

ŠIAULIŲ UNIVERSITETAS  
INFORMATIKOS, MATEMATIKOS IR E.STUDIJŲ INSTITUTAS  
PROGRAMŲ SISTEMŲ KATEDRA

**Žydrūnas Jankevičius**

Informatikos specialybės II kurso ištęstinio skyriaus studentas

**STOCHASTINIO PROGRAMAVIMO UŽDAVINIŲ SPRENDIMAS**  
**KOMPIUTERINĖMIS MATEMATIKOS SISTEMOMIS**  
**THE SOLUTION OF STOCHASTIC PROGRAMMING PROBLEMS USING COMPUTER**  
**MATHEMATICS SYSTEMS**  
**MAGISTRINIS DARBAS**

Darbo vadovas:

Doc. K. Žilinskas

Recenzentas:

Doc. dr. G.Felinskas

Šiauliai, 2015

*„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąrašę pateiktus informacijos šaltinių bei savo tyrimų duomenis“*

Darbo autorius \_\_\_\_\_

(vardas, pavardė, parašas)

# Darbo tikslas ir uždaviniai

## Tikslas

Išanalizuoti kompiuterines matematikos sistemas (KMS) galimybes, sprendžiant stochastinio programavimo uždavinius.

## Uždaviniai

1. Sukurti projektus įvairiose KMS, kurie leistu spręsti stochastinio programavimo uždavinius Monte Karlo metodu.
2. Sudaryti pasirinkto stochastinio programavimo uždavinio KMS modelio sprendimo algoritmą (Monte Karlo metodu).
3. Realizuoti sprendimo algoritmą naudojant KMS Monte Karlo metodu.

Darbo vadovas \_\_\_\_\_

## **Turinys**

1. Įvadas .....	5
2. Teorinė dalis.....	7
2.1 Stochastinio programavimo uždavinys. ....	7
2.2 Tiesinio programavimo uždavinys .....	9
2.3 Netiesinio programavimo uždavinys.....	13
2.4 Dviejų etapų stochastinio programavimo uždavinys.....	14
2.5 Darbo srities analizė .....	16
2.6 Darbinės srities modelis .....	17
3. Projektinė dalis.....	21
3.1 Kompiuterinių matematikos sistemų pasirinkimo analizė .....	21
3.2 MathCad.....	21
3.3 MatLab .....	23
3.4 Maple.....	25
4. Darbo eigos aprašymas.....	28
5. Išvados.....	29
6. Literatūra .....	30
7. Anotacija .....	31
8. Summary .....	32
9. Priedai .....	33

# 1. Įvadas

Stochastinis programavimas, kaip tiesinio programavimo apibendrinimas, atsirado XX a. šeštajame dešimtmetyje dėl techninio, ekonominio ir finansinio planavimo poreikių. Dažniausiai pasitaikantys stochastinio programavimo taikymo sritys:

- 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 parametru 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.

Pagrindinės stochastinio programavimo uždavinių sprendimo problemos: sudėtingas tikslo funkcijos tikslų reikšmių apskaičiavimas bei gautojo taško priklausomumo leistinajai sričiai patikrinimas. Stochastiniame programavime reikia rasti uždavinio optimalų sprendinį, kai uždavinio tikslo funkcijos tikslų reikšmę optimaliame taške apskaičiuoti beveik neįmanoma ir kai yra ribojimai, kurių patikrinti taip pat beveik neįmanoma.

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

Kompiuterinės matematikos sistemos (Mathcad, Matlab, Maple Mathematica ir kt.) yra sukurtos tam, kad jų vartotojai (studentai, inžinieriai, mokslininkai, ir pan.) galėtų pakankamai paprastai atlikti reikalingus matematinius skaičiavimus. Kaip taisyklė, tokias sistemas yra žymiai paprasčiau įsisavinti, nei tradicines programavimo kalbas (C, C++, Java ir t.t.). Dėl šios priežasties kompiuterinės matematikos sistemos yra plačiai taikomos įvairiose srityse, kur reikalingi matematiniai skaičiavimai. Šių paketų pagalba galima atlikti kaip paprastą užduotį (pvz., išspręsti lygčių sistemą, išspręsti diferencialinę lygtį, nubraižyti funkcijos grafiką ir pan.), taip ir sudėtingą (pvz., išspręsti erdvinės santvaros topologijos optimizavimo uždavinį, panaudojant genetinį algoritmą).

Monte Karlo metodas – skaičiavimo algoritmas, pagrįstas statistiniu modeliavimu ir gautų rezultatų apdorojimu statistiniais metodais. Šis metodas leidžia brangiai kainuojančius bandymus

pakeisti modeliavimu kompiuteriais ir labai sumažina tyrimų trukmę. Monte Karlo metodai dažniausiai naudojami fizikinių ir matematinių sistemų modeliavimui, kai neįmanoma gauti tikslių rezultatų naudojant deterministinį algoritmą.

Norint atlikti labai sudėtingą skaičiavimą, reikalaujantį išanalizuoti didelę duomenų erdvę, galima tą patį skaičiavimą atlikti tik su keletu atsitiktinai pasirinktų duomenų. Atsitiktinai parinkti duomenys dažniausiai būna „tipiški“, todėl natūralu tikėtis, kad ir atliktas skaičiavimas ne itin daug skirsis nuo tikslaus. Pavyzdžiui, nežinodami kaip apskaičiuoti apskritimo, nubrėžto kompiuterio ekrane, plotą, galėtume atsitiktinai išdėlioti keliasdešimt taškų. Suskaičiavę, kokia dalis taškų pateko į apskritimą, galėtume apytiksliai pasakyti ir jo plotą. Žinoma, tam turime suprasti, kaip generuojami atsitiktiniai taškai.

## 2. Teorinė dalis

### 2.1 Stochastinio programavimo uždavinys.

Stochastinis programavimas yra matematinio deterministinio programavimo praplėtimas, kai į uždavinio matematinį modelį įvedama neapibrėžtis (atsitiktinumas). Tokie uždaviniai išskyla įvairiose srityse.

Daugelyje verslo ir technikos sričių išskyla uždavinių, susijusių su įvairaus pobūdžio duomenų neapibrėžtimi (nežinomais, nenumatytais kintamaisiais), kuriems spręsti taikomi stochastinio programavimo metodai. Stochastinis tiesinis programavimas – 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 stochastinis tiesinis programavimas uždaviniai formuluojami kaip stochastinio dinaminio programavimo uždaviniai. Vienas plačiausiai taikomų ir dažniausiai sprendžiamų stochastinis tiesinis programavimas uždavinių yra dviejų etapų stochastinio tiesinio programavimo uždavinys. Jei duomenų neapibrėžtis šiame uždavinyje yra aprašoma tolydžiuoju tikimybinio dėsnio, tai jis gali būti suformuluotas kaip netiesinio programavimo su tiesiniais ribojimais uždavinys [3].

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

$$\begin{aligned} f_i(x) &\leq 0, \quad i = 1, 2, \dots, m, \\ x &\in X \subset \mathfrak{R}^n \end{aligned}$$

čia  $f_0 : \mathfrak{R}^n \rightarrow \mathfrak{R}$  yra tikslo funkcija,

$f_i : \mathfrak{R}^n \rightarrow \mathfrak{R}, i = 0, 1, \dots, n$  – yra ribojimų funkcijos.

Jei matematinio programavimo uždavinys yra tiesinis, tikslo funkciją 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

$$\sum_{j=1}^n a_{ij} x_j - b_i \leq 0, \quad i = 1, 2, \dots, m;$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n.$$

Tegul stochastinis tiesinis programavimas uždavinio sprendinys priklauso nuo tam tikros tikimybinės erdvės (įvykių erdvės, atsitiktinių parametrų erdvės)  $(\Omega, F, P)$  elemento  $\omega$ . 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

$$\begin{aligned} f_i(x, \omega) \leq 0, \quad i = 1, 2, \dots, m, \\ x \in X \subset \mathfrak{R}^n \end{aligned}, \tag{2.2}$$

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

$$\begin{aligned} A(\omega)x - b(\omega) \leq 0, \\ x_i \geq 0, \quad i = 1, 2, \dots, n. \end{aligned}$$

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





koeficientai negali būti lygūs nuliui. Vadinasi, apribojimais nusakytoje srityje nėra ekstremumo taškų. Jie gali būti tik ant srities ribų (kraštų), bet ištirti kraštinius taškus negalime, nes funkcijos dalinės išvestinės yra konstantos.

Tiesinio programavimo uždavinių sprendimui reikalingi specialūs metodai.

Apibrėžimai

**1 apibrėžimas.** Bendrasis tiesinio programavimo uždavinys – tai funkcijos

$$f = \sum_{j=1}^n c_j x_j \quad (2.6)$$

maksimalios (minimalios) reikšmės radimas, kai patenkintos sąlygos:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, k; \quad (2.7)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = k + 1, k + 2, \dots, m; \quad (2.8)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, l, \quad l \leq n, \quad (2.9)$$

kur  $a_{ij}$ ,  $b_i$ ,  $c_i$  – duoti pastovūs dydžiai ir  $k \leq m$ .

**2 apibrėžimas.** Funkcija (2.6) vadinama (2.6) – (2.9) uždavinio tikslo funkcija, o (2.7) – (2.9) sąlygos vadinamos uždavinio ribojimais.

**3 apibrėžimas.** Standartinis tiesinio programavimo uždavinys – tai funkcijos maksimalios reikšmės radimas, kai patenkintos (2.7) – (2.9) sąlygos, kur  $k=m$  ir  $l=n$ :

$$f = \sum_{j=1}^n c_j x_j \rightarrow \max,$$

kai

$$\sum_{j=1}^n a_{ij} x_j \leq b_j, \quad i = 1, 2, \dots, m;$$

$$x_j \geq 0, \quad j = 1, 2, \dots, l, \quad l \leq n.$$

**4 apibrėžimas.** Kanoninis tiesinio programavimo uždavinys – tai funkcijos (2.6) maksimalios reikšmės radimas, kai patenkintos (2.8) ir (2.9), sąlygos kur  $k=0$  ir  $l=n$ .

$$f = \sum_{j=1}^n c_j x_j \rightarrow \max,$$

kai

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i=1,2,\dots,m;$$
$$x_j \geq 0, \quad j=1,2,\dots,l, \quad l \leq n.$$

**5 apibrėžimas.** Vektorius  $x = (x_1, x_2, \dots, x_n)$  patenkinantis (2.7) – (2.9) sąlygas, vadinamas uždavinio baziniu (leistiniu) sprendiniu.

**6 apibrėžimas.** Jei uždavinys neturi leistinųjų sprendinių, jis vadinamas neapibrėžtu.

**7 apibrėžimas.** Jei uždavinio tikslo funkcija yra neapibrėžta, jis vadinamas neapibrėžtu.

**8 apibrėžimas.** Bazinis sprendinys  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ , su kuriuo tikslo funkcija įgyja maksimalią (minimalią) reikšmę, vadinamas optimaliuoju sprendiniu.

Tiesinio programavimo uždavinio formuluotė

Bendrasis tiesinio programavimo uždavinys formuluojamas taip: rasti vektorių (skaičių rinkinį)  $X = x_1, x_2, \dots, x_n$  kuris tenkintų tiesinių lygčių ir nelygybių sistemą:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n} \leq b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n} \leq b_2, \\ \dots \\ a_{s1}x_1 + a_{s2}x_2 + \dots + a_{sn} \leq b_s, \\ a_{s+1,1}x_1 + a_{s+1,2}x_2 + \dots + a_{s+1,n} \leq b_{s+1} \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn} \leq b_m, \\ x_1 \geq 0, x_2 \geq 0, \dots, x_k \geq 0, k \leq n \end{cases} \quad (2.10)$$

ir su kuriuo tiesinė funkcija

$$f(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (2.11)$$

įgytų didžiausią (mažiausią) reikšmę.

Trumpai tokį uždavinį galima užrašyti taip:

Rasti

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (2.12)$$

kai

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, s; \quad (2.13)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = s+1, s+2, \dots, m; \quad (2.14)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, k, \quad k \leq n.$$

(2.14) sąlyga vadinama kintamojo neneigiamumo sąlyga arba neneigiamumo reikalavimu.

Vektorius (arba sprendinys)  $X = (x_1, x_2, \dots, x_n)$  tenkinantis visus tiesinio programavimo uždavinio apribojimus, vadinamas leistinuoju vektoriumi arba leistinuoju sprendiniu (planu). Visų leistinųjų sprendinių aibė sudaro uždavinio leistinųjų sprendinių sritį  $X$ . Leistinasis sprendinys  $X^*$ , su kuriuo tikslo funkcija srityje  $X$  įgyja didžiausiąją (mažiausiąją) reikšmę, vadinamas matematinio programavimo uždavinio sprendiniu arba optimaliuoju sprendiniu (planu). Tiesinio programavimo bendrasis uždavinys gali būti užrašytas dar trumpiau – matricų forma:

$$CX \rightarrow \max, \quad (2.15)$$

kai

$$AX = B;$$

$$A'X \leq B';$$

$$X \geq 0,$$

kur

$$C = (c_1, c_2, \dots, c_n);$$

$$X = (x_1, x_2, \dots, x_n);$$

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ a_{s1} & \dots & a_{sn} \end{pmatrix}; B = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_s \end{pmatrix}; \quad (2.16)$$

$$A' = \begin{pmatrix} a_{s+11} & \dots & a_{s+1,n} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ a_{m1} & \dots & a_{mn} \end{pmatrix}; B' = \begin{pmatrix} b_{s+1} \\ \cdot \\ \cdot \\ \cdot \\ b_m \end{pmatrix};$$

### Tiesinio programavimo uždavinio formos

Tam tikrais tiesinio programavimo uždavinio atvejais yra standartinis ir kanoninis uždaviniai. Jei ribojimus sudaro tik nelygybės ir visi kintamieji turi patenkinti neneigiamumo sąlygą ( $k=n$ ), tai turime standartinį tiesinio programavimo maksimizavimo uždavinį:

$$\begin{aligned} CX &\rightarrow \max \\ AX &\leq B \\ X &\geq 0 \end{aligned} \quad (2.17)$$

Jei ribojimus sudaro tik lygybės ir visi kintamieji turi patenkinti neneigiamumo sąlygą ( $k=n$ ), tai turime kanoninį tiesinio programavimo maksimizavimo uždavinį:

$$\begin{aligned} CX &\rightarrow \max \\ AX &= B \end{aligned} \quad (2.18)$$

Taisyklinga standartinio minimizavimo uždavinio užrašymo forma:

$$\begin{aligned} CX &\rightarrow \min \\ AX &\geq B \\ X &\geq 0 \end{aligned} \quad (2.19)$$

### 2.3 Netiesinio programavimo uždavinys

Bendru atveju netiesinio programavimo uždavinys yra rasti didžiausią (mažiausią) tikslo funkcijos reikšmę, kai tenkinamos tam tikros sąlygos:

$$f(x_1, x_2, \dots, x_n) \rightarrow \max(\min) \quad (2.20)$$

kai

$$\begin{cases} g_i(x_1, x_2, \dots, x_n) \leq b_i, i = 1, 2, \dots, k, \\ g_i(x_1, x_2, \dots, x_n) = b_i, i = k + 1, k + 2, \dots, m, \end{cases} \quad (2.21)$$

Kur  $f$  ir  $g_i$  yra žinomos  $n$  kintamųjų funkcijos, o  $b_i$  – duoti skaičiai.

Uždavinio sprendinys yra tam tikras taškas  $X^*(x_1^*, x_2^*, \dots, x_n^*)$  kurio koordinatės tenka (2.21) sąlygas bei bet kuriam kitam taškui  $X(x_1, x_2, \dots, x_n)$  didžiausios reikšmės atveju tenkinama nelygybė

$$f(x_1^*, x_2^*, \dots, x_n^*) \geq f(x_1, x_2, \dots, x_n), \quad (2.22)$$

o mažiausios reikšmės atveju tenkinama nelygybė

$$f(x_1^*, x_2^*, \dots, x_n^*) \leq f(x_1, x_2, \dots, x_n), \quad (2.23)$$

Priminsime, kad jei funkcijos  $f$  ir  $g_i$  yra tiesinės, tai (2.21) – (2.22) uždavinys yra tiesinis. (2.22) sąlygos sudaro ribojimų sistemą. Neneigiamumo sąlygos gali būti įtraukta į ją arba aprašytos tiesiogiai. Euklido erdvėje  $E_n$  ribojimų sistema nusako leistinųjų sprendinių sritį. Skirtingai nei tiesiniu atveju ši sritis ne visuomet yra iškila.

Kai duota leistinųjų sprendinių sritis, tai netiesinio programavimo uždavinio sprendimą sudaro tokio srities taško, per kurį eina didžiausio (mažiausio) lygio hiperplokštuma  $f(x_1, x_2, \dots, x_n) = h$  ieška. Toks taškas gali būti tiek ant leistinųjų sprendinių srities ribos, tiek ir srities viduje.

Netiesinio programavimo uždavinio sprendinio radimo geometrinio metodo algoritmas analogiškas algoritmui tiesiniame programavime [2].

## 2.4 Dviejų etapų stochastinio programavimo uždavinys

Labai dažnai greito 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  $\omega$ , bei vektorius  $b$  taip pat yra atsitiktinis, tai bendru atveju koks bebūtų sprendinys  $x$ , jis negali tenkinti ribojimų sistemos

$$A(\omega)x = b(\omega) \quad (2.24)$$

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

$y = (y_1, y_2, \dots, y_r)$  su ribojimų matricomis  $W(\omega)$ ,  $T(\omega)$  bei laisvųjų narių vektoriumi

$h(\omega) = (h_1(\omega), h_2(\omega), \dots, h_r(\omega))$  ir nagrinėjama sistema

$$\begin{cases} W(\omega)y + T(\omega)x = h(\omega), \\ x \geq 0, \quad y \geq 0, \end{cases} \quad (2.25)$$

kuri yra dviejų etapų stochastinio programavimo uždavinio ribojimų sistema. Vektorius  $x$  randamas prieš tai, kai sužinoma konkreti atsitiktinio parametro  $\omega$  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

duotiesiems  $x$  ir  $\omega$ . Pažymėjus šį vektorių  $y(x, \omega)$ , bendrosios išlaidos plano  $x$  realizavimui lygios

$$F(x) = \sum_{j=1}^n c_j x_j + E \left( \sum_{i=1}^r q_i y_i(x, \omega) \right), \quad (2.26)$$

kai  $x \geq 0$ .

Šiame darbe nagrinėjamas dviejų etapų stochastinio tiesinio programavimo uždavinys :

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

$$\text{kai} \quad Ax = b, \quad x \in \mathcal{R}_+^n,$$

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

Pointegrinė funkcija uždavinio išraiškoje gaunama sprendžiant tiesinio programavimo uždavinį

$$f(x, \xi) = \min_{\substack{Wy+Tx=h \\ y \in R_+^m}} q \cdot y, \quad (2.27)$$

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

## 2.5 Darbo srities analizė

Šios darbo temos tikslas stochastinio programavimo uždavinių sprendimas naudojant kompiuterines matematikos sistemos paketus. Šiems sprendimams atlikti reikia suprojektuoti ir realizuoti pasirinktų matematinių paketų šablonus, gebančius generuoti atsitiktinius dydžius, sudaryti atitinkamas ribojimų, resursų, tikslo funkcijos ir papildomas matricas, bei pateikti galutinius rezultatus. Realizavus suprojektuotą sprendimo algoritmą programiniu kodu pasirinktaisiais kompiuteriniais matematiniais paketais, atlikti rezultatų palyginimą ir analizę.

Dviejų etapų tiesinio stochastinio programavimo uždavinio formuluotė

Minimizuoti

$$cx + E(qy)$$

kai tenkinamos sąlygos

$$Ax = b,$$

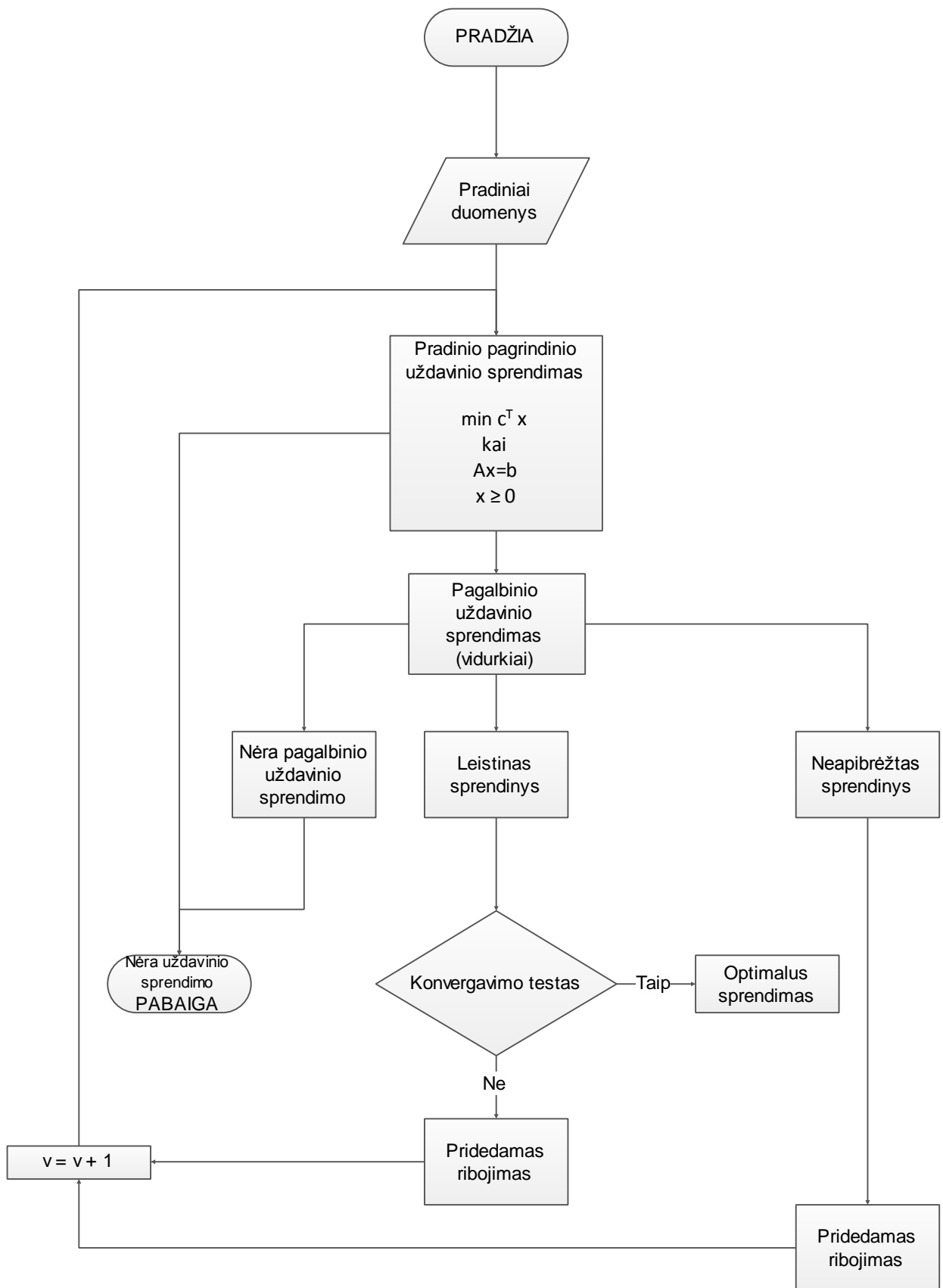
$$Tx + Wy = h,$$

$$x, y \geq 0,$$

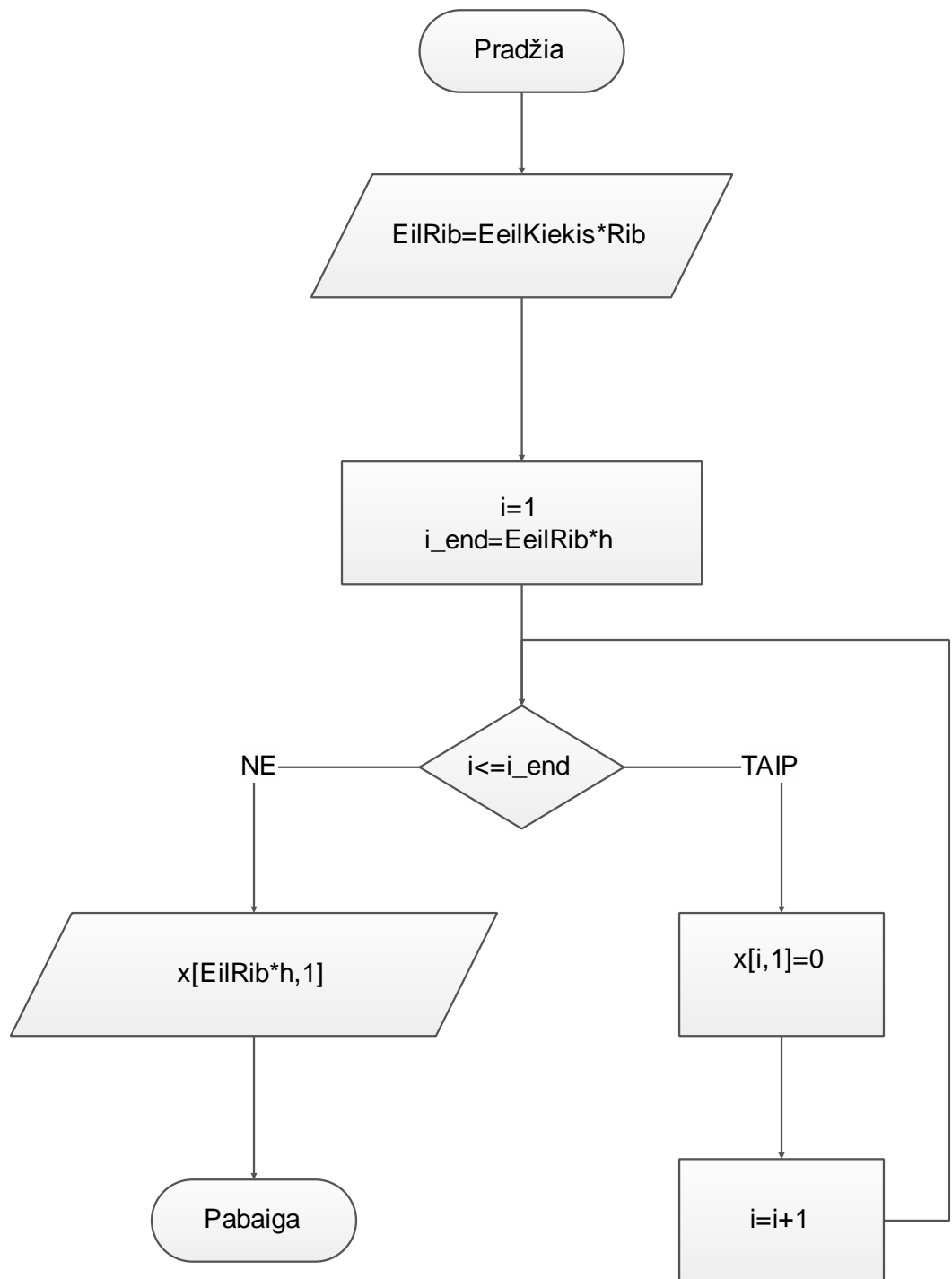
kur  $h$  yra tiesiškai nepriklausomi atsitiktiniai poreikiai, pasiskirstę pagal normalinį dėsnį su žinomais vidurkiais ir dispersijomis.



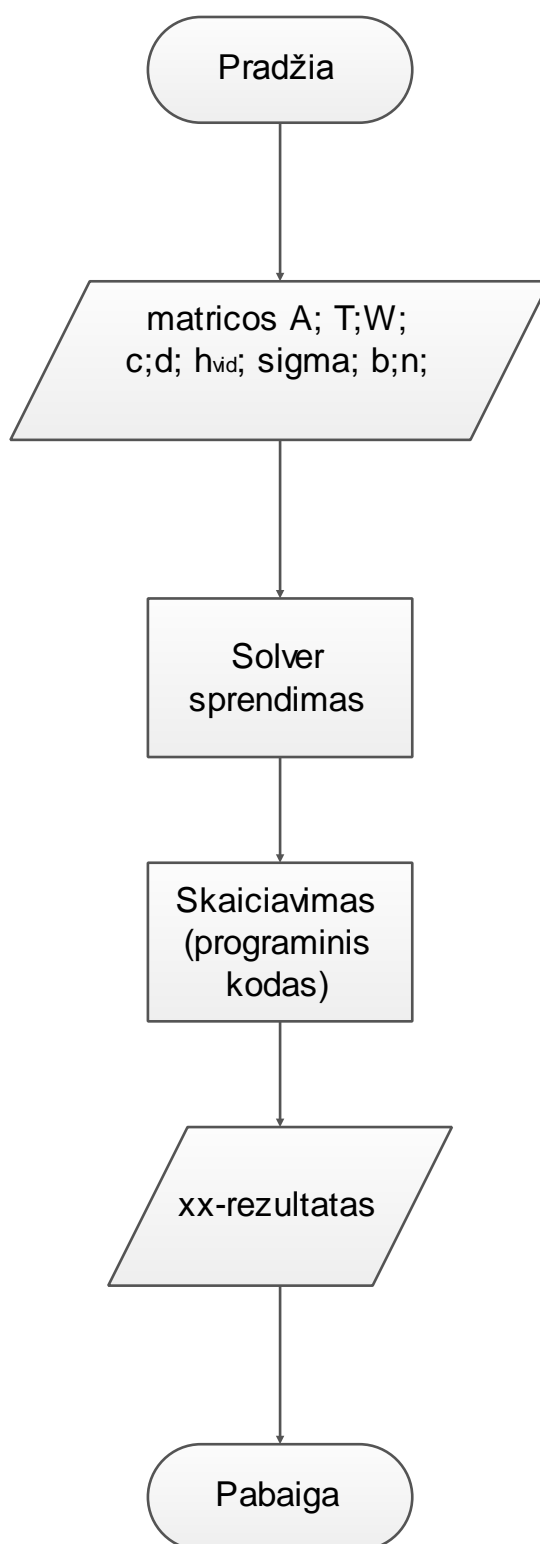
## 2.6 Darbinės srities modelis



1 pav. Sprendinio algoritmo schema

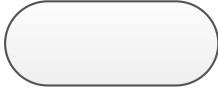


2 pav. Ribojimų matricos sudarymas



3 pav. Uždavinio pagrindinės dalys

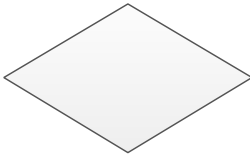
Vaizduojamų blokų ir jų dalių reikšmės blokinėse schemose:



**Pradžia – Pabaiga.** Žymi algoritmo pradžią ir pabaigą.



**Įvedimas ir Išvedimas.** Žymi algoritmo duomenų įvedimą arba išvedimą.



**Sąlyga.** Žymi algoritmo sąlygos tikrinimą. Turi vieną įeities tašką ir du išeities taškus, kurių vienas – TAIP, kitas - NE.



**Skaičiavimas.** Žymi algoritmą tam tikrą skaičiavimą pagal kintamuosius ir/arba nurodytą formulę.

## 3. Projektinė dalis

### 3.1 Kompiuterinių matematikos sistemų pasirinkimo analizė

Kompiuterinės matematikos sistemos pasirinktos PTC MathCad 14, MatLab 8.5 R2015, Maple 14 nes:

- MathCad labai paplitusi mokymosi įstaigose.
- Turi funkcijas ir įrankius reikalingus atlikti esamam darbui.
- MatLab turi didesnę lankstumą, kuris naudingas įvairiems fizikos, biologijos bei kitiems matematiniais modeliams kurti ir tirti.
- Visi pasirinkti matematiniai paketai veikia Windows OS sistemose nuo Windows XP iki Windows 8.

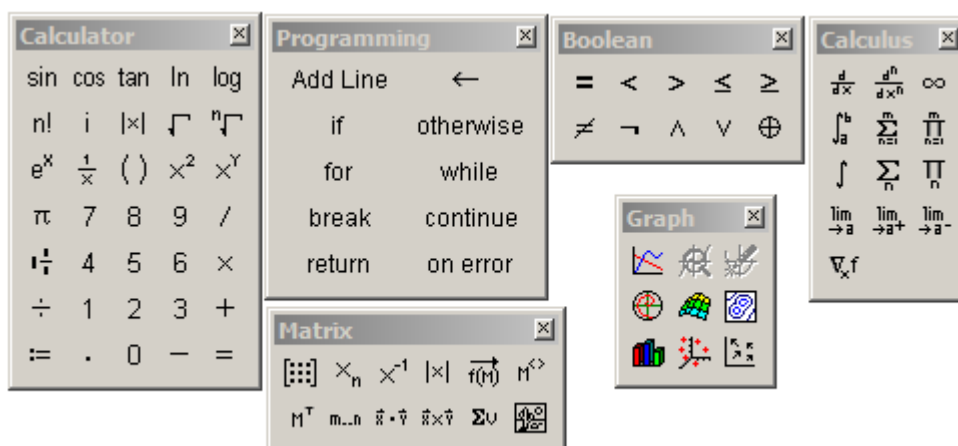
### 3.2 MathCad

MathCad – matematinis paketas, skirta matematiniais skaičiavimams atlikti. Ši programa panaši į skaičiuoklę, turi intuityvią vartotojui draugišką sąsają. Jos galimybės apima įvairias matematikos šakas: kombinatoriką, tikimybių teoriją, algebrą, geometriją ir t. t.

Palyginus su *MatLab* ir *Maple*, *MathCad* pasižymi lengviau išmokstama ir paprasčiau naudojama vartotojo sąsaja, todėl pirmieji žingsniai būna itin lengvi. Tačiau kuriant sudėtingesnius modelius (bent jau biologijoje) gana greitai ši patogi sąsaja nebepadeda: sudėtingus, realius mokslo uždavinius spręsti skirtos sistemos ne paprastesnės kaip *MatLab* ar net tradicinės programavimo kalbos. Paketas turi ir programavimo galimybes, tačiau vartotojo programų darbo sparta ribota, nes tai yra interpretuojanti sistema. Funkcijas MathCad galima rašyti ir C (DLL bibliotekos forma), tačiau paprastai darbo grupė, ėmusi programuoti C pastebi, jog paprasčiau šia kalba užprogramuoti visą užduotį, išskyrus nebent rezultatų vizualizacijai.

Daug geriau *MathCad* pasiteisina tradicinėmis kalbomis sukurtų modelių rezultatų vizualizavimui, kurį lengva derinti su baigiamąja statistine analize ir kitomis panašiomis operacijomis. Šiuo atveju patys modeliai gali būti vykdomi ir superkompiuteriuose, naudojant lygiagretų programavimą (su *MathCad* tai nėra galima). Draugiška sąsaja šiuo atveju naudojama greitai keisti informacijos pateikimo būdus. Tokiu būdu efektyviai išvengiama daug darbo reikalaujančio grafikos programavimo, kuris biologui ar fizikui didelės fundamentalios mokslinės vertės neturi.

MathCad aplinkoje dažniausiai naudojamos įrankių juostos:



4 pav. MathCad įrankių juostos (dažniausiai naudojamos).

- *Calculator* – įrankių juosta skirtas atlikti standartinius skaičiavimo veiksmus.
- *Programming* – įrankių juosta programuoti MathCad programavimo sąsajoje.
- *Boolean* – įrankių juosta atlikti veiksams su nelygybių sistemomis.
- *Matrix* – įrankių juosta atlikti veiksams su matricomis .
- *Graph* – įrankių juosta grafikams atvaizduoti.

#### Privalumai:

- Intuityvi aplinka.
- Paprasta naudoti.
- Nesudėtingas kintamųjų aprašymas (sintaksė).
- Labai paplitęs mokymo įstaigose.
- Lengvas rezultatų atvaizdavimas grafikuose.

#### Trūkumai:

- Paskutinė versija 15.0 (nebeatnaujinama).
- Netinkama kuriant sudėtingus modelius (pvz. biologijoje).
- Nepalaiko lygiagrečiųjų skaičiavimų.

#### MathCad versijos:

- MathCad 2.52      1989 m.
- MathCad 3.1        1992 m.

- MathCad 6.0      1995 m.
- MathCad 13.0    2005 m.
- MathCad 14.0    2007 m. (dirbta šia versija)
- MathCad 15.0    2010 m.

**Naujausioje versijoje papildomos ar pagerintos funkcijos:**

- Dokumento įforminimas – daugiau galimybių teksto maketavimui ir formatavimui.
- Stabilesnis veikimas – suderinama su naujesnėmis Windows OS.
- Kalbos – yra išversta (palaiko) į anglų, vokiečių prancūzų italų, ispanų japonų tradicines kinų ir korėjiečių kalbas.
- Koduote – palaiko (galutinai sutvarkyta) Unicode.
- Galimybė migruoti iš MathCad 14 į MathCad 15.
- 

### 3.3 MatLab

MatLab (iš žodžių MATrix LABoratory) yra daugiaplatformė MathWorks programinė įranga, skirta įvairių mokslo šakų problemoms spręsti, o ypač matematinėms. Kaip galima spręsti iš pavadinimo, turi puikias galimybes darbams su matricomis – būtent toks buvo pirminis šios programos tikslas. Dabar tai didelis galingas paketas, turintis pakankamai lengvai perprantamą programavimo kalbą.

Nors simbolinėms matematinėms manipuliacijoms geriau tinka Maple ir Mathematica produktai, su MathCad pradantis vartotojas gali greičiau gauti pirmuosius rezultatus, MatLab kaip programavimo kalba turi didesnę lankstumą, kuris naudingas įvairiems fizikos, biologijos bei kitiems matematiniais modeliams kurti ir tirti. Svarbus MatLab trūkumas (palyginus su tradicinėmis programavimo kalbomis) yra reikalavimas visiems naudojantiems turėti įkeltas mokamas MatLab bei visų reikalingų jo papildomų modulių kopijas. Tradicinės programavimo kalbos šiuo metu turi atviro kodo ar nemokamus kompiliatorius bei pasiekia didesnę vykdymo greitį, tačiau programavimas jomis ilgiau užtrunka. Bioinformatikiniuose tyrimuose Matlab neretai naudojamas kaip „apžvalgos“ paketas: pasiteisinę matematiniai modeliai vėliau perrašomi C (neretai generuojant žymias kodo dalis su Maple) lygiagrečiam vykdymui superkompiuteriuose.

**Privalumai:**

- Aukšto lygio programavimo kalba techniniams skaičiavimams.
- Interaktyvūs įrankiai duomenų tyrimui, vizualizavimui ir problemų sprendimui.
- Matematinės funkcijos skirtos tiesinei algebrai, statistikai, filtravimui, optimizavimui ir skaičių integravimui.
- Dviejų ir trejų dimensijų funkcijos duomenų vizualizavimui.
- Įrankiai skirti kurti savo vartotojo sąsają.
- Funkcijos skirtos integruoti MatLab pagrindu algoritmus su kitomis išorinėmis sistemomis.

**Trūkumai:**

- Pradedančiajam reikia turėti programavimo pagrindus.
- Darbo įrankiai nėra taip gerai išdėstyti kaip MatCad'e.

**MatLab versijos:**

- MatLab 1.0            1984 m.
- MatLab 2.0            1986 m.
- MatLab 3.0            1987 m.
- ....
- MatLab 7.0            2004 m.
- MatLab 8.0            2012 m.
- MatLab 8.5            2015 m.

**Naujausioje versijoje papildomos ar pagerintos funkcijos:**

## Pagrindinės funkcijos

- Interaktyvi aplinka uždaviniams tirti, dizainas, ir problemų sprendimas.
- Matematinės funkcijos tiesiniai algebros spęsti, statistika, Furjė analizė, filtravimas, optimizavimas, skaitinio integravimas, ir diferencialinė lygtis.
- Built-in grafika, duomenų vizualizavimas.
- Programavimo įrankiai, padidinti našumą
- Įrankiai kurti grafiniai aplinkai (sąsajoms).
- Funkcijos integruojant MATLAB algoritmus su išorinėmis programomis ir kalbomis, tokiomis kaip C, Java, .NET ir Microsoft Excel®



### Skaičių apskaičiavimas

- Interpoliacijos ir regresijos.
- Diferencijavimas ir integravimas.
- Linijinės sistemos lygtis.
- Furjė analizė.
- Paprastųjų Diferencialinė lygtis.
- Sparse matricos.

### Duomenų analizė ir vizualizavimas

- Pagerintas 2D ir 3D duomenų atvaizdavimas
- Didelis pasirinkimas šablonų (grafikų) duomenims atvaizduoti.

### Programavimo ir algoritmų pagerinimas

- Komandinė eilutė - leidžia interaktyviai įvesti duomenis, vykdyti komandas bei matyti programas rezultatus.
- MATLAB redaktorius - teikia redagavimo ir derinimo funkcijas, pavyzdžiui, nustatant ribines vertes ir peržiūrint atskirų eilučių kodus.
- Kodas analizatorius - automatiškai patikrina kodą ir rekomenduoja pakeitimus.

### Aplikacijų kūrimas ir diegimas

- Įrankiai individualiems algoritmams kurti.
- Grafinės sąsajos vartotojui kurti.
- Galimybė diegti ir bendrinti aplikacijas kitiems vartotojams
- MATLAB'o kodą sugeneruoti į C kodą.

## 3.4 Maple

Maple – daugiaplatformis *Waterloo Maple Software* programinis produktas matematiniam skaičiavimams atlikti. Pirmoji versija išleista Ontarijuje Kanadoje 1981 m. Nuo 1988 m. programą komerciniais tikslais kūrė Waterloo Maple Inc. (dar žinoma *Maplesoft* pavadinimu). Paskutinė versija yra Maple 15.

Maple turi labai gerą (vieną geriausių) sistemą įvairioms paprastoms bei diferencialinėms lygtims, jų sistemoms analitiškai spręsti, bei atlikti veiksmus su matricomis. Maple taip pat turi galimybes funkcijų grafikams bei kitai grafiškai informacijai pateikti. Aritmetiniai apskaičiavimai gali būti vykdomi bet koku norimu tikslumu (gali būti šimtai ar tūkstančiai ženklų po kablelio).

Tačiau tiek didelio, tiek ir mažesnio tikslumo aritmetiniai skaičiavimai Maple vykdomi sąlyginai lėčiau ir tai nėra stiprioji šio paketo pusė.

Skirtingai nuo *MathCad*, kuris riboja sukurtų algoritmų perkėlimo galimybes, Maple galima užrašyti bei perskaityti lygtis ir „gryno teksto“ forma, kuri panaši į *FORTRAN*.

Iš gautų sprendinių prireikus generuojamas C ar *FORTRAN* kodas, kurį lengva pritaikyti ir C++ bei Java programose. Skirtingai nuo *MATLAB* ir *MathCad*, kurie patys siūlo skaitmeninio sprendimo galimybes, *Maple* prireikus yra geras įrankis kuriant tradicine programavimo kalba parašytą skaitmeninio užduoties sprendimo programą. Tokie sugeneruoti fragmentai naudingi ir tada, jei kuriama programa bus vykdoma vienu metu daugeliu procesorių superkompiuteryje, arba jei sprendimo algoritmas, jo duomenų struktūros yra tokios, jog ir naudojant „palengvinto programavimo“ paketus programa neužrašoma pastebimai paprasčiau. Šis metodas efektyvus dirbant su matricomis, nes šiuo atveju glaustai užrašomas matricas turintis reiškinys „išskleidžiamas“ į matricų bibliotekos nenaudojantį kodą bei neretai taip pat randami analitiniai matricų inversijos ir kiti sprendiniai. Maple sugeneruotas kodas naudojamas ir taikomiosiose, galutiniam vartotojui skirtose programose.

Maple komandų sistema savo sudėtingumu prilygsta eilinei programavimo kalbai ir jai gerai išmokti reikia maždaug tiek pat laiko. Tačiau kadangi šį paketą neretai naudoja jau programuoti mokantys žmonės, tai dideliu trūkumu nelaikoma.

#### **Privalumai:**

- Didelė įvairių duomenų vizualizacijos galimybės (2D, 3D).
- Ranka rašomų simbolių atpažinimas.
- Turi sąsają su duomenų bazėmis.
- Pradedančiajam didelis uždavinių šablonų kiekis.
- Leidžia uždavinius spręsti dideliu tikslumu.

#### **Trūkumai:**

- Sudėtingesniems ir nestandartiniams uždaviniams spręsti tenka naudotis programavimo galimybėmis.
- Pradedančiajam reikia turėti programavimo pagrindus.
- Skaičiavimai atliekami lėčiau lyginant su kitomis programomis (*MatLab*, *Matematica* ir t.t.)
- Darbo įrankiai nėra taip gerai išdėstyti kaip *MatCad*’e.

### **Maple versijos:**

- Maple 1.0            1981 m.
- Maple 3.3           1985 m. (pirmoji vieša versija).
- ...
- ...
- Maple 14            2010 m. (dirbta šia versija)
- Maple 15            2011 m.
- Maple 16            2012 m.
- Maple 17            2013 m.
- Maple 18            2014 m.
- Maple 19            2014 m.
- Maple 2015         2015 m.

### **Naujausioje versijoje papildomos ar pagerintos funkcijos:**

- Duomenų rinkimas – suteikia galimybę rinkti duomenis realiu laiku ir dideliais kiekiais ir iš karto analizuoti.
- Duomenų atvaizdavimas – atvaizduoti grafikus (2D ir 3D) iš vienos eilutės .
- MapleCloud – galimybė projektą talpinti į debesis ir jį bendrinti.
- Nauji įrankiai - turi interaktyvios mokymo priemonės, kurios gali būti naudojamos iširti sąvokas ir mokyti studentus.
- Matematinės funkcijos – suteikia dar daugiau įrankių matematinėms funkcijos tirti bei skaičiavimams atlikti.
- Statistika – platesnės galimybės statistikos uždaviniams spręsti, bei duomenims atvaizduoti.
- Kodo generavimas – Maple kodą sugeneruoti į R ir JavaScript.
- Importas eksportas – importas iš XML, GPX, SHP, DGML, TGF, GXL, Pajek.
- Kompiuterių tinklas (Grid) – palaiko galimybę atlikti lygiagrečiuosius ir paskirstytus skaičiavimus.

## **4. Darbo eigos aprašymas**

Suprojektuotas stochastinio programavimo uždavinių sprendimo algoritmas blokinėmis schemomis pagal nagrinėtą stochastinio programavimo uždavinio matematinį modelį. Darbo aprašyme pateikiamos trumpos teorinės ir praktinės žinios reikalingos atlikti šiam darbui.

Pagal suprojektuotą algoritmą atliktas matematinio paketo MathCad, MatLab ir Maple sprendimo algoritmas stochastinio programavimo uždaviniams spręsti.

Atliktas skaičiavimas pasirinktųjų matematinų paketų šablonų prie vienodų pradinių duomenų (sugeneruojami atsitiktiniai dydžiai visiems pasirinktiesiems paketams vienodi).

## 5. Išvados

Iškelti tikslai pasiekti ir uždaviniai įvykdyti. Išanalizuota kompiuterinės matematikos sistemos (MathCad, MatLab, Maple) galimybės, sprendžiant stochastinio programavimo uždavinius, sudarytas ir realizuotas sprendimo algoritmas.

Visuose pasirinktuose KMS galima spręsti stochastinio programavimo uždavinius Monte Karlo metodu. Skirtinguose KMS uždavinio sprendimui laiko ir žinių reikia ne vienodai. MathCad'e išspręsti uždavinį nereikia turėti didelių programavimo žinių ir įgūdžių, skirtingai nei Maple ar MatLab. Mathcad yra intuityvi aplinka, paprasta naudotis, nesudėtingas kintamųjų aprašymas (sintaksė), bei lengvas rezultatų atvaizdavimas grafikuose. MatLab ir Maple uždaviniams spręsti reikia turėti programavimo žinių ir įgūdžių, bet perpratus atsiveria didesnės ir lankstesnės galimybės nestandartinėms užduotims spręsti.

## 6. Literatūra

1. G.Steponauskas, Monte karlo metodas, 2008.
2. K. Žilinskas, Stochastinis programavimas, paskaitų konspektas, Šiauliai, 2010.
3. K. Žilinskas, Stochastinio tiesinio programavimo Monte Karlo metodu tyrimas, Daktaro disertacija, Vilnius, 2009.
4. L. Sakalauskas, Statistinis modeliavimas ir analizė, mokomoji knyga, Šiauliai, 2013.
5. User's guide, Matchad 14.0, 2007.
6. User's manual Maple 14, 2010.
7. Т.И. Савелова (2011). Метод Монте-Карло. Москва, 152 с.
8. Михайлов Г. А., Войтишек А. В. Метод Монте-Карло (расширенный лекционный курс). 246с.

## **7. Anotacija**

Darbo tema: „Stochastinio programavimo uždavinių sprendimas kompiuterinėmis matematikos sistemomis“.

Šio darbo tikslas išanalizuoti kompiuterines matematikos sistemas (KMS) galimybes, sprendžiant stochastinio programavimo uždavinius Monte Karlo metodu. Šiam darbui atlikti suprojektuojamas sprendimo algoritmas pagal stochastinio programavimo matematinį modelį. Sudarytam sprendinio algoritmui realizuojamas programinis kodas MathCad, MatLab, Maple matematiniuose paketuose.

## **8. Summary**

The topic: “The solution of stochastic programming problems using computer mathematics systems”.

The aim of this analysis of computer mathematics systems (CMS) technology to solve stochastic programming problems the Monte Carlo method. This work will be designed in accordance with a decision algorithm stochastic mathematical programming model. Composed of the solution algorithm is realized software code MathCad, MatLab, Maple mathematical package.



## 9. Priedai

### 1. Kompaktinis diskas, kuriame yra:

- Matematinio paketo Mathcad sprendimo šablonas;
- Matematinio paketo MatLab sprendimo šablonas;
- Matematinio paketo Maple sprendimo šablonas;
- Diagramos \*.jpg, \*.edx formatais;
- E-šaltinių kopijos;
- Darbo aprašymas \*.pdf, \*.doc ir \*.xdoc formatais;
- Skaidrės \*.ppt, \*.pptx, \*.pps, \*.ppsx.

### 2. Programos, kuriomis atidaromi bylos (failai):

- \*.doc – Microsoft Word 2003;
- \*.docx – Microsoft Word 2007 ir aukštesnės versijos;
- \*.pdf – Adobe Acrobat Reader;
- \*.xmcd – PTC MathCad 14.0;
- \*.m – MatLab;
- \*.edx – Edraw Max 6.8;
- \*.ppt – Microsoft PowerPoint 2003;
- \*.pptx – Microsoft PowerPoint 2007 ir aukštesnės versijos;
- \*.pps – Microsoft PowerPoint 2003;
- \*.ppsx – Microsoft PowerPoint 2007 ir aukštesnės versijos;

3. Kintaieji iš MathCad aplinkos:

$$\underline{A} := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \underline{T} := \begin{pmatrix} 2 & 6 \\ 3 & 3.4 \end{pmatrix} \quad \underline{W} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \underline{c} := \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad \underline{q} := \begin{pmatrix} 7 \\ 12 \end{pmatrix}$$

$$h_{\text{vid}} := \begin{pmatrix} 180 \\ 162 \end{pmatrix} \quad \text{sigma} := \begin{pmatrix} 12 \\ 9 \end{pmatrix} \quad b := 100 \quad \underline{m} := 2$$

$$x_m := 0 \quad y_m := 0 \quad x = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$N_{\min} := 20 \quad N_{\max} := 1000$$

$$\underline{F}(x, y) := c \cdot x + q \cdot y$$

Given

$$A \cdot x \leq b \quad T \cdot x + W \cdot y \geq h_{\text{vid}} \quad x \geq 0 \quad y \geq 0$$

$$\text{pradinis} := \text{Minimize}(F, x, y)$$

$$u_m := 0 \quad \text{eps} := 2 \quad z := 0.001$$

$$f(u, hh, X) := (hh - T \cdot X) \cdot u$$

Given

$$W^T \cdot u \leq q \quad u \geq 0$$

$$\text{dual}(hh, X) := \text{Maximize}(f, u)$$

$$gg_1 := 1$$

$$gg_2 := 1$$

4. Programos kodas iš MathCad:

```

skaiciavimas:=
  x ← pradinis1
  tikslumas ← 100
  AA ← 20
  sk ← 0
  N ← Nmin
  while [(tikslumas > eps) ∧ ((AA > qF(0.95, m, N - m)))]
    sk ← sk + 1
    for k ∈ 1..m
      bbsk, k ← xk
    for j ∈ 1..N
      for k ∈ 1..m
        hk ← norm(1, hvidk, sigmaak)1
      u ← dual(h, x)
      d ← c - T·u
      for k ∈ 1..m
        aaj, k ← dk
      tf ← f(u, h, x) + c·x
      aaj, m+1 ← tf
    for k ∈ 1..m
      gk ← mean(aa⟨k⟩)
  Tf ← mean(aa⟨m+1⟩)

```

```

Tf ← mean(aa⟨m+1⟩)
bbsk,m+1 ← Tf
D ← Stdev(aa⟨m+1⟩)
for ii ∈ 1..m
  for jj ∈ 1..m
    Zii,jj ← cvar(aa⟨ii⟩, aa⟨jj⟩)
ZZ ← |Z|
bbsk,m+5 ← ZZ
Z1 ← Z-1 if ZZ ≥ 10-9
Z1 ←  $\frac{\text{identity}(m)}{100}$  otherwise
AAA ← gT·Z1·g
AA ← AAA ·  $\frac{(N-m)}{m}$ 
bbsk,m+3 ← AA
x ← x - z·g
bbsk,m+6 ← 0 if A·x ≤ b
bbsk,m+6 ← -1 otherwise

```

$$\text{tikslumas} \leftarrow \text{qnorm}(0.95, 0, 1) \cdot \frac{D}{\sqrt{N}}$$

$$\text{bb}_{\text{sk}, m+2} \leftarrow \text{tikslumas}$$

$$\text{NN} \leftarrow m \cdot \frac{\text{qF}(0.95, m, N - m)}{z \cdot \text{AAA}} + m$$

$$\text{bb}_{\text{sk}, m+7} \leftarrow \text{NN}$$

$$N \leftarrow \text{NN}$$

$$N \leftarrow N_{\text{max}} \quad \text{if } \text{NN} \geq N_{\text{max}}$$

$$N \leftarrow N_{\text{min}} \quad \text{if } \text{NN} \leq N_{\text{min}}$$

$$\text{bb}_{\text{sk}, m+4} \leftarrow N$$

bb

5. Skaičiavimo rezultatas:

xx := skaiciavimas

	1	2	3	4	5	6
1	32.143	19.286	195.362	23.036	16.837	$1 \cdot 10^3$
2	32.184	19.314	195.999	4.267	696.199	$1 \cdot 10^3$
3	32.224	19.341	191.452	3.85	652.291	$1 \cdot 10^3$
4	32.262	19.366	196.06	4.121	720.985	$1 \cdot 10^3$
5	32.302	19.393	188.179	3.81	592.892	$1 \cdot 10^3$
6	32.338	19.418	195.418	4.262	597.437	$1 \cdot 10^3$
7	32.375	19.443	184.833	3.761	591.178	$1 \cdot 10^3$
xx = 8	32.411	19.468	182.431	3.634	549.434	$1 \cdot 10^3$
9	32.445	19.491	181.96	3.65	563.959	$1 \cdot 10^3$
10	32.481	19.515	181.344	3.621	519.109	$1 \cdot 10^3$
11	32.515	19.538	181.29	3.584	511.206	$1 \cdot 10^3$
12	32.548	19.561	179.388	3.609	441.284	$1 \cdot 10^3$
13	32.58	19.582	178.208	3.636	427.43	$1 \cdot 10^3$
14	32.611	19.603	177.189	3.438	459.826	$1 \cdot 10^3$
15	32.642	19.624	176.099	3.412	422.775	$1 \cdot 10^3$
16	32.673	19.645	176.891	3.383	447.63	...