzdžiui, jeigu uždavinys duotajai parametro ξ reikšmei yra tiesinio programavimo uždavinys:

$$\sum_{i=1}^{n} c_i x_i \to \min,$$

esant ribojimams

$$Ax \le b(\omega),$$
$$x \ge 0,$$

tai sprendinį galima nusakyti funkcija $x(\omega) = Yb(\omega)$, kur Y – determinuota matrica su nežinomais elementais, kurių reikšmės randamos iki $b(\omega)$ stebėjimo. Paprastai matrica Y randama iš funkcijos

$$F_0(Y) = P\{(c, Yb(\omega)) \ge a\}$$

minimumo, esant ribojimams

$$F_i(Y) = P\{(AYb(\omega))_i \le b_i(\omega)\} \ge p_i, i = 1,2,\dots,m$$

kur $a, p_i, i = 1,2,\dots,m$ tam tikri skaičiai.

2.3.2. Apytikslis pakeitimas

Turime sprendžiamas uždavinys (2.11)-(2.12):

Jei matematinis vidurkis $\overline{\omega} = E\omega$ turi prasmę, tai uždavinį (2.11)-(2.12) galima performuluoti:

minimizuoti

$$f_0\left(x,\bar{\omega}\right), \tag{2.17}$$

esant ribojimams

$$f_i\left(x,\bar{\omega}\right) \le 0,\ i=1,2,\dots,m, \tag{2.18}$$

$$x \in X .$$

Tai įprastas netiesinio programavimo uždavinys. Jei parametro ω reikšmės pasiskirstė „netoli“ $\bar{\omega}$, tai naujasis uždavinys (2.17)-(2.18) pakankamai tiksliai nusako pradinį uždavinį (2.11)-(2.12). Jei funkcijos $f_i\left(x,\omega\right),\ i=1,2,\dots,m$ yra tiesinės parametro ω atžvilgiu, tai šitaip sukonstruoto uždavinio (2.17)-(2.18) sprendinys tikslai nusako pradinio uždavinio (2.11)-(2.12) sprendinį.

Bendruoju atveju uždavinių (2.11)-(2.12) ir (2.17)-(2.18) sprendiniai gali skirtis iš esmės. Pavyzdžiui, tegul reikia minimizuoti funkciją $F\left(x\right)=E\left(\omega x\right)^2+x-1$, kur atsitiktinis dydis su tikimybe ½ įgyja reikšmes ± 1.

Funkcijos $F\left(x\right)$ minimumo taškas yra $x=-\frac{1}{2}$, kai funkcijos $\left(\bar{\omega}x\right)^2+x-1=x+1$ minimumo taškas yra $x=\infty$.

Reikia pabrėžti, kad determinuotą analogą pavyksta rasti ne kiekvienam stochastinio programavimo uždaviniui.

2.3.3. Dviejų etapų stochastinio programavimo uždaviniai baigtinėje aibėje

Jei dviejų etapų uždavinio (2.1)-(2.2) įvykių aibė yra baigtinė, t. y. $\Omega \in \left\{1,2,\dots N\right\}$ ir $p_1, p_2,\dots, p_N$ yra atitinkamos įvykių tikimybės, tai uždavinį galima užrašyti (Ermolyev, (1976)):

rasti planą x ir jo korekciją $y=\left(y_1, y_2,\dots, y_N\right)$ būsenose $1,2,\dots,N$, minimizuojančius tiesinę tikslo funkciją

$$F(x)=\sum_{j=1}^{N} c_j x_j+\sum_{i=1}^{N}(d_i, y_i)p_i ,$$

esant ribojimams

$$
\begin{cases}
A(1)x + B(1)y_1 & = b(1), \\
A(2)x \quad + B(2)y_2 & = b(2), \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\
A(N)x \quad\quad + B(N)y_N & = b(N), \\
x \geq 0,\ y_1 \geq 0,\ y_2 \geq 0, \ldots, y_N \geq 0. &
\end{cases}
$$

Šis dviejų etapų stochastinio programavimo uždavinys susiveda į tiesinio programavimo blokinį uždavinį, kuriam pritaikomi blokinio programavimo metodai.

2.4. Taikomieji stochastinio tiesinio programavimo uždaviniai

Pirmieji stochastinio programavimo metodų taikymai aprašyti modeliuojant Ajovos žemės ūkio ekonomikos planavimą su žemės naudmenų ir darbo jėgos ribojimais (Tintner, (1955)), lėktuvų maršrutų planavimą su baudomis dėl prarastų keleivių (Ferguson, (1956)) ir mazuto gamybą su neapibrėžtimi produktų kainose bei naudojamuose gamybiniuose pajėgumuose (Charnes, (1958)).

Naujesni stochastinio programavimo metodų taikymai aprašyti modeliuojant:

- produkcijos planavimą (Escudero, (1993));
- tvarkaraščių planavimą (Birge, (1996));
- gamybinių pajėgumų plėtimą (Ahmed, (2003));
- elektros energijos gavybos planavimą (Caroe, (1998));
- vandens saugyklų darbo planavimą (Dupačova, (1991));
- telekomunikacijų planavimą (Laguna, (1998));
- cheminių procesų sistemų konstravimą ir optimizavimą (Gupta, (2000));
- finansinę veiklą (Carino, (1998)).

2.4.1. Transporto valdymas

Standartinis transporto valdymo uždavinys yra:

$$
\begin{aligned}
&\sum_{i,j} c_{ij} x_{ij} \to \min, \\
&kai \\
&\sum_{j} x_{ij} = s_i \ \ \forall i, \\
&\sum_{i} x_{ij} = d_j \ \ \forall j, \\
&\quad x_{ij} \geq 0,
\end{aligned}
$$

kur c_{ij} transportavimo kaštai, s_i - krovinių ištekliai siuntimo vietoje i, d_j - krovinių poreikiai pristatymo vietoje j.

Jei krovinių poreikiai bei transportavimo kaštai yra atsitiktiniai dydžiai, tuomet c_{ij} ir d_j yra stochastinio pobūdžio. Turime stochastinio tiesinio programavimo transporto valdymo uždavinį:

$$
\begin{aligned}
&\sum_{i,j} c_{ij}\left(\xi_j\right) x_{ij}\left(\xi_j\right) \to \min, \\
&kai \\
&\sum_{j} x_{ij}\left(\xi_j\right) = s_i \ \ \forall i, \\
&\sum_{i} x_{ij}\left(\xi_j\right) = d_j\left(\xi_j\right) \ \forall j, \\
&x_{ij}\left(\xi_j\right) \geq 0.
\end{aligned}
$$

Ši uždavinio formuluotė ir sprendimo būdai nagrinėjami (Barbarosoglu, (2004)).

2.4.2. Gamybos valdymas

Deterministinis gamybos planavimo modelis pateiktas (Gazmuri, (2001)). Jame aprašoma gamybos įmonės valdymo uždavinys. Įmonė, turinti K įrenginių skirtinguose jos skyriuose, turi per T laiko periodų pagaminti N rūšių produkcijos ir kartu patenkinti žinomus kiekvienos rūšies produkcijos poreikius kiekvienu laiko periodu. (Įmonė gamina įvairių rūšių šaldytuvus. Pradiniame etape iš metalo plokščių štampuojami šaldytuvų sudėtiniai komponentai. Vėliau jie apdorojami ir nudažomi. Kai kurie komponentai papildomi plastikinėmis bei izoliacinėmis sudėtinėmis dalimis ir papildomai sutvirtinami. Paskutiniame etape iš komponentų surenkami įvairių tipų šaldytuvai.)

Efektyviai produkcijos planavimo politikai nustatyti sudaroma tikslo funkcija, maksimizuojanti gaunamą pelną, apibrėžiamą visų pajamų ir visų kaštų skirtumu. Kaštus sudaro žaliavų ir gamybos kaštai, inventoriaus nusidėvėjimo kaštai, neįvykdytų užsakymų kaštai, inventoriaus remonto kaštai.

Tikslo funkcijos parametrai kiekvienam produktui *i* kiekvienu laiko periodu *t* yra:

produkto vieneto kaina p_{it},

gamybos ir žaliavų kaina c_{it}^{ra},

fiksuota mažiausiai vieno pagaminto produkto kaina c_{it}^{se},

inventoriaus vieneto fiksuota nusidėvėjimo kaina c_{il}^{in},

inventoriaus nusidėvėjimo koeficientas r_{it}^{in},

neįvykdyto užsakymo fiksuota kaina(bauda) c_{it}^{ba},

neįvykdyto užsakymo kainos(baudos) kitimo koeficientas r_{it}^{ba}.

Paskutiniame laiko periode $t=T$ dydžiai c_{iT}^{ba} ir r_{iT}^{ba} nusako produkcijos trūkumo poreikių atžvilgiu parametrus.

Uždavinio sprendinys turi tenkinti poreikių, įrengimų pajėgumų bei minimalaus ir maksimalaus produktų krepšelių dydžių ribojimus. Ribojimų funkcijų parametrai yra:

produkto *i* poreikis laiko periodu *t* d_{it},

pradinis inventoriaus *i* lygis u_{i0}^{+},

pradinis produkto *i* užsakymo neįvykdymo lygis u_{i0}^{-},

laikas, reikalingas įrenginiu *k* pagaminti produkto *i* vienetą valandomis r_{ik},

visas galimas įrenginio *k* panaudojimo laikas laiko periodu *t* a_{kt},

produkto *i* laiko periodu *t* minimalaus krepšelio apatinė riba $\overline{x}_{it}^{\min} = \min\limits_{k}\left\{\frac{a_{kt}}{r_{ik}}\right\}$ ir maksimalaus krepšelio viršutinė riba $\overline{x}_{it}^{\max}$.

Uždavinio kintamieji yra:

produkto *i* gamybos laiko periodu *t* lygis x_{it}, $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$,

inventoriaus *i* kiekis laiko periodu *t* u_{it}^{+}, $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$,

produkto *i* neįvykdyto užsakymo lygis u_{it}^{-}, $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$,

binarinis kintamasis γ_{it}, $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$, įgyjanti s reikšmę 1, kai produktas *i* yra gaminamas laiko periodu *t*, ir reikšmę 0priešingu atveju.

Stochastinis gamybos planavimo modelis pateiktas (Bitran, (1984)). Jame poreikių neapibrėžtis modeliuojama duota baigtine atsitiktinių reikšmių scenarijų aibe, kuri gali būti nustatyta ekspertų, ankstesnių gamybos laikotarpių duomenų ekstrapoliavimo būdu. Tegul ši baigtinė scenarijų aibė yra $\Omega = \{1, 2, \ldots, S\}$. Tuomet $d^s = \left(d_{it}^s\right)$ yra scenarijaus poreikių vektorius, kur d_{it}^s žymi scenarijaus $s \in \Omega$ produkto *i* poreikį laiko periodu *t*. Su kiekvienu scenarijumi $s \in \Omega$ susiejamas svorio koeficientas arba tikimybė p_s, $p_s \geq 0$, tokia, kad $\sum_{s \in \Omega} p_s = 1$.

Kiekvienam scenarijui sudaromi inventoriaus kiekio kintamieji $u_{it1}^+, u_{it2}^+, \ldots, u_{itS}^+$ ir neįvykdytų užsakymų lygio kintamieji $u_{it1}^-, u_{it2}^-, \ldots, u_{itS}^-$.

Stochastinio uždavinio matematinis modelis:

$$\max \left(\begin{array}{l} E\left(\sum_{i=1}^{N} \sum_{t=1}^{T} p_{it} \left[x_{it} + u_{i(t-1)s}^+ - u_{its}^+ \right] \right) - \sum_{i=1}^{N} \sum_{t=1}^{T} c_{it}^{ra} x_{it} - \sum_{i=1}^{N} \sum_{t=1}^{T} c_{it}^{se} \gamma_{it} - \\ -E\left(\sum_{i=1}^{N} \sum_{t=1}^{T} c_{it}^{in} u_{its}^+ - \frac{1}{2} \sum_{i=1}^{N} \sum_{t=1}^{T} r_{it}^{in} \left[u_{its}^+ \right]^2 - \sum_{i=1}^{N} \sum_{t=1}^{T} c_{it}^{ba} u_{its}^- - \frac{1}{2} \sum_{i=1}^{N} \sum_{t=1}^{T} r_{it}^{ba} \left[u_{its}^- \right]^2 \right) \end{array} \right),$$

kai

$$x_{it} + u_{i(t-1)s}^+ - u_{its}^+ = d_{it}^s + u_{i(t-1)s}^- - u_{its}^-, \; i = 1, 2, \ldots, N, \;\; t = 1, 2, \ldots, T, \;\; s \in \Omega,$$

$$u_{its}^- \leq u_{i0}^- + \sum_{k=1}^{t} d_{ik}^s, \;\; i = 1, 2, \ldots, N, \;\; t = 1, 2, \ldots, T, \; s \in \Omega,$$

$$\sum_{i=1}^{N} r_{ik} x_{ik} \leq a_{kt}, \; k = 1, 2, \ldots, K, \;\; t = 1, 2, \ldots, T,$$

$$x_i^{\min} \gamma_{it} \leq x_{it} \leq x_i^{\max} \gamma_{it}, \; i = 1, 2, \ldots, N, \;\; t = 1, 2, \ldots, T,$$

$$x_{it} \geq 0, \; \gamma_{it} \in \{0, 1\}, \; i = 1, 2, \ldots, N, \;\; t = 1, 2, \ldots, T \in \{0, 1\},$$

$$u_{its}^+ \geq 0, \;\; u_{its}^- \geq 0, \;\; i = 1, 2, \ldots, N, \;\; t = 1, 2, \ldots, T, \; s \in \Omega,$$

kur matematinė viltis (vidurkis) skaičiuojamas visiems scenarijams $s \in \Omega$.

Šis stochastinis uždavinys nesunkiai pakeičiamas ekvivalenčiu deterministiniu uždaviniu. Jeigu scenarijų aibė nėra baigtinė, atitinkamų funkcijų vidurkių skaičiavimas daug sudėtingesnis.

2.4.3. Tiekimo grandžių vadyba

Planuojant ekonominę įmonės veiklą, svarbūs plano komponentai yra nuolatinis ir pakankamas aprūpinimas žaliavomis bei pagamintos produkcijos savalaikis pristatymas užsakovams. Tiekėjų, gamintojų, sandėlių tinklas bei produkcijos paskirstymo tarp jų kanalai, siekiant įsigyti reikalingas žaliavas, paversti jas galutiniais produktais ir pristatyti šiuos produktus užsakovams, sudaro tiekimo grandinę, grandį (supply chain). Planuojant tiekimo grandinę strateginiu požiūriu, reikia nustatyti tinklo konfigūraciją, atskirų tinklo mazgų skaičių, vietą, pajėgumą bei juose naudojamas technologijas. Planuojant tiekimo grandinę taktiniu požiūriu, reikia nustatyti atskirų operacijų nuoseklumą bei žaliavų ir produkcijos srautus tarp tinklo mazgų, įsigyjant žaliavas, gaminant produkciją ir paskirstant produktus.

Strateginė tiekimo grandinės konfigūracija daro įtaką taktinių operacijų efektyvumui bei reikalauja esminių kapitalo resursų, todėl turi ilgalaikį poveikį įmonės veiklai.

Straipsnyje (Shapiro, (2004)) pateikiama tiekimo grandžių valdymo uždavinio matematinė formuluotė. Tegul turimas tiekimo grandinės tinklas $G=(N,A)$, kur N yra tinklo mazgų aibė ir A tinklo jungčių (lankų) aibė. Aibę N sudaro tiekėjų aibė S, gamybos skyrių aibė P ir užsakovų aibė C, t. y. $N=S\cup P\cup C$. Gamybos padaliniai yra sudėtinių dalių gamybos skyriai M, surinkimo skyriai F ir sandėliai W, t. y. $P=M\cup F\cup W$. Kiekviename gamybos skyriuje $i\in M$ bei surinkimo skyriuje $i\in M$ yra tam tikras kiekis atitinkamų mašinų ar mechanizmų N_i. Aibė P apima ir gamybos skyrius, ir mašinas bei mechanizmus juose. Tegul aibė K yra produktų, judančių tiekimo grandine, aibė.

Tiekimo grandinės organizavimo uždavinio (tinko konstrukcijos) sprendinys nurodo, kokius gamybos padalinius reikia pastatyti ir kokiais įrengimais juos aprūpinti. Sukonstruojamas binarinis kintamasis y_i, įgyjantis reikšmę $y_i=1$, jei yra pastatytas atitinkamas gamybos padalinys arba įsigyjamas atitinkamas įrenginys, bei reikšmę $y_i=0$ kitu atveju. Tiekimo grandinės valdymo uždavinio (produkcijos srautų) sprendinys nusako produkto $k\in K$ gamybos ciklą, kelią nuo tiekėjo iki užsakovo. Tegul x_{ij}^k yra produkto k judėjimas iš tinklo mazgo i į mazgą j , kur $(ij)\in A$. Tuomet aprašyto tiekimo grandinės valdymo uždavinio deterministinis matematinis modelis yra toks:

$$\min\left(\sum_{i\in P}c_i y_i+\sum_{k\in K}\sum_{(ij)\in A}q_{ij}^k x_{ij}^k\right), \tag{2.19}$$

kai

$$y\in Y\subseteq\{0,1\}^{|P|}, \tag{2.20}$$

$$\sum_{i\in N}x_{ij}^k-\sum_{l\in N}x_{jl}^k=0,\quad \forall j\in P,\ \ \forall k\in K, \tag{2.21}$$

$$\sum_{i\in N}x_{ij}^k\geq d_j^k,\quad \forall j\in C,\ \ \forall k\in K, \tag{2.22}$$

$$\sum_{i \in N} x_{ij}^k \leq s_j^k, \quad \forall i \in S, \quad \forall k \in K, \tag{2.23}$$

$$\sum_{k \in K} r_j^k \left(\sum_{i \in N} x_{ij}^k \right) \leq m_j y_j, \quad \forall j \in P, \tag{2.24}$$

$$x \in \Re_+^{|A| \times |K|}. \tag{2.25}$$

Čia c_i - statomo padalinio arba įsigyjamo įrenginio *i* vieneto investicinė kaina, q_{ij}^k - produkto *k*, gamybos padalinyje *i* ir/arba transportavimo lanku (*ij*), vieneto kaina.

Tikslo funkciją (2.19) sudaro bendroji investicijų ir produkcijos gamybos kaina, kurią reikia minimizuoti.

Ribojimas (2.20) nusako loginių priklausomybių bei ribojimų aibę Y: $y_i \leq y_j$ visiems $i \in N_j$ ir $j \in P$ arba $j \in F$, t.y. įrenginį $i \in N_j$ galima įsigyti, jei yra pastatytas padalinys *j*. Šis ribojimas nurodo ir kintamojo *y* binariškumą.

Ribojimas (2.21) nusako produkto *k* judėjimą per kiekvieną gamybos mazgą *j*.

Ribojimas (2.22) reikalauja, kad bendras produkto *k* kiekis, nukreipiamas į užsakovo mazgą *j*, būtų nemažesnis už produkto poreikį šiame mazge (d_j^k).

Ribojimas (2.23) reikalauja, kad bendras produkto *k* kiekis, gaunamas iš tiekėjo mazgo *j*, neviršytų žaliavų kiekio šiame mazge (s_j^k).

Ribojimas (2.24) nurodo reikiamus atitinkamo gamybos mazgo pajėgumus. r_j^k - produkto *k* gamybos mazge *j* kaštai. Visų produktų, gaminamų mazge *j*, kaštai neturi viršyti šio mazgo gamybos pajėgumo m_j, jei šis padalinys yra pastatytas ($y_j = 1$). Jei padalinys *j* nėra pastatytas $y_j = 0$, tai šis ribojimas reikalaujas, kad visi kintamieji $x_{ij}^k = 0$ visiems $i \in N$.

Ribojimas (2.25) nusako kintamųjų x_{ij}^k neneigiamumą kiekvienam produktui $k \in K$ kiekviename tinklo lanke $(ij) \in A$.

Dažnai aukščiau aprašytas modelis (2.19)-(2.25) pateikiamas kompaktiškesne forma:

$$\min\left(c^T y + q^T x\right) \tag{2.26}$$

kai

$$y \in Y \subseteq \{0,1\}^{|P|}, \tag{2.27}$$

$$Nx = 0, \tag{2.28}$$

$$Dx \geq d, \tag{2.29}$$

$$Sx \leq s, \tag{2.30}$$

$$Rx \le My, \tag{2.31}$$

$$x \in \Re_{+}^{|A| \times |K|}. \tag{2.32}$$

Čia vektoriai c, q, d ir s yra atitinkamai investavimo kainų, gamybos/transportavimo kainų, poreikių ir žaliavų vektoriai. Matricos N, D ir S nusako atitinkamai ribojimų (2.21), (2.22) ir (2.23) kairiųjų pusių sumavimo koeficientus. Matrica R nusako koeficientus r_j^k, o matricos M įstrižainė – koeficientus m_j.

Uždavinys(2.26)-(2.32) praplečiamas iki stochastinio uždavinio, pakeičiant deterministinius koeficientus atsitiktiniais. Tegul gamybos/transportavimo kaštai, poreikiai, žaliavų kiekiai ir padalinių gamybiniai pajėgumai yra stochastiniai parametrai, kurių skirstiniai yra žinomi. Tuomet tiekimo grandinės valdymo uždavinio tikslo funkcijos minimumas reiškia investicinių išlaidų sumos ir tikėtinų ateityje gamybos ir transportavimo kaštų sumos mažiausią vertę. Dėl minėtų parametrų neapibrėžties kai kurioms šių parametrų realizacijoms poreikiai gali būti nepatenkinami, todėl sukonstruojamas papildoma baudos kaina *h*, nusakanti papildomus kaštus.

Gaunamas tiekimo grandinės valdymo stochastinio uždavinio matematinis modelis:

$$\min_{y} \quad f(y) = c^T y + E\left(Q(y,\xi)\right), \tag{2.33}$$

kai

$$y \in Y \subseteq \{0,1\}^{|P|}, \tag{2.34}$$

kur $Q(y,\xi)$ yra uždavinio:

$$\min_{x,z}\left(q^T x + h^T z\right), \tag{2.35}$$

kai

$$Nx = 0, \tag{2.36}$$

$$Dx \ge d, \tag{2.37}$$

$$Sx \le s, \tag{2.38}$$

$$Rx \le My, \tag{2.39}$$

$$x \in \Re_{+}^{|A| \times |K|}, \tag{2.40}$$

optimalus sprendinys. $\xi = (q, d, s, M)$ yra atsitiktinis vektorius, nusakantis gamybos ir/arba transportavimo kainas, poreikius, žaliavų kiekius ir gamybos pajėgumus. $Q(y,\xi)$ yra antrojo etapo uždavinio (2.35)-(2.40) optimali reikšmė, kuri yra pirmojo etapo kintamojo y ir atsitiktinio vektoriaus $\xi = (q, d, s, M)$ realizacijos (arba scenarijaus) funkcija. Matematinė viltis (vidurkis) tikslo funkcijoje (2.33) ieškoma atsitiktinio dydžio ξ tikimybės skirstinio, kuris pagal sąlygą yra žinomas, atžvilgiu. Kintamasis z ribojime (2.37) ir kaštų komponentas $h^T z$ antrojo etapo tikslo funkcijoje (2.35) nusako baudas, kylančias dėl atitinkamo poreikio nevykdymo. Šią baudą galima interpretuoti kaip iš kitur perkamo produkto kainą (išorinę kainą). Bendresniu atveju išorinė kaina taip pat gali būti atsitiktinis parametras.

2.4.4. Kiti taikomieji uždaviniai

Logistika. Sandėliuojamų produktų ar jų sudėtinių dalių kiekiai priklauso nuo tikimybiškai apibrėžtų poreikių bei sandėliavimo kainų kitimo (Powell, (2001)).

Elektros energijos gamyba. Elektrinės ar kita elektros energijos tiekimo įmonės administracija turi kiekvieną dieną nuspręsti, kiek elektros energijos gaminti nežinant konkretaus energijos poreikio. Elektros energijos poreikis kinta dėl oro sąlygų (patalpų šildymas ar šaldymas, vandens naudojimas ir pan.), dėl būsimų kuro atsargų sutarčių, dėl elektros energijos pardavimo sutarčių (elektros energijos kainų) kitimo, dėl energetinio tinklo būklės, dėl konkuruojančių įmonių darbo (Caroe, (1998)).

Vandens saugyklų valdymas. Vandens saugyklų talpos bei vamzdynai turi būti valdomi atsižvelgiant į daugelį tarpusavyje nesusijusių faktorių: geriamo vandens ir laistymo-drėkinimo poreikis, elektros energijos gamyba hidroelektrinėse, potvynių kontrolė, žuvų srautų reguliavimas ir pan. Kiekvienas šių faktorių negali būti tiksliai nusakytas, nes priklauso nuo klimato ar oro sąlygų neapibrėžtumo, gyventojų skaičiaus ir jų poreikių kitimo (Dupačova, (1991)).

Gamyba. Gaminamų produktų kiekiai priklauso nuo tikimybiškai apibrėžtų poreikių, gamybos bei realizavimo kainų kitimo (Ahmed, (2003)).

Investicijų valdymas. Lėšos gali būti investuojamos įvairias akcijas, obligacijas, fondus, užsienio valiutą, vertybinius popierius, draudimą ir pan. tiksliai nežinant, ar jų vertė kils ar kris, priklausomai nuo gamybos ir visuomenės poreikių ar politinių sprendimų.

Objektų išdėstymas. Žinant tik tikimybinius poreikių įvertinimus, reikia apsispręsti, kur ir kokio pajėgumo gamybos įmones reikia įrengti, kad šie poreikiai būtų patenkinti, o įmonės neprastovėtų.

Atliekų kontrolė. Reikia apsispręsti, kur ir kokio tipo žaliavų perdirbimo įmones reikia pastatyti, kad užtikrintų gruntinio ir atvirojo vandens kokybę, kai gyventojų skaičiaus kitimas ir būsimi gamybos poreikiai gali būti įvertinti tik apytiksliai ir gali priklausyti nuo neapibrėžtų poveikių (liūtys, sausros, avarijos).

Mechanizmų stabilumas. Palydovinė antena turi būti įtvirtinta pakankamai stabiliai signalų šaltinio atžvilgiu, kai atsitiktiniai įvairaus stiprumo vėjo gūsiai gali ją pakreipti. Reikia nuspręsti, kokia jėgų konfigūracija tikslingiausia, atstatant kaip galima greičiau antenos padėtį ir atsižvelgiant į tai, kad papildomi vėjo gūsiai galimi dar neatstačius reikiamos antenos krypties.

Biologinių sistemų analizė. Optimalios gyvūnų veisimo ir priežiūros strategijos parinkimas tam tikroje biosferos nišoje, siekiant „maksimalaus išgyvenimo“, atsižvelgiant į neapibrėžtus klimato, maisto išteklių, gimstamumo, mirtingumo ir pan. pokyčius. Be to, šie pokyčiai dažnai aprašomi tik teoriniu lygiu.

2.5. Dekompozicijos metodai

2.5.1. Blokinis programavimas ir dekompozicija

Daugelis praktinių matematinio programavimo uždavinių turi didelį skaičių kintamųjų ir (arba) ribojimų. Praktiniams uždaviniams spręsti pakanka sukurti įrangą, leidžiančią patikimai spręsti uždavinius su keliomis dešimtimis arba sudėtingiausiu atveju keliais šimtais kintamųjų arba ribojimų. Dauguma metodų, taikomi stochastinio tiesinio programavimo uždaviniams spręsti, pagrįsti dekompozicijos idėja, kurios esmė yra didelių matavimų pradinio uždavinio išskaidymas, gautųjų pagalbinių uždavinių nepriklausomas sprendimas ir šių dalinių sprendinių susiejimas į bendrąjį pradinio uždavinio sprendinį. Tačiau dekompozicijos metodų taikymas susijęs su didelio skaičiaus papildomų kintamųjų ir ribojimų įvedimu. Pirmąkart dekompozicijos idėją tiesinio programavimo uždaviniams suformulavo G. Dancigas (George Bernard Dantzig) ir F. Vulfas (Phil Wolfe). Vėliau šią idėją plėtojo B. Golšteinas, L. Lesdonas, J. Bendersas ir kt. Plačiausiai taikomi Dantcigo-Vulfo, Benderso, vidinio taško, stochastinės dekompozicijos ir kiti metodai. Esminė dekompozicijos metodo savybė yra ta, kad juos kuriant atsižvelgiama į specialią papildomų kintamųjų ir ribojimų matricos struktūrą. Dalis tokių darbų yra skirti pritaikyti dekompozicijos algoritmus speficinio pavidalo, dažniausiai kelių etapų, uždaviniams spręsti, siekiant išspręsti uždavinius su kiek įmanoma daugiau kintamųjų ir ribojimų (Gondzio, (2007)). Tačiau tokiu būdu gautu uždavinių sprendinių tikslumas nėra pakankamai išsamiai nagrinėtas. Reikia pažymėti, kad dekompozicijos algoritmai kelių etapų uždaviniams spręsti nėra trivialiai pritaikomi dviejų etapų uždaviniams spręsti, nes toks jų pritaikymas susijęs su tam tikra algoritmų adaptacija.

Svarbu pažymėti, kad sprendžiant dviejų etapų STP uždavinius, reikia sukurti algoritmus, kurie tiktų bendro pavidalo uždaviniams ir leistų įvertinti gautų sprendinių tikslumą. Dekompozicijos metodų taikymas šiems uždaviniams yra susijęs su būtinumu saugoti kompiuterio atmintyje didelį skaičių papildomų kintamųjų bei ribojimų. Toliau apžvelgime Dancigo-Vulfo ir Benderso dekompozicijos metodus, taikomus bendro pavidalo stochastinio optimizavimo uždaviniams spręsti.

2.5.2. Stulpelių generavimo metodas

1960 m. G. Dancigas ir F. Vulfas sukūrė metodą didelio matavimo uždaviniams su specialios struktūros ribojimų matrica spręsti. Šis metodas pasirodė efektyviausiu sprendžiant uždavinius, kurių matrica yra blokinė-įstrižaininė ir nedidelis kintamųjų skaičius.

Dekompozicijos metodo ypatybė ta, kad sprendžiamas koordinuojamasis (pagrindinis, master) uždavinys, kurio matrica turi palyginti nedidelį eilučių skaičių bei didelį stulpelių skaičių. Sprendžiant pagrindinį uždavinį nereikia turėti visų užduoties stulpelių išreikštu pavidalu. Jie generuojami naudojant simplekso metodą. Toks skaičiavimo būdas vadinamas stulpelių generavimo metodu. Jo esmė tokia. Tegul turimas tiesinio programavimo uždavinys

$$\begin{aligned} &\sum_{j=1}^{n} c_j x_j \to \max \\ &kai \\ &\sum_{j=1}^{n} A_j x_j = b, \end{aligned} \tag{2.40}$$

kur A_j ir b – m-mačiai vektoriai-stulpeliai ($m < n$).

Tarkime, yra žinomas kažkoks leistinasis bazinis sprendinys x_B ir jį atitinkanti matrica iš bazinių vektorių A. Jei bazinis sprendinys buvo rastas atvirkštinės matricos metodu, tai žinomas ir santykinių įvertinimų vektorius $\lambda = \left[\lambda_j\right] = c_x A_x^{-1}$, kur c_x - tikslo funkcijos esamosios bazės koeficientų vektorius.

Siekiant nustatyti leistinojo bazinio sprendinio x_B pagerinimo galimybę, kiekvienam nebaziniam vektoriui A_j apskaičiuojamas įvertinimas

$$\Delta_j = \lambda A_j - c_j = \sum_{i \in I_B} c_j a_{ij} - c_j . \tag{2.41}$$

Jei $\min \Delta_j = \Delta_s < 0$, tai pradinis sprendinys gali būti pagerintas, į bazę įtraukiant kintamąjį x_s. Tačiau, jei yra didelis kiekis nebazinių stulpelių ($n \geq 10^3$), tai dydžio Δ_s radimas, skaičiuojant įvertinimus Δ_j visiems nebaziniams vektoriams (j=1, 2, ..., n) ir po to juos palyginant, reikalauja daug skaičiavimo resursų ir dažnai yra praktiškai neįgyvendinamas. Stulpelių generavimo metodas rodo, kad tai yra nebūtina.

Tegul visi stulpeliai A_s paimami iš kažkokios iškilosios aibės *S*, kuri nusakoma nelygybių ir lygybių sistema. Tuomet vektorius-stulpelis, kurį būtina įtraukti į bazę, nustatomas sprendžiant pagalbinį uždavinį:

$$\lambda A_j - C\left(A_j\right) \to \min, \tag{2.42}$$

kur $C\left(A_j\right)$ – duota vektoriaus A_j funkcija. Priklausomai nuo aibės S struktūros bei funkcijos $C\left(A_j\right)$ tipo parenkamas efektyviausias šio pagalbinio uždavinio sprendimo metodas. Metodas vadinamas stulpeliu generavimo metodu, nes sprendžiant (2.42) uždavinį naudojamas tik nedidelis kiekis stulpelių, generuojamų esant reikalui. Tuomet žymiai sumažėja atminties, naudojamos saugoti tarpinius rezultatus, kiekis, kas turi esminį pranašumą sprendžiant didelio matavimo uždavinius.

2.5.3. Dancigo - Vulfo dekompozicijos metodas

Tegul tiesinio programavimo uždavinys turi ribojimų blokinę - įstrižaininę matricą

$$\begin{bmatrix} A_1 & A_2 & \dots & A_p \\ B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & B_p \end{bmatrix}$$

Eilutė $\left[A_1,\, A_2,\, \dots,\, A_p\right]$ vadinama jungiančiąja, nes ji jungia visus uždavinio kintamuosius, esančius kokiame nors ribojime arba ribojimų grupėje. Šia ribojimų matrica nusakomas tiesinio programavimo uždavinys:

$$\begin{aligned} & \sum_{i=1}^{p} c_i x_i \to \max \\ & kai \\ & \begin{cases} \sum_{i=1}^{m} A_i x_i = b_0, \\ B_i x_i = b_i,\ i = 1, 2, \dots, p, \\ x_i \geq 0. \end{cases} \end{aligned} \tag{2.43}$$

Reikia pastebėti, kad šiuo pavidalu galima užrašyti bet kokį tiesinio programavimo uždavinį. Kai $p = 1$, ribojimai padalijami į du blokus, pavyzdžiui:

$$f = cx \to \max \tag{2.44}$$

kai

$$A_1 x = b_1, \left(m_1 \ ribojimų \right) \tag{2.45}$$

$$A_2 x = b_2, \left(m_2 \ ribojimų \right) \tag{2.46}$$

$$x \geq 0. \tag{2.47}$$

Tegul iškiloji aibė S_2, nusakoma m_2 ribojimų yra aprėžta, t. y. sudaro daugiamatį daugiakampį. Tuomet jai teisinga kraštinių taškų lema.

1 lema. Tegul uždara aprėžta aibė $R = \{x : Ax = b, x \geq 0\}$ - nėra tuščioji ir $x_i,\ i = 1, 2, \ldots, r$ -yra jos kraštiniai taškai. Tuomet bet kurios šios aibės taškas $x \in R$ gali būti išreikštas aibės R kraštinių taškų iškiląja kombinacija

$$x = \sum_{i=1}^{r} \delta_i x_i,\ \ \delta_i \geq 0,\ \ i = 1, 2, \ldots, r,\ \ \sum_{i=1}^{r} \delta_i = 1. \tag{2.48}$$

Ši lema gali būti apibendrinta neaprėžtai aibei.

2 lema. Tegul aibė $R = \{x : Ax = b, x \geq 0\}$ nėra tuščioji. Bet kuris taškas priklauso aibei R tada ir tik tada, kai jis gali būti išreikštas iškiląja kraštinių taškų kombinacija ir tiesine neaprėžtų aibės briaunų krypčių vektorių kombinacija su neigiamais koeficientais:

$$x = \sum_{i=1}^{r} \delta_i x_i, \tag{2.49}$$

kur

$$\begin{aligned} &\sum_{i=1}^{r} \delta_i \varepsilon_i = 1; \\ &\delta_i \geq 0; \\ &\varepsilon_i = \begin{cases} 1, & jei\ x_i - aibės\ R\ kraštinis\ taškas, \\ 0, & jei\ x_i - aibės\ R\ neaprėžta\ briauna. \end{cases} \end{aligned} \tag{2.50}$$

Remiantis 2-ąja lema kiekvienas aibės S_2 elementas gali būti išreikštas jos kraštiniais taškais

$$x = \sum_{j} \delta_i x_j, \tag{2.51}$$

kur

$$\begin{aligned} &\delta_j \geq 0, \\ &\sum_{j=1}^{n} \delta_j = 1, \end{aligned} \tag{2.52}$$

x_j - daugiakampio S_2 kraštiniai taškai.

Pradinis uždavinys (2.44)-(2.47) įgyja naują formą. Iš visų (2.46)-(2.47) sprendinių reikia parinkti tokį, kuris tenkina sąlygą (2.45) ir kuriame tikslo funkcija (2.44) įgyja didžiausią reikšmę. Tikslo funkcija tampa

$$f = \sum_j \left(c_j x_j \right) \delta_j \to \max, \tag{2.53}$$

o ribojimas (2.45)

$$\sum_j \left(A_1 x_j \right) \delta_j = b_1 . \tag{2.54}$$

Pažymėjus

$$A_1 x_j = P, \quad c_j x_j = z_j \tag{2.55}$$

gaunamas naujas uždavinys:

$$\sum_j z_j \delta_j \to \max \tag{2.56}$$

kai

$$\sum_j P_j \delta_j = b_1, \tag{2.57}$$

$$\sum_j \delta_j = 1, \tag{2.58}$$

$$\delta_j \geq 0. \tag{2.59}$$

Šis uždavinys, ekvivalentus pradiniam uždaviniui (2.44)-(2.47), vadinamas pagrindiniu (koordinuojančiu, master). Jis turi tik $m_1 + 1$ ribojimo eilutę, kai pradinis uždavinys turi $m_1 + m_2$ ribojimų eilutes, ir didelį kiekį stulpelių, kurių skaičius lygus aibės S_2 kraštinių taškų skaičiui. Šie stulpeliai gaunami stulpelių generacijos metodu esant reikalui. Tam reikia, pasinaudojus santykinių įvertinimų vektoriumi $\lambda = \left[\lambda_j \right] = c_x A_x^{-1}$, kiekvienam nebaziniam vektoriui apskaičiuoti dydį Δ_j:

$$\Delta_j = \lambda \times \begin{bmatrix} P_j \\ 1 \end{bmatrix} - z_j . \tag{2.60}$$

Vektorius λ rašomas kitu pavidalu $\lambda = \left[\lambda_1, \lambda_0 \right]$, kur vektorius λ_0 atitinka ribojimus (2.57), o skaičius λ_1 - vienintelį ribojimą (2.58). Todėl gaunama nauja išraiška

$$\Delta_j = \lambda_1 P_j + \lambda_0 - z_j = \left(\lambda_1 A_1 - c_j \right) x_j + \lambda_0 . \tag{2.61}$$

Pagal simplekso metodo taisykles, norint nustatyti įtraukiamą į bazę kintamąjį δ_s, reikia minimizuoti dydį

$$\Delta_s = \left(\lambda_1 A_1 - c_j \right) x_s + \lambda_0 . \tag{2.62}$$

Kadangi tiesinio programavimo uždavinio sprendinys yra kraštinis aprėžtos aibės S_2 taškas, tai šis minimizavimo uždavinys ekvivalentus tokiam pagalbiniam uždaviniui:

$$\begin{aligned} &(\lambda_1 A_1 - C)x \to \min \\ &kai \\ &A_2 x = b_2, \\ &x \ge 0. \end{aligned} \tag{2.63}$$

Radus šio uždavinio sprendinį, patikrinama sąlyga $\Delta_s = \left(\lambda_1 A_1 - c_j\right) x_s + \lambda_0 < 0$. Jei ji įvykdyta, tai vektorių x_s naudinga įtraukti į bazę. Vėliau nustatomos vektoriaus P_s, kurį reikia įtraukti į bazę, komponentės

$$P_s = \begin{bmatrix} A_1 x_s \\ 1 \end{bmatrix}, \tag{2.64}$$

ir atitinkamas tikslo funkcijos koeficientas

$$z_s = c x_s \tag{2.65}$$

Šis metodas ypač efektyvus, kai $p > 1$:

$$f = c_1 x_1 + c_2 x_2 + \ldots + c_p x_p \to \max \tag{2.66}$$

kai

$$A_1 x_1 + A_2 x_2 + \ldots + A_p x_p = b_0, \tag{2.67}$$

$$\begin{aligned} &B_1 x_1 = b_1, \\ &B_2 x_2 = b_2, \\ &\ldots \\ &B_p x_p = b_p, \end{aligned} \tag{2.68}$$

$$x_i \ge 0, \ \ i = 1, 2, \ldots, p. \tag{2.69}$$

Tokiam uždaviniui pagalbinis uždavinys yra toks:

$$\begin{aligned} &\sum_{i=1}^{p} \left(\lambda_1 A_i - c_i\right) x_i \to \min \\ &kai \\ &B_i x_i = b_i, \\ &x_i \ge 0, \ \ i = 1, 2, \ldots, p, \end{aligned} \tag{2.80}$$

kuris dėl tikslo funkcijos (2.66) adityvumo ir ribojimų (2.68) nepriklausomumo suskyla į nepriklausomus uždavinius, kurių kiekvienas yra

$$
\begin{aligned}
&(\lambda_1 A_i - c_i)x_i \to \min \\
&kai \\
&B_i x_i = b_i, \\
&x_i \geq 0.
\end{aligned}
\tag{2.81}
$$

Tegul šių uždavinių sprendiniai yra x_i^0, o jų tikslo funkcijų reikšmės $(\lambda_1 A_i - c_i)x_i^0 = f_i^0$.

Jei $\sum_{i=1}^{p} f_i^0 + \lambda_0 < 0$, tai vektorių $x_0 = \{x_i^0(\lambda_1)\}$, $i = 1, 2, \ldots, p$ galima įtraukti į pagrindinio uždavinio bazę. O jei $\sum_{i=1}^{p} f_i^0 + \lambda_0 \geq 0$, tai esamasis sprendinys yra optimalus.

2.5.4. Dekompozicijos algoritmas

Tegul yra žinomas pradinis leistinasis bazinis sprendinys, kurį atitinka santykinių įvertinimų vektorius $\lambda = [\lambda_1, \lambda_0]$. Kiekviena uždavinio sprendimo iteracija susideda iš dviejų etapų.

Pirmasis etapas:

1. Panaudojus ankstesnės iteracijos įvertinimų vektorių λ_1, suformuojamas ir išsprendžiamas pagalbinis uždavinys (2.81). Randama tikslo funkcijos optimali reikšmė ir atitinkami sprendiniai $x_0 = \{x_i^0(\lambda_1)\}$, $i = 1, 2, \ldots, p$.
2. Randamas mažiausias nebazinių kintamųjų įvertinimas

$$
\min \Delta_j = \sum_{i=1}^{p} f_i^0 + \lambda_0. \tag{2.82}
$$

3. Jeigu $\sum_{i=1}^{p} f_i^0 + \lambda_0 \geq 0$, skaičiavimai baigiami ir nustatomas optimalus uždavinio (2.66)–(2.69) sprendinys

$$
X^* = \sum_j X_j \delta_j, \tag{2.83}
$$

kur $\{\delta_j\}$ - ankstesnis pagrindinio uždavinio bazinis sprendinys, o X_j - aibės S_2 kraštinis taškas, atitinkantis bazinį kintamąjį δ_j.

Jeigu $\sum_{i=1}^{p} f_i^0 + \lambda_0 < 0$, tai pereinama prie antrojo etapo.

Antrasis etapas:

1. Formuojamas stulpelis P_0, kurį reikia įtraukti į uždavinio (2.56)-(2.59) bazę:

$$P_o = \sum_{i=1}^{p} A_i x_j^0 \left(\lambda_1 \right) \tag{2.84}$$

2. Vektorius P_0 įtraukiamas į bazę, atliekamas simplekso metodo žingsnis ir randamas naujas bazinis sprendinys $\left\{ \delta_1^{(k)}, \delta_2^{(k)}, \ldots, \delta_m^{(k)}, \delta_o^{(k)} \right\}$ ir naujasis įvertinimų vektorius $\lambda^{(k)} = \left\{ \lambda_1^{(k)}, \lambda_0^{(k)} \right\}$.
3. Pereinama prie naujos iteracijos pirmojo etapo.

Dancigo – Vulfo dekompozicijos metodas yra dviejų lygių algoritmas, kai iš pradžių sprendžiami pagalbiniai uždaviniai, o vėliau – pagrindinis uždavinys. Jei pagrindinis uždavinys nėra išsigimęs, tai kiekvienoje iteracijoje tikslo funkcijos reikšmė didėja. Kadangi bazių skaičius yra ribotas ir nė viena nepasikartoja dukart, optimalus sprendinys randamas per baigtinį iteracijų skaičių.

Uždavinio sprendimo Dancigo – Vulfo dekompozicijos metodu pavyzdys pateiktas 1 priede.

2.5.5. Benderso dekompozicija dviejų etapų stochastinio programavimo uždaviniui

Dviejų etapų stochastinis tiesinis uždavinys gali būti apibrėžtas šitaip:

$$\begin{aligned} &\min_x \left(c^T x + E_\omega Q\left(x, \omega\right) \right) \\ &kai \\ &Ax = b, \\ &x \geq 0, \end{aligned} \tag{2.85}$$

kur

$$\begin{aligned} &Q\left(x, \omega\right) = \min_y d_\omega^T y \\ &kai \\ &T_\omega x + W_\omega y = h_\omega, \\ &y \geq 0. \end{aligned} \tag{2.86}$$

čia E_ω - matematinė viltis, ω žymi elementarųjį įvykį tikimybinėje erdvėje $\left(\Omega, \Sigma, P_x \right)$. Benderio dekompozicijos metodas taikomas uždaviniams su diskrečiaisiais skirstiniais P. Todėl galima užrašyti

$$E_\omega Q\left(x, \omega\right) = \sum_{\omega \in \Omega} p\left(\omega\right) Q\left(x, \omega\right), \tag{2.87}$$

ir formuluoti pradinio uždavinio deterministinį ekvivalentą kaip didelį tiesinio programavimo uždavinį

$$\begin{aligned}
&\min\left(c^T x+\sum_{\omega} p(\omega) d_{\omega}^T y_{\omega}\right)\\
&kai\\
&Ax=b,\\
&T_{\omega}x+W_{\omega}y_{\omega}=h_{\omega},\\
&x\geq 0,\ y_{\omega}\geq 0.
\end{aligned} \tag{2.88}$$

Ribojimas $T_{\omega}x+W_{\omega}y_{\omega}=h_{\omega}$ turi blokinę struktūrą, vadinama L-struktūra

$$\begin{array}{lllllll}
T_1x & +W_1y_1 & & & & = & h_1\\
T_2x & & +W_2y_2 & & & = & h_2\\
T_3x & & & +W_3y_3 & & = & h_3\\
\vdots & & & & & \vdots & \\
T_kx & & & & +W_ky_k & = & h_k
\end{array} \tag{2.89}$$

Benderso algoritmas šiam uždaviniui išspręsti formuluojamas taip:

1 žingsnis: Pradiniai duomenys

$\nu := 1$ (iteracijos numeris)

$UB := \infty$ (viršutinė riba)

$LB := -\infty$ (apatinė riba)

Sprendžiamas pradinis pagrindinis uždavinys (master):

$$\begin{aligned}
&\min c^T x\\
&kai\\
&Ax=b,\\
&x\geq 0
\end{aligned} \tag{2.90}$$

$\bar{x}^{\nu} := x^*$ (optimalus pagrindinio uždavinio sprendinys)

2 žingsnis: pagalbinis uždavinys (subproblem)

kiekvienam $\omega\in\Omega$

sprendžiamas pagalbinis uždavinys:

$$\begin{aligned}
&\min d_{\omega}^T y_{\omega}\\
&kai\\
&W_{\omega}y_{\omega}=h_{\omega}-T_{\omega}\bar{x}^{\nu}\\
&y_{\omega}\geq 0
\end{aligned} \tag{2.91}$$

$\bar{y}_{\omega}^{\nu} := y_{\omega}^*$ (pagalbinio uždavinio optimalus sprendinys)

$\overline{\pi}_\omega^\nu := \pi_\omega^*$ (pagalbinio uždavinio optimalus dualusis sprendinys)

$$UB := \min\left\{UB, c^T\overline{x}^\nu + \sum p_\omega d_\omega^T \overline{y}_\omega^\nu\right\} \tag{2.92}$$

3 žingsnis: konvergavimo testas

Jeigu $\frac{UB-LB}{1+LB} \le \varepsilon$, tai

Pabaiga: reikiamas tikslumas pasiektas

Grąžinamas sprendinys $\overline{x}^\nu$

4 žingsnis: pagrindinis uždavinys

Sprendžiamas pagrindinis uždavinys (master):

$$\begin{aligned}
&\min\left(c^T x + \theta\right)\\
&kai\\
&Ax = b,\\
&\theta \ge \sum_{\omega\in\Omega} p_\omega\left(-\overline{\pi}_\omega^l\left[T_\omega x + W_\omega \overline{y}_\omega^l - h_\omega\right]\right),\ l = 1,2,\ldots,\nu-1,\\
&x \ge 0
\end{aligned} \tag{2.93}$$

$\overline{x}^\nu := x^*$ (optimalus pagrindinio uždavinio sprendinys)

$\overline{\theta}^\nu := \theta^*$

$LB := c^T\overline{x}^\nu + \overline{\theta}^\nu$

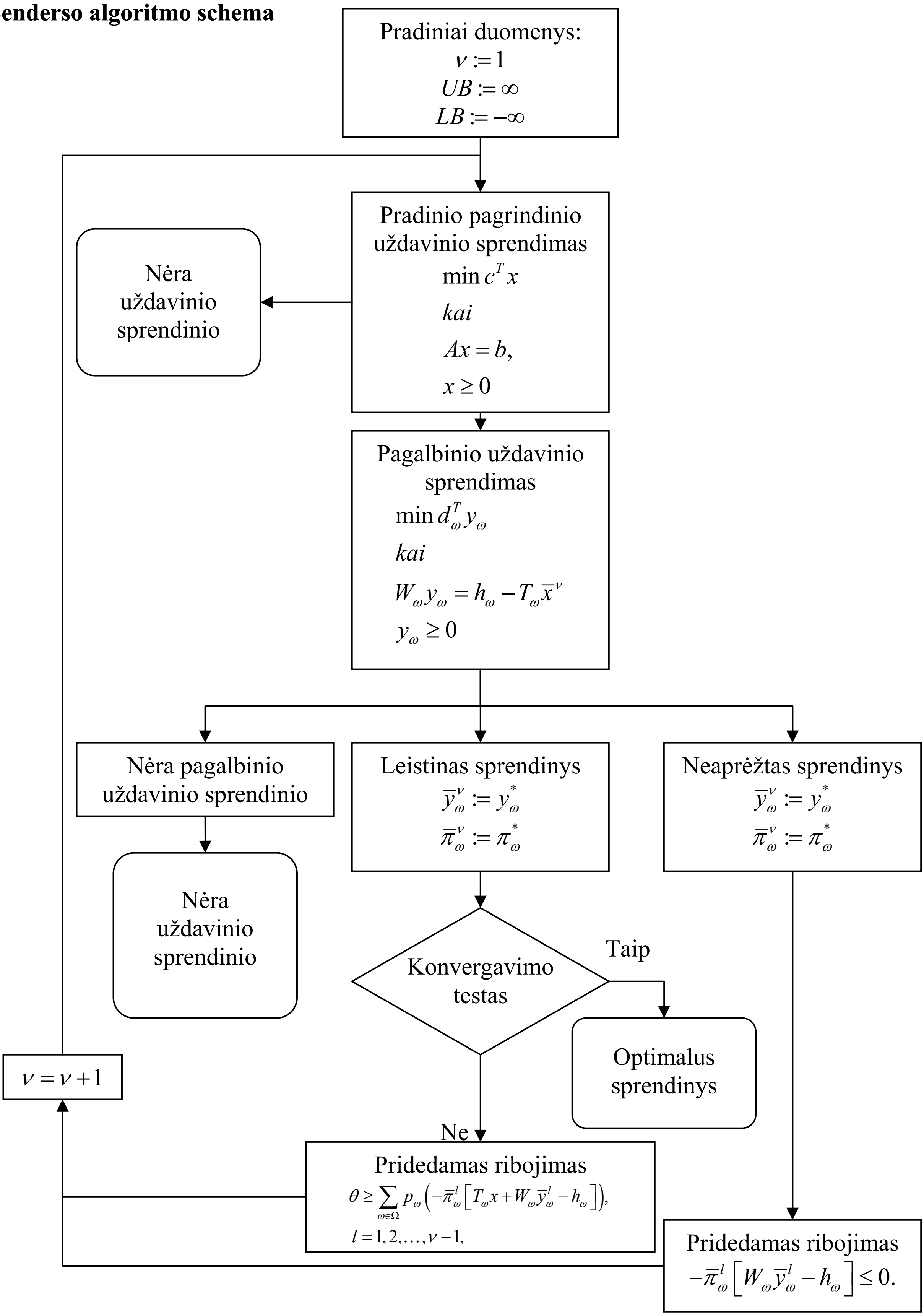
Einama į 2-ąjį žingsnį.

Jei pagalbinis uždavinys neaprėžtas, pagrindiniam uždaviniui reikia pridėti ribojimą $-\overline{\pi}_\omega^l\left[W_\omega \overline{y}_\omega^l - h_\omega\right] \le 0.$

Jei pagalbinis uždavinys neturi sprendinių, pradinis uždavinys taip pat neturi sprendinių.

2.5.6. Benderso algoritmas

Benderso algoritmo schema



2.5.7. Benderso kintamųjų atskyrimo metodas

Tegul nagrinėjamas uždavinys

$$c^T x + f(y) \to \min \tag{2.94}$$

kai

$$Ax + g(y) \geq b \tag{2.95}$$

$$x \geq 0, \ x \in M_y, \tag{2.96}$$

kur *c* ir *x* – n-mačiai vektoriai, *y* – p-matis vektorius, *A* - $m \times n$ matrica, *b* – m-matis vektorius, *f* ir *g* – m-matės funkcijos, M_y - tam tikra aibė erdvėje $\Re^n$.

Kiekvienam fiksuotam y uždavinys (2.94)-(2.96) yra tiesinio programavimo uždavinys, todėl tikslinga atskirti kintamuosius *x* ir *y*. Tegul $P_y \in M_y$ – aibė kintamųjų y reikšmių, su kuriomis tiesinio programavimo uždavinys turi leistinąjį sprendinį.

Fiksuotam y reikia spręsti tiesinio programavimo uždavinį:

$$c^T x \to \min \tag{2.97}$$

kai

$$\begin{aligned} &Ax \geq b - g(y), \\ &x \geq 0. \end{aligned} \tag{2.98}$$

Jam dualus uždavinys:

$$\left(b - g(y)\right)^T \lambda \to \max \tag{2.99}$$

kai

$$\begin{aligned} &A^T \lambda \leq c, \\ &\lambda \geq 0. \end{aligned} \tag{2.100}$$

Kadangi šių uždavinių tikslo funkcijos optimaliosios reikšmės lygios, tai uždavinį (2.94)-(2.96) galima perrašyti:

$$f(y) + \max_{\lambda} \left(b - g(y)\right)^T \lambda \to \min \tag{2.101}$$

kai

$$\begin{aligned} &A^T \lambda \leq c, \\ &\lambda \geq 0. \end{aligned} \tag{2.102}$$

Aibė (2.102) nepriklauso nuo y ir tegul jos kraštiniai taškai yra λ_ν^P, $\nu = 1, 2, \dots, N_P$. Tuomet (2.101) galima perrašyti

$$f(y) + \max_\lambda \left(b - g(y)\right)^T \lambda_\nu^p \to \min \tag{2.103}$$

Gautasis uždavinys ekvivalentus tokiam uždaviniui

$$z \to \min \tag{2.104}$$

kai

$$z \geq f(y) + \left(b - g(y)\right)^T \lambda_\nu^p,\ \nu = 1, 2, \dots, N_P, \tag{2.105}$$

$$\left(b - g(y)\right)^T \lambda_\nu^r \leq 0,\ \nu = 1, 2, \dots, N_r,\ y \in M_y, \tag{2.106}$$

kuris turi didelį ribojimų skaičių. Todėl jam galima taikyti relaksacijos metodą.

Tegul M_y - uždaroji aprėžta aibė, o funkcijos $f(y)$ ir $g(y)$ - iškilosios. Uždavinys (2.104) nagrinėjamas su nedideliu ribojimų skaičiumi:

$$z \to \max \tag{2.107}$$

kai

$$z \geq f(y) + \left(b - g(y)\right)^T \lambda_\nu^p,\ \nu \in I_1, \tag{2.108}$$

$$\left(b - g(y)\right)^T \lambda_\nu^r \leq 0,\ \nu \in I_2,\ y \in M_y \tag{2.109}$$

kur $I_1 \subset [1, N_P]$, $I_2 \subset [1, N_r]$ - indeksų poaibiai.

Tegul (z^0, y^0) - optimalus uždavinio (2.107)-(2.109) sprendinys. Naudojant relaksacijos metodą, reikia patikrinti, ar jis tenkina ir atmestuosius ribojimus (2.105)-(2.106). Tam sukonstruojamas pagalbinis tiesinio programavimo uždavinys:

$$\left(b - g(y)\right)^T \lambda \to \max \tag{2.110}$$

kai

$$\lambda \in M_\lambda = \left\{\lambda : A^T \lambda \leq c,\ \lambda \geq 0\right\}. \tag{2.111}$$

Tegul λ^0 - optimalus uždavinio (2.110)-(2.111) sprendinys kažkuriame aibės M_λ kraštiniame taške. Tuomet būtina patikrinti sąlygą

$$\left(b - g(y^0)\right)^T \lambda_\nu^p \leq z^0 - f(y^0),\ \forall \nu = 1, 2, \dots, N_p. \tag{2.112}$$

Ji bus tenkinama, jeigu bus teisingas optimalumo kriterijus

$$\left(b - g(y^0)\right)^T \lambda^0 \leq z^0 - f(y^0). \tag{2.113}$$

Priešingu atveju, jei tenkinama griežta nelygybė

$$\left(b - g\left(y^0\right)\right)^T \lambda^0 > z^0 - f\left(y^0\right), \tag{2.114}$$

tai šis sąryšis pridedamas prie uždavinio ribojimų. Kiti žingsniai vykdomi pagal bendrąją relaksacijos schemą.

2.5.8. Benderso ir Dancigo - Vulfo dekompozicijos metodų ryšys

Tegul turime tiesinio programavimo uždavinį

$$c^T x + d^T y \to \min \tag{2.115}$$

kai

$$A_1 x + A_0 y \geq b, \tag{2.116}$$

$$x \geq 0,\ y \geq 0 \tag{2.117}$$

ir jam dualųjį uždavinį:

$$b^T \lambda \to \max \tag{2.118}$$

kai

$$A_0^T \lambda \leq d, \tag{2.119}$$

$$A_1^T \lambda \leq c, \tag{2.120}$$

$$\lambda \geq 0. \tag{2.121}$$

Jeigu aibė M_λ yra daugiakampis, tai uždavinys (2.104)-(2.106) uždaviniui (2.115)-(2.117) įgis pavidalą

$$z \to \min \tag{2.122}$$

kai

$$z + \left(A_0 y - b\right)^T \lambda_\nu^p \geq d^T y,\ \nu = 1, 2, \ldots, N_p \tag{2.123}$$

kur $\lambda_\nu^p,\ \nu = 1, 2, \ldots, N_p$ yra daugiakampio M_λ kraštiniai taškai. Benderso metodu parenkamas indeksų ν poaibis. Optimalumo kriterijumi yra lygybė

$$\max\left\{\left(b - A_0 y^0\right)^T \lambda\right\} = z^0 - d^T y^0, \tag{2.124}$$

kai

$$A_\lambda^T \lambda \leq c,\ \lambda \geq 0. \tag{2.125}$$

Tegul uždavinys (2.118)-(1.121) sprendžiamas Dancigo – Vulfo metodu. Kiekvienas daugiakampio M_λ taškas, kurį nusako ribojimai (2.120)-(2.121), gali būti užrašytas kraštinių taškų iškiląja kombinacija

$$\lambda = \sum_{\nu=1}^{N_p} \lambda_\nu \pi_\nu, \ \sum_{\nu=1}^{N_p} \pi_\nu = 1, \ \nu = 1, 2, \ldots, N_p. \tag{2.126}$$

Galima sukonstruoti pagrindinį uždavinį

$$\sum_{\nu=1}^{N_p} \left(b^T \lambda_\nu\right) \pi_\nu \tag{2.127}$$

kai

$$\sum_{\nu=1}^{N_p} \left(A_0 \lambda_\nu\right) \pi_\nu \le d, \tag{2.128}$$

$$\sum_{\nu=1}^{N_p} \pi_\nu = 1, \ \pi_\nu \ge 0, \ \nu = 1, 2, \ldots, N_p. \tag{2.129}$$

Dualusis uždavinys pagrindiniam uždaviniui:

$$d^T y + W \to \min \tag{2.130}$$

kai

$$\begin{gathered} \left(A_0 \lambda_\nu\right)^T y + W \ge b^T \lambda_\nu, \ \nu = 1, 2, \ldots, N_p, \\ y \ge 0. \end{gathered} \tag{2.131}$$

Pakeitus $z = d^T y + W$, gauname uždavinį

$$z \to \min \tag{2.132}$$

kai

$$z + \left(A_0 y - b\right)^T \lambda_\nu^p \ge d^T y, \ \nu = 1, 2, \ldots, N_p \tag{2.133}$$

tiksliai sutampantį su Benderso uždaviniu (2.122)-(2.123).

Naudojant Dancigo – Vulfo metodą nagrinėjamas pagalbinis uždavinys

$$\left(b - A_0 y_0\right)^T \lambda \tag{2.134}$$

kai

$$A_1^T \lambda \le c, \ \lambda \ge 0 \tag{2.135}$$

kur $\{y_0\}$ - uždavinio (2.132)-(2.133) optimalūs dualūs kintamieji. Dancigo – Vulfo metodo optimalumo kriterijus

$$\max_\lambda \left\{ \left(b - A_0 y_0\right)^T \lambda \right\} = W_0 = z_0 - d^T y_0 \tag{2.136}$$

kai

$$A_1^T \lambda \le c, \ \lambda \ge 0 \tag{2.137}$$

visiškai sutampa su Benderso metodo kriterijumi.

Tuo būdu Benderso ir Dancigo – Vulfo metodai yra vienas kitam dualūs.

2.6. Tiesioginiai stochastinio netiesinio optimizavimo metodai

Svarbu pažymėti, kad dviejų etapų stochastinio tiesinio programavimo uždavinio tikslo funkcija, kai atsitiktinis matas tolydusis, o ribojimų matricos yra pilno rango, yra vieno ekstremumo glodžiai diferencijuojama funkcija (Shapiro and Homem de Mello, (1998), Prekopa (2005)). Todėl šį uždavinį galima nagrinėti kaip stochastinio netiesinio programavimo su tiesiniais ribojimais uždavinį. Aptarsime stochastinio netiesinio programavimo metodus, taikytinus šiam uždaviniui spręsti.

Savo struktūra stochastinio programavimo uždavinys primena netiesinio programavimo uždavinį, tačiau tai tik išorinis panašumas. Stochastiniame programavime paprastai neįvykdoma pagrindinė netiesinio programavimo prielaida: neįmanoma tiksliai apskaičiuoti funkcijų $F_i(x,\omega),\ i=1,2,\dots,m$ ir jų išvestinių reikšmes kiekvienam x iš leistinosios srities. Dažniausiai turima informacija apie funkcijų $f_i(x,\omega),\ i=1,2,\dots,m$ reikšmes tam tikroms atsitiktinio parametro ω reikšmėms, gautoms iš atitinkamų bandymų (stebėjimų). Stochastinio netiesinio optimizavimo uždaviniams spręsti dažniausiai taikomi tiesioginiai metodai: stochastinio kvazigradiento projektavimo (atskiru atveju stochastinė aproksimacija) ir Monte Karlo metodas.

2.6.1. Stochastinio kvazigradiento projektavimo metodas

Stochastiniai kvazigradiento metodai yra taikomi spręsti matematinio programavimo uždaviniams su neglodžiomis tikslo funkcijomis bei ribojimų funkcijomis, kai nėra žinoma tiksli informacija apie šias funkcijas arba jų išvestines. Daugumos stochastinio programavimo uždavinių tikslo funkcija yra iškila, bet nėra glodi, nes matematinio vidurkio operacija atliekama su dydžiais, gautais atlikus funkcijos minimizavimo ar maksimizavimo operaciją.

Tarkime reikia minimizuoti iškiląją funkciją

$$F(x_1,x_2,\dots,x_n),$$

esant ribojimams

$$(x_1,x_2,\dots,x_n)\in X\ ,$$

kur aibė X iškila ir uždara. Paprastumo dėlei tarkime, kad tikslo funkcija *F(x)* yra glodžiai diferencijuojama, tačiau tikslias jos ir jos išvestinių reikšmes negalima apskaičiuoti.

Jei žinomi gradiento statistiniai įverčiai, tuomet vietoje tikslių funkcijos *F(x)* gradiento reikšmių galima naudoti šių įverčių vektorius. Stochastiniuose kvazigradiento metoduose naudojami būtent šie atsitiktiniai vektoriai.

Tegul žinoma kažkokia tikimybinė erdvė su elementarių įvykių ω aibe. Nagrinėjama atsitiktinių taškų $x^s(\omega)$, $s = 0,1,\ldots$, nusakomų formule (Ermolyev, (1976)):

$$x^{s+1} = \pi_X\left(x^s - \rho_s\gamma_s\xi^s\right),\ s = 0,1,\ldots, \qquad (2.17)(2.138)$$

seka. Čia π_X - projektavimo į aibę X operatorius, x^0 - laisvai parinktas erdvės E^n taškas, ρ_s - žingsnio ilgis, γ_s - normavimo daugiklis, $\xi^s(\omega)$ - atsitiktinis vektorius, kurio matematinis vidurkis

$$E\left(\xi^s\right) = a_s\widehat{F}_x\left(x^s\right) + b^s,\ s = 0,1,\ldots,$$

kur a_s - teigiamas atsitiktinis dydis, $b^s = \left(b_1^s, b_2^s, \ldots, b_n^s\right)$ - atsitiktinis vektorius, abu priklausantys nuo sekos $\left(x^0, x^1, \ldots, x^s\right)$. Vektorius $\widehat{F}_x\left(x^s\right)$ - funkcijos *F(x)* gradientas taške x^s, t. y. vektorius, kiekvienam z tenkinantis nelygybę

$$F(z) - F\left(x^s\right) \geq \sum_{i=0}^{s}\widehat{F}_x\left(x^i\right)\left(z - x^i\right).$$

Šitaip apibrėžto vektoriaus ξ^s vidurkis išreiškiamas gradiento vektoriumi. Jei $a_s \equiv 1$ ir $b^s \equiv 0$, vektorius ξ^s vadinamas stochastiniu gradientu arba stochastiniu kvazigradientu (2.138) apibrėžta procedūra vadinama stochastinių kvazigradientų projektavimo metodu.

2.6.2. Stochastinės aproksimacijos metodas

Tegul turime uždavinį: minimizuoti funkciją $F(x) = Ef(x,\omega)$.

Pagrindinė stochastinės aproksimacijos metodo idėja: minimizuojant funkciją *F(x)* nusileidimo kryptimi vietoje nežinomo antigradiento $-F_x(x)$ parenkamas funkcijos $f(x,\omega)$ antigradientas, t. y. vietoje įprasto gradiento metodo stochastinėje aproksimacijoje nagrinėjamos iteracinės paieškos procedūros (Robins, (1951)), nusakomos sąryšių

$$x^{s+1} = x^s - \rho_s f_x\left(x^s,\omega\right),\ s = 0,1,\ldots.$$

Jei kokiam tai ω gradientą $f_x(x,\omega)$ dėl kokių tai priežasčių apskaičiuoti sunku, nagrinėjamas metodas, kuriame gradientas apskaičiuojamas skaitiniu būdu

$$x^{s+1} = x^s - \rho_s \sum_{j=1}^{n} \frac{f\left(x^s + \Delta_s e^j, \omega^{s_j}\right) - f\left(x^s, \omega^{s_0}\right)}{\Delta_s} e^j, \; s = 0,1,\ldots,$$

kur e^j - j-tosios ašies ortas, $\omega^{s_v}, v = 0,1,\ldots,n$ - nepriklausomi atsitiktiniai dydžiai, ρ_s - s-tosios iteracijos žingsnio ilgis, Δ_s - poslinkio ašyse dydis. Galima sakyti, kad $\omega^{s_0} = \omega^{s_1} = \ldots = \omega^{s_n} = \omega^s$.

Stochastinės aproksimacijos metodo konvergavimas tiriamas darant prielaidą, kad funkcija *F(x)* turi tolydžias ir aprėžtas antrąsias išvestines. Galima įrodyti, kad esant šiai prielaidai

$$E\left(\sum_{j=1}^{n} \frac{f\left(x^s + \Delta_s e^j, \omega^{s_j}\right) - f\left(x^s, \omega^{s_0}\right)}{\Delta_s} e^j \right) = F_x\left(x^s + V^s \Delta_s\right),$$

kur V^s - tam tikras vektorius, $\left\|V^s\right\| \le const$.

Galima pastebėti, kad stochastinės aproksimacijos metodas yra atskiras stochastinių kvazigradientų projektavimo metodo atvejis, kai

$$\xi^s = \sum_{j=1}^{n} \frac{f\left(x^s + \Delta_s e^j, \omega^{s_j}\right) - f\left(x^s, \omega^{s_0}\right)}{\Delta_s} e^j .$$

Šis metodas apibendrintinas atvejui, kai erdvės E^n matavimų skaičius yra labai didelis ir nusileidimo kryptčiai nustatyti reikia daug laiko bei yra sprendinio ribojimai $x \in X$.

Tuomet apibrėžiamos atsitiktinės kryptys $\beta = \left(\beta_1, \beta_2, \ldots, \beta_n\right)$ su nepriklausomais komponentais, tolygiai pasiskirsčiusiais intervale $\left[-1,1\right]$ ir atsitiktinis vektorius ξ^s nusakomas formule

$$\xi^s = \sum_{k=1}^{K_s} \frac{f\left(x^s + \Delta_s \beta^{s_k}, \omega^{s_k}\right) - f\left(x^s, \omega^{s_0}\right)}{\Delta_s} \beta^{s_k} ,$$

kur β^{s_k}, $k = 1,2,\ldots,K_s$ - nepriklausomos atsitiktinės kryptys s-tojoje iteracijoje, be to, $K_s \ge 1,\ \Delta_s \ge 0$. Jei $K_s = 1$, dydžio ξ^s nustatymui reikia funkciją *F(x)* apskaičiuoti dviejuose taškuose.

Tuomet konstruojama seka, konverguojanti į optimizavimo problemos sprendinį (Kiefer, (1952), Kushner & Yin (2003)):

$$x^{s+1} = \pi_X \left(x^s - \rho_s \sum_{k=1}^{K_s} \frac{f\left(x^s + \Delta_s \beta^{s_k}, \omega^{s_k}\right) - f\left(x^s, \omega^{s_0}\right)}{\Delta_s} \beta^{s_k} \right), \; s = 0,1,\ldots .$$

2.6.3. Monte Karlo metodas

Nagrinėkime netiesinį stochastinio optimizavimo uždavinį:

$$F(x) \equiv Ef(x,\omega) \to \min_{x \in X \subset \Re^n} \tag{2.139}$$

kur tikslo funkcija yra atsitiktinių dydžių funkcijos $f\left(x,\omega\right)$ vidurkis leistinoje srityje $x \in X \subset \Re^n$, kuri yra aprėžta ir uždara tiesinė aibė $X = \left\{ x \middle| Ax = b, x \geq 0 \right\}$. Čia $b \in R^m$, A yra $n \times m$ matrica, $X \neq \varnothing$, $\omega \in \Omega$ yra elementarusis įvykis tikimybinėje erdvėje $\left(\Omega, \Sigma, P_x\right)$, funkcija $f : R^n \times \Omega \to R$ tenkina tam tikras integruotinumo, diferencijuotinumo ir iškilumo sąlygas, matas P_x yra absoliučiai tolydus ir priklauso nuo *x*, t. y., jis gali būti apibrėžtas, įvedus tankio funkciją $p : R^n \times \Omega \to R_+$. Taip pat tarkime, kad galime konstruoti baigtines realizacijų sekas ω ir apskaičiuoti funkcijų *f* ir *p* baigtines reikšmes kiekviename taške $x \in R^n$ bet kuriai $\omega \in \Omega$ realizacijai.

Tarkime, duotas tam tikras pradinis artinys $x^0 \in X \subset \Re^n$, šiame taške sugeneruota tam tikro pradinio ilgio N Monte Karlo imtis, ir apskaičiuoti atitinkami imties įverčiai. Tuomet gradientinės paieškos stochastinė procedūra konstruojama iteraciniu būdu:

$$x^{t+1} = x^t - \rho \cdot \tilde{G}(x^t) \qquad \text{(2.18)(2.140)}$$

čia $\rho > 0$ yra tam tikras daugiklis, reguliuojantis žingsnio ilgį, o $\tilde{G}(x^t)$ – tikslo funkcijos gradiento Monte Karlo įvertis taške x^t. Dažniausiai šis metodas taikomas pasirinkus pakankamai didelio, fiksuoto tūrio imtį. Tačiau pastaruoju atveju, algoritmas garantuoja konvergavimą tik į tam tikrą optimalaus sprendinio aplinką. Be to, nėra aišku, kaip parinkti imties tūrį N bei įvertinti gautų sprendinių tikslumą.

Netiesinio stochastinio programavimo metodų NSP p uždaviniui spręsti mokslinėje literatūroje skiriamas tam tikras dėmesys. T. Homem de Mello (Georgia University of Technology) 1998 m., vadovaujant prof. A. Šapiro, apgynė disertaciją, kurioje remdamasis euristinėmis prielaidomis formulavo imties reguliavimo bei statistinių kriterijų taikymo hipotezei apie gradiento lygybę nuliui tikrinti idėjas. Tačiau šiame darbe nebuvo atsižvelgta į galimybę pasinaudoti sąlygomis tikslo funkcijos pasikliautinojo intervalo ilgiui įvertinti, nebuvo teoriškai išnagrinėtas siūlomo algoritmo konvergavimas, nebuvo atliktas sudaryto algoritmo efektyvumo tyrimas statistinio modeliavimo būdu. Darbo išvados rėmėsi nedidelio pavyzdžio su 10 kintamųjų ir 5 ribojimais sprendimu, beje atliktu nekorektiškai: nagrinėjamo pavyzdžio ribojimų matrica turėjo nulinį stulpelį, nors pagal prielaidas ji turėjo būti pilno rango (Shapiro and Homem de Mello, (1998)). I. Deak (Corvinus University) sukūrė sėkmingos regresijos aproksimacijos techniką (Successive Regression Approximation technique), kurią pritaikė dviejų etapų tiesinio stochastinio programavimo uždaviniams spręsti, kai neapibrėžtis aprašoma tolydžiuoju dėsniu. Jis taip pat sukūrė unikalią dviejų etapų stochastinio programavimo testinių uždavinių duomenų bazę, kuri yra taikoma sukurtiems algoritmams testuoti bei palyginti (žr. 4.3 skyrelį).

2 skyriaus išvados

1. Stochastinio tiesinio programavimo uždaviniai yra plačiai pritaikomi įvairiose srityse: planuojant elektros energijos gamybą, produkcijos gamybą ir transportavimą, personalo valdymą, logistiką, investicijų valdymą, mechanizmų stabilumą, tiekimo grandines, analizuojant chemines, biologines bei visuomenines sistemas.

2. Žinomi stochastinio tiesinio programavimo uždavinių sprendimo metodai remiasi uždavinio suvedimu į determinuotą matematinio programavimo uždavinį, kai atsitiktinių scenarijų erdvė yra diskrečioji ir baigtinė.

3. Jeigu duomenų neapibrėžtis aprašoma absoliučiai tolydžiuoju tikimybiniu dėsniu, tai uždaviniams spręsti taikomi dekompozicijos metodai, besiremiantys tolydžiojo mato diskretizavimu.

4. Diskretizavimo būdu gauti tarpiniai rezultatai turi būti saugomi. Taikant dekompozicijos metodus labai sparčiai auga kompiuterio atminties bei laiko ištekliai didinat diskretizavimo žingsnių skaičių, be to, nevisada galima įvertinti uždavinio sprendimo paklaidą, gaunamą taikant šiuos metodus.

5. Dekompozicijos metodai leidžia išspręsti tiesinio programavimo uždavinį baigtiniu iteracijų skaičiumi, tačiau yra menkai apsaugoti nuo skaičiavimo ir apvalinimo paklaidų, todėl sunkiai pritaikomi uždaviniams su didesniu kintamųjų skaičiumi.

6. Stochastinio netiesinio programavimo ir Monte Karlo metodų taikymas yra susijęs su gauto sprendinio tikslumo įvertinimo bei skaičiavimų apimties sumažinimo problemomis.

3 skyrius. Dviejų etapų stochastinio tiesinio programavimo metodas

3.1. Įvadas

Nagrinėsime dviejų etapų stochastinio tiesinio programavimo uždavinį:

$$F(x) = c \cdot x + E\left\{\min_y [q \cdot y \,|\, W \cdot y + T \cdot x \le h,\ y \in R_+^m]\right\} \to \min\,, \tag{3.1}$$

kai

$$Ax = b, \tag{3.2}$$

čia vektorius h ir matricos W, T gali būti atsitiktiniai.

Jau buvo pažymėta, kad dviejų etapų stochastinio tiesinio programavimo uždavinio tikslo funkcija, kai atsitiktinis matas tolydusis, yra vieno ekstremumo glodžiai diferencijuojama funkcija. (3.1)-(3.2) uždavinio tikslo funkcijos gradientą galima išreikšti per atsitiktinės vektorinės funkcijos matematinę viltį. Iš tikrųjų, remdamiesi dualumu perrašykime tikslo funkciją ekvivalenčiu pavidalu (Ermolyev, (1988), Shapiro, (1998)):

$$F(x) = c \cdot x + E\left\{\max_u [(h - T \cdot x) \cdot u \,|\, u \cdot W^T + q \ge 0, \quad u \in R_+^s]\right\}. \tag{3.3}$$

Remdamiesi matematinės vilties ir maksimumo diferencijavimo taisyklėmis gauname, kad funkcijos F(x) gradientas nusakomas išraiška:

$$\nabla_x F(x) = Eg(x, \xi) \tag{3.4}$$

čia $g(x,\xi) = c - T \cdot u^*$, u^* yra dualaus tiesinio programavimo uždavinio

$$(h - T \cdot x)^T \cdot u^* = \max_u [(h - T \cdot x)^T \cdot u \,|\ u \cdot W^T + q \ge 0, \quad u \in R_+^s] \tag{3.5}$$

3.2. Monte Karlo sekų generavimas

Jei atsitiktiniai vektorius h ir matricos W, T yra diskretieji, tikslo funkciją $F(x)$ galima apskaičiuoti kaip funkcijų $f(x,\cdot)$ svertinį vidurkį bei išspręsti (3.1) uždavinį kaip tiesinio programavimo uždavinį (dažniausiai labai didelį) (Ermolyev, (1988)).

Sprendžiant (3.1) arba (3.2) pavidalo uždavinius tolydžiųjų atsitiktinių kintamųjų atveju daroma prielaida, kad galima konstruoti baigtines atsitiktinio dydžio ξ realizacijų sekas kiekviename taške $x \in D \subset \Re^n$ bei apskaičiuoti atitinkamas funkcijų f ir g reikšmes šioms realizacijoms. Po to nesunku įvertinti Monte Karlo metodu (3.1) uždavinio tikslo funkciją bei jos gradientą, kurie išreiškiami matematinėmis viltimis toje pačioje tikimybinėje erdvėje.

Naudojant didelio tūrio Monte Karlo imtis, kyla klausimas dėl atsitiktinių dydžių patikimumo. Išsamius dažniausiai naudojamų atsitiktinių skaičių generatorių tyrimus atliko David W. Deley (Deley, (1991)). Straipsnyje išnagrinėti atsitiktinių dydžių generatoriai, naudojami programinėse kalbose Pascal ir C. Pateikiami atsitiktinių skaičių pasiskirstymo tyrimo testai (Chi kvadrato, Kolmogorovo-Smirnovo) bei jų taikymo, tiriant keletą dažniausiai naudojamų skaičių generatorių, rezultatai, kurie parodė, kad generuojamų atsitiktinių dydžių sekų savybės nesiskiria nuo iš tikrųjų stochastinio proceso (truly random stochastic process) savybių. Tyrimo rezultatai rodo, kad naudojant 32 bitų procesorių, Pascal kalbos funkcija random leidžia generuoti atsitiktinių dydžių sekas, kurių reikšmės gali pradėti kartotis po $4 \cdot 10^9$ atsitiktinių skaičių generavimo, tačiau atsitiktine tvarka. Analogiško tūrio Monte Karlo imtis patikimai galima generuoti ir su C kalbos funkcija rand(). Darbe buvo atliktas naudojamų atsitiktinių skaičių generatorių testavimas, kuris patvirtino prielaidas apie jų patikimumą. Išsamius atsitiktinių skaičių generatorius atliko I. Deak, kurio tyrimų rezultatas yra sukurta dviejų etapų stochastinio tiesinio programavimo testinių uždavinių duomenų bazė.

Tarkime, kad galima sukonstruoti tam tikro ilgio N Monte Karlo imtį:

$$Y = (y^1, y^2, ..., y^N), \tag{3.6}$$

čia y^i yra nepriklausomi atsitiktiniai dydžiai, pasiskirstę vienodai su tankiu $p(x,\cdot): \Omega \to R_+, x \in D \subset R^n$. Dabar galima įvesti tokius tikslo funkcijos ir jos dispersijos imties įverčius:

$$\tilde{F}(x) = \frac{1}{N}\sum_{j=1}^{N} f(x, y^j), \tag{3.7}$$

$$\tilde{D}^2(x) = \frac{1}{N-1}\sum_{i=1}^{N}\left(f\left(x, y^i\right) - \tilde{F}(x)\right)^2. \tag{3.8}$$

Pastebėsime, kad atvaizdavimas $G: R^n \times \Omega \to R^n$, apskaičiuojamas pagal (3.4), yra tikslo funkcijos stochastinis gradientas, t. y., toks atsitiktinis vektorius, kad $EG(x,w) = \nabla F(x)$. Remdamiesi (3.4) gradientą galime įvertinti ta pačia Monte Karlo imtimi (3.6) be esminių papildomų skaičiavimų :

$$\tilde{g}(x) = \frac{1}{N}\sum_{i=1}^{N} g\left(x, y^i\right),\ x \in D \subset \Re^n. \tag{3.9}$$

Remdamiesi Monte Karlo imtimi galime apskaičiuoti kovariacinę matricą:

$$Z(x) = \frac{1}{N-n}\sum_{i=1}^{N}\left(g\left(x, y^i\right) - \tilde{g}(x)\right)\cdot\left(g\left(x, y^i\right) - \tilde{g}(x)\right)^T, \tag{3.10}$$

kuri toliau bus reikalinga sprendinio tikslumui vertinti.

3.3. Gradientinės paieškos metodas

Aprašysime iteracinę stochastinio optimizavimo procedūrą. Tarkime, duotas tam tikras pradinis artinys $x^0 \in D \subset R^n$, šiame taške sugeneruota tam tikro pradinio ilgio N_0 Monte Karlo imtis (3.6), ir apskaičiuoti atitinkami imties įverčiai (3.7), (3.8), (3.9), (3.10). Tuomet gradientinės paieškos stochastinė procedūra konstruojama iteraciniu būdu:

$$x^{t+1} = x^t - \rho \cdot \tilde{g}(x^t) \qquad (3.11)$$

čia $\rho > 0$ yra tam tikras daugiklis, reguliuojantis žingsnio ilgį.

3.4. Projektavimas į aibę

Gradientinės paieškos procedūros (3.11) eigoje gautas artinys x^{t+1} yra artimesnis (3.1)-(3.2) sprendiniui nei ankstesnės iteracijos metu gautas artinys x^t. Kadangi dviejų etapų stochastinio tiesinio programavimo uždavinys (3.1)-(3.2) turi lygybinius ribojimus, artinys x^{t+1} gali nepriklausyti ribojimų (3.2) sričiai.

Jei artinys nepriklauso ribojimų sričiai, $x^{t+1} \notin D$, tai gautąjį artinį reikia suprojektuoti į ribojimų sritį *D*. Tai atliekama projektavimo operacijos pagalba.

Tarkime, reikia rasti funkcijos $f(x)$ minimumą, kai $x \in D \subset \Re^n$. Turime pradinį artinį $x^0 \in D \subset R^n$ bei funkcijos $f(x)$ gradientą taške x^t: $\tilde{g}(x^t)$. Tuomet gradientinės paieškos procedūrą galima užrašyti:

$$x^{t+1} = \pi_D\left(x^t - \rho \cdot \tilde{g}(x^t)\right), \qquad (3.12)$$

kur $\pi_D(z)$ yra taško *z* projektavimo į aibę *D* operacija (Ermolyev, 1979).

Projektavimo operacija $\pi_D(z)$ turi šias savybes:

- kiekvieną tašką *z* projektavimo operacija atvaizduoja į tašką $\pi_D(z)$, priklausantį aibei *D* $\left(\pi_D(z) \in D\right)$;
- kiekvienam taškui *z* teisinga nelygybė $\left\|x^* - \pi_D(z)\right\|^2 \le \left\|x^* - z\right\|^2$.

Nesunkiai galima įsitikinti, kad projektavimo operacijos $\pi_D(z)$ rezultatu galime laikyti optimizavimo uždavinio

$$\min \|z-x\|^2, \text{ kai } x \in D \tag{3.13}$$

sprendinį, t. y. $\pi_D(z) = \arg\min_{x\in D} \|z-x\|^2$.

Optimizavimo uždavinio (3.13) sprendimo sudėtingumas priklauso nuo aibės *D* tipo.

Jei aibė *D* yra vienmatė atkarpa, t. y. $D = \{x : a \le x \le b\}$, tai

$$\pi_D(z) = \begin{cases} a, & kai \ \ z < a, \\ b, & kai \ \ z > b, \\ z, & kai \ \ z \in D. \end{cases}$$

Jei aibė *D* yra n-matis stačiakampis gretasienis

$$D = \{x = (x_1, x_2, \ldots, x_n) : a_j \le x_j \le b_j,\ j = 1, 2, \ldots, n\}, \text{ tai}$$

$$(\pi_D(z))_j = \begin{cases} a_j, & kai \ \ z_j < a_j, \\ b_j, & kai \ \ z_j > b_j, \\ z_j, & kai \ \ a_j \le z_j \le b_j. \end{cases}$$

Jei aibė *D* duota tiesinių lygčių sistema $Ax = b$, kur A – $m \times n$ matrica, x – n-matis kintamųjų vektorius ir b – m-matis vektorius, taikomas sudėtingesnis projektavimo operatorius:

$$\pi_D(z) = I_n - (AA^T)z(AA^T)^{-1} + Ab(AA^T)^{-1} \tag{3.14}$$

Kur I_n – n-matė vienetinė matrica.

3.5. ε-leistinųjų krypčių metodas

Siekiant išvengti „klajojimo vingiais" (*zigzaging* - angl.) ar „įstrigimo griovyje"(*jamming* – angl.), kiekvienu atveju turi būti įvykdytos tiesinių apribojimų sąlygos. Tuo tikslu naudojamas leistinųjų krypčių metodas.

Tarkime, taškas z priklauso ribojimų sričiai *D*. Nenulinė kryptis *v* vadinama leistinąja, jei einant iš taško *z* šia kryptimi tam tikrą žingsnį, neišeinama iš srities *D*, t. y. pakankamai mažiems $\hat{\rho} > 0$ egzistuoja toks žingsnio ilgis δ, $0 < \rho \le \hat{\rho}$ kad $x + \rho v \in D$

Pavyzdžiui, jeigu sritis D yra skritulys $D = \{x = (x_1, x_2) : x_1^2 + x_2^2 \le 1\}$, tai taške $x = (0,1)$ leistinųjų krypčių aibė yra $V = \{v = (v_1, v_2) : v_2 < 0\}$.

Pateikiame tikslesnį leistinosios krypties apibrėžimą:

Nenulinė kryptis v vadinama leistinąja taške $z \in D$, jei kiek norima mažoje (spindulio ε>0) vektoriaus v aplinkoje egzistuoja kryptis v_{ε} ir skaičius ρ_{ε}, kad taškas $x+\rho_{\varepsilon} v_{\varepsilon} \in D$. (Ermolyev, (1979)).

Tarkime, reikia rasti funkcijos

$$f_0(x) \tag{3.15}$$

minimumą, kai

$$f_i(x) \le 0,\ i, 2, \ldots, m \,. \tag{3.16}$$

Tegul taškas x priklauso ribojimų sričiai ir reikia rasti tokius ρ ir v, kad taškas $x+\rho v$ priklausytų ribojimų sričiai ir kuriame tikslo funkcijos reikšmė būtų mažesnė už $f_0(x)$.

Ribojimų srities (3.16) ribojimai $f_i(x) < 0$, bus įvykdomi taške $x+\rho v$ ir kai žingsnio ilgis ρ yra pakankamai mažas. Todėl konstruojama aktyviųjų ribojimų aibė

$$I(x) = \{i : f_i(x) = 0\} \tag{3.17}$$

Kryptis v leistinoji, jei

$$\left(\left(f_i(x) \right)_x^{'}, v \right) < 0,\ i \in I(x). \tag{3.18}$$

Ši kryptis yra ir tinkamoji, jei

$$\left(\left(f_0(x) \right)_x^{'}, v \right) < 0 \,. \tag{3.19}$$

Norint rasti kryptį v, tenkinančią (3.18) ir (3.19), reikia išspręsti tiesinio programavimo uždavinį:

$$\max\ \rho \tag{3.20}$$

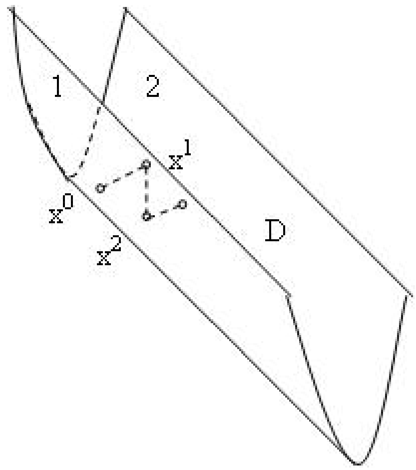
kai

$$\begin{aligned} &\left(\left(f_0(x) \right)_x^{'}, v \right) + r_0 \rho \le 0, \\ &\left(\left(f_i(x) \right)_x^{'}, v \right) + r_i \rho \le 0,\ i \in I(x) \end{aligned} \tag{3.21}$$

kur $r_0, r_i,\ i \in I(x)$ – laisvai parinkti teigiami skaičiai (pavyzdžiui, $r_0 = 1, r_i = 1,\ i \in I(x)$. Kartais pridedama vektoriaus v normavimo sąlyga

$$-1 \le v_j \le 1,\ j = 1, 2, \ldots, n \tag{3.22}$$

Jeigu uždavinio sprendime operuojama tik aktyviųjų ribojimų aibe, neatsižvelgiant į ribojimus, kurie yra beveik aktyvūs (labai mažų dydžių tikslumu), tai judėjimas leistinąja kryptimi gali greitai atvesti prie vadinamojo „griovio“.



1 pav. Ribojimų srities „griovio“ iliustracija

1 pav. pavaizduoti du ribojimų srities D ribojimai 1 ir 2. Pirmajame žingsnyje aktyvus 1 ribojimas, o 2 ribojimas beveik aktyvus. Pirmasis žingsnis greitai atsiremiame į 2 ribojimą. Antrajame žingsnyje aktyvus jau 2 ribojimas, o 1 ribojimas beveik aktyvus. Antrasis žingsnis greitai atsiremia į 1 ribojimą ir t. t. Iteracinio proceso metu vyksta judėjimas smulkiais žingsneliais („zigzagais“).

Siekiant išvengti šitokios situacijos kiekviename iteracijos žingsnyje reikia nagrinėti aktyviųjų ribojimų aibės $I(x)$ plėtinį $I_1(x)$, papildytą beveik aktyvių ribojimų indeksais. Tarkime, yra fiksuotas skaičius $\varepsilon > 0$. Apibrėžiama aibė $I_1(x,\varepsilon) = \{i : -\varepsilon \le f_i(x) \le 0\}$ taške $x \in D$. Tinkamosios krypties nustatymui sprendžiamas (3.20)-(3.21) aibei $I_1(x,\varepsilon)$.

3.6. ε-leistinųjų krypčių metodo taikymas

Aprašysime ε-leistinų krypčių algoritmą uždaviniui (3.1)-(3.2). Jame pasirinktam sprendiniui $x \in X$ nusakoma leistinųjų krypčių aibė:

$$V(x) = \left\{ g \in \Re^n \mid Ag = 0, \forall_{1 \le i \le n} (g_j \ge 0,\ kai\ x_j = 0) \right\} \tag{3.23}$$

Toliau nusakoma vektoriaus g projekcija į tam tikrą sritį Q – g_Q. Tuomet būtinoji optimalumo sąlyga sprendiniui $x \in X$ užrašoma (Bertsekas, (1982)):

$$\nabla F(x)_V = 0 \tag{3.24}$$

Tarkime, duotas tam tikras daugiklis $\hat{\rho} > 0$. Aprašykime funkciją $\rho_x : V(x) \to \Re_+$

$$\rho_x(g) = \min\left\{\hat{\rho},\ \min_{\substack{g_j>0,\\ 1\le j\le n}}\left(\frac{x_j}{g_j}\right)\right\},\ \exists_{1\le j\le n}\left(g_j > 0\right). \tag{3.25}$$

Jeigu $\forall_{1\le j\le n}(g_j \le 0)$, tai $\rho_x(g) = \hat{\rho}$. Tuomet $x + \rho \cdot g \in X$, kai $\rho = \rho_x(g)$ kiekvienam $g \in V$, $x \in X$.

Tegu turime pakankamai mažą dydį $\hat{\varepsilon} > 0$. Tuomet sukonstruojame ε-leistinųjų krypčių aibę

$$V_\varepsilon(x) = \left\{g \in \Re^n \middle| Ag = 0,\ \forall_{1\le i\le n}\left(g_j \le 0,\ if\ \left(0 \le x_j \le \varepsilon_x(g)\right)\right)\right\}, \tag{3.26}$$

kur funkcija $\varepsilon_x : V(x) \to \Re_+$ yra nusakoma

$$\varepsilon_x(g) = \hat{\varepsilon} \cdot \max_{\substack{1\le j\le n\\ g_j>0}}\left\{\min\left\{x_j, \hat{\rho}\cdot g_j\right\}\right\},\ \exists_{1\le j\le n}\left(g_j > 0\right).$$

jeigu $\forall_{1\le j\le n}(g_j \le 0)$, tuomet $\varepsilon_x(g) = 0$.

Stochastinėje paieškos procedūroje ε-leistinųjų krypčių metodas naudojamas tokiu būdu. Tarkime, pradinį paieškos tašką galima rasti sprendžiant deterministinį tiesinį programavimo uždavinį:

$$(x^0, y^0) = \arg\min_{x,y}[c\cdot x + q\cdot y \mid A\cdot x = b,\ W\cdot y + T\cdot x \le h,\ \ y \in R_+^m, x \in R_+^n], \tag{3.27}$$

Gradiento paieškos stochastinės procedūros iteracijos nusakoma sąryšiu:

$$x^{t+1} = x^t - \rho^t \cdot \tilde{G}(x^t), \tag{3.28}$$

kur $\rho^t = \rho_{x^t}(\tilde{G}^t)$ yra iteracijos žingsnio ilgio daugiklis, nusakomas (3.25), ir $\tilde{G}_\varepsilon^t = \tilde{G}(x^t)_{V_\varepsilon\left(x^t\right)}$ yra gradiento įverčio projekcija ε-leistinųjų krypčių aibėje.

3.7. Imties tūrio parinkimas

Iškyla imties (3.6) tūrio parinkimo problema. Kartais imties tūris laikomas pastoviu visoms iteracijoms optimizavimo procese ir parenkamas pakankamai didelis, kad tenkintų reikiamą tikslumą visose iteracijose. Tačiau nėra didelio poreikio optimizavimo pradžioje skaičiuoti įverčius dideliu tikslumu, nes tuomet apskaičiuojama tik apytikslė kryptis optimumo link. Todėl galima imti nedideles imtis optimalios paieškos pradžioje ir tik vėliau, didinant imties tūrį, pasiekti tikslo funkcijos reikšmę reikiamu tikslumu priimant sprendimą apie optimalaus sprendinio radimą (Sakalauskas, (2000), (2002)).

Šia schemą galima realizuoti, jei kiekvienoje kitoje iteracijoje imties tūris yra atvirkščiai proporcingas esamos iteracijos gradiento įverčio normos kvadratui:

$$N^{t+1} \geq \frac{\widehat{\rho} \cdot C}{\rho^t \cdot \left|\tilde{G}_\varepsilon^t\right|^2}, \tag{3.29}$$

čia C>0 yra tam tikra konstanta.

Esant tam tikroms įverčio vidurkio egzistavimo sąlygoms, ši taisyklė garantuoja konvergavimą beveik tikrai į optimalųjį sprendinį. Tarkime, turime pradinę iteraciją $x^0 \in D$ ir $N^0 > 1$, (3.28) ir (3.29) formulės apibrėžia seką $\{x^t, N^t\}_0^\infty$, kai $x^t \in D$, ir egzistuoja dydžiai $\overline{\rho} > 0$, $\varepsilon_0 > 0$, $\overline{C} > 0$ tokie, kad

$$\lim_{t \to \infty} \left| \nabla F\left(x^t\right)_{V_\varepsilon\left(x^t\right)} \right|^2 = 0 \quad (b.t.) \tag{3.30}$$

kai $0 < \widehat{\rho} \leq \overline{\rho}$, $0 < \varepsilon < 1$, $C \geq \overline{C}$. Įrodymas pateiktas (Sakalauskas, 2004).

Žingsnio ilgis ρ gali būti nustatomas eksperimentiniu keliu. Konstantos *C* parinkimas arba stochastinio gradiento normos apskaičiavimo geresnės metrikos radimas reikalauja atskiro tyrimo. Parinkus $C = n \cdot Fish(\gamma, n, N^t - n) \approx \chi_\gamma^2(n)$ (čia $\chi_\gamma^2(n)$ – χ^2 pasiskirstymo su n laisvės laipsnių γ - kvantilis), praktiniam taikymui siūloma tokia imties tūrio reguliavimo taisyklė:

$$N^{t+1} = \min\left(\max\left(\frac{n \cdot Fish\left(\gamma, n, N^t - n\right)}{\rho \cdot \tilde{g}\left(x^t\right)^T \cdot \left(Z\left(x^t\right)\right)^{-1} \cdot \tilde{g}\left(x^t\right)} + n, N_{\min} \right), N_{\max} \right), \tag{3.31}$$

čia $Fish(\gamma, n, N^t - n)$ yra Fišerio pasiskirstymo, turinčio $(n, N^t - n)$ laisvės laipsnių, γ - kvantilis. Gradiento normos įverčio, gauto tokiu būdu, atsitiktinė paklaida neviršija gradiento normos apytikriai su tikimybe $1-\gamma$.

Siekiant išvengti didelių imties tūrio svyravimų, paprastai nurodomi mažiausia ir didžiausia imties tūrio vertės ($N_{\min} \approx 20..100$ ir $N_{\max} \approx 10000..20000$). Pastebėsime, kad $N_{\max}$ taip pat gali būti parinktas pagal ribojimą tikslo funkcijos įverčio pasikliautinojo intervalo ilgiui.

3.8. Statistiniai optimalumo kriterijai

Galimas sprendimas apie optimalaus sprendimo radimą turi būti tiriamas kiekvienoje optimizavimo proceso iteracijoje. Jei visi stacionarūs funkcijos F(x) taškai yra aprėžti, tuomet pasiūlyta procedūra (3.28), (3.31) garantuoja globalų konvergavimą į kokį tai stacionarų tašką.

Kadangi žinome tik tikslo funkcijos ir jos gradiento Monte Karlo įverčius, galime tikrinti tik statistines optimalumo hipotezes. Kadangi šių įverčių stochastinė paklaida priklauso nuo Monte Karlo imties ilgio, sprendimas apie galimą optimalumą gali būti priimtas, jeigu, pirma, nėra pagrindo atmesti hipotezę apie gradiento lygybę nuliui, ir, antra, imties ilgis yra pakankamas įvertinti tikslo funkciją reikiamu tikslumu.

Pastebėkime, kad imties įverčių (3.7) ir (3.9) skirstiniai gali būti aproksimuoti normaliuoju dėsniu remiantis centrine ribine teorema. Todėl būtinąsias pirmos eilės optimalumo (stacionarumo) sąlygas galima patikrinti pritaikius gerai žinomą daugiamatę Hotellingo T^2-statistiką. Optimalumo hipotezė taške x_t gali būti laikoma patvirtinta su tikimybe $1-\gamma$, jei:

$$T_t^2 \equiv \frac{\left(N^t - n\right)\cdot \tilde{g}\left(x^t\right)^T \cdot \left(Z\left(x^t\right)\right)^{-1} \cdot \tilde{g}\left(x^t\right)}{n} \leq Fish\left(\gamma, n, N^t - n\right). \tag{3.32}$$

Toliau galime pasinaudoti asimptotiniu normališkumu dar kartą ir nuspręsti, kad tikslo funkcija yra apskaičiuota tikslumu δ, jeigu jos pasikliautinosios ribos neviršija šio tikslumo:

$$\frac{\eta_\beta \cdot \tilde{D}(x^t)}{\sqrt{N^t}} \leq \delta, \tag{3.33}$$

čia η_β yra normaliojo skirstinio β - kvantilis.

Jei nors viena iš sąlygų (3.32), (3.33) nėra tenkinama, tuomet procedūra kartojama taške, apskaičiuotame pagal (3.28), kur imties (3.6) ilgis yra apskaičiuotas pagal (3.31). Jei abi šios sąlygos yra tenkinamos tam tikroje iteracijoje, tai priežasčių atmesti hipotezę apie optimumo radimą nėra ir, vadinasi, yra pagrindas stabdyti optimizavimą bei priimti sprendimą apie optimumo radimą reikiamu tikslumu. Kaip išplaukia iš ankstesnio skyrelio teoremų, optimizavimas baigsis po baigtinio Monte Karlo imčių skaičiaus generavimo.

3.9. Algoritmai, taikomi antrojo etapo uždaviniams spręsti

Tikslo funkcijos bei jos gradiento įverčiams statistinio modeliavimo būdu gauti reikia daug kartų spręsti antro etapo optimizavimo uždavinį. Optimizavimo algoritmą galima pagerinti atsižvelgiant į tai, kad sprendžiant antrojo etapo uždavinį galima gauti tikslo funkcijos ir jos gradiento įvertį, pritaikius atitinkamą algoritmą, leidžiantį vienu metu gauti dualaus ir tiesioginio uždavinio sprendinį.

Antrojo etapo tiesinio programavimo uždaviniui spręsti sudaryta *Simplekso* programa, derinanti simplekso ir M metodus (Žilinskas K., (2007)). Siekiant sumažinti skaičių apvalinimo paklaidas, M metodas modifikuotas:

1) vietoje labai didelio skaičiaus M naudojama savita laisvųjų kintamųjų įvertinimų užrašymo forma, panaši į kompleksinius skaičius: $\Delta_i = \alpha_i + M\beta_i,\ i = 1, 2, n$;
2) randami ne tik tiesioginiai, bet ir dualieji uždavinio sprendiniai;
3) tikrinamas tiesioginio ir dualaus uždavinių sprendinių tarpusavio suderinamumas.

Uždavinio pradiniai duomenys programai pateikiami formalizuotu pavidalu, nurodant kintamųjų ir ribojimų skaičių, kitus koeficientus pateikiant tam tikra duomenų matrica *D*.

Programa leidžia spręsti didelio matavimo tiesinio programavimo uždavinius su lygybiniais ribojimais, kurių laisvieji nariai gali būti neigiami (didžiausias testuotas uždavinys turėjo 2000 kintamųjų ir 1500 ribojimų).

<u>Algoritmas 3.1.</u> Tiesinio programavimo uždavinio sprendimo algoritmas.

Tikslas: tiesinio programavimo tiesioginio uždavinio bei jam dualaus uždavinio sprendinių ir tikslo funkcijos optimaliosios reikšmės radimas

Pradinės sąlygos (*preconditions*)**:** tiesinio programavimo uždavinio formalizuotas aprašymas.

Galinės sąlygos (*postconditions*)**:** tiesioginio uždavinio sprendinio vektorius, dualaus uždavinio sprendinio vektorius, tikslo funkcijos optimali reikšmė.

1. Inicializuoti M uždavinio matricą, t. y. suformuoti pradinę kintamųjų bazę ir parinkti pradinį leistinąjį sprendinį.
2. **While** leistinasis sprendinys nėra optimalus **do**
   2.1. Rasti sprendžiamąjį stulpelį *sst*.
   2.2. **If** uždavinys neaprėžtas **then** konstatuoti uždavinio neišsprendžiamumą**.**
   2.3. Rasti sprendžiamąją eilutę *seil* ir sprendžiamąjį elementą *sel*.
   2.4. Sukeisti bazinį kintamąjį x_{seil} su laisvąjį kintamąjį x_{sst} vietomis. t. y. suformuoti naują kintamųjų bazę.
   2.5. Atlikti simplekso metodo Žordano žingsnį sprendžiamajam elementui.
   2.6. Apskaičiuoti naująjį leistinąjį sprendinį.
   end while.
3. Apskaičiuoti uždavinio tiesioginį bei dualųjį sprendinius ir tikslo funkcijos reikšmę.

Algoritmas 3.2. Dviejų etapų tiesinio stochastinio programavimo uždavinio sprendimo algoritmas.

Tikslas: rasti dviejų etapų tiesinio stochastinio programavimo uždavinio sprendinį ir tikslo funkcijos reikšmę.

Pradinės sąlygos (*preconditions*)**:** dviejų etapų tiesinio stochastinio programavimo uždavinio formalizuotas aprašymas.

Galinės sąlygos (*postconditions*)**:** dviejų etapų tiesinio stochastinio programavimo uždavinio sprendinio vektorius $x^* = \left(x_1, x_2, \ldots, x_n\right)$, tikslo funkcijos optimali reikšmė $F\left(x^*\right)$.

1. Surašyti uždavinį apibūdinančius parametrus: pirmojo ir antrojo etapų kintamųjų ir ribojimų skaičius n_1, m_1, n_2, m_2, pradinį, minimalų ir maksimalų Monte Karlo imčių ilgius $N_0, N_{\min}, N_{\max}$, tikslo funkcijos tikslumą δ, paieškos žingsnio ilgį α, pasikliautinojo intervalo tikimybę β, optimalumo hipotezės tikimybė $1-\gamma$, maksimalų iteracijų skaičių $iter$.
2. Inicializuoti uždavinio koeficientų matricas A, W, T bei vektorius c, b, q, μ, σ.
3. Nustatyti pradinį gradientinės paieškos tašką $x^0 = \left(x_1^0, x_2^0, \ldots, x_{n_1}^0\right)$ sprendžiant deterministinį uždavinio analogą, kai atsitiktiniai dydžiai pakeičiami jų vidurkiais $h_i = \mu_i,\ i = 1, 2, \ldots, m_1$. (Simplekso programa).
4. Patikrinti pradinį tašką:

   4.1. Patikrinti pirmojo etapo uždavinio ribojimus $Ax = b$ pradiniame taške $x^0 = \left(x_1^0, x_2^0, \ldots, x_{n_1}^0\right)$.

   4.2. **If** pradinis taškas netenkina ribojimų **then** suprojektuoti tašką $x^0 = \left(x_1^0, x_2^0, \ldots, x_{n_1}^0\right)$ į ribojimų sritį D: $\pi_D\left(x^0\right)$. (Algoritmas 3.4).

5. **While** iteracijų skaičius t neviršija maksimalaus dydžio *iter* **do**

   5.1. Patikrinti ribojimus apskaičiuotame taške $x^t = \left(x_1^t, x_2^t, \ldots, x_{n_1}^t\right)$.

   5.2. Apskaičiuoti tikslo funkcijos $f\left(x, y^t\right),\ i = 1, 2, \ldots, N^t$ ir gradiento $g\left(x, y^t\right)$, $i = 1, 2, \ldots, N^t$ reikšmes pagal formules pasinaudojant Monte Karlo imties (3.6) dydžiais. (Algoritmas 3.3).

5.3. Apskaičiuoti tikslo funkcijos ir gradiento įverčius $\tilde{F}\left(x^t\right)$ (3.7) ir $\tilde{g}\left(x^t\right)$ (3.9), gradiento dispersiją $\tilde{D}^2\left(x^t\right)$ (3.8).

5.4. Sudaryti gradiento kovariacijos matricą $Z\left(x^t\right)$ (3.10).

5.5. Suprojektuoti gradientą ε-leistinąja kryptimi $\tilde{g}_\varepsilon^t = \tilde{g}\left(x^t\right)_{V_\varepsilon\left(x^t\right)}$ (3.26).

(Algoritmas 3.5).

5.6. Apskaičiuoti gradiento statistiką T_t^2 (3.32). (Algoritmas 3.6).

5.7. Patikrinti stabdymo sąlygą (3.33):

5.7.1. Apskaičiuoti Fišerio skirstinio kvantilį $Fish\left(\gamma, n_1, N^t - n_1\right)$.

5.7.2. **If** statistika mažesnė už Fišerio kvantilį $T_t^2 \le Fish\left(\gamma, n_1, N^t - n_1\right)$ **and** imties dispersijos pasikliautinosios ribos neviršija tikslumo $\frac{\eta_\beta \cdot \tilde{D}\left(x^t\right)}{\sqrt{N^t}} \le \delta$ **then** gražinti tikslo funkcijos reikšmę $F\left(x^*\right)$ ir uždavinio sprendinį $x^* = \left(x_1, x_2, \ldots, x_n\right)$.

5.8. Apskaičiuoti gradientinės paieškos žingsnio ilgį $\rho_{x^i}\left(g\right)$ (3.25).

(Algoritmas 3.7).

5.9. Rasti sekantį tašką $x^{t+1} = \left(x_1^{t+1}, x_2^{t+1}, \ldots, x_{n_1}^{t+1}\right)$ (3.28)

5.10. Apskaičiuoti Monte Karlo imties ilgį N^{t+1} (3.31).

6. **done**. ◊

<u>Algoritmas 3.3.</u> Dviejų etapų tiesinio stochastinio programavimo uždavinio tikslo funkcijos ir gradiento įverčių bei kovariacijos matricos skaičiavimo algoritmas.

Tikslas: rasti Monte Karlo imties tikslo funkcijos ir gradiento įverčius bei kovariacijos matricą.

Pradinės sąlygos (*preconditions*)**:** pradinis taškas $x^t = \left(x_1^t, x_2^t, \ldots, x_{n_1}^t\right)$, Monte Karlo imties ilgis N^t.

Galinės sąlygos (*postconditions*)**:** dviejų etapų tiesinio programavimo uždavinio tikslo funkcijos įvertis $\tilde{F}\left(x^t\right)$, gradiento įvertis $\tilde{g}\left(x^t\right)$, kovariacijos matrica $Z\left(x^t\right)$.

1. **While** i neviršija Monte Karlo imties ilgio N^t **and** imties dispersijos pasikliautinosios ribos neviršija tikslumo $\dfrac{\eta_\beta \cdot \tilde{D}\left(x^t\right)}{\sqrt{N^t}} \leq \delta$ **do**

   1.1. Sugeneruoti atsitiktinius dydžius $h_i = \sigma_i \cdot Gauss() + \mu_i$, $i = 1, 2, \ldots, m_2$. *Gauss*() yra funkcija, generuojanti standartinio normaliojo skirstino atsitiktinį dydį.

   1.2. Rasti antrojo etapo uždavinio dualųjį sprendinį $u^i = \left(u_1^i, u_2^i, \ldots, u_{m_2}^i\right)$ (Simplekso programa).

   1.3. Apskaičiuoti tikslo funkcijos gradientą $g\left(x^t, u^i\right)$ (3.5).

   1.4. Apskaičiuoti tikslo funkcijos reikšmę $f\left(x^t, u^i\right)$.

2. **done**.

3. Apskaičiuoti tikslo funkcijos įvertį $\tilde{F}\left(x^t\right)$ (3.7).

4. Apskaičiuoti gradiento įvertį $\tilde{g}\left(x^t\right)$ (3.9).

5. Apskaičiuoti dispersijos įvertį $\tilde{D}^2\left(x^t\right)$ (3.8).

6. Sudaryti kovariacijos matricą $Z\left(x^t\right)$ (3.10). ◊

Algoritmas 3.4. Vektoriaus projektavimo į aibę algoritmas.

Tikslas: rasti vektoriaus projekciją aibėje $D = \{x : Ax = b\}$.

Pradinės sąlygos (*preconditions*)**:** pradinis vektorius $z = (z_1, z_2, \ldots, z_n)$, formalizuotas aibės D aprašymas.

Galinės sąlygos (*postconditions*)**:** vektoriaus projekcija aibėje D.

1. Apskaičiuoti matricų sandaugą $A1 = A \cdot A^T$.
2. Rasti matricą $A2$, atvirkštinę matricai $A1$.
3. Rasti projektorių $P1 = I_n - A1 \cdot z \cdot A2$, I_n – n-tosios eilės vienetinė matrica.
4. Rasti projektorių $P2 = A \cdot b \cdot A1$.
5. Sudaryti projektorių $P = P2 + P1$. ◊

Algoritmas 3.5. Gradiento ε-projektavimo algoritmas.

Tikslas: suprojektuoti gradientą ε-leistinosiomis kryptimis $V_\varepsilon(g)$.

Pradinės sąlygos (*preconditions*)**:** pradinis gradientas $g = (g_1, g_2, \ldots, g_n)$, pradinis taškas $x^t = (x_1^t, x_2^t, \ldots, x_{n_1}^t)$, ε-projektavimo parametras.

Galinės sąlygos (*postconditions*)**:** gradiento ε-projekcija .

1. Nustatyti taško $x^t = (x_1^t, x_2^t, \ldots, x_{n_1}^t)$ nulines komponentes ir sudaryti leistinųjų krypčių aibę (aktyvių ribojimų aibę) $V(x)$ (3.23).
2. Suprojektuoti gradientą aktyvių ribojimų aibėje.
3. Sudaryti ε-leistinųjų krypčių aibę (beveik aktyvių ribojimų aibę) $V_\varepsilon(x)$ (3.26)
4. Suprojektuoti gradientą beveik aktyvių ribojimų aibėje. ◊

<u>Algoritmas 3.6.</u> Hotelingo T^2-statistikos skaičiavimo algoritmas.

Tikslas: rasti gradiento Hotelingo T^2-statistiką.

Pradinės sąlygos (*preconditions*)**:** gradientas $g=\left(g_1,g_2,\ldots,g_n\right)$, kovariacijos matrica $Z\left(x^t\right)$.

Galinės sąlygos (*postconditions*)**:** Hotelingo T^2-statistika.

1. Rasti matricą $Z1$, atvirkštinę kovariacijų matricai $Z\left(x^t\right)$.

2. Apskaičiuoti Hotelingo T^2-statistiką (3.33). ◊

<u>Algoritmas 3.7.</u> Gradientinės paieškos žingsnio skaičiavimo algoritmas.

Tikslas: rasti kitą gradientinės paieškos tašką .

Pradinės sąlygos (*preconditions*)**:** pradinis taškas $x^t=\left(x_1^t,x_2^t,\ldots,x_{n_1}^t\right)$, gradientas

$$g=\left(g_1,g_2,\ldots,g_n\right).$$

Galinės sąlygos (*postconditions*)**:** kitas gradientinės paieškos taškas $x^{t+1}=\left(x_1^{t+1},x_2^{t+1},\ldots,x_{n_1}^{t+1}\right)$.

1. Rasti gradientinės paieškos žingsnio ilgį $\rho_{x^i}\left(g\right)$ (3.25).

2. Apskaičiuoti sekantį galimą tašką $x^{t+1}=\left(x_1^{t+1},x_2^{t+1},\ldots,x_{n_1}^{t+1}\right)$ (3.28)

2. **If** taškas nepriklauso ribojimų sričiai D **then** suprojektuoti tašką į sritį D (3.12). ◊

3 skyriaus išvados

1. Sudaryta Monte Karlo įverčių sistema stochastinio tiesinio programavimo uždavinio tikslo funkcijai ir gradientui įvertinti.

2. Sudaryta simplekso algoritmo modifikacija, kuri leidžia išspręsti antrojo etapo optimizavimo uždavinį ir rasti tikslo funkcijos gradiento nepaslinktą įvertį.

3. Sudarytas ε-projektavimo algoritmas stochastiniam gradientui reguliarizuoti.

4. Pasiūlytas metodas dviejų etapų stochastinio tiesinio programavimo uždaviniui spręsti, naudojant baigtines Monte Karlo imčių sekas ir sudarytas jį realizuojantis algoritmas.

5. Įvesta Monte Karlo imčių ilgio parinkimo taisyklė, kai imties ilgis parenkamas atvirkščiai proporcingai gradiento įverčio normos kvadratui.

6. Sudaryta taisyklė garantuoja uždavinio sprendinio konvergavimą beveik tikrai ir leidžia sumažinti skaičiavimų apimtį.

7. Įvesta procedūra statistinei hipotezei apie gauto sprendinio optimalumą patikrinti naudojant statistinius kriterijus.

4 skyrius. Dviejų etapų stochastinio programavimo algoritmų tyrimas

4.1. Stochastinio gradiento įverčio tyrimas

Sprendžiant dviejų etapų stochastinio tiesinio programavimo uždavinį, reikia įvertinti tikslo funkciją bei jos stochastinį gradientą. Kadangi antrojo etapo tikslo funkcija yra stochastinė, gradientui rasti naudotini stochastinio diferencijavimo metodai. Kita vertus, kadangi dviejų etapų STP uždavinio tikslo funkcija yra unimodalinė ir tolydžiai diferencijuojama, ekstremumo aplinkoje ji yra artima kvadratinei funkcijai. Todėl tiriant sudaryto diferencijavimo ir optimizavimo algoritmo efektyvumą galima pasinaudoti testinių uždavinių su tokio pavidalo tikslo funkcijomis tyrimų rezultatais.

4.1.1. Stochastinio diferencijavimo metodai

Panagrinėsime stochastinio diferencijavimo metodus:

- tiesioginio diferencijavimo metodas (differentiation of the integral), kai diferencijuojama pointegralinė funkcija,
- baigtinių skirtumų metodas (finite difference),
- modeliuojamojo pokyčio metodas (simulated perturbation stochastic approximation),
- tikėtinumo santykio metodas (likelihood ratio).

Pirmuoju metodu gradiento įvertis randamas tiesiogiai diferencijuojant tikslo funkcijos sumos narius kintamojo y atžvilgiu

$$g^1(x,y) = \nabla_x f\left(x,y\right)$$

Baigtinių skirtumų metodu kiekviena gradiento įverčio i-toji komponentė randama pagal formulę

$$g_i^2(x,y) = \frac{f(x+\delta\cdot\varsigma_i, y) - f(x,y)}{\delta},$$

kur ς_i yra vektorius, kurio i-toji komponentė lygi 1, o kitos komponentės lygios 0, δ mažas teigiamas dydis.

Modeliuojamojo pokyčio metodu stochastinio gradiento įvertis randamas pagal formulę

$$g^3(x,y) = \frac{\left(f(x+\delta\cdot\nu, y) - f(x-\delta\cdot\nu, y)\right)\cdot\nu}{2\cdot\delta}$$

kur ν yra atsitiktinis vektorius, įgyjantis reikšmes 1 arba -1 su tikimybe p=0.5, δ mažas teigiamas dydis.

Tikėtinumo santykio metodu gradiento įvertis apskaičiuojamas pagal formulę

$$g^4(x,y) = \frac{\left(f(x+d\cdot y) - f(x+d\cdot E\xi)\right)\cdot y}{d},$$

jei $f(x,y) = f(x+d\cdot y)$, $y \sim N(0, d\cdot I_n)$.

4.1.2. SNP tikslo funkcijos gradiento tyrimas

Panagrinėsime testinį uždavinį, kurio tikslo funkcija yra artima kvadratinei minimumo aplinkoje:

$$F(x) \equiv Ef_0(x,\xi) \to \min_{x\in\Re^n}, \tag{4.1}$$

kur

$$f_0(y) = \sum\nolimits_{i=1}^{n} (a_i y_i^2 + b_i \cdot (1-\cos(c_i \cdot y_i))), \tag{4.2}$$

y_i yra normaliai pasiskirstę atsitiktiniai dydžiai, turintys atitinkamus skirstinius $N(x_i, d^2)$, d=0.5, a_i yra atsitiktiniai dydžiai, tolygiai pasiskirstę intervale [2, 5], b_i - intervale [1, 2] bei c_i - intervale [-0.5, 0.5] ir $2 \le n \le 100$.

Yra įrodyta, jog T^2-statistikos, skaičiuojamos nepriklausomiems, vienodai pasiskirsčiusiems (nebūtinai pagal Gauso pasiskirstymo dėsnį) vektoriams, skirstinys asimptotiškai konverguoja į Fišerio skirstinį su n, N-n laisvės laipsnių, kai imties tūris didėja (Fujikoshi, (1997)). Darbe buvo atlikti tyrimai, siekiant nustatyti imties minimalų tūri, kuomet galima pasinaudoti asimptotiniu T^2-statistikos skirstiniu bei įvertinti optimalumo testo, besinaudojančio Fišerio kriterijumi, patikimumą.

Tyrimo metu buvo generuojamos įvairaus ilgio Monte Karlo imtys, N = (50, 100, 200, 500, 1000) bei apskaičiuojama kiekvienos imties T^2-statistika (3.32) žinomame funkcijos optimaliajame taške $x_+ = 0$, panaudojant nagrinėjamus statistinio gradiento įverčio skaičiavimo metodus. ω^2 ir Ω^2 kriterijais buvo tikrinama hipotezė, ar šios statistikos empirinis skirstinys sutampa su Fišerio skirstiniu.

1 ir 2 lentelėse pateikiamos apskaičiuotos ω^2 ir Ω^2 kriterijų reikšmės, naudojant tiesioginio diferencijavimo metodą . Esant tikimybei p=0.05, ω^2 kriterijaus kritinis dydis lygus 0,46, o Ω^2 kriterijaus kritinis dydis - 2,49. Statistikų reikšmės, viršijančios kritinius dydžius yra paryškintos.

1 lentelė. ω^2 kriterijaus reikšmės priklausomybė nuo kintamųjų skaičiaus ir Monte Karlo imties ilgio

n \ N	**50**	**100**	**200**	**500**	**1000**
2	0.30	0.24	0.10	0.08	0.04
3	0.37	0.12	0.09	0.06	0.04
4	0.19	0.19	0.13	0.08	0.04
5	**0.75**	0.13	0.12	0.08	0.06
6	**1.53**	0.34	0.10	0.10	0.08
7	**1.56**	0.39	0.13	0.08	0.09
8	**1.81**	0.42	0.27	0.18	0.10
9	**4.18**	**0.46**	0.26	0.20	0.12
10	**8.12**	**0.56**	**0.53**	0.25	0.17

2 lentelė. Ω^2 kriterijaus reikšmės priklausomybė nuo kintamųjų skaičiaus ir Monte Karlo imties ilgio

n \ N	**50**	**100**	**200**	**500**	**1000**
2	**2.57**	1.14	0.66	0.65	0.42
3	**2.78**	0.82	0.65	0.60	0.27
4	**3.75**	1.17	0.79	0.53	0.31
5	**4.34**	1.46	0.85	0.64	0.36
6	**8.31**	2.34	0.79	0.79	0.76
7	**8.14**	**2.72**	1.04	0.52	0.45
8	**10.22**	**2.55**	1.87	0.89	0.52
9	**20.86**	**2.59**	1.57	1.41	0.78
10	**40.57**	**3.69**	**3.51**	1.56	0.98

Kompiuterinio tyrimo rezultatai rodo, kad Hotelingo statistikos skirstinys gali būti aproksimuojamas Fišerio skirstiniu, parinkus pakankamą Monte Karlo imties tūrį. Tokiu atveju pirmos rūšies statistinė klaida atitinka teorinę reikšmę, išplaukiančią iš stabdymo statistikos pasiskirstymo pagal Fišerio dėsnį. 3 lentelėje pateikta pakankamo Monte Karlo imties ilgio priklausomybė nuo kintamųjų skaičiaus, esant patikimumui α =0,95.

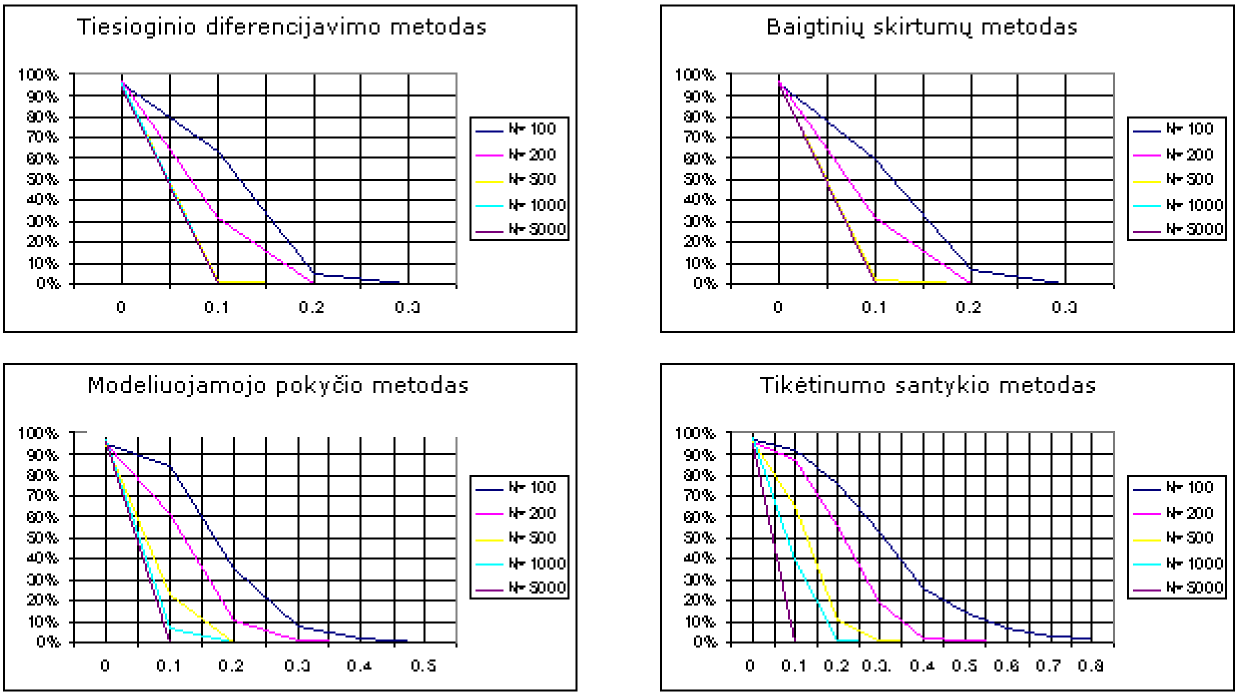
3 lentelė. Pakankamo Monte Karlo imties ilgio priklausomybė nuo kintamųjų skaičiaus

Kintamųjų skaičius	Monte Karlo imties ilgis
10	300
20	1000
40	2200
60	3300
80	4500
100	6000

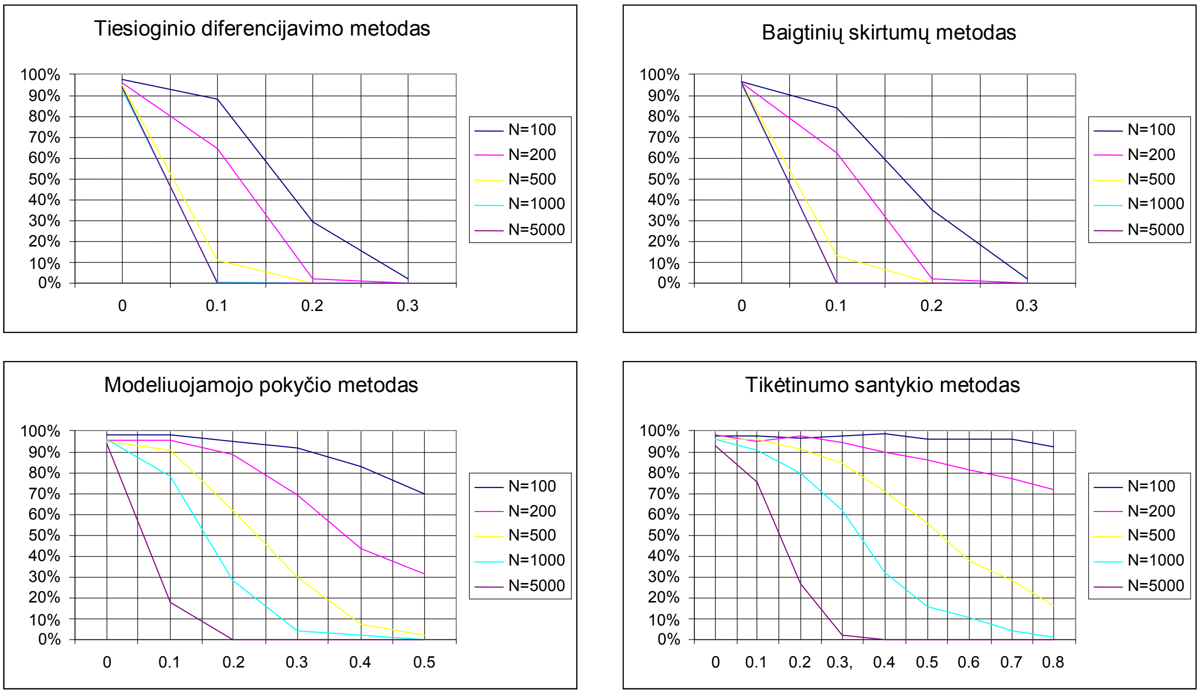
Panašūs rezultatai gaunami ir panaudojus kitus nagrinėjamus stochastinio diferencijavimo metodus. Atliktas tyrimas skirtas patikrinti įvesto kriterijaus patikimumą, tikrinant optimalumo hipotezę, kai ji yra teisinga (pirmos rūšies statistinės klaidos tyrimas).

4.2. Optimalumo hipotezės kriterijaus patikimumo tyrimas

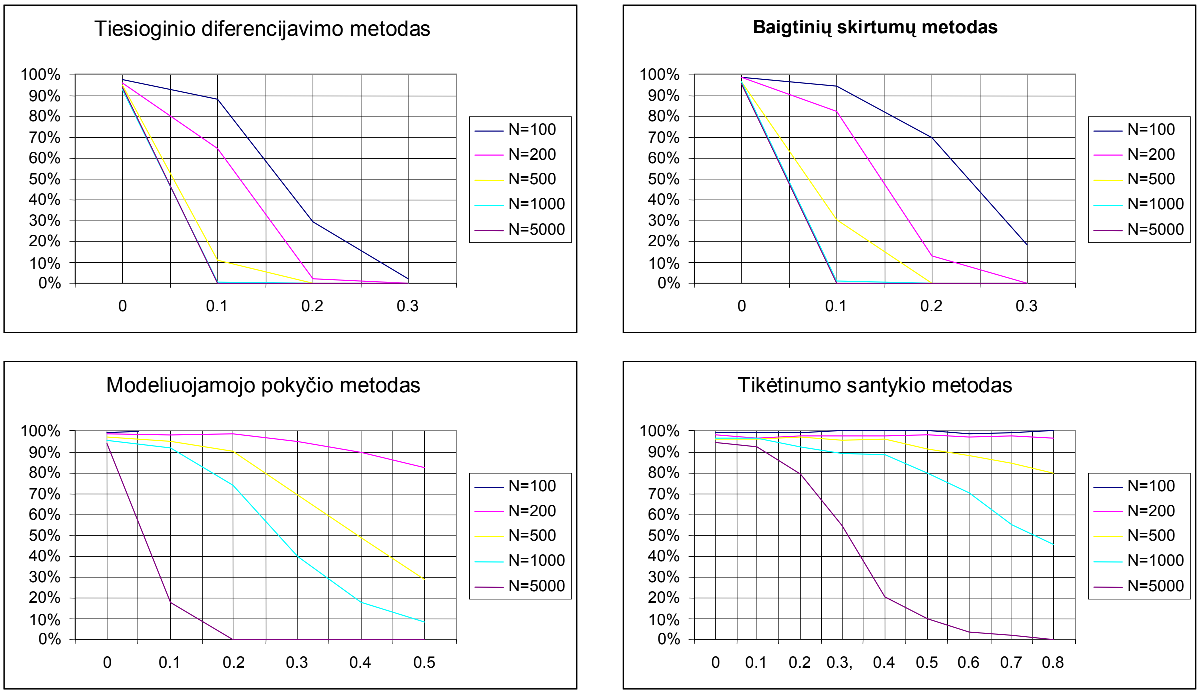
Toliau buvo tiriama tikimybė priimti klaidingą hipotezę (antros rūšies statistinės klaidos tikimybė). Buvo tiriama (4.1)-(4.2) uždavinio optimalumo hipotezės (stochastinio gradiento lygybės nuliui) dažnio priklausomybė nuo kintamųjų skaičiaus optimalaus sprendinio aplinkoje (sprendinio aplinkos spindulys $r = \left|x - x^{+}\right|$). Tyrimo rezultatai, kai kintamųjų skaičius yra $n = 2, 10, 20, 50, 100$ bei patikimumas yra α =0,95, pateikiami 2-6 pav.



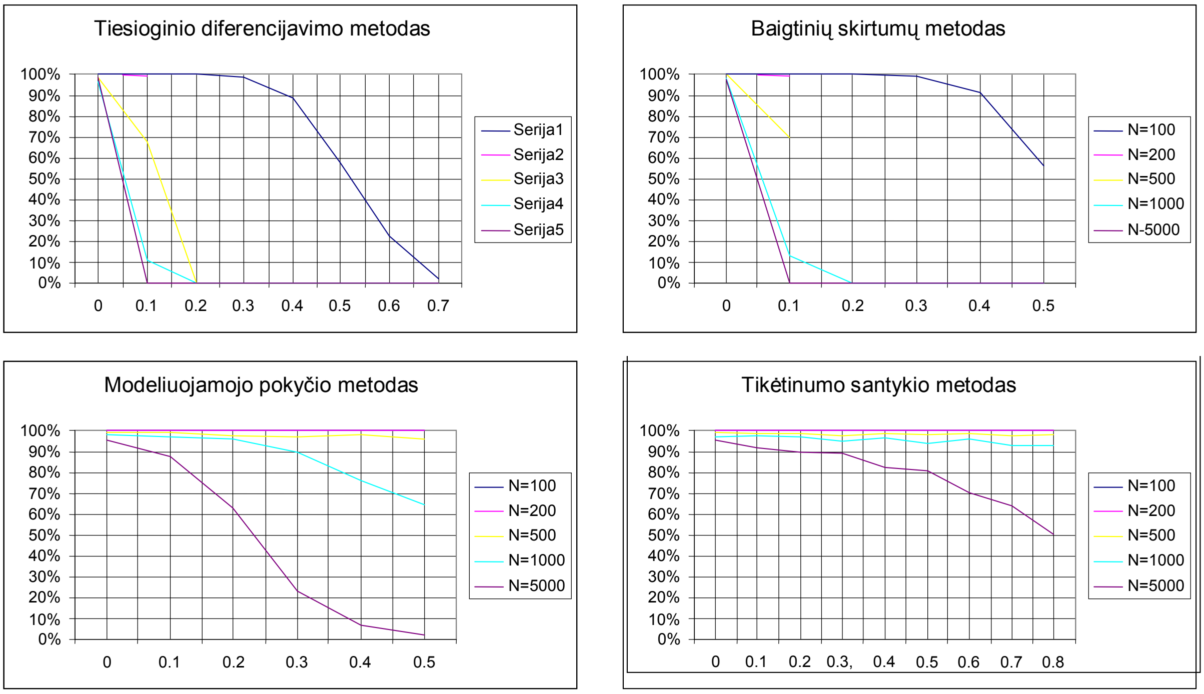
2 pav. Optimalumo hipotezės dažnis, kai kintamųjų skaičius n=2



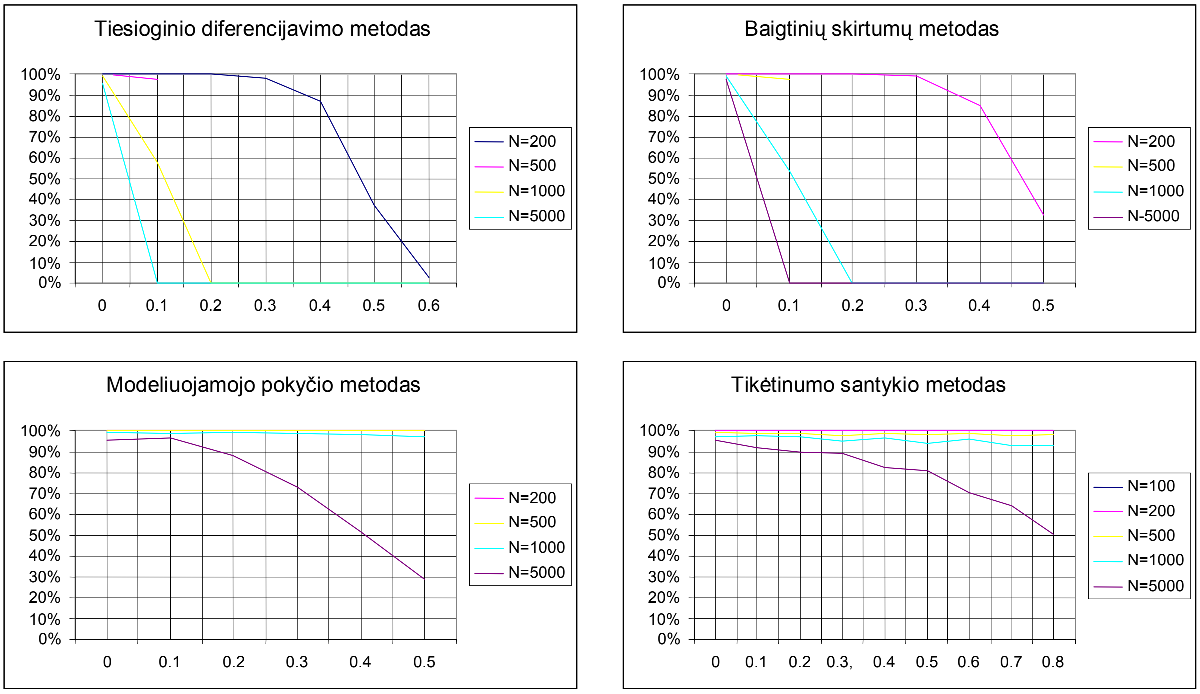
3 pav. Optimalumo hipotezės dažnis, kai kintamųjų skaičius n=10



4 pav. Optimalumo hipotezės dažnis, kai kintamųjų skaičius n=20



5 pav. Optimalumo hipotezės dažnis, kai kintamųjų skaičius n=50



6 pav. Optimalumo hipotezės dažnis, kai kintamųjų skaičius n=100

Iš tyrimų rezultatų aiškėja, kad stochastinio gradiento įverčių, gautų tiesioginio diferencijavimo bei baigtinių skirtumų metodais, Hotelingo statistika leidžia pakankamai patikimai patikrinti optimalumo hipotezę. Šie metodai gali būti taikomi optimalumui testuoti stochastinio optimizavimo uždaviniuose su tikslo funkcija, artima kvadratinei optimumo aplinkoje ir kai hesiano nuosavos reikšmės skiriasi nedaug, o kintamųjų skaičius pakankamai didelis $2 \le n \le 100$. Pastaruoju atveju klaidingai hipotezei atmesti reikalinga tikimybė gali būti užtikrinta atitinkamai parinkus imties tūrį. Kita vertus, įverčių, gautų tiesioginio diferencijavimo ir baigtinių skirtumų metodais, panašumas patvirtina tiesioginio diferencijavimo algoritmo realizavimo korektiškumą.

Modeliuojamojo pokyčio ir tikėtinumo santykio metodai gali būti taikomi testuoti stochastinio optimizavimo uždavinius, kuriuose kintamųjų skaičius nėra didelis $2 \le n \le 20$. Panašios išvados gautos kompiuterinio modeliavimo būdu sprendžiant dviejų etapų STP uždavinį (1 uždavinys) (žr. taip pat (Sakalauskas, Žilinskas, (2006)).

Kadangi baigtinių skirtumų metodas gali pareikalauti daug funkcijos skaičiavimų patogiau naudoti tiesioginio diferencijavimo metodą.

4.3. Testinių tiesinio stochastinio programavimo uždavinių sprendimas

Nagrinėti dviejų etapų tiesinio stochastinio programavimo uždaviniai, turintis pirmame etape n_1 kintamųjų bei m_1 ribojimų ir antrame etape n_2 kintamųjų bei m_2 ribojimų. Duomenys paimti iš internetinės duomenų bazės adresu http://www.math.bme.hu/~deak/twostage/. (nuoroda kreiptasi 2006-01-20).

Dviejų etapų stochastinio tiesinio programavimo užduočių duomenų bazę 2004 m. rugsėjo mėn. sukūrė Budapešto Technologijos ir ekonomikos universitetas (Budapest University of Technology and Economics, Hungary), koordinatorius profesorius Ištvanas Deakas (Istvan Deak), (Deak, (1990)). Duomenų bazėje pateikti sugeneruoti dviejų etapų stochastinio tiesinio programavimo uždaviniai, sugrupuoti pagal pirmojo ir antrojo etapo uždavinių ribojimų ir kintamųjų skaičių. Greta uždavinių sąlygų pateikti jų apytiksliai sprendiniai (together with their approximate solutions). Uždavinių generavimui panaudotos J. Mayer idėjos, aprašytos publikacijoje (Mayer, (1998)).

Uždaviniai pateikiami standartine forma:

$$
\begin{aligned}
&\min\left(cx + E(qy)\right),\\
&kai\\
&Ax = b,\\
&Tx + Wy = h,\\
&x \ge 0,\ y \ge 0,
\end{aligned}
$$

kur h – atsitiktinis vektorius, kurio komponentės nepriklausomi normalinio pasiskirstymo atsitiktiniai dydžiai $h_i \sim N\left(\mu_i, \sigma_i^2\right)$ bei vidurkiai μ_i ir σ_i^2 duoti. Kitos matricos ir vektoriai yra determinuoti, jų komponentų reikšmės pateiktos sąrašu: A – $m_1 \times n_1$ matrica, T – $n_1 \times m_2$ matrica, W – $m_2 \times n_2$ matrica, c – n_1-matis vektorius, b – m_1-matis vektorius, q – n_2-matis vektorius, μ – m_2-matis vektorius, σ^2 – m_2-matis vektorius.

Uždaviniai suskirstyti į atskiras bibliotekas pagal pirmojo etapo uždavinio kintamųjų skaičių n_1 ir antrojo etapo ribojimų skaičių m_2, t. y. pagal matricos T matmenis. Duomenų bazėje yra šios bibliotekos:

L1: 20×20, $n_1 = 20$, $m_2 = 20$;

L2: 40×40, $n_1 = 40$, $m_2 = 40$;

L3: 60×60, $n_1 = 60$, $m_2 = 60$;

L4: 80×80, $n_1 = 80$, $m_2 = 80$.

Kiekvieno duomenų bazės uždavinio struktūra:

a) matricų nenulinių komponentų ir vektorių visų komponentų reikšmės;
b) leistinasis pradinis pirmojo etapo uždavinio sprendinys;
c) apskaičiuotas apytikslis uždavinio sprendinys;
d) apskaičiuota tikslo funkcijos reikšmė ir jos pasikliautinasis intervalas su tikimybe 90%.

1 uždavinys

Dviejų etapų tiesinis stochastinio programavimo užduoties pirmojo etapo uždavinys turi 20 kintamųjų ir 10 ribojimų, o antrojo etapo uždavinys – 30 kintamųjų ir 20 ribojimų. Duomenys pateikti duomenų bazės forma:

```
         THE DIMENSIONS OF THE TWO-STAGE PROBLEM:
         FIRST STAGE HAS   10 ROWS AND   20 VARIABLES
         SECOND STAGE HAS  20 ROWS AND   30 VARIABLES
         THE NUMBER OF INITIALLY GIVEN POINTS=    0

    83 NONZEROS OF MATRIX A (ROW AND COLUMN INDICES)
  ITS DIMENSIONS:  10  20, WITH DENSITY 0.40

    1. (  1,  1)  -3.30|    2. (  1,  2)  -4.00|    3. (  4,  2)   1.80|
    4. (  7,  2)  -5.30|    5. (  9,  2)  -3.50|    6. ( 10,  2)  -8.00|
    7. (  1,  3)  -2.30|    8. (  3,  3)  -3.40|    9. (  4,  3)   3.20|
   10. (  5,  3)   8.70|   11. (  7,  3)  -3.90|   12. (  9,  3)   0.30|
   13. (  2,  4)  -4.00|   14. (  5,  4)   8.30|   15. (  7,  4)  -1.30|
```

```
  16. (  8,  4)   7.40|   17. (  9,  4)   4.20|   18. ( 10,  4)   8.80|
  19. (  2,  5)   5.80|   20. (  4,  5)   7.30|   21. (  7,  5)  -1.60|
  22. (  8,  5)   8.70|   23. (  9,  5)  -2.40|   24. ( 10,  5)  -1.60|
  25. (  9,  6)  -9.10|   26. ( 10,  6)  -1.20|   27. (  1,  7)  -2.00|
  28. (  3,  7)  -6.60|   29. (  6,  7)   6.90|   30. (  7,  7)   1.60|
  31. (  8,  7)  -4.90|   32. ( 10,  7)  -9.10|   33. (  2,  8)  -3.70|
  34. (  3,  8)   8.60|   35. (  4,  8)  -4.50|   36. (  5,  8)  -7.10|
  37. (  6,  8)  -3.20|   38. (  9,  8)  -1.90|   39. (  3,  9)   6.40|
  40. (  5,  9)   8.70|   41. (  6,  9)  -9.10|   42. (  7,  9)  -8.40|
  43. (  8,  9)   4.60|   44. (  9,  9)   6.30|   45. (  1, 10)   8.00|
  46. (  4, 10)  -1.70|   47. (  9, 10)   6.40|   48. ( 10, 10)   1.50|
  49. (  4, 11)   6.70|   50. (  1, 12)   3.10|   51. (  2, 12)   1.20|
  52. (  4, 12)  -3.00|   53. ( 10, 12)   8.10|   54. (  1, 13)   4.10|
  55. (  5, 13)  -0.40|   56. (  7, 13)   5.40|   57. (  8, 13)  -3.80|
  58. (  9, 13)  -6.30|   59. ( 10, 13)   0.10|   60. (  1, 14)   5.10|
  61. (  7, 15)   3.30|   62. ( 10, 16)  -7.90|   63. (  1, 17)  -5.80|
  64. (  3, 17)  -3.90|   65. (  4, 17)  -3.70|   66. (  7, 17)  -9.60|
  67. (  9, 17)  -3.20|   68. ( 10, 17)   4.90|   69. (  2, 18)  -6.60|
  70. (  3, 18)   2.90|   71. (  4, 18)  -8.10|   72. (  6, 18)   8.20|
  73. (  8, 18)  -4.50|   74. ( 10, 18)  -6.90|   75. (  1, 19)   6.80|
  76. (  3, 19)   9.90|   77. (  4, 19)  -5.80|   78. (  5, 19)  -3.00|
  79. (  6, 19)   8.50|   80. (  2, 20)   0.20|   81. (  4, 20)  -7.60|
  82. (  5, 20)   2.90|   83. (  6, 20)   9.80|

 LINEAR PART OF FIRST STAGE OBJECTIVE FUNCTION:
 -0.300  -0.900   1.400  2.700  2.400  -1.400   0.100  -2.900  1.700  1.700
  0.100   0.400  -1.300  0.500  0.000   0.000  -1.100  -0.700  1.500  2.100

 RIGHT HAND SIDE VECTOR OF FIRST STAGE (FOR = CONSTRAINTS)
 3.160 -1.090  1.300 -5.020  3.470  3.320 -3.250  1.150 -0.210  1.760

 A FEASIBLE SOLUTION OF THE FIRST STAGE:
  0.100D+00   0.000D+00   0.000D+00   0.200D+00   0.000D+00
  0.200D+00   0.100D+00   0.100D+00   0.200D+00   0.300D+00
  0.100D+00   0.000D+00   0.200D+00   0.300D+00   0.100D+00
  0.100D+00   0.300D+00   0.000D+00   0.100D+00   0.400D+00

  120 NONZEROS OF MATRIX T (ROW AND COLUMN INDICES)
ITS DIMENSIONS:  20  20, WITH DENSITY 0.30

   1. ( 14,  1)  -5.60|    2. (  3,  2)  -9.20|    3. (  5,  2)  -5.90|
   4. ( 12,  2)   0.50|    5. ( 14,  2)   0.30|    6. ( 16,  2)  -5.00|
   7. (  3,  3)  -5.80|    8. (  4,  3)   0.10|    9. (  6,  3)   2.00|
  10. (  8,  3)  -3.00|   11. ( 14,  3)   7.00|   12. ( 17,  3)   5.10|
  13. ( 20,  3)   1.40|   14. (  3,  4)  -7.50|   15. (  5,  4)  -4.60|
  16. ( 11,  4)   0.50|   17. ( 12,  4)   7.10|   18. (  1,  5)   6.20|
  19. (  3,  5)  -7.80|   20. (  4,  5)   8.50|   21. (  5,  5)   1.70|
  22. (  6,  5)  -5.30|   23. (  9,  5)   8.20|   24. ( 13,  5)   4.30|
  25. ( 14,  5)  -4.10|   26. ( 17,  5)   3.90|   27. ( 18,  5)   8.80|
  28. ( 20,  5)  -5.30|   29. (  1,  6)  -1.00|   30. (  2,  6)  -0.30|
  31. (  4,  6)  -0.40|   32. (  6,  6)   2.40|   33. (  8,  6)   9.00|
  34. (  9,  6)  -2.80|   35. ( 10,  6)   3.60|   36. ( 11,  6)   5.90|
  37. ( 12,  6)  -0.30|   38. ( 13,  6)  -5.50|   39. ( 15,  6)  -3.20|
  40. ( 16,  6)  -6.30|   41. ( 18,  6)   2.40|   42. ( 19,  6)  -2.00|
  43. ( 20,  6)  -2.70|   44. (  6,  7)  -5.20|   45. ( 14,  7)  -0.90|
  46. ( 17,  7)   6.00|   47. ( 20,  7)   2.70|   48. (  3,  8)   2.20|
  49. (  4,  8)  -3.60|   50. (  6,  8)   5.30|   51. (  8,  8)  -5.90|
  52. ( 12,  8)  -0.70|   53. ( 13,  8)   0.60|   54. ( 20,  8)  -0.80|
  55. (  3,  9)  -0.60|   56. (  4,  9)   9.10|   57. (  5,  9)  -6.00|
  58. (  7,  9)   0.80|   59. (  8,  9)  -0.50|   60. ( 10,  9)  -4.60|
  61. ( 16,  9)  -1.10|   62. ( 18,  9)  -0.70|   63. ( 19,  9)   6.30|
```

```
  64. (  1, 10)   6.60|   65. (  5, 10)  -6.70|   66. (  8, 10)  -6.30|
  67. (  9, 10)   2.10|   68. ( 10, 10)  -8.70|   69. ( 11, 10)   1.10|
  70. ( 12, 10)   6.80|   71. ( 13, 10)  -7.30|   72. ( 16, 10)  -1.50|
  73. ( 17, 10)  -1.30|   74. ( 12, 11)   0.60|   75. ( 17, 11)  -7.70|
  76. (  1, 12)   1.00|   77. (  5, 12)   9.30|   78. (  6, 12)  -4.70|
  79. (  8, 12)   9.70|   80. ( 11, 12)   6.80|   81. (  6, 13)  -4.60|
  82. (  8, 13)  -8.30|   83. ( 12, 13)   9.70|   84. ( 14, 13)  -4.80|
  85. ( 20, 13)  -6.70|   86. (  6, 14)   8.50|   87. ( 14, 14)  -4.20|
  88. (  1, 15)  -3.30|   89. (  2, 15)   6.20|   90. (  3, 15)   0.80|
  91. (  4, 15)   3.60|   92. (  5, 15)  -8.30|   93. (  6, 15)  -1.80|
  94. ( 14, 15)  -0.90|   95. ( 17, 15)  -7.60|   96. ( 20, 15)  -3.60|
  97. (  3, 16)   0.80|   98. ( 12, 16)  -5.50|   99. ( 14, 16)  -1.10|
 100. (  1, 17)  -4.40|  101. (  4, 17)   1.30|  102. (  6, 17)  -9.20|
 103. ( 11, 17)   1.40|  104. ( 14, 17)  -8.90|  105. ( 17, 17)  -3.30|
 106. ( 18, 17)   4.30|  107. ( 20, 17)  -4.20|  108. (  3, 18)   1.30|
 109. (  4, 18)  -8.50|  110. (  6, 18)   1.70|  111. ( 11, 18)   1.50|
 112. ( 13, 18)  -7.40|  113. ( 14, 18)  -6.60|  114. ( 15, 18)   3.40|
 115. ( 16, 18)   4.00|  116. ( 17, 18)  -5.70|  117. ( 14, 19)  -3.10|
 118. ( 17, 19)   8.00|  119. (  5, 20)   8.90|  120. (  6, 20)   6.70|

  62 NONZEROS OF MATRIX W (ROW AND COLUMN INDICES)
ITS DIMENSIONS:  20  30, WITH DENSITY 0.10

   1. (  2,  1)  -9.60|    2. (  5,  1)   7.90|    3. ( 16,  1)   3.40|
   4. (  2,  2)  -9.10|    5. (  4,  2)   1.95|    6. (  6,  2)   7.00|
   7. ( 10,  2)  -6.70|    8. (  2,  3)   8.60|    9. (  4,  4)   3.15|
  10. (  3,  5)   2.00|   11. ( 15,  6)  -1.90|   12. (  1,  7)   1.50|
  13. ( 10,  7)  -4.70|   14. ( 12,  8)  -5.80|   15. ( 20,  8)  -3.90|
  16. (  4,  9)   3.40|   17. ( 19,  9)   1.00|   18. ( 15, 10)   2.60|
  19. (  8, 11)  -5.05|   20. ( 15, 11)  -8.40|   21. ( 10, 12)   3.60|
  22. ( 19, 13)   1.40|   23. ( 14, 14)  -2.00|   24. (  8, 15)   1.80|
  25. ( 18, 16)   1.50|   26. (  9, 17)  -5.10|   27. (  8, 18)   0.90|
  28. ( 17, 18)   3.95|   29. ( 20, 18)  -4.40|   30. (  7, 19)   2.50|
  31. ( 15, 19)   5.80|   32. ( 17, 19)   4.40|   33. (  2, 20)   2.10|
  34. ( 14, 20)   2.00|   35. (  5, 21)  -8.80|   36. ( 19, 21)  -1.90|
  37. (  9, 22)  -0.30|   38. ( 20, 22)   8.30|   39. ( 13, 23)   8.30|
  40. ( 16, 23)  -4.30|   41. (  9, 24)   5.40|   42. (  6, 25)  -4.80|
  43. (  8, 25)   1.50|   44. ( 11, 25)  -1.60|   45. ( 12, 25)   6.00|
  46. (  1, 26)  -1.50|   47. (  2, 26)   8.00|   48. (  6, 26)  -2.20|
  49. (  7, 26)  -2.50|   50. ( 11, 26)  -6.40|   51. ( 12, 26)  -0.20|
  52. ( 13, 26)  -8.30|   53. ( 16, 26)   0.90|   54. ( 18, 26)  -1.50|
  55. (  4, 27)  -8.50|   56. ( 17, 27)  -8.35|   57. (  3, 28)  -2.00|
  58. ( 10, 28)   7.80|   59. (  5, 29)   0.90|   60. (  8, 30)   0.85|
  61. ( 11, 30)   8.00|   62. ( 19, 30)  -0.50|

 OBJECTIVE FUNCTION OF THE SECOND STAGE PROBLEM
 6.200  9.630  8.780  7.060  7.190  2.660  6.790  8.190  8.130  6.820
 8.810  3.740  1.790  1.250  8.040  6.380  3.850  1.060  7.690  9.350
 0.280  7.650  1.490  8.500  1.800  7.730  3.560  6.670  1.220  8.120

 EXPECTED VALUE OF RANDOM RHS OF THE SECOND STAGE PROBLEM
-4.300  0.300  4.700 -3.800  3.400  2.300 -4.200  0.300 -3.100 -1.500
 2.900 -2.400  4.700  0.300 -0.900  3.300 -0.700  2.300  4.400  0.300

 STANDARD DEVIATION OF RIGHT HAND SIDE VECTOR
 1.290  0.090  1.410  1.140  1.020  0.690  1.260  0.090  0.930  0.450
 0.870  0.720  1.410  0.090  0.270  0.990  0.210  0.690  1.320  0.090

 XPOINT
   0.000000    0.000000    0.410936    0.107915    0.020078
   0.000000    0.000000    0.280964    0.038423    0.085044
```

```
   0.000000    0.161090    0.000000    0.758401    0.551627
   0.266730    0.309641    0.000000    0.125513    0.357334

 FUNCVAL = 182.94234  (+- 0.066)
```

Duomenų bazėje pateiktas žinomas uždavinio sprendinys

0.000000	0.000000	0.410936	0.107915	0.020078
0.000000	0.000000	0.280964	0.038423	0.085044
0.000000	0.161090	0.000000	0.758401	0.551627
0.266730	0.309641	0.000000	0.125513	0.357334

bei tikslo funkcijos reikšmė 182.94234 ± 0.066

Pasiūlytu algoritmu gautas uždavinio sprendinys

0.000000	0.000000	0.438473	0.105014	0.027074
0.000000	0.000000	0.297970	0.029859	0.104539
0.000000	0.170788	0.000000	0.737453	0.580798
0.279175	0.325201	0.000000	0.127925	0.352842

bei tikslo funkcijos reikšmė 182.59248 ± 0.033.

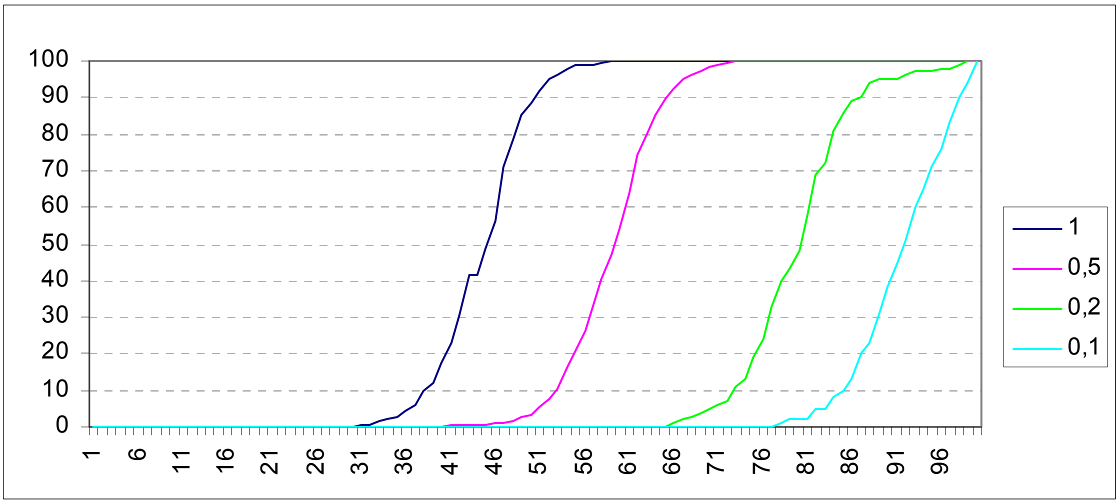
Žemiau pateikti tyrimo rezultatai. Uždavinys spręstas 400 kartų pasiūlytu metodu, kai pradiniai duomenys: $\gamma = \beta = 0.95$, $\mu = 0.99$, $\varepsilon = 0.1; 0.2; 0.5; 1.0$, $N^0 = N_{min} = 100$, maksimalus iteracijų skaičius $t_{\max} = 100$, imčių generavimas buvo nutraukiamas, kai tikslo funkcijos pasikliautinojo intervalo ilgis tapdavo mažesnis už pageidaujamą tikslumą ε.

7 pav. parodyta skaičiavimų sustabdymo po t iteracijų dažnio, kai įvykdoma stabdymo sąlyga, priklausomybė nuo pageidaujamo tikslumo ε. Vidutinė imties ilgio, tikslo funkcijos, pasikliautinojo intervalo ilgio ir Hotelingo statistikos kitimas, esant įvairioms tikslumo reikšmėms ($\varepsilon = 0{,}1;\ 0{,}2;\ 0{,}5;\ 1$), pateikti 8–11 pav., iliustruoja proceso konvergavimą bei jo pobūdį. Nutraukimo sąlyga buvo patenkinama nors kartą visoms optimizuojamoms imtims. Todėl gali būti padaryta išvada apie algoritmo konvergavimą bei sprendinio radimą pageidaujamu tikslumu visoms optimizavimo imtims.

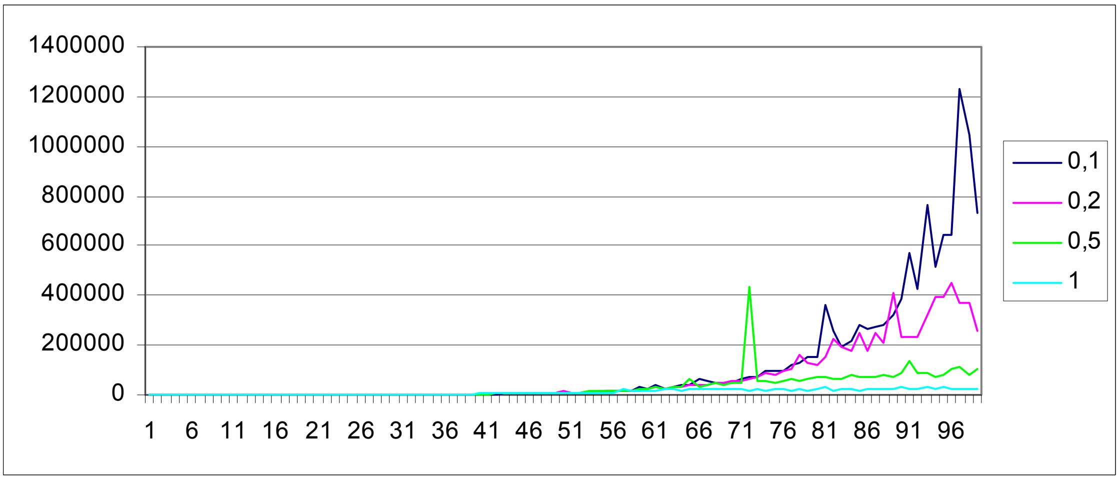
Yra įrodyta, kad netiesinio stochastinio optimizavimo uždavinių sprendimas nagrinėjamu metodu garantuoja konvergavimą su tikimybe 1 į optimumą tiesiniu greičiu, kai imties tūris apytikriai didėja geometrine progresija iteracijų metu (Sakalauskas, (2002)). Pastaruoju atveju santykis $\sum_{j=1}^{t} \frac{N^j}{N^t}$ turėtų nepriklausyti nuo pageidaujamo tikslumo. 12 pav. pateiktas santykio

$\sum_{j=1}^{t} \frac{N^j}{N^t}$ kitimas. 4 lentelėje tikslo funkcijos reikšmės F ir santykio $\sum_{j=1}^{t} \frac{N^j}{N^t}$ priklausomybė nuo pageidaujamo tikslumo ε, kurios iliustruoja minėtos teorinės išvados teisingumą.

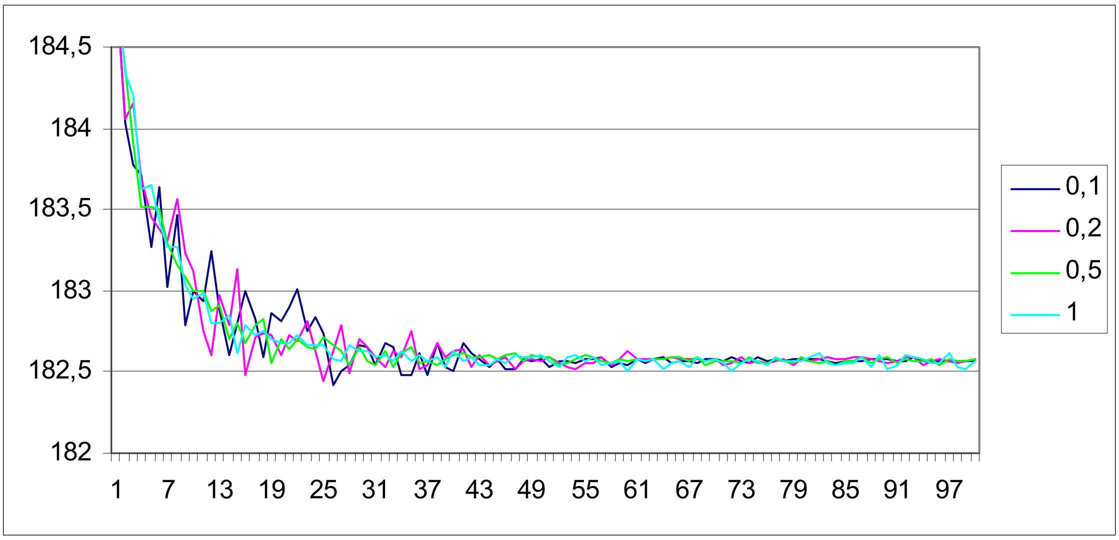
Panašios išvados gautos sprendžiant taikomąjį darbo organizavimo uždavinį (4.5.1) skyrelis



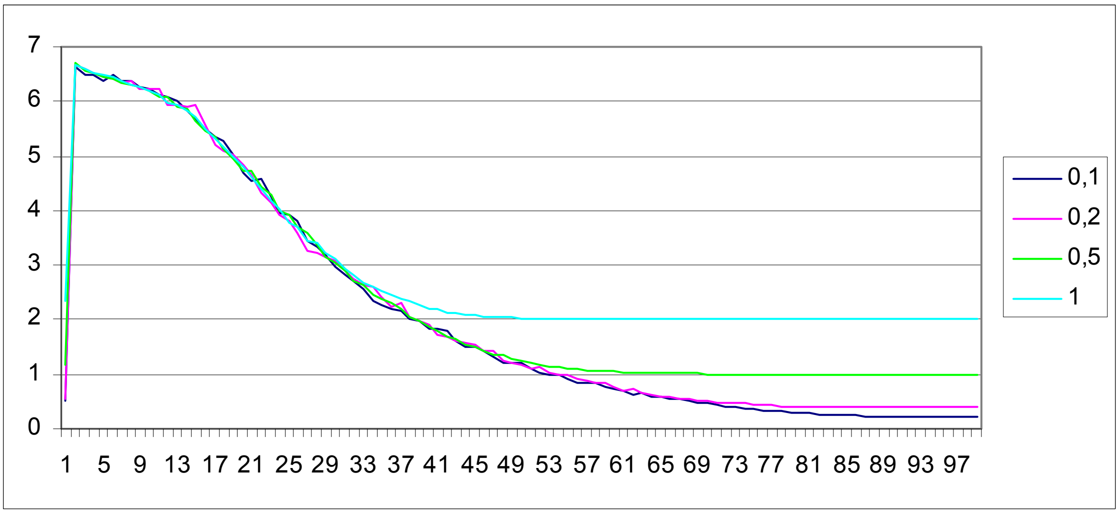
7 pav. Sustojimo dažnio priklausomybė nuo pageidaujamo tikslumo ε.



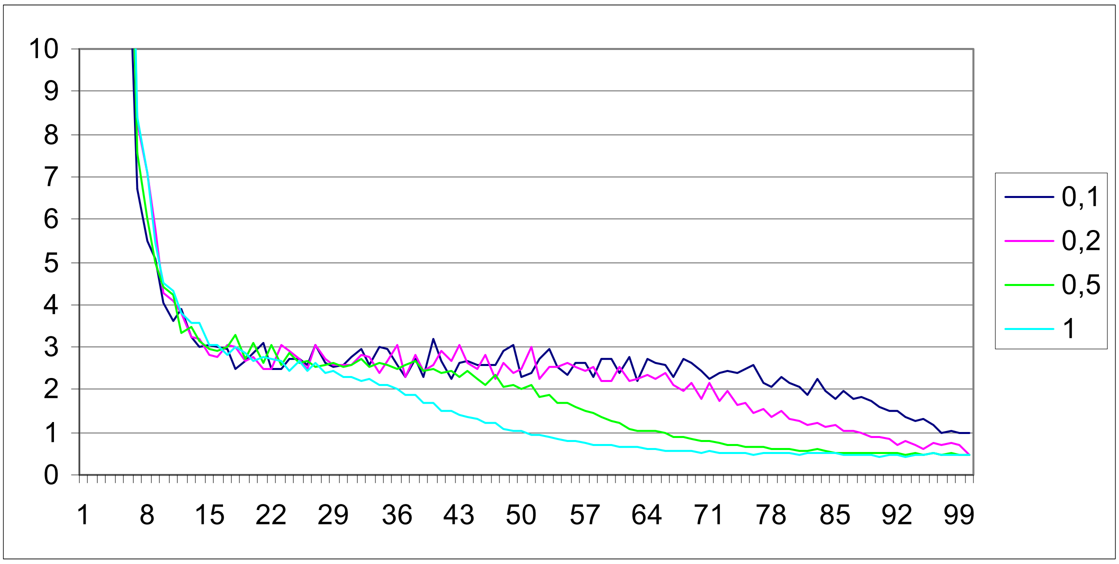
8 pav. Imties ilgio priklausomybė nuo pageidaujamo tikslumo ε.



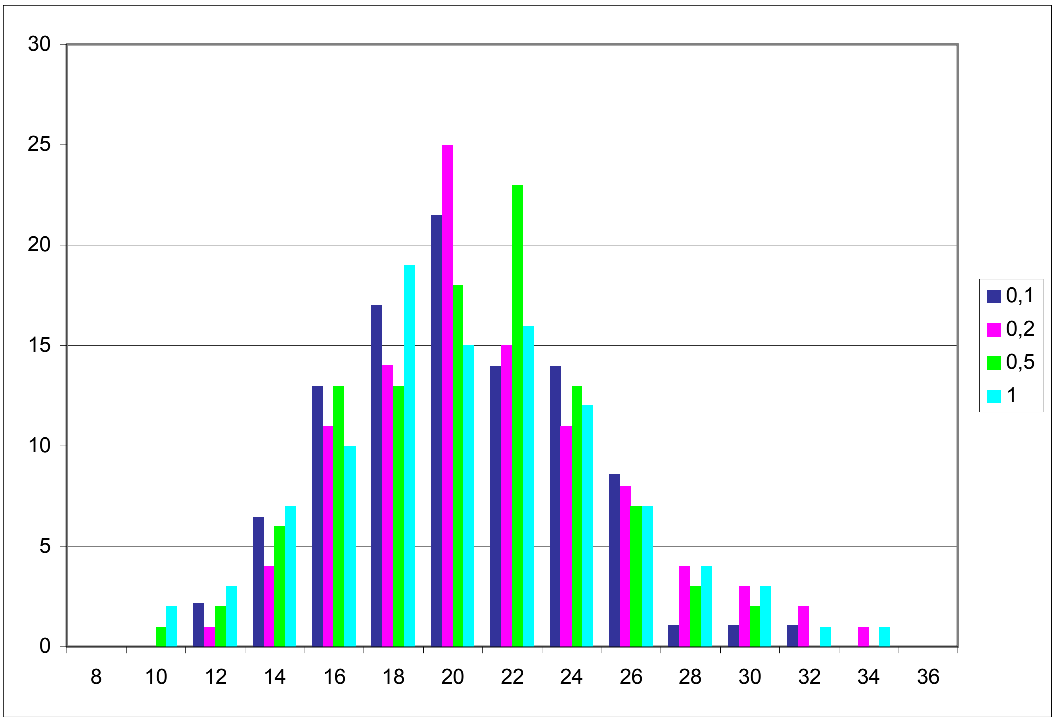
9 pav. Tikslo funkcijos reikšmės priklausomybė nuo pageidaujamo tikslumo ε.



10 pav. Pasikliautinojo intervalo ilgio priklausomybė nuo pageidaujamo tikslumo ε.



11 pav. Hotelingo statistikos priklausomybė nuo pageidaujamo tikslumo ε.



12 pav. Santykio $\sum_{j=1}^{t} \frac{N^j}{N^t}$ priklausomybė nuo pageidaujamo tikslumo ε.

4 lentelė. Tikslo funkcijos reikšmės ir santykio $\sum_{j=1}^{t} \frac{N^j}{N^t}$ priklausomybė nuo pageidaujamo tikslumo ε.

Pasikliautinasis intervalas, ε	Tikslo funkcijos reikšmė, F	Santykis, $\sum_{j=1}^{t} \frac{N^j}{N^t}$
0.1	182.6101	20.14
0.2	182.6248	19.73
0.5	182.7186	19.46
1	182.9475	19.43

2 uždavinys

Dviejų etapų tiesinis stochastinio programavimo užduoties pirmojo etapo uždavinys turi 20 kintamųjų ir 10 ribojimų, o antrojo etapo uždavinys – 30 kintamųjų ir 20 ribojimų. Uždavinio sąlyga pateikta 2 priede.

Duomenų bazėje pateiktas žinomas uždavinio sprendinys:

0.000000	2.491040	0.000000	1.140885	0.290554
0.000000	0.033435	0.023088	0.406078	0.000000
0.000000	0.625346	0.000000	0.000000	0.000000
0.084560	0.119499	0.281248	0.192027	0.152020

bei tikslo funkcijos reikšmė 266.68373 ± 0.187.

Pasiūlyto algoritmo pagalba gautas uždavinio sprendinys:

0.000000	2.596710	0.000000	1.195545	0.281074
0.000000	0.032765	0.020928	0.406148	0.000000
0.000000	0.638956	0.000029	0.000000	0.000000
0.083820	0.118879	0.282258	0.191927	0.150970

bei tikslo funkcijos reikšmė 266.22764 ± 0.066.

3 uždavinys

Dviejų etapų tiesinis stochastinio programavimo užduoties pirmojo etapo uždavinys turi 60 kintamųjų ir 30 ribojimų, o antrojo etapo uždavinys – 60 kintamųjų ir 90 ribojimų. Uždavinio sąlyga pateikta 2 priede.

Duomenų bazėje pateiktas žinomas uždavinio sprendinys:

0.1969	0.0000	0.0260	0.2892	0.1560	0.0000	0.0000	0.0172	0.0000	0.0793
0.0000	0.0000	0.0000	0.0499	0.2118	0.0881	0.0000	0.0425	0.0967	0.0501
0.6735	0.0000	0.0619	0.0203	0.0000	0.0000	0.1780	0.0000	0.0414	0.3646
0.3090	0.3593	0.0000	0.0000	0.0000	0.0899	0.2509	0.1090	0.0375	0.0000
0.0656	0.0450	0.0501	0.0000	0.0000	0.1950	0.0633	0.0000	0.0000	0.0633
0.6750	0.0026	0.0000	0.0000	0.2335	0.3067	0.0000	0.8165	0.0641	0.0000

bei tikslo funkcijos reikšmė 300.84160 ± 0.039.

Pasiūlyto algoritmo pagalba gautas uždavinio sprendinys:

0.1940	0.0000	0.0224	0.2781	0.1651	0.0000	0.0000	0.0187	0.0000	0.0804
0.0000	0.0000	0.0000	0.0594	0.2019	0.0837	0.0000	0.0395	0.0984	0.0536
0.6311	0.0000	0.0636	0.0342	0.0000	0.0000	0.1745	0.0000	0.0318	0.3610
0.3057	0.3738	0.0000	0.0000	0.0000	0.0812	0.2623	0.0915	0.0385	0.0000
0.0651	0.0367	0.0592	0.0000	0.0000	0.1919	0.0666	0.0000	0.0000	0.0680
0.6700	0.0051	0.0000	0.0000	0.2367	0.3040	0.0000	0.7703	0.0563	0.0000

bei tikslo funkcijos reikšmė 300.66896 ± 0.033.

4 uždavinys

Dviejų etapų tiesinis stochastinio programavimo užduoties pirmojo etapo uždavinys turi 80 kintamųjų ir 40 ribojimų, o antrojo etapo uždavinys – 80 kintamųjų ir 120 ribojimų. Uždavinio sąlyga pateikta 2 priede.

Duomenų bazėje pateiktas žinomas uždavinio sprendinys:

0.0000	0.0000	0.2811	0.9275	0.0000	0.0000	0.0000	0.0000	0.1368	0.2596
0.2590	0.0000	0.1567	0.0727	0.0311	0.3451	0.0000	0.0917	0.0262	0.0276
0.3560	0.0000	0.0853	0.0000	0.6380	0.0000	0.6734	0.0000	0.1170	0.0723
0.0000	0.1558	0.1571	0.0000	0.1542	0.0000	0.0653	0.1334	0.0000	0.0000
0.0000	0.0000	0.0548	0.0000	0.1130	0.0000	0.0153	0.0735	0.1191	0.0000
0.0401	0.0251	0.0519	0.0000	0.0319	0.0000	0.0229	0.0000	0.0406	0.1074
0.0101	0.0000	0.0000	0.0000	0.3822	0.0000	0.0624	0.0000	0.0368	0.0186

bei tikslo funkcijos reikšmė 586.32985 ± 0.327.

Pasiūlyto algoritmo pagalba gautas uždavinio sprendinys:

0.1451	0.0000	0.0000	0.0704	0.1418	0.0000	0.0020	0.2071	0.0000	0.2293
0.0000	0.1992	0.2146	0.0387	0.0175	0.0000	0.0000	0.1091	0.1159	0.1331
0.0000	0.0000	0.0126	0.0529	0.8415	0.0000	0.2138	0.0671	0.1740	0.1710
0.0000	0.2721	0.1630	0.0000	0.0589	0.0000	0.1285	0.1268	0.0000	0.0000
0.2703	0.0000	0.0351	0.0000	0.6207	0.2879	0.1270	0.1280	0.1745	0.0375
0.0567	0.0751	0.1603	0.0000	0.0000	0.0000	0.0244	0.4942	0.0539	0.1429
0.0639	0.0000	0.0000	0.1672	0.0913	0.0000	0.0000	0.0619	0.0000	0.0149
0.1967	0.1349	0.0000	0.0000	0.0000	0.0110	0.2659	0.0000	0.0938	0.0000

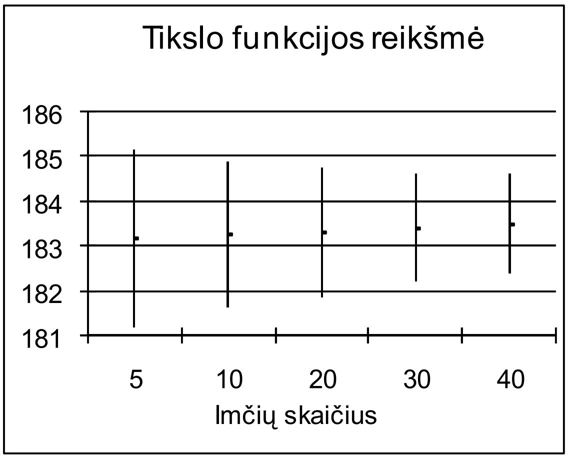
bei tikslo funkcijos reikšmė 475.01266 ± 0.99999.

4.4. Metodų palyginimas

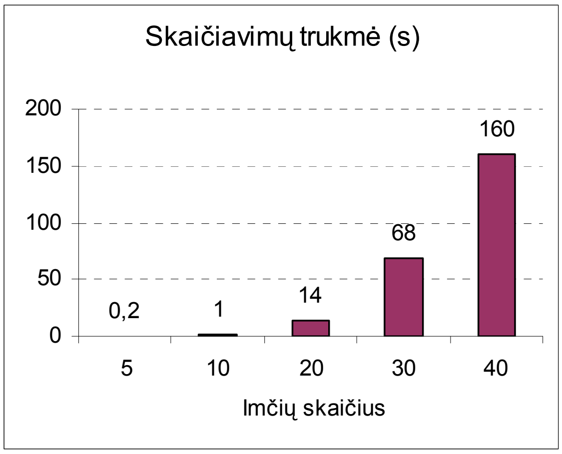
Siekiant palyginti skirtingus tiesinio stochastinio programavimo metodus, 1-ojo pavyzdžio uždavinys buvo sprendžiamas Dancigo – Vulfo dekompozicijos metodu bei Simplekso programa. Uždavinys buvo suformuluotas kaip determinuotas tiesinio programavimo uždavinys, paėmus įvairius atsitiktinių dydžių h imčių (realizacijų) kiekius (5, 10, 20, 30, 40). Simplekso programa uždavinys buvo spręstas, norint pasitikrinti gautus rezultatus. Dancigo – Vulfo dekompozicijos metodu, pasiūlytu Monte Karlo metodu ir Simplekso programa spręstų uždavinių atsitiktinių dydžių vektoriai buvo vienodi.

4.4.1. Dancigo - Vulfo dekompozicijos metodo taikymo rezultatai

Išsprendus uždavinį Dancigo – Vulfo dekompozicijos metodu 400 kartų su atitinkamu imčių skaičiumi gauti atitinkami rezultatai. 13 pav. pateikta tikslo funkcijos reikšmės bei pasikliautinojo intervalo vidurkio priklausomybė nuo imčių skaičiaus. 14 pav. pateikta vidutinės skaičiavimų trukmės priklausomybė nuo imčių skaičiaus.



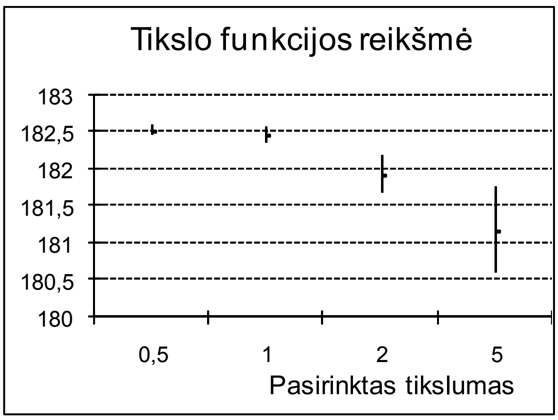
13 pav. Tikslo funkcijos reikšmės bei pasikliautinojo intervalo vidurkio priklausomybė nuo imčių skaičiaus.



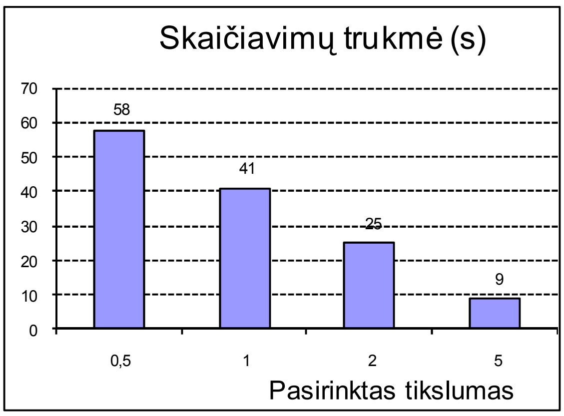
14 pav. Vidutinės skaičiavimų trukmės priklausomybė nuo imčių skaičiaus.

4.4.2. Monte Karlo metodo taikymo rezultatai

Išsprendus uždavinį pasiūlytu Monte Karlo metodu 400 kartų gauti atitinkami rezultatai. 15 pav. pateikta tikslo funkcijos reikšmės bei pasikliautinojo intervalo vidurkio priklausomybė nuo pasirinkto tikslumo. 16 pav. pateikta vidutinės skaičiavimų trukmės priklausomybė nuo pasirinkto tikslumo.



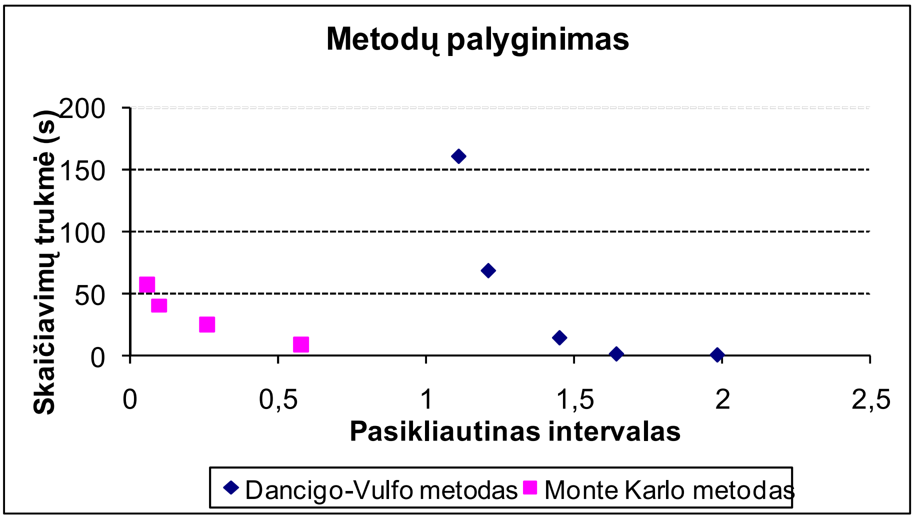
15 pav. Tikslo funkcijos reikšmės bei pasikliautinojo intervalo vidurkio priklausomybė nuo pasirinkto tikslumo.



16 pav. Vidutinės skaičiavimų trukmės priklausomybė nuo pasirinkto tikslumo.

4.4.3. Dancigo - Vulfo dekompozicijos ir Monte Karlo baigtinių imčių metodų palyginimas

Gautų rezultatų palyginimui sudaryta vidutinės skaičiavimų trukmės priklausomybė nuo tikslo funkcijos pasikliautinojo intervalo ilgio Dancigo – Vulfo bei Monte Karlo metodams, pateikta 17 pav.



17 pav. Vidutinės skaičiavimų trukmės priklausomybė nuo tikslo funkcijos pasikliautinojo intervalo ilgio

Skaitiniai tyrimai parodė, kad pagrindinis Dancigo - Vulfo metodo trūkumas yra tai, kad uždavinio suformavimui yra reikalingas didelis atsitiktinių dydžių masyvas, proporcingas stochastinių duomenų realizacijų skaičiui. Šis duomenų masyvas turi būti saugomas kompiuterio atmintyje uždavinio sprendimo eigoje.

Sprendžiant suformuotą uždavinį simplekso metodu, sprendinys gaunamas po didelio iteracijų skaičiaus, kuris yra proporcingas stochastinių duomenų realizacijų skaičiui. Stochastinių duomenų realizacijų skaičių riboja kompiuterio atmintis ir programavimo kalbų naudojamų masyvų ribojimai. Dėl mažo galimo duomenų realizacijų skaičiaus stochastinio uždavinio sprendinio tikslumas mažėja. Šis metodas naudotas kaip testinis, dekompozicijos metodo tikslumui įvertinti.

Sprendžiant suformuotą uždavinį Dancigo - Vulfo dekompozicijos metodu, reikia mažiau iteracijų. Tačiau uždavinio sprendinio tikslumas labai sumažėja, nes atskiras duomenų realizacijas aprašančių pagalbinių uždavinių sprendiniai nevienodai įtakoja pirmojo etapo kintamuosius.

Skaitiniai tyrimai leidžia daryti išvadą, kad pasiūlytas stochastinių tiesinių uždavinių sprendimo metodas, realizuojantis iteracinį algoritmą baigtinėmis Monte-Karlo imčių sekomis dviejų etapų tiesinio programavimo uždaviniui spręsti, leidžia efektyviai gauti tikslesnius sprendinius.

4.5. Taikomųjų uždavinių sprendimas

4.5.1. Darbo organizavimo uždavinys

Šiame uždavinyje darbdavys turi parinkti nuolatinių darbuotojų skaičių kelių lygių darbų organizavimo hierarchinėje struktūroje. Papildomai reikia įvertinti nuolatinių darbuotojų viršvalandžius bei laikinai samdomos darbo jėgos poreikį priklausomai nuo iš anksto nežinomos paslaugų, kurias įmonė turės pateikti, apimties. Uždavinio detalų aprašymą galima rasti (Ermolyev, Wets, (1988)).

4.5.1.1. Uždavinio formuluotė

Šiame pavyzdyje nagrinėjamas darbo jėgos poreikis trijų lygių struktūros (1 lygis – vykdytojai ir darbininkai, 2 – tarpinis, 3 – vadovaujantys darbuotojai). Darbuotojų skaičių kiekviename lygyje žymėkime $\left(x_1, x_2, x_3\right)$.

Tad reikia rasti x = $\left(x_1, x_2, x_3\right)$ minimizuojantį vidutinius darbo jėgos kaštus:

$$F(x,z) = \sum_{j=1}^{3} c_j \cdot x_j + \sum_{t=1}^{12} E \min_{y,z}\left(\sum_{j=1}^{3} (q_j \cdot y_{j,t} + r_j \cdot z_{j,t})\right)$$

kai

$x_j \geq 0$, $y_{j,t} \geq 0$, $z_{j,t} \geq 0$,

$\sum_{j=1}^{3} (y_{j,t} + z_{j,t}) \geq w_t - \alpha_t \sum_{j=1}^{3} x_j$, t=1,...,12,

$y_{j,t} \leq 0,2 \cdot a_t x_j$, j=1, 2, 3 t=1,...,12,

$\gamma_{j-1}(x_j + y_{j-1,t} + z_{j-1,t}) - (x_j + y_{j-1,t} + z_{j-1,t}) \geq 0$, j=2, 3, t=1,...,12.

čia

x_j – j lygio pastovių darbuotojų skaičius, j = 1, 2, 3

$y_{j,t}$ – viršvalandžių kiekis,

$z_{j,t}$ – laikinos pagalbos dydis,

c_j – j lygio pastovių darbuotojų užmokestis, j = 1, 2, 3

q_j – viršvalandžių kaina,

r_j – laikinosios pagalbos kaina,

w_t – paslaugų poreikis t laikotarpiu,

α_t – prognozuojamas darbuotojų neatvykimo santykis laikotarpiu t,

γ_{j-1} – reikalingas j lygio ir j-1 lygio darbuotojų santykis.

Pradiniai duomenys:

c = [7.03, 4.53, 3.44],

q = [9.59, 6.18, 4.69],

r =[11.70, 9.95, 5.78],

α = [0.8943, 0.8917, 0.8948, 0.9086, 0.9032, 0.8842, 0.8513, 0.8798, 0.8871, 0.9043, 0.8606, 0.8341],

γ = [0.6, 0.2],

w_t – nepriklausomi t = 1,…,12 dydžiai $N\left(\overline{\mu_t}, \overline{\sigma_t^2}\right)$, čia $\overline{\sigma}_t^2 = \eta \cdot \mu_t$.

$\overline{\mu}$ = [11975, 11740, 12169, 13132, 13525, 12598, 13503, 14168, 12602, 11807, 11334, 10410],

η – atsitiktinio paslaugų poreikio variacija 0, 1, 10, 20, 30.

4.5.1.2. Uždavinio sprendimo rezultatai

Sprendimo rezultatai, gauti panaudojant Monte Karlo metodą, pateikti 5 lentelėje bei 18–21 pav. 5 lentelėje matome optimalų darbo jėgos poreikį visuose trijuose lygiuose bei bendrus darbo jėgos kaštus priklausomai nuo atsitiktinio paslaugų poreikio variacijos koeficiento η (pasikliautinasis intervalas – 100 dolerių (USD), darbo jėgos kaštai pateikti tūkst. dolerių).

5 lentelė. Optimalus darbo jėgos poreikis ir bendri darbo jėgos kaštai (priklausomai nuo variacijos η).

	Pastovių darbuotojų skaičius			Darbo jėgos kaštai (pasikliautinas intervalas 0,1)
η	X_1	X_2	X_3	F
0	9222	5533	1106	94,899
1	9222	5533	1106	94,899
10	9376	5616	1106	96,832

Yra žinoma, kad siūlomas metodas užtikrina konvergavimą tiesiniu greičiu (Sakalauskas, 2000, 2002). Todėl bendras Monte Karlo imčių skaičius $\sum_{j=1}^{t} N^{j}$, reikalingas gauti optimalųjį sprendinį, yra apytiksliai proporcingas imčių skaičiui N^{t}, reikalingam išspręsti uždavinį norimu tikslumu. Santykis $\sum_{j=1}^{t} \frac{N^{j}}{N^{t}}$ labiausiai priklauso nuo tikslo funkcijos Hesiano teigiamo apibrėžtumo ir beveik nepriklauso nuo pasirinkto tikslumo ε (Sakalauskas, 2002). 6 lentelė parodo, kaip šis santykis priklauso nuo parametrų ε ir η.

6 lentelė. Santykio $\sum_{j=1}^{t} \frac{N^{j}}{N^{t}}$ priklausomybė nuo tikslumo ε ir variacijos η.

ε	0,05	0,1	0,2
η=10	16.1	10.7	10.9
η=30	21.4	21.3	20.2

Galima padaryti išvadą, kad optimizavimui reikiamu tikslumu reikia tik keletą kartų daugiau kompiuterinių resursų, palyginus su resursų kiekiu, reikalingu apskaičiuoti vieną tikslo funkcijos reikšmę tokiu tikslumu.

18–21 pav. pavaizduotos tikslo funkcijos įverčio, jo pasikliautinojo intervalo ilgio, statistikos $\frac{T_t^2}{Fish(\mu, n, N^t - n)}$, bei imties tūrio priklausomybės nuo iteracijų skaičiaus t, kai η = 10, o reikiamas pasikliautinas intervalas ε = 100 USD. Gauti rezultatai iliustruoja sukurto metodo konvergavimą bei skaičiavimų apimtį, reikalingus uždaviniui išspręsti reikiamu tikslumu.

18 pav. Tikslo funkcijos $\tilde{F}(x^{t})$ priklausomybė nuo iteracijų skaičiaus.

19 pav. Tikslo funkcijos pasikliautinojo intervalo priklausomybė nuo iteracijų skaičiaus.

20 pav. Statistikos $T_t^2 / Fish(\mu, n, N^t - n)$ priklausomybė nuo iteracijų skaičiaus.

21 pav. Imties ilgio priklausomybė nuo iteracijų skaičiaus.

4.5.2. Investavimo ir elektros energijos gavybos uždavinys

Šiame uždavinyje energetikos kompanija turi parinkti tinkamą energetinę struktūrą, kuri minimizuoja investavimo išlaidų ir elektros energijos gamybos kaštų sumą. Galimi keturių elektros energijos gamybos stočių tipai, energijos poreikis – atsitiktinis dydis, turintis žinomą skirstinį. Uždavinio detalų aprašymą galima rasti (Freund, (2004)).

4.5.2.1. Uždavinio formuluotė

Tegul kompanija planuoja investuoti lėšas į elektros energijos gavybą bei tiekti reikiamą elektros energijos kiekį. Energijos gavybos stotys pastatomos pradiniame etape, elektros energija bus tiekiama 15 metų. Energijos stočių statybai investuojama iki 10 mlrd. dolerių. Statomos keturių tipų elektrinės: termofikacinės(dujų), kietojo kuro(anglies), branduolinio kuro ir hidroelektrinės. Investavimo projekto tikslas – mažiausios investavimo ir elektros gavybos išlaidos visiškai patenkinant elektros energijos poreikius. Reikia rasti tokią energetinę struktūrą, kuri minimizuoja investavimo išlaidų ir elektros energijos gamybos kaštų per 15 metų sumą.

Elektros energijos stočių statybai reikalingos lėšos pateiktos 7 lentelėje.

7 lentelė. Investicinės išlaidos elektros energijos stočių statybai

Elektros stotis	**1 GW pajėgumo stoties statybos kaina (mln. dolerių)**
Termofikacinė	110
Kietojo kuro	180
Branduolinė	450
Hidroelektrinė	950

Energijos poreikis D yra atsitiktinis dydis $N\left(\mu,\ 0{,}5\right)$. Energijos poreikis per metus skirstomas į 5 laikotarpius (pagal klimato sąlygas) su skirtinga trukme (8 lentelė).

8 lentelė. Elektros energijos poreikis per metus

Laikotarpis	Elektros energijos poreikis (μ ,GW)	Trukmė (h val.)
1	26,0	490
2	21,5	730
3	17,3	2190
4	13,9	3260
5	11,1	2090

Elektros energijos gamybos kainos pirmaisiais metais pateikti 9 lentelėje. Jei pagaminta elektros energija nepatenkina vartotojų poreikių, papildoma energija perkama iš išorinių šaltinių.

Elektros energijos gamyba kasmet brangsta 1%. Dėl tam tikrų vietinių sąlygų hidroelektrinės gali pagaminti ne daugiau kaip 5 GW energijos.

9 lentelė. Elektros energijos kaina

Elektros stotis	Elektros energijos 1 kWh gamybos kaina (centais)
Termofikacinė	3,92
Kietojo kuro	2,44
Branduolinė	1,40
Hidroelektrinė	0,40
Išoriniai šaltiniai	15,0

Turime stochastinio programavimo uždavinį:

$$\sum_{i=1}^{4} c_i x_i + E\left\{\min_{y}\left(\sum_{i=1}^{5}\sum_{j=1}^{5}\sum_{k=1}^{15} q_i h_j r_k y_{ijk\omega}\right)\right\} \to \min,$$

kai

$$\sum_{i=1}^{4} c_i x_i \le 10000,$$

$$x_4 \le 5.0,$$

$$y_{ijk\omega} \le x_i,\ \ i = 1,2,3,4,\ \ \forall\, j,k,\omega,$$

$$\sum_{i=1}^{5} y_{ijk\omega} \ge D_{jk\omega},\ \ \forall\, j,k,\omega,$$

$$x \ge 0,\ y \ge 0,$$

kur

$x = (x_1, x_2, x_3, x_4)$ – elektros energijos gamybos stočių pajėgumų (GW) vektorius,

$y_{ijk\omega}$ – elektros energijos kiekis (GW), kurį *i*-tojo tipo elektros energijos gamybos stotis pagamina *k*-tųjų metų *j*-tąjį laikotarpį, kai elektros energijos poreikis $D_{jk\omega}$,

c_i – *i*-tojo tipo elektros energijos gamybos stoties statybos kaina (1 GW pajėgumui),

q_i – *i*-tojo tipo stoties pagamintos energijos kaina,

h_j – *j*-tojo laikotarpio ilgis(val.),

r_k – elektros energijos gamybos kainos padidėjimas *k*-taisiais metais,

$D_{jk\omega}$ – elektros energijos poreikis *k*-tųjų metų *j*-tąjį laikotarpį, kuris yra atsitiktinis dydis $N(\mu_j, 0{,}5)$.

4.5.2.2. Uždavinio sprendimo rezultatai

Investavimo ir elektros energijos gavybos uždavinys sprendimo rezultatai, gauti panaudojant Monte Karlo metodą, pateikti 10 lentelėje. Uždavinys spręstas 2 metų laikotarpiui. Antrojo etapo uždavinys turi 50 ribojimų ir 100 kintamųjų.

10 lentelė. Elektros stočių pajėgumas

	Gautas sprendinys	**Žinomas sprendinys**
Elektros stotis	**Pajėgumas (GW)**	**Pajėgumas (GW)**
Termofikacinė	4,66	4,67
Kietojo kuro	4,57	4,59
Branduolinė	4,68	4,67
Hidroelektrinė	5,00	5,00

Bendra investicijų ir elektros energijos gamybos kaina $16483{,}718 \pm 29{,}985$ mln. dolerių.

4 skyriaus išvados

1. Kelių stochastinio gradiento įverčių savybės ištirtos statistinio modeliavimo būdu palyginant tiesioginio diferencijavimo (sprendžiant dualųjį uždavinį), baigtinių skirtumų, modeliuojamojo pokyčio metodas, ir tikėtinumo santykio metodus.

2. Statistinio modeliavimo rezultatai parodė, kad įvertis, gautas sprendžiant dualųjį uždavinį arba baigtiniu skirtumų metodu leidžia duotu tikslumu patikrinti hipotezę apie tikslo funkcijos gradiento lygybę nuliui, kai kintamųjų skaičius $n < 100$.

3. Sudaryto stochastinio tiesinio programavimo algoritmo efektyvumas ištirtas statistinio modeliavimo būdu, sprendžiant uždavinius iš standartinės tiesinio programavimo testinių pavyzdžių duomenų bazės.

4. Sudarytas algoritmas leido išspręsti visus duomenų bazės testinius uždavinius bei pagerinti kai kurių uždavinių žinomus sprendimo rezultatus.

5. Statistinio modeliavimo rezultatai patvirtino teorines išvadas apie sukurto metodo konvergavimą bei konvergavimo greičio eilę.

6. Statistinio eksperimento rezultatai parodė, kad bendra skaičiavimo apimtis, reikalinga uždaviniui išspręsti duotu tikslumu, yra tiesiogiai proporcinga skaičiavimų kiekiui, reikalingam įvertinti vieną tikslo funkcijos reikšmę tokiu tikslumu.

7 Sudarytas algoritmas nereikalauja papildomų kompiuterio atminties išteklių, palyginus su žinomais dekompozicijos metodais, taikomais tokiems uždaviniams spręsti.

8. Sudarytas stochastinių tiesinių uždavinių sprendimo metodas leidžia gauti tikslesnius sprendinius, palyginus su žinomais dekompozicijos metodais, panaudojant mažiau skaičiuojamųjų resursų.

5 skyrius. Išvados

Stochastinio tiesinio programavimo uždaviniai yra plačiai pritaikomi įvairiose srityse: planuojant elektros energijos gamybą, produkcijos gamybą ir transportavimą, personalo valdymą, logistiką, investicijų valdymą, mechanizmų stabilumą, tiekimo grandines, analizuojant chemines, biologines bei visuomenines sistemas.

Sprendžiant darbe iškeltus uždavinius yra pasiekti tokie rezultatai:

1. Pasiūlytas dviejų etapų stochastinio tiesinio programavimo Monte Karlo metodas ir sudarytas jį realizuojantis iteracinis algoritmas.

2. Sudaryti ir ištirti stochastinio diferencijavimo algoritmai:
   - dualaus uždavinio sprendimo,
   - baigtinių skirtumų,
   - modeliuojamojo pokyčio,
   - tikėtinumo santykio.

3. Sudarytas stochastinio gradiento ε-projektavimo algoritmas bei ε-projektavimo algoritmas stochastiniam gradientui reguliarizuoti.

4. Sudaryta ir ištirta Monte Karlo imčių tūrio parinkimo taisyklė, garantuojanti konvergavimą bei racionaliai panaudojanti skaičiuojamuosius išteklius.

5. Sudarytas algoritmas optimalumo hipotezei patikrinti, taikant statistinius kriterijus.

6. Sudarytas algoritmas dviejų etapų stochastinio tiesinio programavimo uždaviniui spręsti naudojant baigtines Monte Karlo imčių sekas.

7. Sukurti algoritmai ir programinių modulių biblioteka Delphi, FreePascal, MathCad, C++ priemonėmis, realizuojanti stochastinio diferencijavimo bei dviejų etapų stochastinio tiesinio programavimo algoritmą Monte Karlo metodu.

8. Sudaryti algoritmai yra pritaikyti taikomiesiems uždaviniams spręsti:
   - personalo organizavimo valdymas;
   - investavimo ir elektros energijos gavybos planavimas.

Remiantis gautais rezultatais bei atliktais tyrimais, galima daryti išvadas:

1. Sprendžiant testinius bei taikomuosius uždavinius parodyta, jog sudarytas dviejų etapų stochastinio tiesinio programavimo baigtinėmis Monte Karlo imčių serijomis algoritmas leidžia surasti dviejų etapų STP uždavinių sprendinius reikiamu tikslumu, panaudojant priimtinus kompiuterinio laiko ir atminties išteklius.

3. Statistinio modeliavimo rezultatai parodė, kad stochastinio gradiento įvertis, gautas sprendžiant dualųjį uždavinį, leidžia duotu patikimumu patvirtinti hipotezę apie tikslo funkcijos gradiento lygybę nuliui, kai kintamųjų skaičius $n \leq 100$.

4. Sudaryto algoritmo taikymas leido išspręsti visus standartinės duomenų bazės testinius uždavinius bei pagerinti kai kurių uždavinių žinomus sprendimo rezultatus.

5. Statistinio modeliavimo rezultatai patvirtino teorines išvadas apie sukurto metodo konvergavimo greičio eilę. Statistinio eksperimento rezultatai parodė, kad bendra skaičiavimo apimtis, reikalinga uždaviniui išspręsti, yra tiesiai proporcinga skaičiavimų kiekiui, reikalingam įvertinti vieną tikslo funkcijos reikšmę duotu tikslumu.

8. Sudarytas algoritmas nereikalauja papildomų kompiuterio atminties išteklių palyginus su žinomais dekompozicijos metodais, taikomais tokiems uždaviniams spręsti.

9. Sudarytas stochastinių tiesinių uždavinių sprendimo metodas, leidžia gauti tikslesnius sprendinius palyginus su žinomais dekompozicijos metodais, panaudojant mažiau skaičiuojamųjų resursų.

Literatūra

1. Ahmed S., King A., Parija G. (2003a). A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. // *Journal of Global Optimization*, 26, p. 3-24.
2. Appelgren L. H. (1969). Column generation algorithm for a ship scheduling problem. // *Transportation Science,* 3, p. 53–68.
3. Barahona F., Bermon S., Gunluk O., Hood S. (2005). Robust capacity planning in semiconductor manufacturing. // *Naval Research Logistics*, 52, p. 459–468.
4. Barbarosoglu G., Arda Y. (2004). A two-stage stochastic programming framework for transportation planning in disaster response, // *Journal of the Operational Research Society*, 55, p. 43–53.
5. Bellman R., Zadeh L. (1970). Decision maling in a fuzzy environment. // *Management Science*, 17, p. 141-161
6. Benders J.F. (1962). Partitioning procedures for solving mixed variables programming problems. // *Numersiche Mathematik*, 4, p. 238-252.
7. Bertsekas D. P. (1982). Constrained optimization and Lagrange multiplier methods. // *Computer Science and Applied Mathematics*, Academic Press, London.
8. Bhattacharya, R.N., Ranga Rao R. (1976). Normal Approximation and Asymptotic Expansions. John Wiley, New York, London, Toronto.
9. Birge J. R., (1985). Decomposition and partitioning methods for multistage stochastic linear programs. // *Operations Research*, 33, p. 989-1007.
10. Birge J. R., Dempster M. (1996). Stochastic programming approaches to stochastic scheduling. // *Journal of Global Optimization*, 9(3-4), p. 417-451.
11. Birge J.R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. // *Operations Research*, 33, p. 989-1007.
12. Bitran G., Yanasee H. (1984). Deterministic Approximations to Stochastic Production Problems. // *Operations Research*, Vol. 32, No. 5, p. 999 – 1018.
13. Carino D., Kent T., Myers D. H., Stacy C., Sylvanus M., Turner A., Watanabe K., Ziemba W. T. (1994). The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming, // *Interfaces*, Vol. 24, No. 1, p. 29–49.

14. Carino D., Ziemba W. (1998). Formulation of the Russel-Yasuda Kasai financial planning model. // *Operations Research,* 46, p. 433-449.
15. Caroe C., Tind J. (1998). L-shaped decomposition of two-stage stochastic programs with integer recourse. // *Mathematical Programming*, 83, p. 451-464.
16. Charnes A., Cooper W., Symonds G. (1958). Cost horizons and certainty equivivalents: An approach to stochasting programming of heating oil. // *Management Science*, 6, p. 235-263.
17. Dantzig G. B. (1963). Linear Programming and Extensions. Princeton. Princeton U P.
18. Dantzig G. B., Orden A., Wolfe P. (1955). The Generalized Simplex Method for Minimizing a Linear Form under Inequality Restraints. // *Pacific Journal of Mathematics*. Vol. 8, p. 183-195.
19. Deak I. (1990). Random number generators and simulation. // *Mathematical Methods of Operations Research,* Akademiai Kiado, Budapest, p. 301-334.
20. Deak I. (2006). Two-stage stochastic problems with correlated normal variables: computational experiences. // Annals of Operations Research, Vol. 142, no 1, p. 79-97.
21. Deley W. D. (1991). Computer Generated Random Numbers. (on-line version: http://world.std.com/~franl/crypto/random-numbers.html).
22. Delft C., Vial J.-Ph. (2004). A practical implementation of stochastic programming: an application to the evaluation of option contracts in supply chains. // *Automatica*, 40, p. 743–756
23. Delgado M., Herrera F., Verdegay J., Vila M. (1993). Post-optimality analysis on the membership function of a fuzzy linear programming problem. // *Fuzzy Sets and Systems*, 53, p. 289-297.
24. Dennis J. E., Shnabel R. B. (1996). Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM, Philadelphia.
25. Dupač V. (1988). Stochastic approximation. In Handbook of Statistics. // *Nonparametric methods*. Eds. Krishnaiah P.R., Sen P.K. Nord Holand, N.Y.
26. Dupačova J., Gaivoronski A., Kos Z., Shantai T. (1991). Stochastic programming in water management: A case study and a comparison of solution techniques. // *European Journal of Operational Research*, 52, p. 28-44.
27. Dvoretzky A. (1956). On stochastic approximation. // *Proc. 3rd Berkeley Symposium of Mathematical Statistics and Probability*, J.Neumann(eds.), University of California Press, Berkeley, v. I, p. 39-55.

28. Ermolyev Yu. and R.Wets. (1988). Numerical Techniques for Stochastic Optimization. Springer-Verlag, Berlin.
29. Ermolyev, Yu.M. (1976). Methods of Stochastic Programming. Nauka, Moscow. 240 pp. (in Russian).
30. Ermolyev, Yu.M. (1979). Mathematical methods of operational research. Vishcha shkola, Kiev. 310 pp. (in Russian).
31. Escudero J., Kamesaman P., King A., Wets R. (1993). Production planning via scenario modeling. // *Annals of Operations Research*, 43, p. 311-335.
32. Ferguson A., Dantzig G. (1956). The allocation of aircraft to routes: An example of linear programming under uncertain demands. // *Management Science*, 3, p. 45-73
33. Floudas C.A., Pardalos P.M., Adjiman C.S., Esposito W.R., Gumus Z., Harding S.T., Klepeis J.L., Meyer C.A., Schweiger C.A. (1999). Handbook of Test Problems for Local and Global Optimization. Kluwer Academic Publishers.
34. Fourer R., Gay D. M., Kernighan B. W. (1992). AMPL: A Modeling Language for Mathematical Programming, Scientific Press, San Francisco, CA.
35. Freund R. M. (2004). Optimization under Uncertainty, Masachusetts Institute of Technology, p. 18-27.
36. Fujokoshi Y. (1997). An asymptotic expansion for the distribution of Hotelling's T2-statistic under nonnormality. // *Journal of Multivariate Analysis***,** Volume 61 , Issue 2, p. 187 - 193
37. Gassman H. I. (1989). Optimal harvest of a forest in the presence of uncertainty. // *Canadian Journal of Forest Research*, Vol 19, pp. 1267-1274.
38. Gazmuri P., Maturana S. (2001). Developing and implementation a production planning DSS for CTI using structured modeling. // *Interfaces*, Vol. 31, No. 4, p. 22-36.
39. Gondzio J., Grothey A. (2007). Solving Nonlinear Portfolio Optimization Problems with the Primal-Dual Interior Point Method. // European Journal of Operational Research, Vol. 181, No 3, p. 1019-1029.
40. Graves S. C., Jordan W. C. (1991). Principles of the Benefits of Manufacturing Process Flexibility, General Motors Laboratories Research Publication.
41. Guldman J. (1983). Supply, storage and service reliability decisions by gas distribution utilities: a chance-constrained approach. // *Management Science*, 29 (8), 884-906.
42. Gupta A., Maranas C. (2000). A two-stage modeling and solution framework for multisite midterm planning under demand uncertainty. // *Industrial & Engineering Chemistry Research*, 39, p. 3799-3813.

43. Higle J. L., Sen S. (1995). Recourse constrained stochastic programs. // *Annals of Operations Research*, Vol. 56, p. 157–175.
44. Higle J. L., Sen S. (1999). Statistical approximations for stochastic linear programming problems. // *Annals of Operations Research*, 85, p. 173-192.
45. Higle J. L., Sen, S. (1991). Stochastic decomposition: An algorithm for two-stage linear programs with recourse. // *Mathematics of Operations Research*, Vol. 16, No. 3, p. 650–669.
46. Ho J. K. (1977). Nested Decomposition of a Dynamic Energy Model. // *Management Science*, Vol. 23, No 9. p. 1022-1026.
47. Ho J. K. (1980). A successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem. // *Transportation Science*, 14(4), p. 295-305.
48. Horst R., Pardalos P.M., Van Thoai N. (2000). Introduction to Global Optimization. // *Nonconvex Optimization and its Applications*, vol. 48, Kluwer Academic Publishers.
49. Huang Y., Baetz B., Huang G., Liu L. (2002). Violation analysis for solid waste management systems: An interval fuzzy programming approach. // *Journal of Environmental Management*, 65, p. 431-446.
50. Kall P. (1976). Stochastic Linear Programming, Springer-Verlag, Berlin.
51. Karnopp D. C. (1963). Random search techniques for optimization problems. // *Automatica*, Vol. 1(1), p. 111–121.
52. King J. (1988). Stochastic Programming problems: Examples from the Literature. // In: *Ermolyev, Ju, and Wets, R. (eds.), Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin.
53. Klein Haneveld W. K. (1986). Duality in Stochastic Linear and Dynamic Programming, Lecture notes in economics and mathematical systems, No. 274, Springer-Verlag, Berlin.
54. Kleywegt A.J., Shapiro A., Homem-de Mello T. (2001). The sample average approximationmethod for stochastic discrete optimization. // *SIAM Journal on Optimization,* Vol. 12, p. 479–502.
55. Korn G. A., Korn T. M. (2003). Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review. Dover publications, INC., Mineola, New York.
56. Krishnaiah P.R., Lee, J.C. (1980). Handbook of Statistics, vol. 1 (Analysis of Variance). North-Holland, N.Y.

57. Kushner H.J., Yin G. G. (2003). Stochastic Approximation and Recursive Algorithms and Applications. Springer: N.Y., Heidelberg, Berlin.
58. Kusy M. I., Ziemba W. T. (1986). A Bank Asset and Liability Management Model. // *Operations Research*, 34(3), p. 356-376.
59. Laguna M. (1998). Applying robust optimization to capacity expantion of one location in telecommunications with demand uncertainty. // *Management Science*, 44, p. 101-110.
60. Louveaux F., Smeers Y. (1988). Optimal investments for electricity generation: a stochastic model and a test-problem. // *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, pp. 33-64.
61. Lustig I., Mulvey J., Carpenter T. (1991). Formulating stochastic programs for interior point methods. // *Operations Research*, 39, 5, p. 757-770.
62. Marti K. (1988). Descent stochastic quasigradient methods. // In *Ermolyev, Yu., and Wets R. Numerical Techniques for Stochastic Optimization*. Springer-Verlag, Berlin, p. 393-400.
63. Marti K. (1989). Optimal semi-stochastic approximation procedures. //. Z. Angew. Mathematical Mechanics, 69, No.4, p. 67-69.
64. Marti K. (1996). Differentiation formulas for probability functions: The transformation method. // *Mathematical Programming*, 75(2), p. 201-220.
65. Marti K. (2005). Stochastic Optimization Methods. Springer: N.Y.
66. Masse P., Gibrat R. (1957). Application of linear programming to investments in the electric power industry. // *Management Science*, 3, p. 149–166.
67. Mikhalevitch V.S., Gupal A.M., Norkin, V.I. (1987). Methods of Nonconvex Optimization. Nauka: Moscow (in Russian).
68. Miller W., Leung L., Azhar T., Sargent S. (1997). Fuzzy production planning model for fresh tomato packing. // *International Journal of Production Economics*, 53, p. 227-238);
69. Mulvey J. M., Vladimirou H. (1989). Stochastic network optimization model for investment planning. // *Annals of Operations Research*, Vol. 20, p. 187.
70. Nayak R., Panda R. (2001). Integrated mangement of a canal command in a river delta using multi-objective techniques. // *Water Resources Management*, 15, p. 383-401.
71. Nelder J. A., Mead R. (1965). A Simplex Method for Function Minimization. // *Computer Journal*, vol. 7, p. 308-313.
72. Nemura A. (2007). Neaiškių sistemų panaudojimas elektros energetikoje // *Mokslas ir technika*, Nr. 7-8, p. 19, 34-36

73. Nesterov Y., Vial J.-Ph. (2001) Confidence level solutions for stochastic programming. Tech. rep., Department of Management Studies, University of Geneva.
74. Nurminski E.A. (1979). Numerical methods for solving deterministic and stochastic minimax problems. Naukova Dumka: Kiev (in Russian).
75. Nürnberg, R, Römisch, W. (2002). A two-stage planning model for power scheduling in ahydro-thermal system under uncertainty. // *Optimization in Engineering*, Vol. 3, p. 355–378.
76. Olsen P. (1976). Multistage stochastic programming with recourse: The equivalent deterministic problem. *SIAM Journal on Optimization*, 14, p. 495–517.
77. Parra M., Terol A., Uria M. (2001). A fuzzy goal programming approach to portfolio selection. // *European Journal of Operational research*, 133, p. 287-297.
78. Pickenss J., Hof J. (1991). Fuzzy goal programming in forestry: An application with special solution problems. // *Fuzzy Sets and Systems*, 39, p. 239-246.
79. Polyak B.T. (1987) Introduction to optimization. Translations Series in Mathematics and Engineering. Optimization Software, Inc., Publications Division, New York.
80. Powell W. B.. Topaloglu H. (2001). Stochastic Programming in Transportation and Logistics, A. Ruszczynski and A. Shapiro, Handbooks in OR & MS, Vol. 10, Elsevier Science B.V., p. 555 – 597.
81. Prekopa A. (1995). Stochastic Programming, Kluwer.
82. Rardin R. L. (1998). Optimization in Operations Research. Prentice Hall, New Jersey.
83. Rastrigin L. A. (1968). Statistical search methods. Nauka, Moscow. (In Russian).
84. Riis M., Andersen K. A.. (2004). Multiperiod capacity expansion of a telecommunications connection with uncertain demand. // *Computers & Operations Research*, 31, p. 1427–1436.
85. Robins H., Monro S. (1951). A stochastic approximation method. // *Annals of Mathematical Statistics*, vol. 22, No 3, p.400-407.
86. Rubinstein R, Shapiro A. (1993) Discrete events systems: sensitivity analysis and stochastic optimization by the score function method. Wiley & Sons, N.Y.
87. Sakalauskas L. (2000). Nonlinear stochastic optimization by Monte-Carlo method. // *Informatica*, vol. 11, No 4, p. 455-468.
88. Sakalauskas L. (2002). Nonlinear stochastic programming by Monte-Carlo estimators. // *European Journal on Operational Research*, vol. 137, p. 558 – 573.

89. Sakalauskas L. (2004). Application of the Monte-Carlo method to nonlinear stochastic optimization with linear constraints. // *Informatica*, 15(2), p. 271-282.
90. Sakalauskas L., Žilinskas K. (2005a). Monte Karlo metodo taikymas netiesiniam stochastiniam programavimui. // *Informacijos mokslai*, 34 tomas, p. 326-330.
91. Sakalauskas L., Žilinskas K. (2005b). Stochastinio tiesinio optimizavimo algoritmas Monte Karlo imčių serijomis. // *Informacinės technologijos – 2005*. Konferencijos pranešimų medžiaga, Kaunas, Technologija, T. 2, p. 425-431.
92. Sakalauskas L., Žilinskas K. (2005c). Stochastinio tiesinio optimizavimo Monte Karlo imčių serijomis algoritmas. // *Matematika ir matematinis modeliavimas 1*, p. 20-25.
93. Sakalauskas L., Žilinskas K. (2006a). Application of statistical criteria to optimality testing in Stochastic Programming“. // *Information Technology And Control*, Technologija, 35(1) tomas, p. 34-56.
94. Sakalauskas L., Žilinskas K. (2006b). Application of the Monte-Carlo Method to Stochastic Linear Programming. // *Series on Computers and Operations Research*, Vol. 7*, Computer Aided Methods in Optimal Design and Operations*, p. 39-48.
95. Sakalauskas L., Žilinskas K. (2006c). Two-stage Stochastic Linear Programming with recourse by series of Monte-Carlo estimators. // *Technological and economic development of economy*, XII tomas, Nr.4, p. 314-320.
96. Sakawa M., Nishizaki I., Uemura Y. (2002). A decentralized two-level transportation problem in a housing material manufacturer: Interactive fuzzy programming. // *European Journal of Operational Research*, 141, p. 167-185.
97. Santoso T., Goetschalckx M., Shapiro A. (2004).. A stochastic programming approach for supply chain network design under uncertainty.
98. Sen S. (2001) Stochastic Programming: Computational issues and Challenges. // *Encyklopedia of OR/MS*, Gass S., and Harris C. (eds.), Kluwer Academic Publishers, Dordrecht, p. 784-789.
99. Sen S., Mai J., Higle J. L. (1993). Solution of Large Scale Stochastic Programs with Stochastic Decomposition Algorithms, Technical Report, SIE Dept., University of Arizona, Tucson, AZ, 85721.
100. Shapiro A. (2000) Stochastic Programming by Monte-Carlo simulation methods. Stochastic Programming E-Print Series.
101. Shapiro A. (2002). Stochastic Programming by Monte Carlo Simulation Methods. Georgia Institute of Technology, Atlanta.

102. Shapiro A., Homem de Mello T. (2000). On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. // *SIAM Journal on Optimization*, 11, p. 70-86.

103. Shapiro A., Homem-de-Mello T. (1998). A simulation-based approach to two-stage stochastic programming with recourse. // *Mathematical Programming*, 81, p. 301–325.

104. Singh K. J., Philpott A. B., Wood R. K. (2004). Column-generation for survivable design of electricity distribution networks. In review, Department of Engineering Science, University of Auckland, New Zealand.

105. Spall J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. // *IEEE Transactions on Automatic Control*, vol.37, p. 332-341.

106. Spall J. C. (2003). Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control, Wiley, Hoboken, NJ.

107. Swart W., Smith C., Holderby T. (1975). Expansion Planning for a Large Dairy Farm, Chapter 8 in Studies in Linear Programming, H. Salkin and J. Saha, eds. North Holland, Amsterdam.

108. Tanaka H., Asai K. (1984). Fuzzy linear programming problems with fuzzy numbers. // *Fuzzy Sets and Systems,* 13, p. 1-10.

109. Tintner G. (1955). Stochastic linear programming with applications to agricultural economics. – Proceedings of the Second Symposium in Linear Programming, National Bureau of Standards, Washington, p. 197-228.

110. Wasan M.T. (1969). Stochastic Approximation. Cambridge. Transactions in Mathematics and Mathematical Physics. Cambridge University Press, Cambridge.

111. Wets R. J-B. (1974). Stochastic programs with fixed recourse: The equivalent deterministic program. // *SIAM Review,* Vol. 16, No. 3, p. 309–339.

112. Yasunori Fujikoshi, (1997). An asymptotic expansion for the distribution of Hotelling's T2-statistic under nonnormality. // *Journal of Multivariate Analysis*, vol. 61, p. 187-193.

113. Yudin D. B. (1965). Qualitative methods for analysis of complex systems (Количестенные методы анализа сложных систем). Izv. AN SSSR, ser. Technicheskaya Kibernetika, No 1, p. 3-13.

114. Yudin D. B. (1974). Mathematical methods of control in a case of uncertainty of information. Sov. Radio, Moscow (in Russian).

115. Zierer T.K., Mitchell W.A., White T.R. (1976). Practical applications of linear programming to Shell's distribution problems. // *Interfaces*, Vol. 6, no. 4, p. 13-26.
116. Žilinskas A. (2000). Matematinis programavimas, VDU leidykla, Kaunas, p. 169-194.
117. Žilinskas A., Zhigljavsky A. (1991). Methods of the global extreme searching. Nauka, Moscow. (In Russian).
118. Žilinskas K. (2007). Matematinis programavimas, I dalis. Tiesinis programavimas, Šiaulių universitetas, Šiauliai.
119. Zimmermann H. (1978). Fuzzy programming and linear programming with several objective functions. // *Fuzzy Sets and Systems*, 1, p. 45-55.
120. Zimmermann H. (1991). Fuzzy set theory and its applications, Boston: Kluwer Academic Publishers.