

**LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTISKĀS ANALĪZES KATEDRA**

DIPLOMDARBS

**MODIFICĒTĀ SIMPLEKSA METODE KVAADRĀTISKĀS
PROGRAMMĒŠANAS UZDEVUMA RISINĀŠANAI
UN TĀS PIELIETOJUMS
OPTIMĀLĀ INVESTORA PORTFEĻA NOTEIKŠANAI**

Juris Breidaks

Darba vadītāja: Dr.math., asoc.prof.

S. Asmuss

**Rīga
2006**

Anotācija

Šajā darbā ir apskatīts optimālā investora portfeļa uzdevums. Šis uzdevums pēc nostādnes ir kvadrātiskās programmēšanas uzdevums. Ar Vulfa – Daugaveta metodes palīdzību uzdevums tiek reducēts uz lineārās programmēšanas uzdevumu ar dažiem nelineāriem papildus nosacījumiem. Uzdevuma risināšanai tika uzrakstīta programma, kas realizē simpleksa algoritma modifikāciju, kura ievēro nelineāros nosacījumus.

Annotation

This work dwells upon the task of the Optimum Investment Portfolio. According to its formulation, it is the quadratic programming problem. With the help of Wolf – Daugavet method this problem is transformed into the linear programming problem with several additional non-linear conditions. To solve the resulting problem we offer program that contains simplex – algorithm with the modification taking into account non-linear conditions.

Аннотация

В этой работе рассматривается задача нахождения оптимального портфеля инвестора. По постановке эта задача является задачей квадратичного программирования. С помощью метода Вульфа-Даугавета она сводится к задаче линейного программирования с несколькими дополнительными нелинейными условиями. Для решения этой задачи была составлена программа, которая реализует модификацию симплекс-алгоритма, учитывающую нелинейные условия.

Satura rādītājs

1. Ievads.....	6
1.1. Uzdevuma apraksts	6
1.2. Modeļa apraksts	6
1.3. Darba struktūra.....	7
2. Vērtspapīru portfeļu teorija.....	8
2.1. Kas ir vērtspapīri?	8
2.2. Ienesīgums un risks	8
2.3. Vērtspapīru portfeļa jēdziens	10
2.4. Minimālā riska portfeļa noteikšana	12
3. Kvadrātiskās programmēšanas uzdevums	13
3.1. Matemātiskās programmēšanas uzdevums	13
3.2. Lineārās programmēšanas uzdevums	14
3.3. Kvadrātiskās programmēšanas uzdevums	15
3.4. Optimālā investora portfeļa uzdevuma pieraksts kvadrātiskās programmēšanas uzdevuma kanoniskā formā	16
4. Vulfa – Daugaveta metode kvadrātiskās programmēšanas uzdevuma risināšanai.....	17
4.1. Izliktās programmēšanas uzdevums	17
4.2. Kvadrātiskās programmēšanas uzdevuma atrisinājuma eksistence	19
4.3. Lagranža funkcija.....	20
4.4. Kvadrātiskās programmēšanas uzdevuma atrisinājuma eksistences nepieciešamie un pietiekamie nosacījumi	21
4.5. Vienādojumu sistēma kvadrātiskās programmēšanas uzdevuma atrisināšanai.....	23
4.6. Vulfa – Daugaveta metodes apraksts	24
4.7. Palīguzdevums kvadrātiskās programmēšanas uzdevuma atrisināšanai	24
5. Simpleksa metode	27
5.1. Simpleksa metodes teorētiskie pamati.....	27
5.2. Simpleksa metodes algoritms	29
5.3. Modificētā simpleksa metode	34
6. Programmas	35
6.1. Lineārās programmēšanas uzdevuma risināšanas programma	35
6.2. Kvadrātiskās programmēšanas uzdevuma risināšanas programma.....	41
Secinājumi	47
Izmantotā literatūra	48
Pielikumi	49

1. Ievads

1.1. Uzdevuma apraksts

Daudzus plānošanas un projektēšanas uzdevumus iespējams formulēt kā kvadrātiskās programmēšanas uzdevumus. Apskatīsim minimālā riska portfeļa uzdevumu. Pieņemsim, ka investors vēlas izveidot vērtspapīru portfeli. Veidojot portfeli, investoram interesē gan ienesīgums, gan risks.

Literatūrā ir aprakstīts optimālā investora portfeļa uzdevums, kas ir kvadrātiskās programmēšanas uzdevums ar lineāriem ierobežojumiem. Ir zināms, ka investors, veidojot portfeli līdzekļu iegūšanai tālākām investīcijām, nevar pārdot vērtspapīru, kas tam nemaz nepieder. Šo uzdevumu ir iespējams uzrakstīt sekojošā veidā:

$$\begin{aligned} \sigma_p^2 &= \sum_{i=1}^H \sum_{j=1}^H x_i x_j \sigma_{ij} \rightarrow \min \\ &\begin{cases} 0 \leq x_i \leq 1, & i = 1, 2, \dots, H \\ \sum_{i=1}^H x_i = 1 \end{cases} \end{aligned} \quad (1.1.1)$$

Darba mērķis ir uzrakstīt datorprogrammu, kura risina kvadrātiskās programmēšanas uzdevumus, kā arī pielietot šo programmu, lai minimizētu investora portfeļa risku.

1.2. Modeļa apraksts

Apzīmēsim ar

i – vērtspapīra veidu, j – vērtspapīra veidu, t – laika perioda indeksu, T – laika periodu, H – vērtspapīru skaitu.

Ja R_{it} ir i -tā veida vērtspapīra ienesīgums t -tajā laika periodā,

tad $\bar{R}_i = \frac{1}{T} \sum_{t=1}^T R_{it}$ – i -tā veida vērtspapīra vidējais ienesīgums T -tā laika periodā.

Ar x_i apzīmēsim i -tā vērtspapīra daļu (svaru) vērtspapīru portfeli;

un ar $x = (x_1, x_2, \dots, x_M)$ – investīciju plānu.

Šeit $\forall i = \overline{1, H} \quad 0 \leq x_i \leq 1$ un $\sum_{i=1}^H x_i = 1$.

Apzīmēsim ar s_{ij} kovariāciju starp i -tā veida vērtspapīru un j – tā veida vērtspapīru

$$s_{ij} = \text{cov}(R_i, R_j) \stackrel{\text{def}}{=} \frac{1}{T} \sum_{t=1}^T [(R_{it} - \bar{R}_i)(R_{jt} - \bar{R}_j)] . \quad (1.2.1)$$

Sastādīsim vērtspapīru ienesīguma dispersijas jeb riska kvadrāta funkciju

$$s_p^2 = \sum_{i=1}^H \sum_{j=1}^H s_{ij} x_i x_j . \quad (1.2.2)$$

Tā ir funkcija, kas jāminimizē.

1.3. Darba struktūra

Darbs sastāv no ievada, 5 nodaļām, nobeiguma, literatūras saraksta un pielikuma.

Ievadā formulēts darba uzdevums un īsi aprakstīta darba struktūra.

Otrā nodaļa satur īsu aprakstu par vērtspapīru portfeļiem. Šajā daļā tiek aprakstīta minimālā riska portfeļa problēma.

Trešajā nodaļā ir aprakstīti kvadrātiskās programmēšanas uzdevumi kā matemātiskās programmēšanas uzdevumu speciālgadījums.

Ceturtā nodaļa satur Vulfa – Daugaveta metodes aprakstu. Metode ļauj kvadrātiskās programmēšanas uzdevumu aizvietot ar minimizācijas uzdevumu ar lineāru mērķa funkciju ar lineāriem un arī nelineāriem ierobežojumiem. Aprakstītā metode dod iespēju kvadrātiskās programmēšanas uzdevuma atrisinājuma meklēšanai izmantot nedaudz modificētu simpleksa algoritmu.

Piektā nodaļā ir apskatīti divi algoritmi: simpleksa metodes algoritms un modificētās simpleksa metodes algoritms.

Sestā nodaļa satur divas Visual Basic programmas. Pirmā programma atrisina lineārās programmēšanas uzdevumus ar simpleksa metodi, bet otrā programma - kvadrātiskās programmēšanas uzdevumus ar modificēto simpleksa metodi. Otrā programma izmantota, lai minimizētu vērtspapīru portfeļu riska kvadrātu.

2. Vērtspapīru portfeļu teorija

2.1. Kas ir vērtspapīri?

No ieguldītāja viedokļa vērtspapīrs ir dokuments, kas tā īpašniekam dod noteiktas finansiāla rakstura tiesības, kuras jāapmierina vērtspapīra izlaidējam jeb emitentam. No vērtspapīra izlaidēja jeb emitenta viedokļa vērtspapīrs ir privāttiesību dokuments, kas apliecina emitenta saistības pret dokumenta valdītāju un var būt par publiskās apgrozības objektu, nemainoties tajā nostiprināto tiesību apjomam. Savukārt no profesionālā finanšu tirgus dalībnieka viedokļa vērtspapīrs ir finanšu tirgus instruments, ko iespējams izmantot darījumos peļņas gūšanas nolūkā¹.

Akcijas ir vērtspapīri, kas apliecina tā īpašnieka līdzdalību akciju sabiedrības pamatkapitālā (atbilstoši akcijas nominālvērtībai) un dod tiesības uz zināmu sabiedrības peļņas daļu (dividendi). Fondu biržas akcijas tiek tirgotas atbilstoši to kursam – t.i. tirgus cenai par vienu akciju attiecīgajā brīdī. Šis kurss tiek noteikts vadoties no akciju pieprasījuma un piedāvājuma tirgū attiecīgajā brīdī, līdz ar to tas var atšķirties no nominālvērtības.

Akcijas ir visizplatītākais publiski tirgojamo vērtspapīru veids. To kopvērtība finanšu tirgos pārsniedz jebkuru citu atsevišķi ņemtu veidu vērtspapīrus. Statistiskie dati liecina, ka vidēji pasaulē ik gadu no jauna tiek emitētas akcijas aptuveni viena procenta vērtībā no jau eksistējošām. No visām akcijām apmēram puse ir fizisko personu valdījumā, bet pārējās pārvalda pensiju fondi, savstarpējo ieguldījumu fondi un apdrošināšanas sabiedrības².

2.2. Ienesīgums un risks

Divi galvenie vērtspapīrus raksturojošie parametri ir to ienesīgums un risks. Tagad aplūkosim, kā šie lielumi tiek aprēķināti skaitliski. Vispirms aplūkosim obligācijas, bet pēc tam akcijas³.

Akciju ienesīgums kādam fiksētam laika periodam, jeb precīzāk to ienesīguma likme tiek aprēķināta saskaņā ar formulu

$$R = \frac{C_b - C_s + Div_{s,b}}{C_s}, \quad (2.2.1)$$

kur

¹ Ģ. Apsītis, I. Aščuks, U. Cērps, G. Kokorēvičs, Ģ. Ozols, A. Sedlenieks, H. Zuļģis, "Vērtspapīru tirgus zinības" Otrais papildinātais izdevums, Rīga, Jumava, 2006, 21.lpp.

² J. Vucāns "Vērtspapīru portfeļi un to vadīšana" 1.lekcija, 2005, 3. lpp.

³ J. Vucāns "Vērtspapīru portfeļi un to vadīšana" 1.lekcija, 2005, 5. lpp.

C_s - akciju tirgus kurss perioda sākumā

C_b - akciju tirgus kurss perioda beigās

$Div_{s,b}$ - starp sākuma un beigu momentiem izmaksāto dividenžu kopējais lielums.

Iepriekšējā formula ir pārrakstāma veidā

$$R = \frac{C_b - C_s}{C_s} + \frac{Div_{s,b}}{C_s}, \quad (2.2.2)$$

kur pirmais saskaitāmais vienādības labajā pusē raksturo ienesīgumu no akciju cenu svārstībām, bet otrais saskaitāmais ienesīgumu no dividendēm.

Lai iegūtu atbilstošo gada ienesīguma likmi, perioda ienesīguma likmei būtu jāaprēķina atbilstošā efektīvā likme. Tā kā tas finansu matemātikā ir pieņemts, tad efektīvā likme tiek aprēķināta kā atbilstošā gada salikto procentu likme.

$$R_{gada} = (1 + R)^{\frac{1}{T}} - 1, \quad (2.2.3)$$

kur T – perioda ilgums.

Parasti aprēķinot vērtspapīru sagaidāmos ienesīgumus un riskus rīkojas sekojoši:

Izvēlas gadu ilgu laika periodu, gadā – $T=52$ nedēļas (apmēram). Katrai pilnai nedēļai aprēķina apskatāmā vērtspapīra kursa vidējo vērtību. Starp jebkurām divām blakus nedēļām aprēķina vērtspapīra ienesīgumu nedēļu ilgā laika periodam, izmantojot nedēļu vidējās cenas. Tām nedēļām, kuru laikā ir izmaksātas dividendes, tiek ņemts vērā arī to apjoms. Tādējādi, ja ņem vērā vēl arī datus par iepriekšējā gada pēdējo nedēļu, tad, teiksim i-tā veida vērtspapīram tiek iegūtas nedēļu skaitam T atbilstošas nedēļas ienesīguma vērtības $R_{i1}; R_{i2}; \dots; R_{iT}$.

No šīs i-tā veida vērtspapīra nedēļu ienesīgumu virknes izrēķina nedēļas vidējo ienesīgumu gadu ilgā laika periodā

$$\bar{R}_i = \frac{1}{T} \sum_{t=1}^T R_{it}. \quad (2.2.4)$$

Tādējādi nedēļas ienesīgumu var uzskatīt par gadījuma lielumu, kuram aprakstītajā veidā ir izveidota izlase. Tad $R_{i1}; R_{i2}; \dots; R_{iT}$ ir gadījuma lieluma vērtības, kas atbilst izlasei, bet $\bar{R}_i$ ir izlases vidējā vērtība, kuru pieņem arī par gadījuma lieluma vidējo vērtību. Nedēļas ienesīguma kā gadījuma lieluma svārstības raksturo ar tā izlases standartnovirzi

$$s_i = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (R_{it} - \bar{R}_i)^2}. \quad (2.2.5)$$

To pieņem par i-tā veida vērtspapīra ienesīguma svārstību riska mēru.

2.3. Vērtspapīru portfeļa jēdziens

Vērtspapīru portfelis veidojas, iegādājoties vairākus finansu aktīvus (piemēram, akcijas) atbilstoši kādai noteiktai proporcijai¹.

Modelēšanas kontekstā tas nozīmē, ka no katras vērtspapīros ieguldītās naudas vienības kādu daļu x_1 iegulda viena veida vērtspapīros, daļu x_2 otra veida vērtspapīros utt.

Ja tiek veidots portfelis no H dažāda veida vērtspapīriem, ņemot tos portfelī atbilstoši proporcijām $x_1, x_2, \dots, x_H$, kur

$$x_1 + x_2 + \dots + x_H = 1, \quad (2.3.1)$$

tad t -tajā pagātnes laika periodā šāda portfeļa ienesīgums būtu bijis

$$R_{pt} = x_1 R_{1t} + x_2 R_{2t} + \dots + x_H R_{Ht}, \quad (2.3.2)$$

kur R_{it} - i -tā veida vērtspapīra ienesīgums t -tajā laika periodā.

Tātad visos pagātnes laika periodos portfeļa ienesīguma struktūra būtu izsakāma kā lineāra kombinācija no atsevišķo vērtspapīru ienesīgumiem:

$$R_p = x_1 R_1 + x_2 R_2 + \dots + x_H R_H, \quad (2.3.3)$$

kur $R_p, R_1, R_2, \dots, R_H$ ir gadījuma lielumi, kas raksturo, atbilstoši, portfeļa un tā sastāvā ieejošo vērtspapīru ienesīgumus.

Līdz ar to, izmantojot matemātiskās cerības E linearitātes īpašību, iegūstam:

$$\overline{R_p} = E(R_p) = E\left(\sum_{i=1}^H x_i R_i\right) = \sum_{i=1}^H E(x_i R_i) = \sum_{i=1}^H x_i E(R_i) = \sum_{i=1}^H x_i \overline{R_i}. \quad (2.3.4)$$

Šo vispārīgo formulu arī turpmāk vienmēr izmantosim portfeļa sagaidāmā ienesīguma noteikšanai.

Otru vērtspapīru portfeli raksturojošo lielumu – *dispersiju* (jeb *variāciju*) noteikt ir nedaudz sarežģītāk, tāpēc iesāksim ar visvienkāršāko – 2 vērtspapīru portfeļa gadījumu, pēc tam vispārināsim rezultātu arī H vērtspapīru gadījumam.

Tātad, ņemot vērā dispersijas definīciju, dispersiju portfeļa ienesīgumam izsakām kā

$$s_p^2 \stackrel{\text{def}}{=} E((R_p - \overline{R_p})^2). \quad (2.3.5)$$

Divu vērtspapīru gadījumā izsakās sekojoši:

$$\begin{aligned} s_p^2 &= E((R_p - \overline{R_p})^2) = E\left(\left[(x_1 R_1 + x_2 R_2) - (x_1 \overline{R_1} + x_2 \overline{R_2})\right]^2\right) = \\ &= E\left(\left[x_1 (R_1 - \overline{R_1}) + x_2 (R_2 - \overline{R_2})\right]^2\right) \end{aligned} \quad (2.3.6)$$

¹ J. Vucāns "Vērtspapīru portfeli un to vadīšana" 2.lekcija, 2005, 1. lpp.

Nemot vērā summas kvadrāta formulu, iegūstam:

$$\begin{aligned} s_p^2 &= E\left(\left[x_1(R_1 - \bar{R}_1) + x_2(R_2 - \bar{R}_2)\right]^2\right) = \\ &= E\left(x_1^2(R_1 - \bar{R}_1)^2 + 2x_1x_2(R_1 - \bar{R}_1)(R_2 - \bar{R}_2) + x_2^2(R_2 - \bar{R}_2)^2\right). \end{aligned} \quad (2.3.7)$$

Izmantojot jau pieminētās matemātiskās cerības īpašības, iegūstam

$$\begin{aligned} s_p^2 &= x_1^2 E\left[(R_1 - \bar{R}_1)^2\right] + 2x_1x_2 E\left[(R_1 - \bar{R}_1)(R_2 - \bar{R}_2)\right] + x_2^2 E\left[(R_2 - \bar{R}_2)^2\right] = \\ &= x_1^2 s_1^2 + 2x_1x_2 E\left[(R_1 - \bar{R}_1)(R_2 - \bar{R}_2)\right] + x_2^2 s_2^2. \end{aligned} \quad (2.3.8)$$

Kā redzams, tad vidējais saskaitāmais satur divu ienesīgumu R_i un R_j kovariāciju:

$$\text{cov}(R_i, R_j) \stackrel{\text{def}}{=} E\left[(R_i - \bar{R}_i)(R_j - \bar{R}_j)\right], \quad i \neq j. \quad (2.3.9)$$

Lietojot apzīmējumus:

$$\text{cov}(R_i, R_j) \equiv s_{ij}, \quad i \neq j;$$

$$s^2(R_i) = s_i^2 \equiv s_{ii},$$

varam uzrakstīt t.s. variāciju (dispersiju) – kovariāciju matricu

$$\begin{pmatrix} s_{11} & s_{12} & \dots & s_{1H} \\ s_{21} & s_{22} & \dots & s_{2H} \\ \dots & \dots & \dots & \dots \\ s_{H1} & s_{H2} & \dots & s_{HH} \end{pmatrix},$$

kur visi matricas elementi formāli ir izteikti kā

$$s_{ij} = E\left[(R_i - \bar{R}_i)(R_j - \bar{R}_j)\right]. \quad (2.3.10)$$

Līdz ar to iegūstam, ka divu vērtspapīru portfeļa gadījumā, *portfeļa* dispersija ir izteikta kā

$$s_p^2 = x_1^2 s_1^2 + x_2^2 s_2^2 + 2x_1x_2 s_{12}. \quad (2.3.11)$$

Tā kā kovariācijas izteiksmē ieiet reizinājums no divu aktīvu ienesīgumu novirzēm (no to vidējām vērtībām), kuras katra var būt gan pozitīva, gan negatīva, tad arī reizinājums var būt gan pozitīvs, gan negatīvs. Kovariācija atspoguļo abu apskatāmo aktīvu ienesīgumu izmaiņu savstarpējo saistību, t.i. šo izmaiņu mijiedarbību.

Vispārinot šo formulu H vērtspapīru gadījumam, var iegūt vispārīgu formulu no tiem sastādītā portfeļa ienesīguma dispersijas aprēķināšanai:

$$s_p^2 = \sum_{i=1}^H x_i^2 s_i^2 + \sum_{i=1}^H \sum_{\substack{j=1 \\ j \neq i}}^H s_{ij} x_i x_j = \sum_{i=1}^H \sum_{j=1}^H s_{ij} x_i x_j \quad (2.3.12)$$

2.4. Minimālā riska portfeļa noteikšana

Ir zināms, ka investors, veidojot portfeli, līdzekļu iegūšanai tālākām investīcijām nevar pārdot vērtspapīru, kas tam nemaz nepieder. Lai noteiktu minimālā riska portfeļa konstrukciju (atbilstošos svaru koeficientus investīcijām atsevišķos vērtspapīros) no H vērtspapīriem veidotam portfelim un, jāminimizē portfeļa riska kvadrāta vērtība¹, par mainīgajiem uzskatīsim svaru koeficientus x_i , kur $i = 1, 2, \dots, H$.

$$s_p^2 = \sum_{i=1}^H \sum_{j=1}^H x_i x_j s_{ij} \rightarrow \min, \quad (2.4.1)$$

pie nosacījumiem

$$\left(\sum_{i=1}^K x_i \right) - 1 = 0, \quad (2.4.2)$$

$$0 \leq x_i \leq 1, \quad i = 1, 2, \dots, H, \quad (2.4.3)$$

Svaru koeficients x_i ir i -tā vērtspapīra proporcija vērtspapīru portfelī. Tā kā visi svāri ir nenegatīvi, tad nosacījums $x_i \geq 0$, $i = 1, 2, \dots, H$ ir nepieciešams, lai minimizējot portfeļa riska kvadrāta vērtību neiegūtu negatīvus svarus.

Pieņemsim, ka vērtspapīru portfelis sastāv tikai no viena vērtspapīru, tas nozīmē, ka šī vērtspapīra svārs vērtspapīru portfelī ir 1. Tas nozīmē, ka ikviens no H vērtspapīriem var veidot portfeli, bet tas nozīmē, ka svārs ir 1, t.i., $x_i \leq 1$, $i = 1, 2, \dots, H$.

Tātad minimālā riska portfeļa problēma ir izskatās sekojoši:

$$\begin{aligned} \sigma_p^2 &= \sum_{i=1}^H \sum_{j=1}^H x_i x_j \sigma_{ij} \rightarrow \min \\ &\begin{cases} 0 \leq x_i \leq 1, & i = 1, 2, \dots, H \\ \sum_{i=1}^H x_i = 1 \end{cases} \end{aligned} \quad (2.4.4)$$

Kā redzams, tad problēma ir kvadrātiskās programmēšanas uzdevums ar lineāriem ierobežojumiem.

¹ J. Vucāns "Vērtspapīru portfeli un to vadīšana" 4.lekcija, 2005, 5. lpp.

3. Kvadrātiskās programmēšanas uzdevums

3.1. Matemātiskās programmēšanas uzdevums

Matemātiskā programmēšana ir matemātikas nozare, kas izstrādā teoriju un skaitliskās metodes daudzdimensiju ekstrēma uzdevumu ar ierobežojumiem risināšanai. Matemātiskās programmēšanas uzdevums pēc būtības ir optimizācijas uzdevums. Funkciju, kuru optimizē, sauc par mērķa funkciju. Mērķa funkcija apraksta kritēriju, pēc kura plānojam uzlabot situāciju. Ekonomiskās iespējas parādās kā ierobežojumu sistēma. Viss kopā ir uzdevuma matemātiskais modelis. Matemātiskais modelis iekļauj sevī:

- uzdevuma plānu: nezināmo parametru kopumu, iedarbojoties uz kuru sistēmu var uzlabot,
- mērķa funkciju, kura ļauj izvēlēties vislabāko variantu no iespējamo variantu kopas.
(Vislabākais variants ir mērķa funkcijas ekstrēms, tas var būt peļņa, izstrādājumu skaits, kā arī zaudējumi, atkritumi ...)

Šie nosacījumi seko no resursu ierobežojuma, ko cilvēki var lietot jebkurā laika momentā, no tehnoloģiskā un ražošanas procesa nosacījumiem, kā arī no preces pieprasījuma. Dažkārt pieprasījums pārsniedz realizācijas iespējas. Matemātiski ierobežojums izpaužas kā vienādojumu vai nevienādību sistēma, kas veido iespējamo atrisinājumu apgabalu (ekonomiskās iespējas apgabalu). Plānu, kas apmierina nosacījumu sistēmu, sauc par atbalsta plānu. Ja atbalsta plāns piegādā mērķa funkcijai ekstremālo vērtību, tad to sauc par optimālo plānu. Optimāls atrisinājums var eksistēt un būt viens vienīgs, bet dažkārt tas neeksistē, vai pieder galīgai vai bezgalīgai atrisinājumu kopai.

Jebkuru uzdevumu, kurā tiek meklēts funkcijas minimums vai maksimums, sauc par matemātiskās programmēšanas uzdevumu un noformulē sekojošā veidā:

$$z = f(x) \rightarrow \min \quad (3.1.1)$$

$$g_i(x) \leq b_i, \quad i = 1, 2, \dots, m, \quad (3.1.2)$$

kur $x = (x_1, x_2, x_3, \dots, x_n)$, ir jāatrod vektora x vērtības, kas apmierina m nevienādības (3.1.2) (nosacījumi (3.1.2) daļēji var būt pierakstīti vienādību formā) un minimizē vai maksimizē funkciju (3.1.1).

Nosacījumus (3.1.2) sauc par *ierobežojumiem* (ekonomiskām iespējām), kur var būt jebkura no zīmēm $\leq, =, \geq$ (dažādiem ierobežojumiem var būt dažādas zīmes), bet funkciju (3.1.1) sauc par *mērķa funkciju*. Lielumi m un n nav saistīti savā starpā, tāpēc tie var pieņemt jebkuras naturālas vērtības.

3.2. Lineārās programmēšanas uzdevums

Interese par matemātiskās programmēšanas praktisko pielietojanu ir augusi kopš 1947. gada, kad Dž. Dancings izstrādāja simpleksa metodi lineārās programmēšanas vispārīgā uzdevuma atrisināšanai¹. Ja formulās (3.1.1) un (3.1.2)

$$g_i(x) = g_i(x_1, x_2, \dots, x_n) = \sum_{j=1}^n a_{ij} x_j, \quad i = 1, 2, \dots, m, \quad (3.2.1)$$

un

$$f(x) = f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n c_j x_j, \quad (3.2.2)$$

kur a_{ij} un c_j ir zināmas konstantes, tad uzdevumu sauc par lineāru pie nosacījuma, ka nav citu ierobežojumu, izņemot nosacījumu par dažu vai visu mainīgo nenegativitāti. Parasti lineārās programmēšanas vispārīgā uzdevumā tiek prasīts, lai visi mainīgie būtu nenegatīvi, t.i.,

$$x_j \geq 0, \quad j = 1, 2, \dots, n, \quad (3.2.3)$$

jo tāda uzdevuma forma ir ērtāka skaitliskai rēķināšanai. Uzdevumu, kurā daži mainīgie var būt ar patvaļīgu zīmi, var viegli pārveidot uzdevumā ar ierobežojumiem $(3.2.3)^2$.

Tādā veidā, lineārās programmēšanas uzdevums ir atrast n mainīgo x_j nenegatīvas vērtības, kas apmierina m ierobežojumus

$$\sum_{j=1}^n a_{ij} x_j \{ \leq, =, \geq \} b_i$$

un maksimizē jeb minimizē lineāru funkciju

$$z = \sum_{j=1}^n c_j x_j$$

Jebkuru citu matemātiskās programmēšanas uzdevumu uzskatīsim par nelineāru³.

¹ Хедли Дж. Нелинейное и динамическое программирование. М.: Мир, 1967, 10 лрр.

² Хедли Дж. Нелинейное и динамическое программирование. М.: Мир, 1967, 11 лрр.

³ Хедли Дж. Нелинейное и динамическое программирование. М.: Мир, 1967, 11 лрр.

3.3. Kvadrātiskās programmēšanas uzdevums

Daudzus plānošanas un projektēšanas uzdevumus iespējams formulēt kā kvadrātiskās programmēšanas uzdevumus. Kvadrātiskā programmēšana ir nelineārās programmēšanas nozare, kas pēta tādus uzdevumus, kuru mērķa funkcija ir otrās pakāpes funkcija, bet nosacījumu sistēma sastāv no lineārām nevienādībām.

Kvadrātiskās programmēšanas uzdevumu klase pieder pie nelineārās programmēšanas uzdevumu klases ar ierobežojumiem. Vispārīgā formā nelineārās programmēšanas uzdevumu var pierakstīt šādi:

$$\begin{aligned} z &= f(x_1, x_2, x_3, \dots, x_n) \rightarrow \min \\ \left\{ \begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n \end{aligned} \right. \end{aligned} \quad (3.3.1)$$

Kvadrātiskās programmēšanas uzdevumu var pierakstīt sekojošā formā:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j + \sum_{i=1}^n c_i x_i &\rightarrow \min \\ \left\{ \begin{aligned} \sum_{j=1}^n a_{ij} x_j - b_i &\leq 0, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n \end{aligned} \right. \end{aligned} \quad (3.3.2)$$

Mērķa funkcija ir lineārās un kvadrātiskās formas summa, bet vektora x koordinātēm ierobežojumi ir lineārā formā.

Pārrakstīsim kvadrātiskās programmēšanas uzdevumu (2.3.2) matricu formā:

$$\begin{aligned} x^T D x + c^T x &\rightarrow \min \\ \left\{ \begin{aligned} A x - b &\leq 0 \\ x &\geq 0 \end{aligned} \right. \end{aligned} \quad (3.3.3)$$

kur $D=(d_{ij})$ ir simetriska matrica ($d_{ij}=d_{ji}$); $A=(a_{ij})$ – matrica $m \times n$; $b=(b_i)$ – n -tās kārtas vektors. Šeit nevienādība $x \geq 0$, nozīmē, ka visām vektora x koordinātēm jābūt nenegatīvām.

3.4. Optimālā investora portfeļa uzdevuma pieraksts kvadrātiskās programmēšanas uzdevuma kanoniskā formā

Apskatīsim kvadrātiskās programmēšanas uzdevumu:

$$\begin{aligned} \sigma_p^2 &= \sum_{i=1}^H \sum_{j=1}^H x_i x_j \sigma_{ij} \rightarrow \min \\ &\begin{cases} 0 \leq x_i \leq 1, \quad i = 1, 2, \dots, H \\ \sum_{i=1}^H x_i = 1 \end{cases} \end{aligned} \quad (3.4.1)$$

Ievēdīsim jaunus apzīmējumus:

$$\begin{aligned} n &= H, \quad k = H, \\ d_{ij} &= \sigma_{ij} \quad i = \overline{1, n}, \quad j = \overline{1, n}. \end{aligned}$$

Vektora c komponentes ir vienādas ar 0.

Lineārā daļa izskatās sekojoši

$$\sum_{i=1}^n c_i x_i \quad (3.4.2)$$

Ievērojot jaunus apzīmējumus, uzdevumu (3.4.1) var pārrakstīt sekojoši

$$\begin{aligned} \sigma_p^2 &= \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j + \sum_{i=1}^n c_i x_i \rightarrow \min \\ &\begin{cases} x_i \leq 1, \quad i = 1, 2, \dots, k \\ \sum_{i=1}^n x_i = 1 \\ x_i \geq 0, \quad i = 1, 2, \dots, n \end{cases} \end{aligned}$$

4. Vulfa – Daugaveta metode kvadrātiskās programmēšanas uzdevuma risināšanai

4.1. Izliektās programmēšanas uzdevums

Pieņemsim, ka kvadrātiskās programmēšanas uzdevums dots sekojošā formā:

$$\begin{aligned} & x^T D x + c^T x \rightarrow \min \\ & \begin{cases} A x - b \leq 0 \\ x \geq 0, \end{cases} \end{aligned} \quad (4.1.1)$$

kur $D=(d_{ij})$ ir pozitīvi pusdefinita simetriska matrica ($d_{ij}=d_{ji}$); $A=(a_{ij})$ – matrica $m \times n$, kurai visi elementi ir nenegatīvi, un katrā rindā ir vismaz viens pozitīvs elements; $b=(b_i)$ – n -tās kārtas vektors, kura elementi ir nenegatīvi. Nevienādība $x \geq 0$, nozīmē, ka katra vektora x koordināte ir nenegatīva.

Silvestra kritērija papildinājums.¹ Lai simetriska $n \times n$ – matrica A būtu pozitīvi definita, t.i., eksistē $v \geq 0$ tāds, ka $A \geq vI$, nepieciešami un pietiekami, lai visi tās “lejušlidošie” galvenie minori būtu pozitīvi.

Ar “lejušlidošiem” galvenajiem matricas A minoriem saprot $\det A$ un determinantus matricām, kuras iegūst izsvītrojot pakāpeniski no matricas A n -to rindiņu un n -to kolonnu; n -to un $n-1$ – rindiņu un n -to un $n-1$ – kolonnu; utt.

Nevienādība ir ekvivalenta formai:

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j + \sum_{i=1}^n c_i x_i \rightarrow \min \\ & \begin{cases} \sum_{j=1}^n a_{ij} x_i - b_i \leq 0, & i = 1, \dots, k \\ \sum_{j=1}^n a_{ij} x_i - b_i = 0, & i = k + 1, \dots, m \\ x_i \geq 0, & i = 1, \dots, n \end{cases} \end{aligned} \quad (4.1.2)$$

Piezīme. Ja $k = 0$, tad visi nosacījumi ir uzrakstīti vienādojumu formā, ja $k = m$, tad visi nosacījumi ir uzrakstīti nevienādību formā.

$$\text{Apzīmēsim} \quad f(x) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j + \sum_{i=1}^n c_i x_i \quad \text{un} \quad g_i(x) = \sum_{j=1}^n a_{ij} x_i - b_i,$$

$R_+^n = \{x = (x_1, x_2, \dots, x_n) \in R^n \mid x_i \geq 0\}$, tad minimizācijas uzdevumu var pierakstīt šādā formā:

¹ Raitums U. “Optimizācijas metodes: lekciju kurss”, Rīga, Latvijas Universitāte, 2002, 25 lpp.

$$\begin{aligned} f(x) \rightarrow \min \\ \begin{cases} g_i(x) \leq 0, & i = 1, \dots, k \\ g_i(x) = 0, & i = k+1, \dots, m \\ x \in R_+^n \end{cases} \end{aligned} \quad (4.1.3)$$

Apzīmēsim ar Q kopu, kas apmierina uzdevuma nosacījumus, tas ir, $Q = \bigcap_{i=1}^m Q_i$, kur $Q_i = \{x \in R_+^n \mid g_i(x) \leq 0\}$, $i = 1, \dots, k$ un $Q_i = \{x \in R_+^n \mid g_i(x) = 0\}$, $i = k+1, \dots, m$.

Noskaidrosim, vai kopa Q ir ierobežota?

Mūsu gadījumā uzdevumā ir dots, ka $\forall i = 1, \dots, n \quad 0 \leq x_i \leq 1$ un $\sum_{i=1}^n x_i = 1$, tad iegūstam, ka kopa Q ir ierobežota un slēgta.

Tad ikvienu no $m-k$ vienādībām aizvietojam ar divām nevienādībām $\leq, \geq$. Tad iegūstam, ka mums ir $2m-k$ nevienādības.

Tālāk apskatīsim uzdevumu, kur visi nosacījumi ir uzrakstīti nevienādību formā:

$$\begin{aligned} f(x) \rightarrow \min \\ \begin{cases} g_i(x) \leq 0 & i = 1, \dots, m \\ x \in R_+^n \end{cases} \end{aligned} \quad (4.1.4)$$

Teorēma (par izliektību)¹. Ja $M \subset R^n$ ir izliekta vaļēja kopa, $f: M \rightarrow R$ un $f \in C^2(M)$, tad sekojošās īpašības:

f ir izliekta funkcija kopā M ;

$$f''(x) \geq 0 \quad \forall x \in M$$

ir ekvivalentas.

Noskaidrosim, vai funkcijas $f, g_i, i = 1, \dots, n$ ir izliektas funkcijas?

Atrādīsim funkcijas f otrās kārtas atvasinājumu matricu

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \dots & d_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & d_{n3} & \dots & d_{nn} \end{pmatrix} = D \geq 0 \quad (4.1.5)$$

¹ Raitums U. "Optimizācijas metodes: lekciju kurss", Rīga, Latvijas Universitāte, 2002, 23 lpp.

Tā kā teorēmas (par izliektību) nosacījumi izpildās, tad seko, ka f ir izliekta.

Tā kā funkcijas g_i ir lineāras, tad teorēmas (par izliektību) nosacījumi izpildās. Seko, ka funkcijas $g_i, \forall i = \overline{1, m}$ ir izliektas.

Tā kā $\forall i = \overline{1, m}$ funkcija g_i ir izliekta, tad kopa Q_i ir izliekta.

Tā kā Q ir $\bigcap_{i=1}^m Q_i$, tad $Q \subset R_+^n$ ir izliekta kopa.

1. definīcija². Par **izliektās programmēšanas uzdevumu** sauc problēmu

$$\begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0, \quad i = 1, \dots, m \\ x \in R_+^n, \end{cases} \quad (4.1.6)$$

kur kopa $Q \subset R_+^n$ ir izliekta un funkcijas $f : Q \rightarrow R, g_j : Q \rightarrow R, j = 1, \dots, m$, ir izliektas.

Pēc 1. definīcijas esam ieguvuši, ka mūsu problēma ir izliektās programmēšanas uzdevums.

4.2. Kvadrātiskās programmēšanas uzdevuma atrisinājuma eksistence

Funkcijas f un $g_j, j = \overline{1, m}$ ir nepārtrauktas un nepārtraukti atvasināmas $\forall x \in R_+^n$.

Pārbaudīsim, vai kopa ir netukša kopa?

Viegli pārlicināties, ka vektors $x_* = (0, 0, 0, \dots, 0)$ apmierina visus nosacījumus.

Teorēma (par problēmas atrisinājuma eksistenci)¹. Ja

- (i) funkcijas $f : Q \rightarrow R, g_j : Q \rightarrow R, j = 1, \dots, m$, ir nepārtrauktas;
- (ii) eksistē elements $x_* \in Q$ tāds, ka $g_j(x_*) \leq 0, j = 1, \dots, m$;
- (iii) kopa Q ir slēgta;
- (iv) izpildās viens no sekojošiem nosacījumiem:

kopa Q ir ierobežota,

² Raitums U. "Optimizācijas metodes: lekciju kurss", Rīga, Latvijas Universitāte, 2002, 37 lpp.

¹ Raitums U. "Optimizācijas metodes: lekciju kurss", Rīga, Latvijas Universitāte, 2002, 37 lpp.

$$f(x) \xrightarrow{|x| \rightarrow \infty} \infty,$$

$$\text{kādam } j \quad g_j(x) \xrightarrow{|x| \rightarrow \infty} \infty,$$

tad problēmai (4.1.4) eksistē atrisinājums.

Tā kā problēmai (4.1.4) ir spēkā visi teorēmas (par problēmas atrisinājuma eksistenci) nosacījumi, tad tai eksistē atrisinājums.

4.3. Lagranža funkcija

Definīcija 2¹. Funkciju L

$$L_1 : R^m \times R^n \rightarrow R, \quad L_1 = L_1(l^1, x), \quad l = (l_1^1, \dots, l_m^1), \quad x = (x_1, \dots, x_n),$$

$$L_1(l, x) = f(x) + \sum_{j=1}^m l_j^1 g_j(x),$$

sauc par problēmas

$$\begin{cases} f(x) \rightarrow \min \\ x \in R_+^n \\ g_j(x) \leq 0, \quad j = 1, \dots, m \end{cases} \quad (4.3.1)$$

normālo Lagranža funkciju, bet koeficientus $\lambda_1^1, \lambda_2^1, \dots, \lambda_m^1$ par **Lagranža reizinātājiem**.

Tagad varam sastādīt normālo Lagranža funkciju mūsu minimizācijas uzdevumam

$$L_1(l^1, x) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j + \sum_{i=1}^n c_i x_i + \sum_{j=1}^m l_j^1 \left(\sum_{i=1}^n a_{ij} x_i - b_i \right), \quad \text{kur } \lambda = (\lambda_1^1, \lambda_2^1, \dots, \lambda_{m-1}^1, \lambda_m^1), \text{ un}$$

$$x = (x_1, x_2, x_3, \dots, x_n).$$

¹ Raitums U. "Optimizācijas metodes: lekciju kurss", Rīga, Latvijas Universitāte, 2002, 34 lpp.

4.4. Kvadrātiskās programmēšanas uzdevuma atrisinājuma eksistences nepieciešamie un pietiekamie nosacījumi

Izveidosim jaunu funkciju $\tilde{f}(x) = -f(x)$, tagad mūsu problēma uzrakstās sekojoši:

$$\begin{aligned} &\tilde{f}(x) \rightarrow \max \\ &\begin{cases} x \in R_+^n \\ g_j(x) \leq 0, \quad j = 1, \dots, m \end{cases} \end{aligned} \quad (4.4.1)$$

Teorēma (nepieciešamais nosacījums)¹ Lai $\tilde{f}(x)$ sasniegtu maksimumu robežpunktā

$\tilde{x} \in Q$, kur $Q = \bigcap_{i=1}^m \{x \in R_+^n \mid g_i(x) \leq 0\}$, ir nepieciešams, ka eksistē vektori $l \geq 0$ un

$v \geq 0$ tādi, ka $\frac{\partial \tilde{f}(\tilde{x})}{\partial \tilde{x}_k} = \sum_{i=1}^m l_i \frac{\partial g_i(\tilde{x})}{\partial \tilde{x}_k} - v_k$, $k = \overline{1, n}$ un ja $\tilde{x}_k > 0$, tad $v_k = 0$, un ja $g_i(\tilde{x}) < 0$, tad

$$l_i = 0$$

No teorēmas par nepieciešamajiem nosacījumiem iegūstam, ka:

$$v_k = -\frac{\partial \tilde{f}(\tilde{x})}{\partial \tilde{x}_k} + \sum_{i=1}^m l_i \frac{\partial g_i(\tilde{x})}{\partial \tilde{x}_k} = -\left(-\frac{\partial f(\tilde{x})}{\partial \tilde{x}_k}\right) + \sum_{i=1}^m l_i \frac{\partial g_i(\tilde{x})}{\partial \tilde{x}_k}, \quad k = \overline{1, n}$$

$$v_k = \frac{\partial f(\tilde{x})}{\partial \tilde{x}_k} + \sum_{i=1}^m l_i \frac{\partial g_i(\tilde{x})}{\partial \tilde{x}_k} = 2 \sum_{i=1}^n d_{ik} x_i + c_k + \sum_{i=1}^m l_i a_{ik}, \quad k = \overline{1, n}$$

Izveidosim jaunu funkciju $\tilde{g}_i(x) = -g_i(x)$, $\forall i = \overline{1, m}$, tad mūsu problēma uzrakstās sekojoši:

$$\begin{aligned} &\tilde{f}(x) \rightarrow \max \\ &\begin{cases} x \in R_+^n \\ \tilde{g}_j(x) \geq 0, \quad j = 1, \dots, m \end{cases} \end{aligned} \quad (4.4.2)$$

Izveidosim normālo Lagranža funkciju problēmai (3.4.2):

$$L(l, x) = \tilde{f}(x) + \sum_{j=1}^m l_j \tilde{g}_j(x) = -\sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j - \sum_{i=1}^n c_i x_i - \sum_{j=1}^m l_j \left(\sum_{i=1}^n a_{ij} x_i - b_j \right), \quad \text{kur}$$

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_{m-1}, \lambda_m), \text{ un } x = (x_1, x_2, x_3, \dots, x_n).$$

¹ Thomas L. Seaty, Joseph Bram, "Nonlinear Mathematics", McCraw - Hill Book Company, New York, San Francisko, Toronto, London, 117 lpp.

Teorēma (pietiekamais un nepieciešamais nosacījums)¹ Lai $\tilde{x}$ būtu maksimālās nelineārās programmēšanas problēmas atrisinājums, ir nepieciešami un pietiekami, lai eksistētu tāds vektors $\tilde{l}$, ka $\tilde{x}$ un $\tilde{l}$ apmierina sekojošus nosacījumus :

$$\nabla_x L(\tilde{l}, \tilde{x}) \leq 0 \quad \nabla_x L(\tilde{l}, \tilde{x}) \cdot \tilde{x}^T = 0 \quad \tilde{x} \geq 0$$

$$\nabla_l L(\tilde{l}, \tilde{x}) \geq 0 \quad \nabla_l L(\tilde{l}, \tilde{x}) \cdot \tilde{l}^T = 0 \quad \tilde{l} \geq 0$$

$$\frac{\partial L(x, l)}{\partial x_k} = -2 \sum_{i=1}^n d_{ik} x_i - c_k - \sum_{i=1}^m l_i a_{ik} = - \left(2 \sum_{i=1}^n d_{ik} x_i + c_k + \sum_{i=1}^m l_i a_{ik} \right) = -v_k, \quad k = \overline{1, n}$$

Ievēdīsim apzīmējumu

$$w_s := \frac{\partial L(x, l)}{\partial l_s} = - \left(\sum_{k=1}^n a_{sk} x_k - b_s \right) = - \sum_{k=1}^n a_{sk} x_k + b_s, \quad s = \overline{1, m}$$

No teorēmas par pietiekamajiem un nepieciešamajiem nosacījumiem izriet, ka ir jāizpildās sekojošiem nosacījumiem:

$$\begin{cases} l \geq 0, \\ w \geq 0, \\ x \geq 0, \\ v \geq 0, \\ v \cdot x^T = 0, \\ w \cdot l^T = 0. \end{cases} \quad (4.4.3)$$

Ja $g_i(x) = \sum_{k=1}^m a_{ik} x_k - b_i = -w_i < 0$ un ir zināms, ka reizinājums $w_i \cdot l_i = 0$, tad

secinām, ka $l_i = 0$, $i = \overline{1, m}$, tas nozīmē, ka teorēmas par nepieciešamajiem nosacījumiem papildnosacījums ir jau iekļauts sistēmā (4.4.3). Ja $x_j > 0$ un ir zināms, ka reizinājums $v_j \cdot x_j = 0$, tad secinām, ka $v_j = 0$, $j = \overline{1, n}$, tas nozīmē, ka teorēmas par nepieciešamajiem nosacījumiem papildnosacījums ir jau iekļauts sistēmā (4.4.3).

No teorēmas par nepieciešamo nosacījumu un teorēmas par pietiekamajiem un nepieciešamajiem nosacījumiem, iegūstam uzdevuma atrisināšanas pietiekamos un nepieciešamos nosacījumus, kas veidos sekojošu vienādojumu sistēmu:

¹ Thomas L. Seaty, Joseph Bram, "Nonlinear Mathematics", McCraw - Hill Book Company, New York, San Francisco, Toronto, London, 119 lpp.

$$\left\{ \begin{array}{ll} -\sum_{k=1}^n a_{sk} x_s + b_s = w_s, & s = \overline{1, m} \\ 2\sum_{i=1}^n d_{ik} x_i + c_k + \sum_{i=1}^m \lambda_i a_{ik} = v_k, & k = \overline{1, n} \\ \lambda_i \geq 0, & i = \overline{1, m} \\ w_i \geq 0, & i = \overline{1, m} \\ x_k \geq 0, & k = \overline{1, n} \\ v_k \geq 0, & k = \overline{1, n} \\ w_i \cdot \lambda_i = 0, & i = \overline{1, n} \\ v_k \cdot x_k = 0, & k = \overline{1, m} \end{array} \right. \quad (4.4.4)$$

4.5. Vienādojumu sistēma kvadrātiskās programmēšanas uzdevuma atrisināšanai

$$\left\{ \begin{array}{ll} -\sum_{k=1}^n a_{sk} x_s + b_s = w_s, & s = \overline{1, m} \\ 2\sum_{i=1}^n d_{ik} x_i + c_k + \sum_{i=1}^m \lambda_i a_{ik} = v_k, & k = \overline{1, n} \\ \lambda_i \geq 0, & i = \overline{1, m} \\ w_i \geq 0, & i = \overline{1, m} \\ x_k \geq 0, & k = \overline{1, n} \\ v_k \geq 0, & k = \overline{1, n} \\ w_i \cdot \lambda_i = 0, & i = \overline{1, n} \\ v_k \cdot x_k = 0, & k = \overline{1, m} \end{array} \right. \quad (4.5.1)$$

Lai ērtāk būtu lietot iegūto vienādojumu sistēmu, pārrakstīsim to matricu formā:

$$\left\{ \begin{array}{l} Ax + w = b \\ 2Dx - v + A\lambda = -c, \\ \lambda \geq 0, \\ w \geq 0, \\ x \geq 0, \\ v \geq 0. \end{array} \right. \quad (4.5.2)$$

$$\left\{ \begin{array}{l} w \cdot l^T = 0, \\ v \cdot x^T = 0. \end{array} \right. \quad (4.5.3)$$

4.6. Vulfa – Daugaveta metodes apraksts

Vulfs^{1,2,3} piedāvāja vienādojumu sistēmu kvadrātiskās programmēšanas uzdevuma atrisināšanai pārvērst par nenegatīva palīgmainīgā minimizācijas uzdevumu; papildmainīgo pieskaita lineāro vienādojumu kreisajai pusei. Pie nosacījuma, ka sākotnējai sistēmai atrisinājums eksistē, apskatītajam minimizācijas uzdevumam būs nulles atrisinājums, kas atbilst palīgmainīgā nulles vērtībai. Pie tam, pārējo mainīgo vērtības dos sākuma sistēmas atrisinājumu.

Šādā veidā pārveidots minimizācijas uzdevums ir lineārās programmēšanas uzdevums ar dažiem papildus nelineāriem ierobežojumiem. Šādi pārveidotā uzdevuma priekšrocība ir tā, ka tā risināšanai eksistē labi izstrādāta simpleksa metode. Lai realizētu simpleksa metodi, jāastāda sākotnējais plāns. Lai tas būtu izdarīts, ir lietderīgi izmantot nevis vienu, bet vairākus nenegatīvus papildmainīgos, un rēķināt uzdevumu par to summu minimizēšanu.

Daugavets⁴ pamatoja metodi gadījumam, kad kvadrātiskās formas matrica ir pozitīvi pusdefinīta.

4.7. Palīguzdevums kvadrātiskās programmēšanas uzdevuma atrisināšanai

Tagad mums ir vienādojumu un nevienādību sistēma, kuras atrisinājums dos mums arī sākotnēja uzdevuma (4.1.4) atrisinājumu, un ir zināms, ka problēmas (2.4.4) atrisinājums eksistē. Sistēmu (4.5.1) var pārveidot tā, lai risinātu nevis vienādojumu un nevienādību sistēmu, bet minimizācijas uzdevumu ar lineāru mērķa funkciju un lineāriem ierobežojumiem, kuru var risināt ar simpleksa metodi.

Nemot $\mathbf{x} = \mathbf{0}$, $\mathbf{n} = \mathbf{0}$, $\mathbf{l} = \mathbf{0}$ no sistēmas (4.5.1) iegūsim

$$\begin{aligned} 0 &= \mathbf{b} - \mathbf{w} \\ 0 &= -\mathbf{c} \end{aligned} \tag{4.7.1}$$

Tā kā vektoram $\mathbf{c}$ nav uzlikti nekādi ierobežojumi, tā koordinātes var būt gan negatīvas, gan pozitīvas, gan arī vienādas ar nulli. Tāpēc, lai sistēma (3.5.1) būtu spēkā pie $\mathbf{x} = \mathbf{0}$, $\mathbf{n} = \mathbf{0}$, $\mathbf{l} = \mathbf{0}$, otrajā vienādojumā ir jāievieš palīgmainīgais. Tas varētu būt vektors $\mathbf{z}=(z_1, z_2, \dots, z_n)$, kur $z_i \geq 0$, $\forall i=1, \dots, n$. Lai nodrošināt zīmes vektora $\mathbf{z}$ komponentēm, ievieš diagonālmatrixu $\mathbf{E}$,

¹ Thomas L. Seaty, Joseph Bram, "Nonlinear Mathematics", McCraw - Hill Book Company, New York, San Francisko, Toronto, London, 129 lpp.

² Кюнц Г.П., Крелле В. Нелинейное программирование. М.: Сов. радио, 1965, 161 lpp.

³ Хедли Дж. Нелинейное и динамическое программирование. М.: Мир, 1967, 230 lpp.

⁴ Даугавет В. А. Модификация метода Вулфа. Журнал вычислительной математики и математической физики, 1981, 21, 2, 504 lpp.

kuras elementi ir vienādi ar +1, ja $c_i < 0$, un ar -1, ja $c_i > 0$. Tad sistēmas (4.7.1) otrais vienādojums izskatīsies šādi: $Ez = -c$, un tāpēc $e_{ii}z_i$ zīme ir atkarīga no vektora c attiecīgās komponentes zīmes: ja $c_i < 0$, tad z_i jābūt pozitīvai, lai izpildītos $e_{ii}z_i = -c_i$, un, otrādi, ja $c_i > 0$, tad $d_{ii}z_{ii}$ jābūt negatīvai; ja savukārt $c_i = 0$, tad attiecīgā z_i zīmi varam izvēlēties pēc patikas.

Ar jaunievestajiem mainīgajiem sistēma (4.5.1) izskatās šādi:

$$\left\{ \begin{array}{ll} \sum_{k=1}^n a_{sk} x_s + w_s = b_s, & s = \overline{1, m} \\ 2 \sum_{i=1}^n d_{ik} x_i - v_k + \sum_{i=1}^m \lambda_i a_{ik} + e_{kk} z_k = -c_k, & k = \overline{1, n} \\ \lambda_i \geq 0, & i = \overline{1, m} \\ w_i \geq 0, & i = \overline{1, m} \\ x_k \geq 0, & k = \overline{1, n} \\ v_k \geq 0, & k = \overline{1, n} \\ w_i \cdot \lambda_i = 0, & i = \overline{1, n} \\ v_k \cdot x_k = 0, & k = \overline{1, m} \\ z_k \geq 0, & k = \overline{1, n} \end{array} \right. \quad (4.7.2)$$

Ja pieņem, ka mainīgie z_i tiecas uz nulli, tad atkal iegūsim sistēmu (4.5.1). Tāpēc sistēmas (4.5.1) vietā varam risināt minimizācijas uzdevumu ar mērķa funkciju $\sum_{i=1}^n z_i$ un ierobežojumiem (4.7.2). Tā kā $z_i \geq 0$, tad mērķa funkcijas minimums būs vienāds ar nulli, un sasniedzot to, mēs iegūsim sistēmas (4.5.1) atrisinājumu un līdz ar to uzdevuma (4.1.3) atrisinājumu. Šo minimizācijas uzdevumu var atrisināt ar simpleksa algoritmu ar papildnosacījumiem:

$$\left\{ \begin{array}{ll} w_i \cdot \lambda_i = 0, & \forall i = \overline{1, n} \\ v_k \cdot x_k = 0, & \forall k = \overline{1, m} \end{array} \right. \quad (4.7.3)$$

Šo nosacījumu izpildi var panākt, nepieļaujot, lai x_k un v_k vienlaicīgi neatrastos bāzē, tas pats ir spēkā arī pārim w_i un λ_i .

Tātad, lai atrisinātu kvadrātiskās programmēšanas uzdevumu

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_i x_j + \sum_{i=1}^n c_i x_i \rightarrow \min \\ & \begin{cases} \sum_{j=1}^n a_{ij} x_j - b_i \leq 0, & i = 1, \dots, k \\ \sum_{j=1}^n a_{ij} x_j - b_i = 0, & i = k+1, \dots, m \\ x_j \geq 0 & j = 1, \dots, n \end{cases} \end{aligned} \quad (4.7.4)$$

ir jāatrod atrisinājums lineārās programmēšanas uzdevumam

$$\begin{aligned} & \sum_{i=1}^n z_i \rightarrow \min \\ & \begin{cases} \sum_{j=1}^n a_{sj} x_j = b_s - w_s & s = 1, \dots, m \\ 2 \sum_{i=1}^n d_{ik} x_i - v_k + \sum_{i=1}^m \lambda_i a_{ik} + e_{kk} z_k = -c_k, & k = 1, \dots, n \end{cases} \end{aligned} \quad (4.7.5)$$

ar nosacījumiem

$$\begin{cases} x_k \geq 0, & k = 1, \dots, n \\ v_k \geq 0, & k = 1, \dots, n \\ \lambda_i \geq 0, & i = 1, \dots, m \\ w_i \geq 0, & i = 1, \dots, m \\ z_k \geq 0, & k = 1, \dots, m \end{cases} \quad (4.7.6)$$

un papildus nosacījumiem

$$\begin{cases} w_i \cdot \lambda_i = 0, & i = \overline{1, m} \\ v_k \cdot x_k = 0, & k = \overline{1, n} \end{cases} \quad (4.7.7)$$

Tā kā mērķa funkcija un ierobežojumi (4.7.5) un (4.7.6) ir lineāri, šo uzdevumu var risināt ar simpleksa metodi, kura ir papildināta ar nosacījumiem (4.7.7). Kā bāzes mainīgos ņem $z_1, z_2, \dots, z_n, w_1, w_2, \dots, w_n$. Lai realizētos nosacījumi (4.7.7), ir jāseko: ja x_k ir bāzē, tad v_k nedrīkst atrasties bāzē nākamajā simpleksa algoritma realizācijas solī un otrādi, ja v_k ir bāzē, tad x_k nedrīkst atrasties bāzē nākamajā solī. Tāds pats noteikums ir spēkā arī pārī w_k un λ_k .

5. Simpleksa metode

5.1. Simpleksa metodes teorētiskie pamati

Lineārās programmēšanas uzdevumus var atrisināt ar vairākām metodēm. Visbiežāk lineārās programmēšanas uzdevumu atrisināšanai lieto *simpleksa metodi*, kuru literatūrā sauc arī par *plāna pakāpeniskas uzlabošanas metodi*.

Tā ir universāla lineārās programmēšanas uzdevumu atrisināšanas metode, kuru izmantojot var atrast optimālo plānu vai pārliecināties, ka nav atrisinājuma.

Uzdevuma risināšana sākas ar atbalsta plāna noteikšanu. Dažiem uzdevumiem atbalsta plānu var noteikt samērā vienkārši, bet vispārīgā gadījumā to atrod ar īpašām metodēm. Vienlaicīgi iespējams konstatēt, kad uzdevuma nosacījumu sistēma ir nesaderīga.

Pēc tam kad atrasts atbalsta plāns, izmantojot īpašu pazīmi, noskaidro, vai tas ir optimāls. Ja atbalsta plāns ir optimāls, tad uzdevums ir atrisināts, ja ne, tad, izmantojot speciālu algoritmu, atrod jaunu atbalsta plānu ar lielāku lineārās formas vērtību nekā iepriekšējam. Iespējams konstatēt arī lineārās formas neierobežotību. Uzdevumu risināšanas process turpinās tik ilgi, kamēr atrod optimālo plānu vai konstatē, ka nav atrisinājuma.

Apskatīsim lineārās programmēšanas uzdevumu kanoniskā formā¹:

$$\begin{aligned} f &= \sum_{j=1}^n c_j x_j \rightarrow \max \\ \sum_{j=1}^n x_j a_j &= b \\ x_j &\geq 0, \quad (j = 1, 2, \dots, n). \end{aligned}$$

Pieņemsim, ka ir zināms atbalsta plāns

$$x = (x_1; x_2; \dots; x_m; \underbrace{0; \dots; 0}_{n-m})^T,$$

$$x_i > 0 \quad (i = 1, 2, \dots, m).$$

Tā bāzi veido lineāri neatkarīga vektoru sistēma a_i ($i = 1, 2, \dots, m$). No tā seko

$$J_B = \{1, 2, \dots, m\}, \quad B = (a_1; a_2; \dots; a_m), \quad x_B = (x_1; x_2; \dots; x_m)^T.$$

Tā kā B ir nesingulāra matrica, tad iegūst

$$x_B = B^{-1}b$$

Izteiksim vektorus a_j ($j = 1, 2, \dots, n$) kā atbalsta plāna bāzes vektoru a_i ($i = 1, 2, \dots, m$) lineārās kombinācijas. Izvirzījuma koeficientus apzīmēsim ar y_{ij} ($i = 1, 2, \dots, m, j = 1, 2, \dots, n$).

¹ D.Kļaviņš, P.Zelčs "Operāciju pētīšanas matemātiskās metodes", Rīga, "Zvaigzne", 1979, 52 lpp.

Vektorus, kuru koordinātes ir a_j izvirzījuma koeficienti, apzīmēsim ar

$$y_j = (y_{1j}; y_{2j}; \dots; y_{mj})^T.$$

Tad saskaņā ar pieņemtajiem apzīmējumiem

$$\sum_{j=1}^n y_{ij} a_j = a_i \quad (j=1,2,\dots,n) \quad (5.1.1)$$

jeb

$$By_j = a_j \quad (j=1,2,\dots,n) \quad (5.1.1)$$

No (5.1.1) iegūst

$$y_j = B^{-1}a_j \quad (j=1,2,\dots,n). \quad (5.1.2)$$

Pēc formulas (5.1.2) var aprēķināt vektoru a_j izvirzījuma koeficientus atbalsta plāna bāzē. Ja $j \in J_B$, tad attiecīgie izvirzījuma koeficientu vektori y_j ir vienības vektori.

Apzīmēsim ar $c_B = (c_1; c_2; \dots; c_m)$ vektoru, kura koordinātes ir vektora c koordinātes, kurām indeksi pieder kopai J_B .

Izmantojot izvirzījuma un lineārās formas koeficientus, katram vektoram a_j var aprēķināt īpašu rādītāju, kuru sauc par vektora novērtējumu. Vektora novērtējumus apzīmē ar Δ_j un aprēķina pēc formulas

$$\Delta_j = c_B y_j - c_j \quad (j=1,2,\dots,n) \quad (5.1.3)$$

jeb

$$\Delta_j = \sum_{i=1}^m c_i y_{ij} - c_j \quad (j=1,2,\dots,n) \quad (5.1.3)$$

Ja $j \in J_B$, tad $\Delta_j = 0$, jo šajā gadījumā y_j ir vienības vektori. Tātad bāzes vektoru novērtējumi ir vienādi ar nulli.

Atkarībā no Δ_j un y_j koordinātu vērtībām iespējams viens no gadījumiem.

1. teorēma: Visu vektoru novērtējumi ir nenegatīvi: $\Delta_j \geq 0$ ($j = 1, 2, \dots, n$).
2. teorēma: Vismaz viena vektora novērtējums ir negatīvs, un katram negatīvam novērtējumam Δ_j atbilst vektors y_j , kuram ir vismaz viena pozitīva koordināte.
3. Eksistē vismaz viens negatīvs novērtējums, piemēram, $\Delta_s < 0$, kam atbilst vektors y_s , kuram nav nevienas pozitīvas koordinātes ($y_s \leq 0$).

5.2. Simpleksa metodes algoritms

Pieņemsim, ka dots lineārās programmēšanas uzdevums kanoniskā formā un tā nosacījumu sistēmas matrica satur m -tās kārtas vienības matricu. Šādā gadījumā uzdevumam var noteikt atbalsta plānu, nelietojot īpašas metodes.

Apskatīsim uzdevuma atrisināšanas secību, kuru sauksim par *simpleksa metodes algoritmu (parasto algoritmu)*. Uzdevuma risināšana notiek īpašās tabulās, kuras sauc par *iterāciju tabulām*. Katra šāda tabula atbilst noteiktam atbalsta plānam. Pāreja no vienas tabulas uz otru notiek, izslēdzot no dotā atbalsta plāna bāzes vienu vektoru un ievēdot tā vieta citu. Jaunas tabulas elementus nosaka, izpildot *Žordāna — Gausa* (*C. Jordan — C. F. Gauss*) izslēgšanas metodes vienu soli ar noteiktā kārtībā izvēlētu *galveno elementu*. Saskaņā ar teorēmām 1. un 2. no iepriekšējā paragrāfa katrā iterācijā pārbauda, vai atbalsta plāns ir optimāls.

Ja lineārās programmēšanas uzdevuma nosacījumu sistēma satur m -tās kārtas vienības matricu, tad varam pieņemt, ka atbalsta plāns ir ¹

$$x = (x_1; x_2; \dots; x_m; \underbrace{0; \dots; 0}_{n-m})^T$$

kura bāzi veido vienības vektori a_i ($i = 1, 2, \dots, m$). Tāpēc $B = (a_1 \parallel a_2 \parallel \dots \parallel a_m) = E_m$, kur E_m — m -tās kārtas vienības matrica. Ja $B = E_m$, tad arī $B^{-1} = E_m$.

Saskaņā ar formulām $x_B = B^{-1}b$ un (5.1.2) iegūst

$$x_B = B^{-1}b = E_m b = b,$$

$$y_j = B^{-1}a_j = E_m a_j = a_j \quad (j=1, 2, \dots, n).$$

Tātad 1. iterācijā atbalsta plāna bāzes koordinātes ir vienādas ar brīvajiem locekļiem un vektoru izvērījuma koeficienti ir vienādi ar pašu vektoru koordinātēm.

Atbilstoši atbalsta plānam x sastāda tabulu 1. Tabulā 1. ieraksta bāzes vektorus a_i ($i=1, 2, \dots, m$), vektoru $x_B = b$, $y_j = a_j$ ($j=1, 2, \dots, n$), c un $c_B = (c_1; c_2; \dots; c_m)$ koordinātes. Tabulas 2. apakšējā rindā ieraksta lineārās formas vērtību atbalsta plānam x :

$$f(x) = c_B x_B$$

un vektoru novērtējumus Δ_j ($j=1, 2, \dots, n$), ko aprēķina pēc formulas (5.1.3):

$$\Delta_j = \sum_{i=1}^m c_i y_{ij} - c_j \quad (j=1, 2, \dots, n).$$

¹ D.Kļaviņš, P.Zelčs "Operāciju pētīšanas matemātiskās metodes", Rīga, "Zvaigzne", 1979, 60 lpp.

Tā kā $J_B = \{1, 2, \dots, m\}$, tad $\Delta_j = 0$ ($j = 1, 2, \dots, m$).

Pēc tam kad aizpildīta tabula, katrā iterācijā izpilda šādus simpleksa metodes parastā algoritma soļus.

1. Pārbauda, vai atbalsta plāns ir optimāls. Ja $\Delta_j \geq 0$ ($j = 1, 2, \dots, n$), tad saskaņā ar teorēmu 1 x ir optimālais plāns. Pretējā gadījumā x nav optimālais plāns.

2. Pārliecinās, vai uzdevumam ir optimālais plāns. Ja eksistē $\Delta_s < 0$, kuram atbilst $y_{is} \leq 0$ ($i = 1, 2, \dots, m$), tad saskaņā ar teorēmu 2 uzdevumam nav optimālā plāna. Pretējā gadījumā, izslēdzot no bāzes vienu vektoru un tā vietā ievēdot citu vektoru, iespējams atrast jaunu atbalsta plānu x' ar lielāku lineārās formas vērtību nekā dotajam.

Tabula 1

c_B^T	Bāze	x_B	c_1	c_2	$\dots$	c_m	c_{m+1}	$\dots$	c_j	$\dots$	c_n
			y_1	y_2	$\dots$	y_m	y_{m+1}	$\dots$	y_j	$\dots$	y_n
c_1	a_1	x_1	1	0	$\dots$	0	$y_{1,m+1}$	$\dots$	y_{1j}	$\dots$	y_{1n}
c_2	a_2	x_2	0	1	$\dots$	0	$y_{2,m+1}$	$\dots$	y_{2j}	$\dots$	y_{2n}
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
c_i	a_i	x_i	0	0	$\dots$	0	$y_{i,m+1}$	$\dots$	y_{ij}	$\dots$	y_{in}
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
c_m	a_m	x_m	0	0	$\dots$	1	$y_{m,m+1}$	$\dots$	y_{mj}	$\dots$	y_{mn}
		$f(x)$	0	0	$\dots$	0	Δ_{m+1}	$\dots$	Δ_j	$\dots$	Δ_n

3. Vektoru, kurš jāievēd bāzē, atrod, nosakot $\min_j \Delta_j < 0$. Pieņemsim, ka $\min_j \Delta_j = \Delta_n < 0$. Tas nozīmē, ka bāzē jāievēd vektors a_n .

4. Vektoru, kurš jāizslēdz no bāzes, atrod, aprēķinot q_0 :

$$q_0 = \min_i \frac{x_i}{y_{in}}$$

$$y_{in} > 0$$

Pieņemsim, ka

$$q_0 = \min_i \frac{x_i}{y_{in}} = \frac{x_m}{y_{mn}}$$

$$y_{mn} > 0.$$

Tas nozīmē, ka no bāzes jāizslēdz vektors a_m .

5. Aprēķina x'_B , y_j ($j=1, 2, \dots, n$), $f(x')$ un Δ_j ($j=1, 2, \dots, n$). Jaunā atbalsta plāna x' bāzi veido vektori a_i ($i=1, 2, \dots, m-1$), a_n tāpēc $c'_B = (c_1; c_2; \dots; c_m)$.

Jaunā atbalsta plāna bāzes koordinātes aprēķina pēc formulām

$$\left. \begin{aligned} x'_i &= x_i - \frac{x_m}{y_{mn}} y_{in} \\ (i &= 1, 2, \dots, m-1) \\ x'_n &= \frac{x_m}{y_{mn}} \end{aligned} \right\} \quad (5.2.1)$$

y'_j koordinātu aprēķināšanai, pamatojoties uz (5.1.1), izmanto šādas izteiksmes:

$$\sum_{i=1}^{m-1} y_{ij} a_i + y_{mj} a_m = a_j \quad (j=1, 2, \dots, n) \quad (5.2.2)$$

$$\sum_{i=1}^{m-1} y_{in} a_i + y_{mn} a_m = a_n \quad (5.2.3)$$

Tā kā $y_{mn} > 0$, tad no (5.2.3) iegūst

$$a_m = \frac{1}{y_{mn}} \left(a_n - \sum_{i=1}^{m-1} y_{in} a_i \right) \quad (5.2.4)$$

Ievieto (5.2.4) vienādībā (5.2.2). Rezultātā iegūst

$$\sum_{i=1}^{m-1} y_{ij} a_i + \frac{y_{mj}}{y_{mn}} \left(a_n - \sum_{i=1}^{m-1} y_{in} a_i \right) = a_j \quad (j=1, 2, \dots, n)$$

jeb

$$\sum_{i=1}^{m-1} \left(y_{ij} - \frac{y_{mj}}{y_{mn}} y_{in} \right) a_i + \frac{y_{mj}}{y_{mn}} a_n = a_j \quad (5.2.5)$$

Tā kā a_i ($i=1, 2, \dots, m-1$), a_n veido jaunā atbalsta plāna bāzi, tad no (5.2.5) izriet, ka vektoru a_j ($j=1, 2, \dots, n$) izvirzījuma koeficientus jaunajā bāzē var aprēķināt pēc formulām

$$\left. \begin{aligned} y'_{ij} &= y_{ij} - \frac{x_m}{y_{mn}} y_{in} \\ (i &= 1, 2, \dots, m-1) \\ y'_{nj} &= \frac{y_{mj}}{y_{mn}} \end{aligned} \right\} \quad (5.2.6)$$

Atzīmēsim, ka y'_{ij} ($i=1, 2, \dots, m-1$), y'_n ir vienības vektori.

Lineārās formas vērtība jaunajam atbalsta plānam ir

$$f(x') = \sum_{i=1}^{m-1} c_i x'_i + c_n x'_n \quad (5.2.7)$$

Šo formulu izmanto, lai pārbaudītu, vai aprēķini izdarīti pareizi. $f(x')$ atrod pēc formulas:

$$f(x') = f(x) - \frac{y_n}{y_{mn}} \Delta_n \quad (5.2.8)$$

Vektoru novērtējumi jaunajam atbalsta plānam ir

$$\Delta_j = \sum_{i=1}^{m-1} c_i y'_{ij} + c_n y'_{nj} - c_j \quad (j=1, 2, \dots, n) \quad (5.2.9)$$

Formulu (5.2.9) izmanto, lai pārbaudītu, vai aprēķini izdarīti pareizi. Δ_j aprēķināšanas formulu iegūst, ja pārveido (5.2.9), izmantojot (5.2.6) un (5.1.3):

$$\begin{aligned} \Delta_j &= \sum_{i=1}^{m-1} c_i \left(y_{ij} - \frac{y_{mj}}{y_{mn}} y_{in} \right) + c_n \frac{y_{mj}}{y_{mn}} - c_j + c_m y_{mj} - c_m y_{mj} = \\ &= \sum_{i=1}^{m-1} c_i y_{ij} + c_m y_{mj} - c_j - \frac{y_{mj}}{y_{mn}} \left(\sum_{i=1}^{m-1} c_i y_{in} + c_m y_{mj} - c_n \right) = \\ &= \sum_{i=1}^m c_i y_{ij} - c_j - \frac{y_{mj}}{y_{mn}} \left(\sum_{i=1}^{m-1} c_i y_{in} - c_n \right) = \Delta_j - \frac{y_{mj}}{y_{mn}} \Delta_n \end{aligned}$$

Tātad

$$\Delta_j = \Delta_j - \frac{y_{mj}}{y_{mn}} \Delta_n \quad (j=1, 2, \dots, n) \quad (5.2.10)$$

Tabulas (1) kolonnu, kurā atrodas $\min_j \Delta_j = \Delta_n < 0$, sauc par *galveno kolonnu*, bet rindu, kuras elementus dalot ir noteikts q_0 , sauc par *galveno rindu*. Elementu, kas atrodas galvenās kolonnas un galvenās rindas krustojumā, sauc par *galveno elementu*. Tabulā 1 galvenais elements ir y_{mn} .

Izpildot Žordāna-Gausa izslēgšanas metodes vienu soli ar galveno elementu y_{mn} , iegūst tādus pašus rezultātus kā ar formulām (5.2.1), (5.2.6), (5.2.8) un (5.2.9). Vienā solī, kurā nosaka tabulas 2 elementus, izpilda šādas darbības: 1) dala tabulas 1 galvenās rindas elementus ar galveno elementu, 2) aizvieto ar nullēm tabulas 1 galvenās kolonnas elementus,

izņemot galveno elementu, kura jaunā vērtība ir skaitlis 1, 3) pārrēķina pārējos tabulas 1. elementus pēc *taisnstūra likuma*.

Taisnstūra likumu formulē šādi:

jāizveido taisnstūri, kura vienā virsotnē atrodas tabulas 1 elements (I), otrā virsotnē — elements galvenajā rindā (II), trešā virsotnē — elements galvenajā kolonnā (III), bet ceturtā virsotnē atrodas galvenais elements (IV), tad attiecīgo tabulas 2 elementu (I') var aprēķināt pēc šādas shēmas:

$$I' = I - \frac{II \times III}{IV} \quad (5.2.11)$$

Lietderīgi ievērot vēl šādu norādījumu, kas izriet no (5.2.11) :

ja galvenajā rindā ir elements 0, tad tam atbilstošās kolonnas elementus tabulā 2 ieraksta bez izmaiņām, bet, ja galvenajā kolonnā ir elements 0, tad bez izmaiņām ieraksta atbilstošās rindas elementus.

Tālāk izpilda nākošo iterāciju. Procesu turpina tik ilgi, kamēr atrod optimālo plānu vai pārliecinās, ka uzdevumam nav atrisinājuma, tāpēc ka lineārā forma nav ierobežota.

Tabula 2

c_B^T	Bāze	x_B	c_1	c_2	...	c_{m-1}	c_m	...	c_j	...	c_{n-1}	c_n
			y'_1	y'_2	...	y'_{m-1}	y'_m	...	y'_j	...	y'_{n-1}	y'_n
c_1	a_1	x'_1	1	0	...	0	y'_{1m}	...	y'_{1j}	...	$y'_{1,n-1}$	0
c_2	a_2	x'_2	0	1	...	0	y'_{2m}	...	y'_{2j}	...	$y'_{2,n-1}$	0
.	.	.	.	.	.	.	.	.	.	.	.	.
c_i	a_i	x'_i	0	0	...	0	y'_{im}	...	y'_{ij}	...	$y'_{i,n-1}$	0
.	.	.	.	.	.	.	.	.	.	.	.	.
c_{m-1}	a_{m-1}	x'_{m-1}	0	0	...	1	$y'_{m-1,m}$		$y'_{m-1,j}$		$y'_{m-1,n-1}$	0
c_n	a_n	x'_n	0	0	...	0	y'_{nm}	...	y'_{nj}	...	$y'_{n,n-1}$	1
$f(x')$			0	0	...	0	Δ'_m	...	Δ'_j	...	Δ'_{n-1}	0

Lai atrisinātu lineārās programmēšanas uzdevumus, tika uzrakstīta datorprogramma, kura tiks aprakstīta nākamajā nodaļā “Programmas”.

5.3. Modificētā simpleksa metode

Mūsu uzdevums (4.7.5), (4.7.6), (4.7.7) atšķiras no lineāras programmēšanas uzdevuma tikai ar diviem vienkāršiem nelineāriem nosacījumiem (4.7.7) .

$$\begin{cases} w_i \cdot \lambda_i = 0, & i = \overline{1, m} \\ v_k \cdot x_k = 0, & k = \overline{1, n} \end{cases} \quad (5.3.1)$$

Uzdevuma risināšanai tiek piedāvāta simpleksa metodes modifikācija, kas balstās uz Vulfa-Daugaveta metodes.

Sākotnējais plāns.

- apskatīsim x un w koordināšu vērtību kā kaut kādu kombināciju, ņemot vērā

$$\text{nosacījumu } \sum_{j=1}^n a_{sj} x_j = b_s - w_s$$

- attiecīgos v un λ elementus definēsim tā, lai $w_i \cdot \lambda_i = 0, i = \overline{1, m}, v_k \cdot x_k = 0, k = \overline{1, n}$ (programmā $\lambda_i = 0, i = \overline{1, m}, v_k = 0, k = \overline{1, n}$)

- tā kā problēmā $\forall k = \overline{1, n} \quad c_k = 0$, tad $e_{kk} = 1$.

- par nenulles z_i elementiem ņemsim vērtības

$$2 \sum_{i=1}^n d_{ik} x_i - v_k + \sum_{i=1}^m \lambda_i a_{ik} + e_{kk} z_k = -c_k, \quad k = 1, \dots, n \quad (\text{programmā})$$

$$z_k = -2 \sum_{i=1}^n d_{ik} x_i + v_k - \sum_{i=1}^m \lambda_i a_{ik}$$

Iterācijas solis.

Katra metodes iterācija ir simpleksa tabulas transformācija, ņemot vērā papildnosacījumus (5.3.1) .

Tie ir jāpārbauda pēc galvenās kolonnas un rindas izvēles. (iepriekšējā paragrāfā aprakstītas shēmas pēc 4. soļa). Ir pierādīts¹, ka tabulas transformācija nav iespējama papildnosacījumu dēļ, tad mēs iegūstam uzdevuma atrisinājumu. Algoritms beidz savu darbu, kad vektora z komponentu vērtības būs vienādas ar nulli.

Lai atrisinātu mūsu uzdevumu, tika uzrakstīta kvadrātiskās programmēšanas datorprogramma. Šī programma pārbauda, vai matricas D galvenie minoru ir pozitīvi.

¹ Даугавет В. А. Модификация метода Вулфа. Журнал вычислительной математики и математической физики, 506 lpp.

6. Programmas

6.1. Lineārās programmēšanas uzdevuma risināšanas programma

Matemātiskā sistēma Maple ļauj ar simpleksa algoritma palīdzību maksimizēt vai minimizēt lineāru mērķa funkciju pie lineāriem nosacījumiem, tas ir risināt lineārās programmēšanas uzdevumu. Uzdevumu (4.7.5), (4.7.6) var risināt ar tādu simpleksa algoritmu, kurš ir papildināts ar nosacījumiem (4.7.7). Tā kā šie nosacījumi ir nelineāri, tad simpleksa algoritmu, kas ir iebūvēts programmā Maple, kvadrātiskās programmēšanas uzdevuma risināšanai izmantot nevar.

Lineārās programmēšanas uzdevuma risināšanas programma ir uzrakstīta ar mērķi, lai varētu vēlāk to modificēt par kvadrātiskās programmēšanas uzdevuma risināšanas programmu.

Turpmāk apskatīsim lineārās programmēšanas uzdevuma risināšanas programmu dažādos lineārās programmēšanas uzdevumu piemēros.

1. uzdevums

$$\begin{aligned} f &= -5x_1 + 2x_2 \rightarrow \max \\ \begin{cases} -x_1 + x_2 \leq -3 \\ 2x_1 + 3x_2 \leq 15 \end{cases} \end{aligned} \quad (6.1.1)$$

Pārveidosim teksta formātā doto uzdevumu, lai turpmāk varētu apstrādāt ar lineārās programmēšanas uzdevuma risināšanas programmu^{1,2,3}. 1.pielikumā ir attēloti pirmā uzdevuma dati un to struktūra.



6.1.1.att. Mazais programmas logs

Atvērot programmu 23.pielikumā, parādās logs, kas ir redzams 6.1.1.attēlā, kuru turpmāk tekstā sauksim par mazo programmas logu. 6.1.1.attēlā izvēloties pogu “Rādīt

¹ Microsoft Visual Basic Help

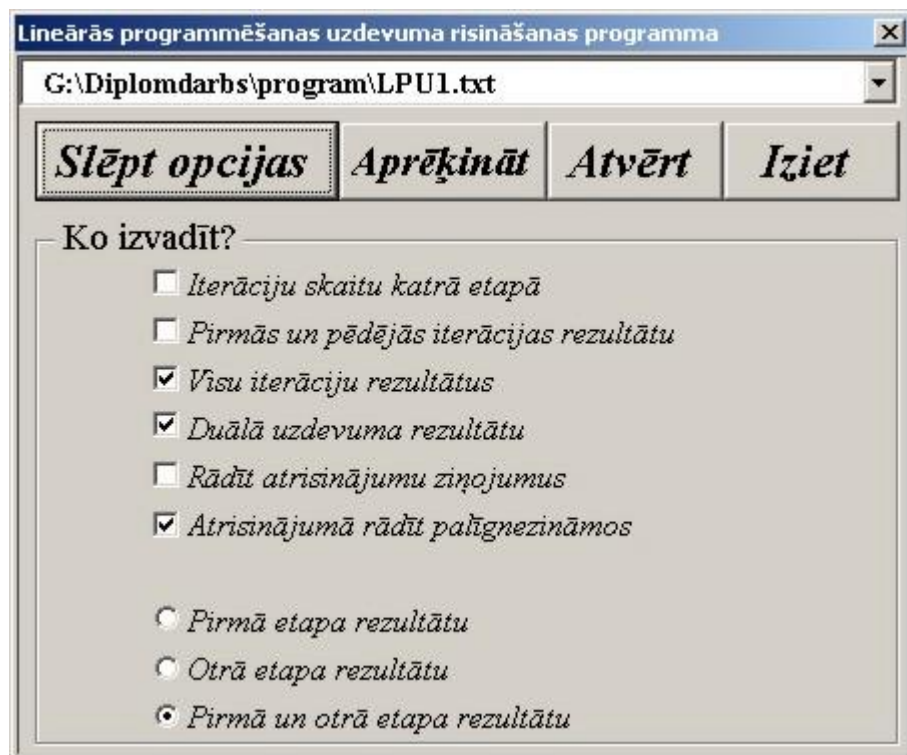
² Visual Basic 6.0: Пер. с англ. - СПб.: БХВ-Петербург, 2003

³ <http://www.freevbcode.com/listcode.asp?Category=7>

opcijas”, parādās lielāks logs, kas ir redzams 6.1.3.attēlā, ko nosauksim par paplašināto logu. Šajā logā ir redzams, kādas izvēlnes ir pēc noklusējuma. Ar pogu “Aprēķināt” turpmāk veiks datu pārbaudi, vai ir tāda datu struktūra kā 1.pielikumā, kā arī nepieciešamos aprēķinus, bet ar pogu “Atvērt” var izmantot tikai gadījumā, kad ir bijusi pielietota poga “Aprēķināt”. Poga “Iziet” ir nepieciešama, lai varētu iziet no lineārās programmēšanas uzdevuma programmas, kā arī var saglabāt visas nepieciešamās darba burtnīcas. Šajā 6.1.1.attēlā ir redzams adreses logs, kurā ir norādīts, kur atrodas pirmā uzdevuma dati. Kad lietotājs izvēlas pogu “Aprēķināt” vai “Atvērt”, tad programma veic pārbaudi, vai lietotājs ir ievadījis korektu adresi un faila nosaukumu, kurā glabājas pirmā uzdevuma dati. Ja kādā no pārbaudēm ir atrasta kļūda, tad lietotājam tiek paziņots kļūdu ziņojums, piemēram, 6.1.2.attēlā.



6.1.2.att. Kļūdu ziņojums



6.1.3.att. 1.uzdevuma paplašinātais programmas logs

Ja 6.1.3.attēlā izvēlne “Visu iterāciju rezultātus” tiks izvēlēta, tas nozīmē, ka tiks izvadītas visu iterāciju tabulas, bet, ja tā netiks izvēlēta, tad tiks izvadīts tikai pirmās un pēdējās iterācijas rezultāts, jo būs aktivizēta izvēlne “Pirmās un pēdējās iterācijas rezultātu”.

Par pirmo etapu uzskata palīguzdevumu, kurā maksimizē jaunu mērķa funkciju h , kurai nezināmo skaits ir vienāds sākotnējā uzdevuma negatīvo koeficientu skaitu un bāzes mainīgos izsaka ar citiem mainīgajiem, kas nav iekļauti bāzē.

Par otro etapu uzskata uzdevumu, kad sākotnējā uzdevumā ir vismaz viens negatīvs koeficients b , tad šim uzdevumam ir nepieciešams izveidot palīguzdevuma tabulu, kā arī sākotnējā uzdevuma mērķa funkcijā tiek izteikti tie mainīgie, kas ir iekļauti bāzē. Ja sākotnējā uzdevumā nav negatīvu b koeficientu, tad problēma tiek risināta tikai otrajā etapā.

Ja 6.1.3.attēlā izvēlne “Duālā uzdevuma rezultātu” tiks izvēlēta, tas nozīmē, ja uzdevumam ir 2.etapa iterāciju rezultātu tabulas, tad no pēdējās tabulas tiks nolasīts duālā uzdevuma atrisinājums, bet, ja ir tikai 1.etapa iterāciju rezultātu tabulas, tad netiks izvadīts duālā uzdevuma atrisinājums.

Ja 6.1.3.attēlā izvēlne “Pirmā un otrā etapa rezultātu” tiks izvēlēta, tas nozīmē, ka uzdevumam tiks izvadītas gan pirmā, gan arī otrā etapa iterāciju rezultātu tabulas, bet ir iespējams izvēlēties, lai izvadītu tikai pirmā etapa iterāciju rezultātu tabulas, tad izvēlas izvēlni “Pirmā etapa rezultātu”, bet lai izvadītu tikai otrā etapa iterāciju rezultātu tabulas, tad izvēlas izvēlni “Otrā etapa rezultātu”. Ja tiks izvēlēta izvēlne “Pirmā etapa rezultātu”, tad izvēlni “Duālā uzdevuma rezultātu” nebūs iespējams izvēlēties. Lai izvadītu tikai otrā etapa iterāciju rezultātu tabulas, tad izvēlas izvēlni “Otrā etapa rezultātu”.

Ja 6.1.3.attēlā izvēlne “Atrisinājumā rādīt palīgnezināmos” tiks izvēlēta, tas nozīmē, ka atrisinājumā tiks izvadīti gan pamatmainīgie, gan palīgmainīgie, Ja netiks izvēlēta izvēlne “Atrisinājumā rādīt palīgnezināmos”, tad atrisinājumā tiks izvadīti tikai pamatmainīgie. Ja tiks izvēlēta izvēlne “Pirmā etapa rezultātu”, tad izvēlne “Atrisinājumā rādīt palīgnezināmos” tiks aktivizēta un šo nebūs iespējams atslēgt. Ja tiks izvēlēta izvēlne “Pirmā un otrā etapa rezultātu” un uzdevumam ir 1. etaps, tad 1. etapa rezultātu tabulā tiks izvadīti gan pamatmainīgie, gan palīgmainīgie.

Turpināsim analizēt 1.uzdevuma problēmu.

Lai varētu veikt aprēķinus, ir nepieciešams nospiegt pogu “Aprēķināt” un dati tiek saglabāti speciālā datu failā.

1.uzdevumā ir negatīvs viens b koeficients. Šo uzdevumu ir nepieciešams sākt risināt ar 1. etapu, t.i., palīguzdevumu. Ja, maksimizējot palīguzdevuma mērķa funkciju, rezultātā iegūst 0, tas nozīmē, ka sākotnējam uzdevumam ir atrisinājums, bet ja nav atrisinājuma, tad

tiek izvadīts ziņojums: “Uzdevumam nav atrisinājuma, jo sākotnējam uzdevumam plānu kopa ir tukša.”

Lai izvadītu rezultātus MS Excel vidē, pielietosim pogu “Atvērt”, kura izveda datus pēc iepriekš uzstādītām prasībām. Pēc tam nepieciešams aizvērt programmas logu, ko var izdarīt nospiežot pogu “Iziet”. Ja tiek nospiesta poga “Iziet”, tad tas ir nepieciešams, lai datus varētu saglabāt, kā arī vēlāk varētu gan apskatīt, gan arī apstrādāt MS Excel vidē. Šī uzdevuma 1. etapa atrisinājums ir apskatāms 2.pielikumā, bet 3.pielikumā ir redzams 2.etapa atrisinājums.

2.uzdevums

$$\begin{aligned} f &= 6x_1 + 2x_2 \rightarrow \max \\ \begin{cases} -12x_1 + 3x_2 \leq 6 \\ -10x_1 + 2x_2 \leq 9 \\ -3x_1 + x_2 \leq 5 \end{cases} \end{aligned} \quad (6.1.2)$$

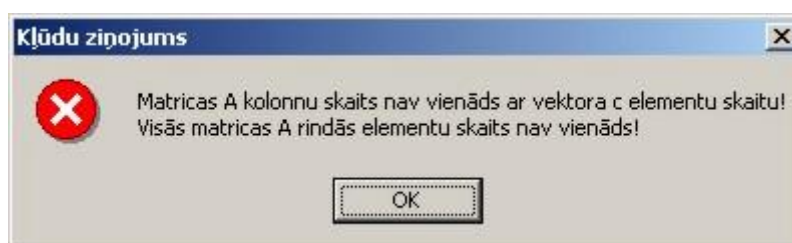
Pārveidosim teksta formātā doto uzdevumu, lai turpmāk varētu apstrādāt ar lineārās programmēšanas uzdevuma programmu.



6.1.4. att. 2.uzdevuma mazais logs

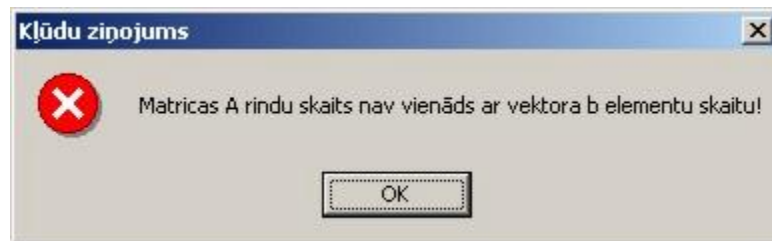
2.uzdevumam izmantosim mazo logu 6.1.4.attēlā un slēptās izvēlnes – 6.1.3. attēlā.

Ja 2.uzdevuma datu struktūrai tiks izmantoti 4.pielikumā attēlotie dati un tiks pielietota poga “Aprēķināt”, tad tiks izvadīts 6.1.5. attēla kļūdu ziņojums.



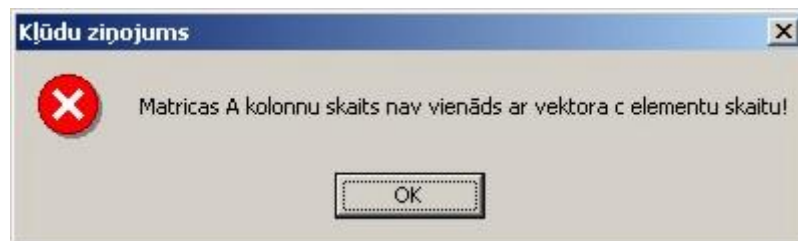
6.1.5. att. 2.uzdevuma kļūdu ziņojums

Ja 2.uzdevuma datu struktūrai tiks izmantoti 5.pielikumā attēlotie dati un tiks pielietota poga “Aprēķināt”, tad tiks izvadīts 6.1.6. attēla kļūdu ziņojums.



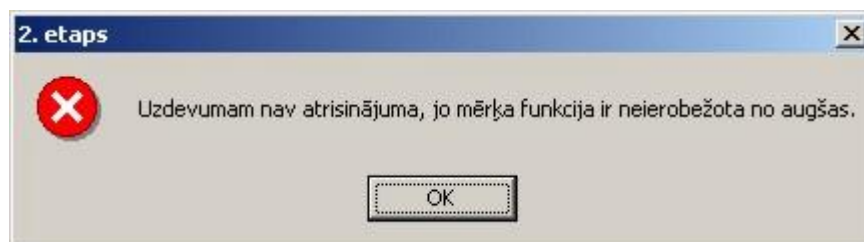
6.1.6. att. 2.uzdevuma kļūdu ziņojums

Ja 2.uzdevuma datu struktūrai tiks izmantoti 6.pielikumā attēlotie dati un tiks pielietota poga “Aprēķināt”, tad tiks izvadīts 6.1.7. attēla kļūdu ziņojums.



6.1.7. att. 2.uzdevuma kļūdu ziņojums

Ja 2.uzdevuma datu struktūrai tiks izmantoti 7.pielikumā attēlotie dati, tad netiks izvadīts kļūdu ziņojums un programma veiks aprēķinus. Veicot aprēķinus, programma ir konstatējusi, ka uzdevumam nav atrisinājuma un izvada ziņojumu 6.1.8.attēlā.



6.1.8. att. 2.uzdevuma 2.etapa kļūdu ziņojums

No 6.1.8. attēla varam secināt, ka 2.etapā pēdējā iterācijā eksistē kāda kolonna, kurā visi elementi ir negatīvi, tātad uzdevumam nav atrisinājuma, jo mērķa funkcija ir neierobežota no augšas.

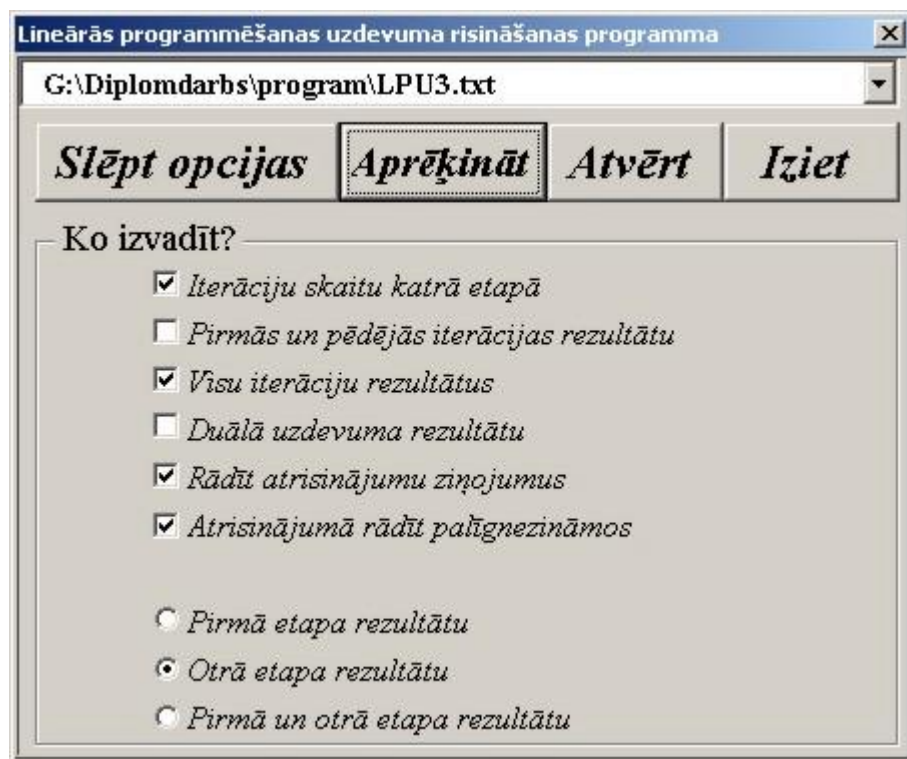
Lai izvadītu rezultātus MS Excel vidē, pielietosim pogu “Atvērt”, kura izvada datus pēc iepriekš uzstādītām prasībām. Pēc tam nepieciešams aizvērt programmas logu, ko var izdarīt

nospiežot pogu “Iziet”. Tā ir nepieciešams darīt, lai varētu saglabāt, lai vēlāk varētu gan apskatīt, gan arī apstrādāt MS Excel vidē. Šī uzdevuma 2.etapa atrisinājums ir apskatāms 8.pielikumā.

3. uzdevums

$$\begin{aligned} f &= 3x_1 + 2x_2 \rightarrow \max \\ \begin{cases} x_1 + 2x_2 \leq 6 \\ 2x_1 + x_2 \leq 8 \\ -x_1 + x_2 \leq 1 \\ x_2 \leq 2 \end{cases} \end{aligned} \quad (6.1.3)$$

Pārveidosim teksta formātā doto uzdevumu, lai turpmāk varētu apstrādāt ar lineārās programmēšanas uzdevuma programmu. 9.pielikumā ir attēloti 3.uzdevuma dati.

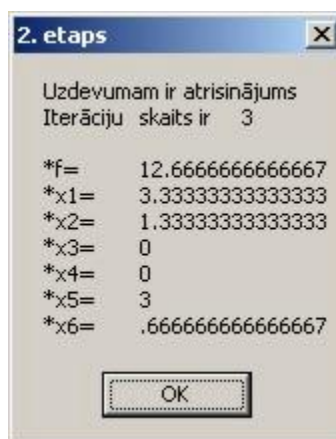


6.1.9.att. 3.uzdevuma paplašinātais programmas logs

Ja 6.1.9.attēlā izvēlne “Iterāciju skaitu katrā etapā” tiks izvēlēta, tas nozīmē, ka tiks izvadīts iterāciju skaits katrā etapā, bet, ja šī izvēlne netiks aktivizēta, tad netiks izvadīts iterāciju skaits katrā etapā.

Ja 6.1.9.attēlā izvēlne “Rādīt atrisinājumu ziņojumus” tiks izvēlēta, tas nozīmē, ka tiks rādīti katra etapa atrisinājumu ziņojumi, piemēram, 6.1.10.attēlā. Ja tiks vai ir izvēlēta izvēlne “Iterāciju skaitu katrā etapā”, tad atrisinājumu ziņojumos arī tiks izvadīts iterāciju skaits katrā etapā, bet, ja tiks vai arī jau ir izvēlēta izvēlne “Duālā uzdevuma rezultātu” un uzdevumam 2.etapā ir atrisinājums, tad arī tiks izvadīts ziņojums ar duālā uzdevuma atrisinājumu.

Pēc tam, kad poga “Aprēķināt” ir tikusi izvēlēta, tiek izvadīts 6.1.10.attēls. Lai varētu veikt aprēķinus, ir nepieciešams nospiegt pogu “Aprēķināt”, un dati tiek saglabāti speciālā datu failā.



6.1.10. attēls. 3.uzdevuma 2.etapa atrisinājuma ziņojums

Lai izvadītu rezultātus MS Excel vidē, pielietosim pogu “Atvērt”, kura izveda datus pēc iepriekš uzstādītām prasībām. Pēc tam nepieciešams aizvērt programmas logu, ko var izdarīt nospiežot pogu “Iziet”. Tā ir nepieciešams darīt, lai varētu saglabāt, lai vēlāk varētu gan apskatīt, gan arī apstrādāt MS Excel vidē. Šī uzdevuma 2.etapa atrisinājums ir apskatāms 10.pielikumā.

6.2. Kvadrātiskās programmēšanas uzdevuma risināšanas programma

Iepriekšējā nodaļā tika apskatīta lineārās programmēšanas uzdevuma risināšanas programma, un pievienojot papildus nosacījumus, varam risināt kvadrātiskās programmēšanas uzdevumus. Kvadrātiskās programmēšanas uzdevuma risināšanas programmas pogas un izvēlnes, kas saglabājušās no lineārās programmēšanas uzdevuma risināšanas programmas, ir aprakstītas iepriekšējā nodaļā, kā arī tiks aprakstītas papildus iespējas. Šajā nodaļā tiks apskatīti trīs uzdevumi.

1. uzdevums

$$f = \sum_{i=1}^3 \sum_{j=1}^3 d_{ij} x_i x_j \rightarrow \min$$
$$\begin{cases} 0 \leq x_i \leq 1, & i = \overline{1,3} \\ \sum_{j=1}^3 x_i = 1 \end{cases} \quad (6.2.1)$$

kur

$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Pārveidosim teksta formātā doto uzdevumu, lai turpmāk varētu apstrādāt ar kvadrātiskās programmēšanas uzdevuma risināšanas programmu^{1,2,3}. 11.pielikumā ir attēloti pirmā uzdevuma dati.



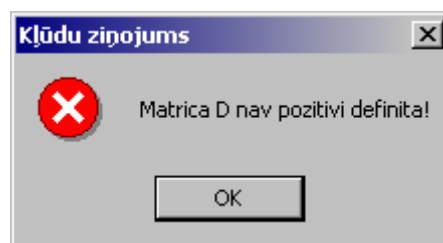
6.2.1.att. Mazais programmas logs

Atvērot programmu, parādās mazais programmas logs, kas ir redzams 6.2.1.attēlā. 6.2.1.attēlā izvēloties pogu “Rādīt opcijas”, parādās paplašinātais logs, kas ir redzams 6.2.3.attēlā. Šajā logā ir redzams, kādas izvēlnes ir pēc noklusējuma. Ar pogu “Aprēķināt” turpmāk veiks datu pārbaudi, vai ir tāda datu struktūra kā 11.pielikumā, kā arī nepieciešamos aprēķinus, bet ar pogu “Atvērt” var izmantot tikai gadījumā, kad ir bijusi pielietota poga “Aprēķināt”. Bet poga “Iziet” ir nepieciešama, lai varētu iziet no lineārās programmēšanas uzdevuma programmas, kā arī var saglabāt visas nepieciešamās darba burtnīcas. Šajā 6.2.1.attēlā ir redzams adreses logs, kurā ir norādīts, kur atrodas pirmā uzdevuma dati. Kad lietotājs izvēlas pogu “Aprēķināt” vai “Atvērt”, tad programma veic pārbaudi, vai lietotājs ir ievadījis korektu adresi un faila nosaukumu, kurā glabājas pirmā uzdevuma dati. Ja kādā no pārbaudēm ir atrasta kļūda, tad lietotājam tiek paziņots kļūdu ziņojums, piemēram, 6.2.2.attēlā.

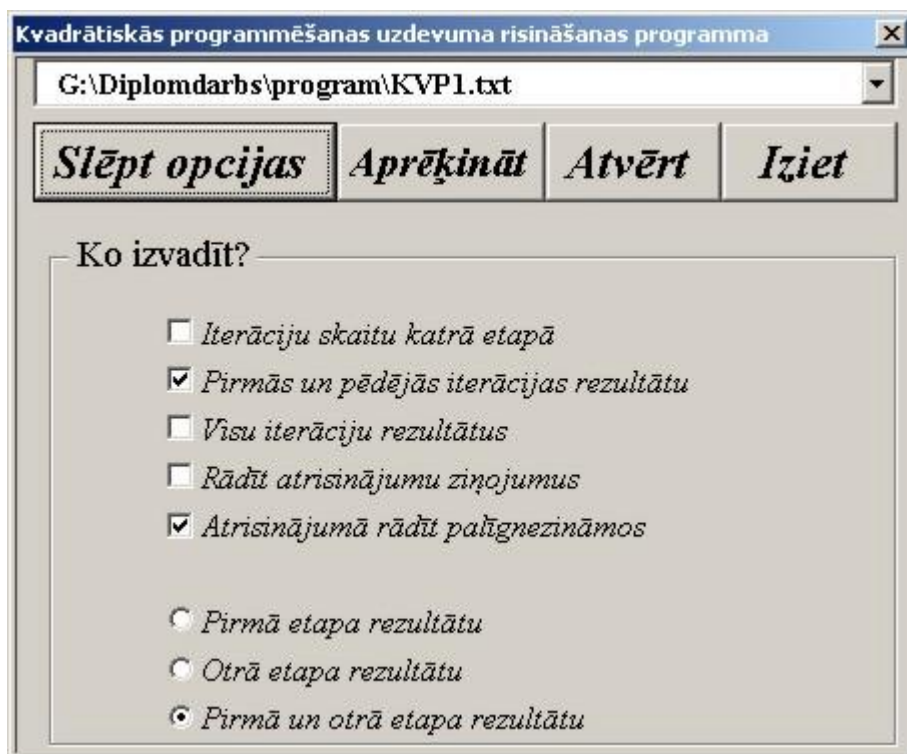
¹ Microsoft Visual Basic Help

² Visual Basic 6.0: Пер. с англ. - СПб.: БХВ-Петербург, 2003

³ <http://www.freevbcode.com/listcode.asp?Category=7>



6.1.2.att. Kļūdu ziņojums



6.2.3.att. 1.uzdevuma paplašinātais programmas logs

Salīdzinot 6.2.3.attēlu ar 6.1.3. attēlu, ir redzams, ka ir izslēgta izvēlne “Duālā uzdevuma rezultātu”. Ja 6.2.3.attēlā sākotnēji ir aktivizēta izvēlne “Pirmās un pēdējās iterācijas rezultātu”, bet 6.2.3. attēlā ir aktivizēta izvēlne “Visu iterāciju rezultātus”.

Ja tiks izslēgta izvēlne “Atrisinājumā rādīt palīgmmainīgos”, tad 2.etapā rezultātā tiks izvadīti x_i , $i = \overline{1, n}$.

Tā kā 6.2.2. attēlā tiek izvadīts kļūdu ziņojums par to, ka matrica D nav pusdefinita, bet tas nozīmē, ka neizpildās būtisks nosacījums.

2. uzdevums

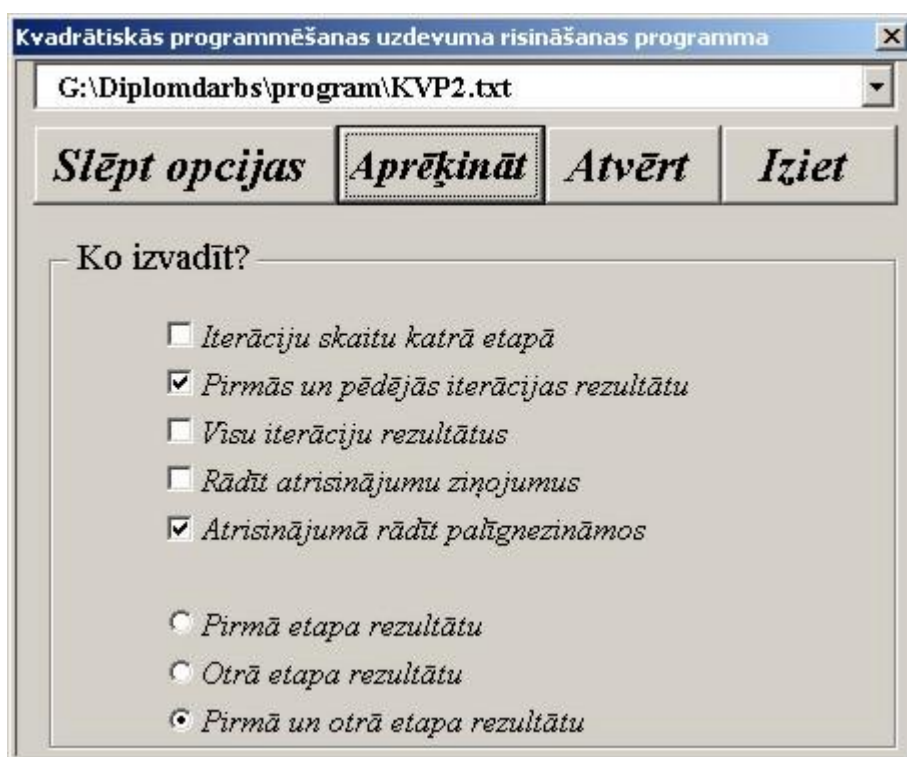
$$f = \sum_{i=1}^5 \sum_{j=1}^5 d_{ij} x_i x_j \rightarrow \min$$

$$\begin{cases} 0 \leq x_i \leq 1, & i = \overline{1,5} \\ \sum_{j=1}^5 x_j = 1 \end{cases} \quad (6.2.2)$$

kur

$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} \end{pmatrix} = \begin{pmatrix} 19 & 2 & 9 & 11 & 40 \\ 2 & 11 & 0 & -1 & 29 \\ 9 & 0 & 119 & 0 & 73 \\ 11 & -1 & 0 & 118 & 45 \\ 40 & 29 & 73 & 45 & 1121 \end{pmatrix}$$

Pārveidosim teksta formātā doto uzdevumu, lai turpmāk varētu apstrādāt ar kvadrātiskās programmēšanas uzdevuma risināšanas programmu. 12.pielikumā ir attēloti 2.uzdevuma dati, kur ir dots sistēmas vienādojumu skaits, ko apzīmēsim ar ψ , un matrica A ir nevienādību formā. , izņemot pēdējās ψ rindas ir vienādojumu formā.



6.2.3.att. 2.uzdevuma paplašinātais programmas logs

Tā kā 2.uzdevumā ir viens vienādojums, t.i., $x_1 + x_2 + x_3 + x_4 + x_5 = 1$, tad to kvadrātiskās programmēšanas programma pārveido par divām nevienādībām, t.i.,

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 1 \\ x_1 + x_2 + x_3 + x_4 + x_5 \geq 1 \end{cases} \quad (6.2.3)$$

jeb

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \leq 1 \\ -x_1 - x_2 - x_3 - x_4 - x_5 \leq -1 \end{cases} \quad (6.2.4)$$

2. uzdevumā 1.vienādojumā vietā tagad būs divas nevienādības un uzdevums būs jārisina ar 1.etapu, jo tagad vektorā b viena komponente ir negatīva.

Lai varētu veikt aprēķinus, ir nepieciešams nospiegt pogu “Aprēķināt”, un dati tiek saglabāti speciālā datu failā.

Lai izvadītu rezultātus MS Excel vidē, pielietosim pogu “Atvērt”, kura izvada datus pēc iepriekš uzstādītām prasībām. Pēc tam nepieciešams aizvērt programmas logu, ko var izdarīt nospiežot pogu “Iziet”. Tā ir nepieciešams darīt, lai varētu saglabāt, lai vēlāk varētu gan apskatīt, gan arī apstrādāt MS Excel vidē. Šī uzdevuma 1. etapa atrisinājums ir apskatāms 13.pielikumā, bet 14.pielikumā ir redzams 2.etapa atrisinājums.

3. uzdevums

Rīgas Fondu Biržā¹ tiek pārdotas akcijas. Apskatīsim “Eesti Telekom”, “Latvijas Gāze”, “Harju Elekter”, “Vilniaus Vingis”, “SAF tehnika”, “Pieno žvaigždēs” akciju cenas no 2005. gada 2.maija līdz 2006. gada 7.maijam, kā ir veikts pieņēmums, ka šiem 6 uzņēmumiem gada laika netiek maksātas dividendes. Ir nepieciešams atrast minimālo tirgus portfeli, kuru var veidot tikai no iepriekš minēto uzņēmumu akcijām.

15.pielikumā ir redzamas akciju vidējās cenas nedēļā. Šīs cenas tika aprēķināta

$$\text{vidējā akcijas cena} = \frac{\text{kopējais apgrozījumu nedēļā}}{\text{akciju skaitu nedēļā}} \quad (6.2.5)$$

16.pielikumā ir aprēķināts akciju ienesīgums ar formulu 2.2.1, kā arī 17.pielikumā kovariācijas matrica ar formulu 2.3.10.

Tad no 2.4. nodaļas varam uzrakstīt sekojošu minimizācijas uzdevumu

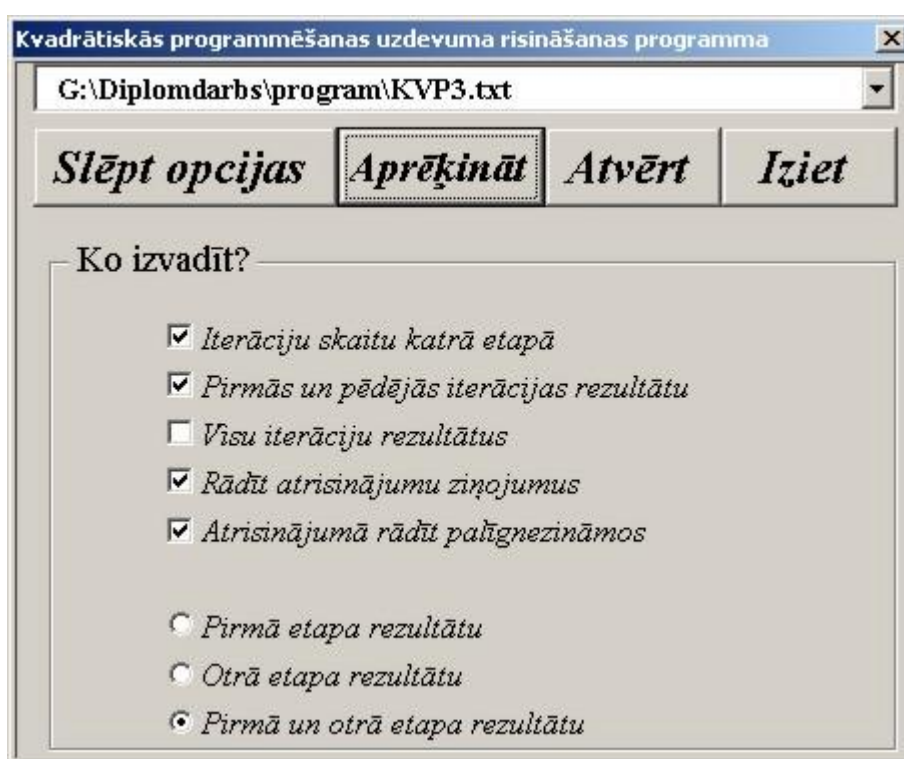
¹ <http://market.lv.omxgroup.com/?pg=stats&lang=lv>

$$f = \sum_{i=1}^6 \sum_{j=1}^6 d_{ij} x_i x_j \rightarrow \min$$

$$\begin{cases} 0 \leq x_i \leq 1, & i = \overline{1,6} \\ \sum_{j=1}^6 x_j = 1 \end{cases} \quad (6.2.6)$$

kur matricas D elementi ir 16.pielikumā un x_i ir i-tā vērtspapīra svars portfelī.

Pārveidosim teksta formātā doto uzdevumu, lai turpmāk varētu apstrādāt ar kvadrātiskās programmēšanas uzdevuma risināšanas programmu. 17.pielikumā ir attēloti 3.uzdevuma dati.



6.2.4.att. 3.uzdevuma paplašinātais programmas logs

Lai varētu veikt aprēķinus, ir nepieciešams nospiegt pogu “Aprēķināt”, kā arī tiek izvadīti gan pirmā etapa ziņojums – 19.pielikumā, gan otrā etapa ziņojums – 20.pielikumā. Dati tiek saglabāti speciālā datu failā.

Lai izvadītu rezultātus MS Excel vidē, pielietosim pogu “Atvērt”, kura izvada datus pēc iepriekš uzstādītām prasībām. Pēc tam nepieciešams aizvērt programmas logu, ko var izdarīt nospiežot pogu “Iziet”. Tā ir nepieciešams darīt, lai varētu saglabāt, lai vēlāk varētu gan apskatīt, gan arī apstrādāt MS Excel vidē. Šī uzdevuma 1. etapa atrisinājums ir apskatāms 21.pielikumā, bet 22.pielikumā ir redzams 2.etapa atrisinājums.

Secinājumi

Šajā diplomdarbā tika izstrādāta gan teorētiskā, gan praktiskā daļa. Vulfa – Daugaveta metode tika pielietota diplomdarba izstrādē, lai noteiktu minimālo investora portfeli.

Diplomdarbā tika apskatīts tirgus portfeļa minimizācijas uzdevums, kas tika reducēts uz lineāras programmēšanas uzdevumu ar dažiem nelineāriem nosacījumiem. Modificēts simpleksa algoritms tika piedāvāts uzdevuma risināšanai. Tika izstrādāta speciālā programmatūra uzstādītā uzdevuma atrisināšanai.

3. kvadrātiskās programmēšanas uzdevuma 2.etapa rezultāts ir salīdzināts ar Microsoft Excel palīgīdzekli “Solver”, un rezultāti sakrīta.

Diplomdarba izstrādāšanas laikā parādījās laba iespēja pielietot universitātē iegūtās teorētiskās zināšanas dažādos priekšmetos praktisko uzdevumu risināšanai. Tika arī apgūti teorijas papildjautājumi par vērtspapīru portfeļiem, kā arī padziļinātas praktiskās iemaņas darbā ar Microsoft Visual Basic, Microsoft Excel.

Izmantotā literatūra

1. Даугавет В. А. Модификация метода Вулфа. Журнал вычислительной математики и математической физики, 1981, 21, 2, 504 – 508.lpp.
2. D.Kļaviņš, P.Zelčs “Operāciju pētīšanas matemātiskās metodes”, Rīga, “Zvaigzne”, 1979
3. Хедли Дж. Нелинейное и динамическое программирование. М.: Мир, 1967
4. Ģ. Apsītis, I. Aščuks, , U. Cērps, G. Kokorēvičs, Ģ. Ozols, A. Sedlenieks, H. Zuļģis, “Vērtspapīru tirgus zinības” Otrais papildinātais izdevums, Rīga, Jumava, 2006
5. J. Vucāns “Vērtspapīru portfeļi un to vadīšana” lekciju konspekti, 2005
6. Кюнц Г.П., Крелле В. Нелинейное программирование. М.: Сов. радио, 1965
7. <http://market.lv.omxgroup.com/?pg=stats&lang=lv>
8. <http://www.freevbcode.com/listcode.asp?Category=7>
9. Microsof Visual Basic Help
10. Raitums U. “Optimizācijas metodes: lekciju kurss”, Rīga, Latvijas Universitāte, 2002
11. Thomas L. Seaty, Joseph Bram, “Nonlinear Mathematics”, McCraw - Hill Book Company, New York, San Francisko, Toronto, London.
12. Visual Basic 6.0: Пер. с англ. - СПб.: БХВ-Петербург, 2003.

Pielikumi

1. pielikums

Pirmais lineārās programmēšanas uzdevuma ievadāmo datu piemērs

Transponets vektors c ir

$(-5 \quad 2)$.

Matrica A ir

$(-1 \quad 1)$

$(2 \quad 3)$.

Transponets vektors b ir

$(-3 \quad 15)$.

2. pielikums

Pirmā lineārās programmēšanas uzdevuma 1.etapa rezultāti

bāze	h	x1	x2	x3	x4	u1	b
u1	0	1	-1	-1	0	1	3
x4	0	2	3	0	1	0	15
h	1	-1	1	1	0	0	-3

bāze	h	x1	x2	x3	x4	u1	b
x1	0	1	-1	-1	0	1	3
x4	0	0	5	2	1	-2	9
h	1	0	0	0	0	1	0

Uzdevumam ir atrisinājums

*h= 0

*x1= 3

*x2= 0

*x3= 0

*x4= 9

*u1= 0

Pirmā lineārās programmēšanas uzdevuma 2.etapa rezultāti

bāze	f	x1	x2	x3	x4	B
x1	0	1	-1	-1	0	3
x4	0	0	5	2	1	9
f	1	0	3	5	0	-15

Uzdevumam ir atrisinājums

$$\begin{aligned}
 *f &= -15 \\
 *x_1 &= 3 & *x_3 &= 0 \\
 *x_2 &= 0 & *x_4 &= 9
 \end{aligned}$$

Duālajam uzdevumam ir atrisinājums

$$\begin{aligned}
 *g &= -15 \\
 *y_1 &= 5 & *y_3 &= 0 \\
 *y_2 &= 0 & *y_4 &= 3
 \end{aligned}$$

Otrā lineārās programmēšanas uzdevuma pirmais nepareizi ievadīto datu piemērs

Transponets vektors c ir

$$(6 \quad 2).$$

Matrica A ir

$$\begin{pmatrix} -12 & 3 \\ -10 & 2 & 1 \\ -3 & 1 \end{pmatrix}.$$

Transponets vektors b ir

$$(6 \quad 9 \quad 5).$$

Otrā lineārās programmēšanas uzdevuma otrais nepareizi ievadīto datu piemērs

Transponets vektors c ir

$$(6 \quad 2).$$

Matrica A ir

$$(-12 \quad 3)$$

$$(-10 \quad 2)$$

$$(-3 \quad 1).$$

Transponets vektors b ir

$$(6 \quad 9).$$

Otrā lineārās programmēšanas uzdevuma trešais nepareizi ievadīto datu piemērs

Transponets vektors c ir

$$(6 \quad 2 \quad 1).$$

Matrica A ir

$$(-12 \quad 3)$$

$$(-10 \quad 2)$$

$$(-3 \quad 1).$$

Transponets vektors b ir

$$(6 \quad 9 \quad 5).$$

Otrā lineārās programmēšanas uzdevuma pareizi ievadīts datu piemērs

Transponets vektors c ir

$$(6 \quad 2).$$

Matrica A ir

$$\begin{pmatrix} -12 & 3 \end{pmatrix}$$

$$\begin{pmatrix} -10 & 2 \end{pmatrix}$$

$$\begin{pmatrix} -3 & 1 \end{pmatrix}.$$

Transponets vektors b ir

$$(6 \quad 9 \quad 5).$$

Otrā lineārās programmēšanas uzdevuma 2.etapa rezultāti

bāze	f	x1	x2	x3	x4	x5	B
x3	0	-12	3	1	0	0	6
x4	0	-10	2	0	1	0	9
x5	0	-3	1	0	0	1	5
f	1	-6	-2	0	0	0	0

Uzdevumam nav atrisinājuma, jo mērķa funkcija ir neierobežota no augšas.

Trešā lineārās programmēšanas uzdevuma ievadāmo datu piemērs

Transponets vektors c ir

$$(3 \quad 2).$$

Matrica A ir

$$(1 \quad 2)$$

$$(2 \quad 1)$$

$$(-1 \quad 1)$$

$$(0 \quad 1).$$

Transponets vektors b ir

$$(6 \quad 8 \quad 1 \quad 2).$$

Trešā lineārās programmēšanas uzdevuma 2.etapa rezultāti

bāze	f	x1	x2	x3	x4	x5	x6	b
x3	0	1	2	1	0	0	0	6
x4	0	2	1	0	1	0	0	8
x5	0	-1	1	0	0	1	0	1
x6	0	0	1	0	0	0	1	2
f	1	-3	-2	0	0	0	0	0

bāze	f	x1	x2	x3	x4	x5	x6	b
x3	0	0	1 1/2	1	- 1/2	0	0	2
x1	0	1	1/2	0	1/2	0	0	4
x5	0	0	1 1/2	0	1/2	1	0	5
x6	0	0	1	0	0	0	1	2
f	1	0	- 1/2	0	1 1/2	0	0	12

bāze	f	x1	x2	x3	x4	x5	x6	b
x2	0	0	1	2/3	- 1/3	0	0	1 1/3
x1	0	1	0	- 1/3	2/3	0	0	3 1/3
x5	0	0	0	-1	1	1	0	3
x6	0	0	0	- 2/3	1/3	0	1	2/3
f	1	0	0	1/3	1 1/3	0	0	12 2/3

Uzdevumam ir atrisinājums

Iterāciju skaits ir 3

$$*f = 12 \frac{2}{3}$$

$$*x_1 = 3 \frac{1}{3}$$

$$*x_2 = 1 \frac{1}{3}$$

$$*x_3 = 0$$

$$*x_4 = 0$$

$$*x_5 = 3$$

$$*x_6 = \frac{2}{3}$$

11. pielikums

Pirmā kvadrātiskās programmēšanas uzdevuma ievadāmo datu piemērs

Sistēmas vienādojumu skaits ir 1.

Transponētais vektors c ir

$$(0 \ 0 \ 0).$$

Matrica A ir

$$(1 \ 0 \ 0)$$

$$(0 \ 1 \ 0)$$

$$(0 \ 0 \ 1)$$

$$(1 \ 1 \ 1).$$

Matrica D ir

$$(1 \ 2 \ 0)$$

$$(2 \ 1 \ 0)$$

$$(0 \ 0 \ 1).$$

Transponētais vektors b ir

$$(1 \ 1 \ 1 \ 1).$$

Otrā kvadrātiskās programmēšanas uzdevuma ievadāmo datu piemērs

Sistēmas vienādojumu skaits ir 1.

Transponētais vektors c ir

$$(0 \ 0 \ 0 \ 0 \ 0).$$

Matrica A ir

$$(1 \ 0 \ 0 \ 0 \ 0)$$

$$(0 \ 1 \ 0 \ 0 \ 0)$$

$$(0 \ 0 \ 1 \ 0 \ 0)$$

$$(0 \ 0 \ 0 \ 1 \ 0)$$

$$(0 \ 0 \ 0 \ 0 \ 1)$$

$$(1 \ 1 \ 1 \ 1 \ 1).$$

Matrica D ir

$$(19 \quad 2 \quad 9 \quad 11 \quad 40)$$

$$(2 \quad 11 \quad 0 \quad -1 \quad 29)$$

$$(9 \quad 0 \quad 119 \quad 0 \quad 73)$$

$$(11 \quad -1 \quad 0 \quad 118 \quad 45)$$

$$(40 \quad 29 \quad 73 \quad 45 \quad 1121).$$

Otrā kvadrātiskās programmēšanas uzdevuma 1.etapa rezultāti

bāze	h	x1	x2	x3	x4	x5	w1	w2	w3	w4	w5	w6	w7	v1	v2	v3
w1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
w2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
w3	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
w4	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
w5	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
w6	0	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
u1	0	1	1	1	1	1	0	0	0	0	0	0	-1	0	0	0
z1	0	38	4	18	22	80	0	0	0	0	0	0	0	-1	0	0
z2	0	4	22	0	-2	58	0	0	0	0	0	0	0	0	-1	0
z3	0	18	0	238	0	146	0	0	0	0	0	0	0	0	0	-1
z4	0	22	-2	0	236	90	0	0	0	0	0	0	0	0	0	0
z5	0	80	58	146	90	2242	0	0	0	0	0	0	0	0	0	0
h	1	-1	-1	-1	-1	-1	0	0	0	0	0	0	1	0	0	0

bāze	v4	v5	L1	L2	L3	L4	L5	L6	L7	z1	z2	z3	z4	z5	u1	b
w1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
w2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
w3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
w4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
w5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
w6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
u1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
z1	0	0	1	0	0	0	0	1	-1	1	0	0	0	0	0	0
z2	0	0	0	1	0	0	0	1	-1	0	1	0	0	0	0	0
z3	0	0	0	0	1	0	0	1	-1	0	0	1	0	0	0	0
z4	-1	0	0	0	0	1	0	1	-1	0	0	0	1	0	0	0
z5	0	-1	0	0	0	0	1	1	-1	0	0	0	0	1	0	0
h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1

bāze	h	x1	x2	x3	x4	x5
w1	0	0	0	0	0	-0.434921656
w2	0	0	0	0	0	-0.10080842
w3	0	0	0	0	0	-0.352543823
w4	0	0	0	0	0	-0.111726102
w5	0	0	0	0	0	1
w6	0	0	0	0	0	0
v5	0	0	0	0	0	-2085.566652
x2	0	0	1	0	0	0.10080842
L7	0	0	0	0	0	-54.26598035
x3	0	0	0	1	0	0.352543823
x4	0	0	0	0	1	0.111726102
x1	0	1	0	0	0	0.434921656
h	1	0	0	0	0	0

bāze	w1	w2	w3	w4	w5	w6
w1	1	0	0	0	0	0
w2	0	1	0	0	0	0
w3	0	0	1	0	0	0
w4	0	0	0	1	0	0
w5	0	0	0	0	1	0
w6	0	0	0	0	0	1
v5	0	0	0	0	0	0
x2	0	0	0	0	0	0
L7	0	0	0	0	0	0
x3	0	0	0	0	0	0
x4	0	0	0	0	0	0
x1	0	0	0	0	0	0
h	0	0	0	0	0	0

bāze	w7	v1	v2	v3	v4	v5
w1	0.283176492	0.024238231	-0.017606665	-0.003022961	-0.003608606	0
w2	0.632924148	-0.017606665	0.019807139	-0.001327749	-0.000872726	0
w3	0.041491787	-0.003022961	-0.001327749	0.004255973	9.47367E-05	0
w4	0.042407574	-0.003608606	-0.000872726	9.47367E-05	0.004386594	0
w5	0	0	0	0	0	0
w6	1	0	0	0	0	0
v5	-54.26598035	-0.434921656	-0.10080842	-0.352543823	-0.111726102	1
x2	-0.632924148	0.017606665	-0.019807139	0.001327749	0.000872726	0
L7	-14.97222206	0.283176492	0.632924148	0.041491787	0.042407574	0
x3	-0.041491787	0.003022961	0.001327749	-0.004255973	-9.47367E-05	0
x4	-0.042407574	0.003608606	0.000872726	-9.47367E-05	-0.004386594	0
x1	-0.283176492	-0.024238231	0.017606665	0.003022961	0.003608606	0
h	0	0	0	0	0	0

bāze	L1	L2	L3	L4	L5	L6
w1	-0.024238231	0.017606665	0.003022961	0.003608606	0	0
w2	0.017606665	-0.019807139	0.001327749	0.000872726	0	0
w3	0.003022961	0.001327749	-0.004255973	-9.47367E-05	0	0
w4	0.003608606	0.000872726	-9.47367E-05	-0.004386594	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
v5	0.434921656	0.10080842	0.352543823	0.111726102	-1	0
x2	-0.017606665	0.019807139	-0.001327749	-0.000872726	0	0
L7	-0.283176492	-0.632924148	-0.041491787	-0.042407574	0	-1
x3	-0.003022961	-0.001327749	0.004255973	9.47367E-05	0	0
x4	-0.003608606	-0.000872726	9.47367E-05	0.004386594	0	0
x1	0.024238231	-0.017606665	-0.003022961	-0.003608606	0	0
h	0	0	0	0	0	0

bāze	L7	z1	z2	z3	z4	z5
w1	0	-0.024238231	0.017606665	0.003022961	0.003608606	0
w2	0	0.017606665	-0.019807139	0.001327749	0.000872726	0
w3	0	0.003022961	0.001327749	-0.004255973	-9.47367E-05	0
w4	0	0.003608606	0.000872726	-9.47367E-05	-0.004386594	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
v5	0	0.434921656	0.10080842	0.352543823	0.111726102	-1
x2	0	-0.017606665	0.019807139	-0.001327749	-0.000872726	0
L7	1	-0.283176492	-0.632924148	-0.041491787	-0.042407574	0
x3	0	-0.003022961	-0.001327749	0.004255973	9.47367E-05	0
x4	0	-0.003608606	-0.000872726	9.47367E-05	0.004386594	0
x1	0	0.024238231	-0.017606665	-0.003022961	-0.003608606	0
h	0	0	0	0	0	0

bāze	u1	b
w1	-0.283176492	0.716823508
w2	-0.632924148	0.367075852
w3	-0.041491787	0.958508213
w4	-0.042407574	0.957592426
w5	0	1
w6	-1	0
v5	54.26598035	54.26598035
x2	0.632924148	0.632924148
L7	14.97222206	14.97222206
x3	0.041491787	0.041491787
x4	0.042407574	0.042407574
x1	0.283176492	0.283176492
h	1	0

Uzdevumam ir atrisinājums

*f= 0	
*x1= 0.515936092	*v4= 0
*x2= 0.127739289	*v5= 0
*x3= 0	*v6= 0
*x4= 0.025144388	*L1= 0
*x5= 0.172097011	*L2= 0
*x6= 0.159083219	*L3= 0
*w1= 0.484063908	*L4= 0
*w2= 0.872260711	*L5= 0
*w3= 1	*L6= 0
*w4= 0.974855612	*L7= 0
*w5= 0.827902989	*L8= 0.000283025
*w6= 0.840916781	*z1= 0
*w7= 0	*z2= 0
*w8= 0	*z3= 0
*v1= 0	*z4= 0
*v2= 0	*z5= 0
*v3= 0.00064926	*z6= 0

Otrā kvadrātiskās programmēšanas uzdevuma 2.etapa rezultāti

bāze	f	x1	x2	x3	x4	x5	w1
w1	0	0	0	0	0	-0.434921656	1
w2	0	0	0	0	0	-0.10080842	0
w3	0	0	0	0	0	-0.352543823	0
w4	0	0	0	0	0	-0.111726102	0
w5	0	0	0	0	0	1	0
w6	0	0	0	0	0	0	0
v5	0	0	0	0	0	-2085.566652	0
x2	0	0	1	0	0	0.10080842	0
L7	0	0	0	0	0	-54.26598035	0
x3	0	0	0	1	0	0.352543823	0
x4	0	0	0	0	1	0.111726102	0
x1	0	1	0	0	0	0.434921656	0
f	1	0	0	0	0	0	0

bāze	w2	w3	w4	w5	w6	w7	v1
w1	0	0	0	0	0	0.283176492	0.024238231
w2	1	0	0	0	0	0.632924148	-0.017606665
w3	0	1	0	0	0	0.041491787	-0.003022961
w4	0	0	1	0	0	0.042407574	-0.003608606
w5	0	0	0	1	0	0	0
w6	0	0	0	0	1	1	0
v5	0	0	0	0	0	-54.26598035	-0.434921656
x2	0	0	0	0	0	-0.632924148	0.017606665
L7	0	0	0	0	0	-14.97222206	0.283176492
x3	0	0	0	0	0	-0.041491787	0.003022961
x4	0	0	0	0	0	-0.042407574	0.003608606
x1	0	0	0	0	0	-0.283176492	-0.024238231
f	0	0	0	0	0	0	0

bāze	v2	v3	v4	v5	L1	L2	L3
w1	-0.017606665	-0.003022961	-0.003608606	0	-0.024238231	0.017606665	0.003022961
w2	0.019807139	-0.001327749	-0.000872726	0	0.017606665	-0.019807139	0.001327749
w3	-0.001327749	0.004255973	9.47367E-05	0	0.003022961	0.001327749	-0.004255973
w4	-0.000872726	9.47367E-05	0.004386594	0	0.003608606	0.000872726	-9.47367E-05
w5	0	0	0	0	0	0	0
w6	0	0	0	0	0	0	0
v5	-0.10080842	-0.352543823	-0.111726102	1	0.434921656	0.10080842	0.352543823
x2	-0.019807139	0.001327749	0.000872726	0	-0.017606665	0.019807139	-0.001327749
L7	0.632924148	0.041491787	0.042407574	0	-0.283176492	-0.632924148	-0.041491787
x3	0.001327749	-0.004255973	-9.47367E-05	0	-0.003022961	-0.001327749	0.004255973
x4	0.000872726	-9.47367E-05	-0.004386594	0	-0.003608606	-0.000872726	9.47367E-05
x1	0.017606665	0.003022961	0.003608606	0	0.024238231	-0.017606665	-0.003022961
f	0	0	0	0	0	0	0

bāze	L4	L5	L6	L7	z1	z2	z3
w1	0.003608606	0	0	0	-0.024238231	0.017606665	0.003022961
w2	0.000872726	0	0	0	0.017606665	-0.019807139	0.001327749
w3	-9.47367E-05	0	0	0	0.003022961	0.001327749	-0.004255973
w4	-0.004386594	0	0	0	0.003608606	0.000872726	-9.47367E-05
w5	0	0	0	0	0	0	0
w6	0	0	0	0	0	0	0
v5	0.111726102	-1	0	0	0.434921656	0.10080842	0.352543823
x2	-0.000872726	0	0	0	-0.017606665	0.019807139	-0.001327749
L7	-0.042407574	0	-1	1	-0.283176492	-0.632924148	-0.041491787
x3	9.47367E-05	0	0	0	-0.003022961	-0.001327749	0.004255973
x4	0.004386594	0	0	0	-0.003608606	-0.000872726	9.47367E-05
x1	-0.003608606	0	0	0	0.024238231	-0.017606665	-0.003022961
f	0	0	0	0	1	1	1

bāze	z4	z5	b
w1	0.003608606	0	0.716823508
w2	0.000872726	0	0.367075852
w3	-9.47367E-05	0	0.958508213
w4	-0.004386594	0	0.957592426
w5	0	0	1
w6	0	0	0
v5	0.111726102	-1	54.26598035
x2	-0.000872726	0	0.632924148
L7	-0.042407574	0	14.97222206
x3	9.47367E-05	0	0.041491787
x4	0.004386594	0	0.042407574
x1	-0.003608606	0	0.283176492
f	1	1	0

Uzdevumam ir atrisinājums

*f= 0	*v3= 0
*x1= 0.283176492	*v4= 0
*x2= 0.632924148	*v5= 54.26598035
*x3= 0.041491787	*L1= 0
*x4= 0.042407574	*L2= 0
*x5= 0	*L3= 0
*w1= 0.716823508	*L4= 0
*w2= 0.367075852	*L5= 0
*w3= 0.958508213	*L6= 0
*w4= 0.957592426	*L7= 14.97222206
*w5= 1	*z1= 0
*w6= 0	*z2= 0
*w7= 0	*z3= 0
*v1= 0	*z4= 0
*v2= 0	*z5= 0

**Akciju vidējās nedēļas cenas Rīgas Fondu biržā
no 2005. gada 2.maija līdz 2006. gada 7.maijam**

Nedēļas numurs	Datums	Vidējās nedēļas cenas (kursi)					
		Eesti Telekom	Latvijas Gāze	Harju Elekter	Vilniaus Vingis	SAF tehnika	Pieno žvaigždēs
1	2005.05.02-2005.05.08	8.09899	9.83064	13.0884	1.4505	19.7551	1.46876
2	2005.05.09-2005.05.15	8.04662	9.54942	4.98528	1.4503	19.4506	1.45383
3	2005.05.16-2005.05.22	8.07743	9.56443	4.55743	1.448	19.2097	1.45305
4	2005.05.23-2005.05.29	8.19646	9.56276	4.43804	1.4481	18.7016	1.48034
5	2005.05.30-2005.06.05	7.58399	9.58937	4.31975	1.447	18.3956	1.4864
6	2005.06.06-2005.06.12	7.48822	9.57979	4.40936	1.4442	18.3956	1.51668
7	2005.06.13-2005.06.19	7.39344	9.8338	3.72845	1.4043	17.9755	1.5202
8	2005.06.20-2005.06.26	7.38665	10.4578	4.14433	1.4047	17.9755	1.52286
9	2005.06.27-2005.07.03	7.38485	10.5284	4.4445	1.379	17.9311	1.5207
10	2005.07.04-2005.07.10	7.35845	10.5455	4.33359	1.303	18.0883	1.51295
11	2005.07.11-2005.07.17	7.32209	10.6732	4.41718	1.3037	18.0705	1.50656
12	2005.07.18-2005.07.24	7.30612	10.7194	4.51217	1.2864	18.0497	1.50025
13	2005.07.25-2005.07.31	7.36691	10.4249	4.51735	1.2934	18.1016	1.48466
14	2005.08.01-2005.08.07	7.36427	10.217	4.47646	1.3119	18.1661	1.44703
15	2005.08.08-2005.08.14	7.35933	10.318	4.52803	1.3177	18.1661	1.44249
16	2005.08.15-2005.08.21	7.35718	10.3308	4.50809	1.3179	17.8779	1.43689
17	2005.08.22-2005.08.28	7.35059	10.2803	4.49726	1.3199	18.0606	1.43953
18	2005.08.29-2005.09.04	7.35351	10.4122	4.491	1.3241	18.0703	1.39593
19	2005.09.05-2005.09.11	7.39529	10.5994	4.35001	1.3886	18.4525	1.40577
20	2005.09.12-2005.09.18	7.3803	10.7247	4.2985	1.4602	18.4798	1.40597
21	2005.09.19-2005.09.25	7.4778	10.7301	4.32416	1.4609	18.6649	1.402
22	2005.09.26-2005.10.02	7.58877	10.9104	4.31349	1.4723	18.925	1.38867
23	2005.10.03-2005.10.09	7.82059	11.0692	4.45182	1.5071	19.2088	1.37487
24	2005.10.10-2005.10.16	7.60973	11.4843	4.36182	1.3838	18.872	1.36011
25	2005.10.17-2005.10.23	7.44633	11.5879	4.30911	1.2923	18.8554	1.34851
26	2005.10.24-2005.10.30	7.37233	11.9162	4.30737	1.2923	16.4118	1.3657
27	2005.10.31-2005.11.06	7.58907	12.5021	4.30054	1.2496	16.1854	1.34574
28	2005.11.07-2005.11.13	7.61696	12.735	4.25635	1.2513	16.5464	1.36351
29	2005.11.14-2005.11.20	7.71739	12.7264	4.12914	1.1851	16.5654	1.33944
30	2005.11.21-2005.11.27	7.75428	12.724	3.89701	1.0642	16.5765	1.32934
31	2005.11.28-2005.12.04	7.75541	12.6603	3.96648	1.1202	16.6127	1.29369
32	2005.12.05-2005.12.11	7.7019	12.1474	4.05339	1.143	16.363	1.29236
33	2005.12.12-2005.12.18	7.55117	12.4992	4.10704	1.1669	16.4353	1.32175
34	2005.12.19-2005.12.25	7.51618	12.8565	4.04455	1.1803	16.4179	1.31358
35	2005.12.26-2006.01.01	7.57787	13.7911	4.0462	1.2526	16.4559	1.31748

Nedēļas numurs	Datums	Vidējās nedēļas cenas (kursi)					
		Eesti Telekom	Latvijas Gāze	Harju Elekter	Vilniaus Vingis	SAF tehnika	Pieno žvaigždēs
36	2006.01.02-2006.01.08	7.66677	14.4507	4.1917	1.1743	16.7899	1.52586
37	2006.01.09-2006.01.15	7.7761	15.602	4.31262	1.1585	16.7606	1.34092
38	2006.01.16-2006.01.22	7.75674	15.4642	4.1864	1.1584	16.5133	1.37627
39	2006.01.23-2006.01.29	7.71436	15.3016	4.14348	1.1566	16.5086	1.36898
40	2006.01.30-2006.02.05	7.74556	13.9418	4.03193	1.118	16.283	1.31909
41	2006.02.06-2006.02.12	7.73476	14.1653	3.873	1.0948	15.6516	1.30407
42	2006.02.13-2006.02.19	7.74826	13.9947	3.96259	0.9847	15.5093	1.2916
43	2006.02.20-2006.02.26	7.75006	14.0684	3.92459	0.8904	15.5093	1.27536
44	2006.02.27-2006.03.05	7.76345	13.6253	3.95459	0.9128	15.0617	1.27876
45	2006.03.06-2006.03.12	7.86166	13.6877	3.89885	0.9413	14.6566	1.28828
46	2006.03.13-2006.03.19	7.76494	13.4381	3.64448	1.0353	14.2582	1.25813
47	2006.03.20-2006.03.26	7.81121	14.3718	3.59783	1.1773	14.2287	1.25812
48	2006.03.27-2006.04.02	7.87989	14.5806	3.93365	1.3663	14.087	1.32574
49	2006.04.03-2006.04.09	7.91376	14.6447	3.91952	1.1445	13.9491	1.32702
50	2006.04.10-2006.04.16	7.90308	14.9512	4.07225	1.4446	14.0153	1.28804
51	2006.04.17-2006.04.23	8.11566	14.9722	3.87529	1.449	13.9852	1.23218
52	2006.04.24-2006.04.30	8.15547	14.9582	3.99875	1.4625	13.9803	1.22439
53	2006.05.01-2006.05.07	8.11617	14.9728	3.86895	1.4763	13.9441	1.18815
Vidējie lielumi		7.65744	12.2127	4.36423	1.2768	16.9465	1.3779

Akciju nedēļas ienesīgums

Nedēļas numurs	Nedēļas ienesīgums					
	Eesti Telekom	Latvijas Gāze	Harju Elekter	Vilniaus Vingis	SAF tehnika	Pieno žvaigždēs
1	-0.00651	-0.02945	-1.62541	-0.00010	-0.01566	-0.01027
2	0.00381	0.00157	-0.09388	-0.00161	-0.01254	-0.00054
3	0.01452	-0.00017	-0.02690	0.00006	-0.02717	0.01843
4	-0.08076	0.00278	-0.02738	-0.00074	-0.01663	0.00408
5	-0.01279	-0.00100	0.02032	-0.00197	0.00000	0.01996
6	-0.01282	0.02583	-0.18262	-0.02839	-0.02337	0.00231
7	-0.00092	0.05967	0.10035	0.00025	0.00000	0.00175
8	-0.00024	0.00671	0.06754	-0.01861	-0.00248	-0.00142
9	-0.00359	0.00162	-0.02559	-0.05835	0.00869	-0.00512
10	-0.00497	0.01196	0.01892	0.00059	-0.00098	-0.00424
11	-0.00219	0.00431	0.02105	-0.01347	-0.00115	-0.00421
12	0.00825	-0.02825	0.00115	0.00541	0.00286	-0.01050
13	-0.00036	-0.02035	-0.00913	0.01412	0.00355	-0.02601
14	-0.00067	0.00979	0.01139	0.00440	0.00000	-0.00315
15	-0.00029	0.00124	-0.00442	0.00013	-0.01612	-0.00390
16	-0.00090	-0.00491	-0.00241	0.00150	0.01011	0.00183
17	0.00040	0.01267	-0.00139	0.00320	0.00054	-0.03123
18	0.00565	0.01766	-0.03241	0.04645	0.02072	0.00700
19	-0.00203	0.01168	-0.01198	0.04903	0.00148	0.00014
20	0.01304	0.00050	0.00593	0.00045	0.00992	-0.00283
21	0.01462	0.01653	-0.00247	0.00774	0.01374	-0.00960
22	0.02964	0.01434	0.03107	0.02311	0.01477	-0.01004
23	-0.02771	0.03614	-0.02063	-0.08912	-0.01784	-0.01085
24	-0.02194	0.00895	-0.01223	-0.07074	-0.00088	-0.00861
25	-0.01004	0.02755	-0.00040	0.00000	-0.14889	0.01259
26	0.02856	0.04687	-0.00159	-0.03419	-0.01399	-0.01483
27	0.00366	0.01828	-0.01038	0.00131	0.02182	0.01303
28	0.01301	-0.00067	-0.03081	-0.05585	0.00115	-0.01797
29	0.00476	-0.00019	-0.05957	-0.11362	0.00067	-0.00760
30	0.00015	-0.00503	0.01751	0.04999	0.00218	-0.02756
31	-0.00695	-0.04222	0.02144	0.02000	-0.01526	-0.00103
32	-0.01996	0.02815	0.01306	0.02046	0.00440	0.02224
33	-0.00466	0.02779	-0.01545	0.01131	-0.00106	-0.00622
34	0.00814	0.06777	0.00041	0.05779	0.00231	0.00296
35	0.01160	0.04564	0.03471	-0.06675	0.01989	0.13656

Nedēļas numurs	Nedēļas ienesīgums					
	Eesti Telekom	Latvijas Gāze	Harju Elekter	Vilniaus Vingis	SAF tehnika	Pieno žvaigždēs
36	0.01406	0.07379	0.02804	-0.01362	-0.00175	-0.13792
37	-0.00250	-0.00892	-0.03015	-0.00009	-0.01497	0.02568
38	-0.00549	-0.01062	-0.01036	-0.00156	-0.00029	-0.00532
39	0.00403	-0.09754	-0.02767	-0.03450	-0.01385	-0.03782
40	-0.00140	0.01578	-0.04104	-0.02121	-0.04034	-0.01152
41	0.00174	-0.01219	0.02261	-0.11178	-0.00917	-0.00965
42	0.00023	0.00524	-0.00968	-0.10591	0.00000	-0.01273
43	0.00172	-0.03252	0.00759	0.02451	-0.02972	0.00266
44	0.01249	0.00455	-0.01430	0.03028	-0.02764	0.00738
45	-0.01246	-0.01857	-0.06980	0.09079	-0.02794	-0.02396
46	0.00592	0.06497	-0.01297	0.12065	-0.00207	-0.00001
47	0.00872	0.01432	0.08537	0.13832	-0.01006	0.05101
48	0.00428	0.00437	-0.00360	-0.19378	-0.00989	0.00096
49	-0.00135	0.02050	0.03751	0.20772	0.00472	-0.03026
50	0.02619	0.00140	-0.05082	0.00309	-0.00215	-0.04534
51	0.00488	-0.00094	0.03087	0.00922	-0.00035	-0.00636
52	-0.00484	0.00098	-0.03355	0.00936	-0.00259	-0.03050
53						
Vidējais nedēļas ienesīgums	-0.00008	0.00766	-0.03700	-0.00163	-0.00699	-0.00459
Izlases standartnovirze (risks)	0.01625	0.02851	0.04141	0.06409	0.02447	0.03245

Kovariācijas koeficienti starp vērtspapīriem

	Eesti Telekom	Latvijas Gāze
Eesti Telekom	0.00025006412381169700	0.00004091116998360230
Latvijas Gāze	0.00004091116998360230	0.00079381175207780600
Harju Elekter	0.00029932371862721900	0.00134484121627173000
Vilniaus Vingis	0.00004301515286340110	0.00019581907099275800
SAF tehnika	0.00007472351605764720	0.00003682215392498860
Pieno žvaigždēs	-0.00004194017562349330	0.00004867354676285760

	Harju Elekter	Vilniaus Vingis
Eesti Telekom	0.00029932371862721900	0.00004301515286340110
Latvijas Gāze	0.00134484121627173000	0.00019581907099275800
Harju Elekter	0.05123928852951180000	0.00044470686697837700
Vilniaus Vingis	0.00044470686697837700	0.00387085410472910000
SAF tehnika	0.00043408735175889400	0.00002763723221469370
Pieno žvaigždēs	0.00033966027906931600	-0.00004891167667799530

	SAF tehnika	Pieno žvaigždēs
Eesti Telekom	0.00007472351605764720	-0.00004194017562349330
Latvijas Gāze	0.00003682215392498860	0.00004867354676285760
Harju Elekter	0.00043408735175889400	0.00033966027906931600
Vilniaus Vingis	0.00002763723221469370	-0.00004891167667799530
SAF tehnika	0.00056626517674801200	0.00000068338956191495
Pieno žvaigždēs	0.00000068338956191495	0.00099347717137761200

Trešā kvadrātiskās programmēšanas uzdevuma ievadāmo datu piemērs

Sistēmas vienādojumu skaits ir 1.

Transponētais vektors c ir

$$(0 \ 0 \ 0 \ 0 \ 0 \ 0).$$

Matrica A ir

$$(1 \ 0 \ 0 \ 0 \ 0 \ 0)$$

$$(0 \ 1 \ 0 \ 0 \ 0 \ 0)$$

$$(0 \ 0 \ 1 \ 0 \ 0 \ 0)$$

$$(0 \ 0 \ 0 \ 1 \ 0 \ 0)$$

$$(0 \ 0 \ 0 \ 0 \ 1 \ 0)$$

$$(0 \ 0 \ 0 \ 0 \ 0 \ 1)$$

$$(1 \ 1 \ 1 \ 1 \ 1 \ 1).$$

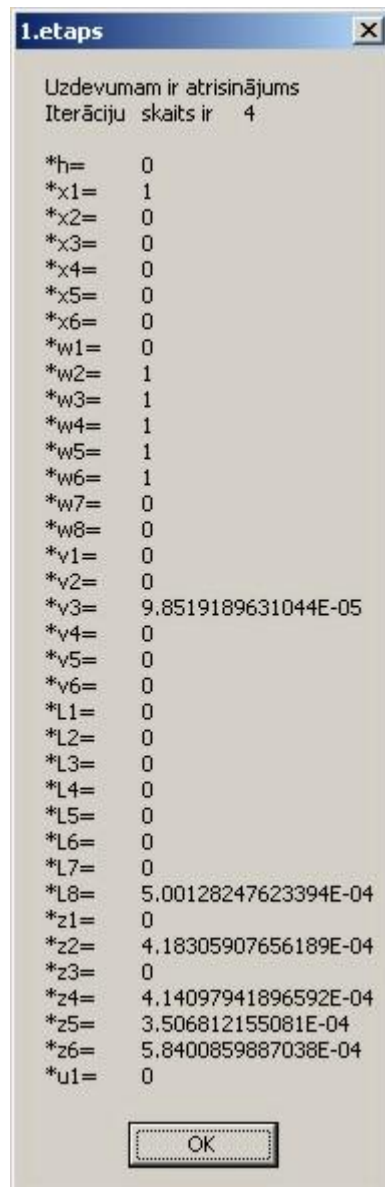
Matrica D ir

(0.00025006412381169700	0.00004091116998360230	0.00029932371862721900
0.00004301515286340110	0.00007472351605764720	-0.00004194017562349330)
(0.00004091116998360230	0.00079381175207780600	0.00134484121627173000
0.00019581907099275800	0.00003682215392498860	0.00004867354676285760)
(0.00029932371862721900	0.00134484121627173000	0.05123928852951180000
0.00044470686697837700	0.00043408735175889400	0.00033966027906931600)
(0.00004301515286340110	0.00019581907099275800	0.00044470686697837700
0.00387085410472910000	0.00002763723221469370	-0.00004891167667799530)
(0.00007472351605764720	0.00003682215392498860	0.00043408735175889400
0.00002763723221469370	0.00056626517674801200	0.00000068338956191495)
(-0.00004194017562349330	0.00004867354676285760	0.00033966027906931600
-0.00004891167667799530	0.00000068338956191495	0.00099347717137761200).

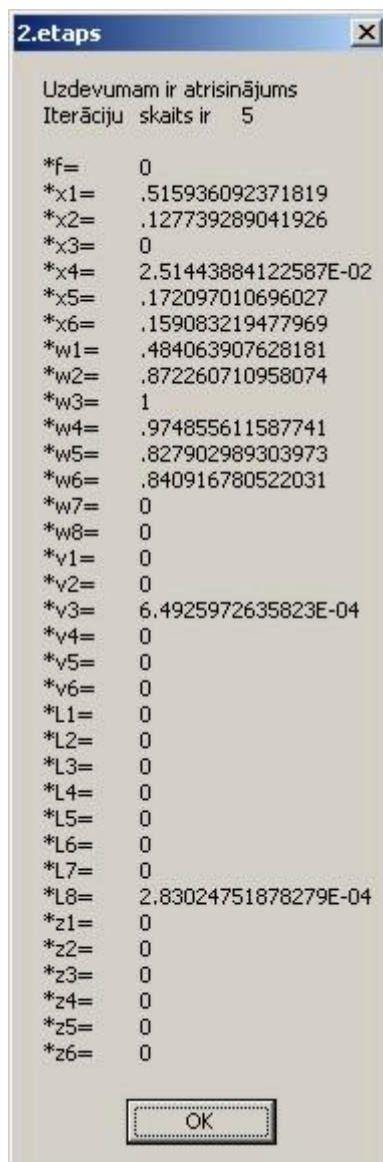
Transponētais vektors b ir

$$(1 \ 1 \ 1 \ 1 \ 1 \ 1).$$

Trešā kvadrātiskās programmēšanas uzdevuma 1.etapa ziņojums



Trešā kvadrātiskās programmēšanas uzdevuma 2.etapa ziņojums



Trešā kvadrātiskās programmēšanas uzdevuma 1.etapa rezultāti

bāze	h	x1	x2	x3	x4	x5
w1	0	1	0	0	0	0
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	0	0	0	0	0	0
w7	0	1	1	1	1	1
u1	0	1	1	1	1	1
z1	0	0.000500128	8.18223E-05	0.000598647	8.60303E-05	0.000149447
z2	0	8.18223E-05	0.001587624	0.002689682	0.000391638	7.36443E-05
z3	0	0.000598647	0.002689682	0.102478577	0.000889414	0.000868175
z4	0	8.60303E-05	0.000391638	0.000889414	0.007741708	5.52745E-05
z5	0	0.000149447	7.36443E-05	0.000868175	5.52745E-05	0.00113253
z6	0	-8.38804E-05	9.73471E-05	0.000679321	-9.78234E-05	1.36678E-06
h	1	-1	-1	-1	-1	-1

bāze	x6	w1	w2	w3	w4	w5
w1	0	1	0	0	0	0
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	1	0	0	0	0	0
w7	1	0	0	0	0	0
u1	1	0	0	0	0	0
z1	-8.38804E-05	0	0	0	0	0
z2	9.73471E-05	0	0	0	0	0
z3	0.000679321	0	0	0	0	0
z4	-9.78234E-05	0	0	0	0	0
z5	1.36678E-06	0	0	0	0	0
z6	0.001986954	0	0	0	0	0
h	-1	0	0	0	0	0

bāze	w6	w7	w8	v1	v2	v3
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	1	0	0	0	0	0
w7	0	1	0	0	0	0
u1	0	0	-1	0	0	0
z1	0	0	0	-1	0	0
z2	0	0	0	0	-1	0
z3	0	0	0	0	0	-1
z4	0	0	0	0	0	0
z5	0	0	0	0	0	0
z6	0	0	0	0	0	0
h	0	0	1	0	0	0

bāze	v4	v5	v6	L1	L2	L3
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
u1	0	0	0	0	0	0
z1	0	0	0	1	0	0
z2	0	0	0	0	1	0
z3	0	0	0	0	0	1
z4	-1	0	0	0	0	0
z5	0	-1	0	0	0	0
z6	0	0	-1	0	0	0
h	0	0	0	0	0	0

bāze	L4	L5	L6	L7	L8	z1
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
u1	0	0	0	0	0	0
z1	0	0	0	1	-1	1
z2	0	0	0	1	-1	0
z3	0	0	0	1	-1	0
z4	1	0	0	1	-1	0
z5	0	1	0	1	-1	0
z6	0	0	1	1	-1	0
h	0	0	0	0	0	0

bāze	z2	z3	z4	z5	z6	u1	b
w1	0	0	0	0	0	0	1
w2	0	0	0	0	0	0	1
w3	0	0	0	0	0	0	1
w4	0	0	0	0	0	0	1
w5	0	0	0	0	0	0	1
w6	0	0	0	0	0	0	1
w7	0	0	0	0	0	0	1
u1	0	0	0	0	0	1	1
z1	0	0	0	0	0	0	0
z2	1	0	0	0	0	0	0
z3	0	1	0	0	0	0	0
z4	0	0	1	0	0	0	0
z5	0	0	0	1	0	0	0
z6	0	0	0	0	1	0	0
h	0	0	0	0	0	0	-1

bāze	h	x1	x2	x3	x4	x5
w1	0	0	-1	-1	-1	-1
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	-0.002509341	-0.10178141	-0.000704864	-0.000620208
x1	0	1	1	1	1	1
z2	0	0	0.001924107	0.002509341	0.000723914	0.000342503
L8	0	0	0.000418306	-9.85192E-05	0.000414098	0.000350681
z4	0	0	0.000723914	0.000704864	0.008069776	0.000319925
z5	0	0	0.000342503	0.000620208	0.000319925	0.001333765
z6	0	0	0.000599533	0.000664682	0.000400155	0.000435928
h	1	0	0	0	0	0

bāze	x6	w1	w2	w3	w4	w5
w1	-1	1	0	0	0	0
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	1	0	0	0	0	0
w7	0	0	0	0	0	0
v3	-0.000664682	0	0	0	0	0
x1	1	0	0	0	0	0
z2	0.000599533	0	0	0	0	0
L8	0.000584009	0	0	0	0	0
z4	0.000400155	0	0	0	0	0
z5	0.000435928	0	0	0	0	0
z6	0.002654843	0	0	0	0	0
h	0	0	0	0	0	0

bāze	w6	w7	w8	v1	v2	v3
w1	0	0	1	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	1	0	0	0	0	0
w7	0	1	1	0	0	0
v3	0	0	-9.85192E-05	-1	0	1
x1	0	0	-1	0	0	0
z2	0	0	-0.000418306	1	-1	0
L8	0	0	-0.000500128	1	0	0
z4	0	0	-0.000414098	1	0	0
z5	0	0	-0.000350681	1	0	0
z6	0	0	-0.000584009	1	0	0
h	0	0	0	0	0	0

bāze	v4	v5	v6	L1	L2	L3
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	0	1	0	-1
x1	0	0	0	0	0	0
z2	0	0	0	-1	1	0
L8	0	0	0	-1	0	0
z4	-1	0	0	-1	0	0
z5	0	-1	0	-1	0	0
z6	0	0	-1	-1	0	0
h	0	0	0	0	0	0

bāze	L4	L5	L6	L7	L8	z1
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	0	0	0	1
x1	0	0	0	0	0	0
z2	0	0	0	0	0	-1
L8	0	0	0	-1	1	-1
z4	1	0	0	0	0	-1
z5	0	1	0	0	0	-1
z6	0	0	1	0	0	-1
h	0	0	0	0	0	0

bāze	z2	z3	z4	z5	z6	u1
w1	0	0	0	0	0	-1
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	-1
v3	0	-1	0	0	0	9.85192E-05
x1	0	0	0	0	0	1
z2	1	0	0	0	0	0.000418306
L8	0	0	0	0	0	0.000500128
z4	0	0	1	0	0	0.000414098
z5	0	0	0	1	0	0.000350681
z6	0	0	0	0	1	0.000584009
h	0	0	0	0	0	1

bāze	b
w1	0
w2	1
w3	1
w4	1
w5	1
w6	1
w7	0
v3	9.85192E-05
x1	1
z2	0.000418306
L8	0.000500128
z4	0.000414098
z5	0.000350681
z6	0.000584009
H	0

Uzdevumam ir atrisinājums

Iterāciju skaits ir 4

*h= 0	*v4= 0
*x1= 1	*v5= 0
*x2= 0	*v6= 0
*x3= 0	*L1= 0
*x4= 0	*L2= 0
*x5= 0	*L3= 0
*x6= 0	*L4= 0
*w1= 0	*L5= 0
*w2= 1	*L6= 0
*w3= 1	*L7= 0
*w4= 1	*L8= 0.000500128
*w5= 1	*z1= 0
*w6= 1	*z2= 0.000418306
*w7= 0	*z3= 0
*w8= 0	*z4= 0.000414098
*v1= 0	*z5= 0.000350681
*v2= 0	*z6= 0.000584009
*v3= 9.85192E-05	*u1= 0

Trešā kvadrātiskās programmēšanas uzdevuma 2.etapa rezultāti

bāze	f	x1	x2	x3	x4	x5
w1	0	0	-1	-1	-1	-1
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	-0.002509341	-0.10178141	-0.000704864	-0.000620208
x1	0	1	1	1	1	1
z2	0	0	0.001924107	0.002509341	0.000723914	0.000342503
L8	0	0	0.000418306	-9.85192E-05	0.000414098	0.000350681
z4	0	0	0.000723914	0.000704864	0.008069776	0.000319925
z5	0	0	0.000342503	0.000620208	0.000319925	0.001333765
z6	0	0	0.000599533	0.000664682	0.000400155	0.000435928
f	1	0	-0.003590057	-0.004499095	-0.00951377	-0.002432121

bāze	x6	w1	w2	w3	w4	w5
w1	-1	1	0	0	0	0
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	1	0	0	0	0	0
w7	0	0	0	0	0	0
v3	-0.000664682	0	0	0	0	0
x1	1	0	0	0	0	0
z2	0.000599533	0	0	0	0	0
L8	0.000584009	0	0	0	0	0
z4	0.000400155	0	0	0	0	0
z5	0.000435928	0	0	0	0	0
z6	0.002654843	0	0	0	0	0
f	-0.00409046	0	0	0	0	0

bāze	w6	w7	w8	v1	v2	v3
w1	0	0	1	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	1	0	0	0	0	0
w7	0	1	1	0	0	0
v3	0	0	-9.85192E-05	-1	0	1
x1	0	0	-1	0	0	0
z2	0	0	-0.000418306	1	-1	0
L8	0	0	-0.000500128	1	0	0
z4	0	0	-0.000414098	1	0	0
z5	0	0	-0.000350681	1	0	0
z6	0	0	-0.000584009	1	0	0
f	0	0	0.001767094	-4	1	0

bāze	v4	v5	v6	L1	L2	L3
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	0	1	0	-1
x1	0	0	0	0	0	0
z2	0	0	0	-1	1	0
L8	0	0	0	-1	0	0
z4	-1	0	0	-1	0	0
z5	0	-1	0	-1	0	0
z6	0	0	-1	-1	0	0
f	1	1	1	4	-1	0

bāze	L4	L5	L6	L7	L8	z1
w1	0	0	0	0	0	0
w2	0	0	0	0	0	0
w3	0	0	0	0	0	0
w4	0	0	0	0	0	0
w5	0	0	0	0	0	0
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	0	0	0	1
x1	0	0	0	0	0	0
z2	0	0	0	0	0	-1
L8	0	0	0	-1	1	-1
z4	1	0	0	0	0	-1
z5	0	1	0	0	0	-1
z6	0	0	1	0	0	-1
f	-1	-1	-1	0	0	5

bāze	z2	z3	z4	z5	z6	b
w1	0	0	0	0	0	0
w2	0	0	0	0	0	1
w3	0	0	0	0	0	1
w4	0	0	0	0	0	1
w5	0	0	0	0	0	1
w6	0	0	0	0	0	1
w7	0	0	0	0	0	0
v3	0	-1	0	0	0	9.85192E-05
x1	0	0	0	0	0	1
z2	1	0	0	0	0	0.000418306
L8	0	0	0	0	0	0.000500128
z4	0	0	1	0	0	0.000414098
z5	0	0	0	1	0	0.000350681
z6	0	0	0	0	1	0.000584009
f	0	1	0	0	0	-0.001767094

bāze	f	x1	x2	x3	x4	x5
w1	0	0	0	0.36813697	0	0
w2	0	0	0	-1.309053164	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0.03307958	0	0
w5	0	0	0	-0.158447579	0	0
w6	0	0	0	0.066284193	0	0
w7	0	0	0	0	0	0
v3	0	0	0	-0.098465654	0	0
x1	0	1	0	-0.36813697	0	0
x2	0	0	1	1.309053164	0	0
L8	0	0	0	-0.00064926	0	0
x4	0	0	0	-0.03307958	1	0
x5	0	0	0	0.158447579	0	1
x6	0	0	0	-0.066284193	0	0
f	1	0	0	0	0	0

bāze	x6	w1	w2	w3	w4	w5
w1	0	1	0	0	0	0
w2	0	0	1	0	0	0
w3	0	0	0	1	0	0
w4	0	0	0	0	1	0
w5	0	0	0	0	0	1
w6	0	0	0	0	0	0
w7	0	0	0	0	0	0
v3	0	0	0	0	0	0
x1	0	0	0	0	0	0
x2	0	0	0	0	0	0
L8	0	0	0	0	0	0
x4	0	0	0	0	0	0
x5	0	0	0	0	0	0
x6	1	0	0	0	0	0
f	0	0	0	0	0	0

bāze	w6	w7	w8	v1	v2	v3
w1	0	0	0.515936092	1174.698417	-331.18766	0
w2	0	0	0.127739289	-331.18766	588.7160657	0
w3	0	0	0	0	0	0
w4	0	0	0.025144388	-61.24331105	-43.23066254	0
w5	0	0	0.172097011	-585.8099	-105.1283976	0
w6	1	0	0.159083219	-196.4575458	-109.1693456	0
w7	0	1	1	0	0	0
v3	0	0	-0.00064926	0.36813697	-1.309053164	1
x1	0	0	-0.515936092	-1174.698417	331.18766	0
x2	0	0	-0.127739289	331.18766	-588.7160657	0
L8	0	0	-0.000283025	0.515936092	0.127739289	0
x4	0	0	-0.025144388	61.24331105	43.23066254	0
x5	0	0	-0.172097011	585.8099	105.1283976	0
x6	0	0	-0.159083219	196.4575458	109.1693456	0
f	0	0	0	0	0	0

bāze	v4	v5	v6	L1	L2	L3
w1	-61.24331105	-585.8099	-196.4575458	-1174.698417	331.18766	0
w2	-43.23066254	-105.1283976	-109.1693456	331.18766	-588.7160657	0
w3	0	0	0	0	0	0
w4	128.8301424	-17.58878485	-6.767383974	61.24331105	43.23066254	0
w5	-17.58878485	816.1476199	-107.6205375	585.8099	105.1283976	0
w6	-6.767383974	-107.6205375	420.0148128	196.4575458	109.1693456	0
w7	0	0	0	0	0	0
v3	0.03307958	-0.158447579	0.066284193	-0.36813697	1.309053164	-1
x1	61.24331105	585.8099	196.4575458	1174.698417	-331.18766	0
x2	43.23066254	105.1283976	109.1693456	-331.18766	588.7160657	0
L8	0.025144388	0.172097011	0.159083219	-0.515936092	-0.127739289	0
x4	-128.8301424	17.58878485	6.767383974	-61.24331105	-43.23066254	0
x5	17.58878485	-816.1476199	107.6205375	-585.8099	-105.1283976	0
x6	6.767383974	107.6205375	-420.0148128	-196.4575458	-109.1693456	0
f	0	0	0	0	0	0

bāze	L4	L5	L6	L7	L8	z1
w1	61.24331105	585.8099	196.4575458	0	0	-1174.698417
w2	43.23066254	105.1283976	109.1693456	0	0	331.18766
w3	0	0	0	0	0	0
w4	-128.8301424	17.58878485	6.767383974	0	0	61.24331105
w5	17.58878485	-816.1476199	107.6205375	0	0	585.8099
w6	6.767383974	107.6205375	-420.0148128	0	0	196.4575458
w7	0	0	0	0	0	0
v3	-0.03307958	0.158447579	-0.066284193	0	0	-0.36813697
x1	-61.24331105	-585.8099	-196.4575458	0	0	1174.698417
x2	-43.23066254	-105.1283976	-109.1693456	0	0	-331.18766
L8	-0.025144388	-0.172097011	-0.159083219	-1	1	-0.515936092
x4	128.8301424	-17.58878485	-6.767383974	0	0	-61.24331105
x5	-17.58878485	816.1476199	-107.6205375	0	0	-585.8099
x6	-6.767383974	-107.6205375	420.0148128	0	0	-196.4575458
f	0	0	0	0	0	1

bāze	z2	z3	z4	z5	z6	b
w1	331.18766	0	61.24331105	585.8099	196.4575458	0.484063908
w2	-588.7160657	0	43.23066254	105.1283976	109.1693456	0.872260711
w3	0	0	0	0	0	1
w4	43.23066254	0	-128.8301424	17.58878485	6.767383974	0.974855612
w5	105.1283976	0	17.58878485	-816.1476199	107.6205375	0.827902989
w6	109.1693456	0	6.767383974	107.6205375	-420.0148128	0.840916781
w7	0	0	0	0	0	0
v3	1.309053164	-1	-0.03307958	0.158447579	-0.066284193	0.00064926
x1	-331.18766	0	-61.24331105	-585.8099	-196.4575458	0.515936092
x2	588.7160657	0	-43.23066254	-105.1283976	-109.1693456	0.127739289
L8	-0.127739289	0	-0.025144388	-0.172097011	-0.159083219	0.000283025
x4	-43.23066254	0	128.8301424	-17.58878485	-6.767383974	0.025144388
x5	-105.1283976	0	-17.58878485	816.1476199	-107.6205375	0.172097011
x6	-109.1693456	0	-6.767383974	-107.6205375	420.0148128	0.159083219
f	1	1	1	1	1	0

Uzdevumam ir atrisinājums

Iterāciju skaits ir 5

*f= 0	
*x1= 0.515936092	*v4= 0
*x2= 0.127739289	*v5= 0
*x3= 0	*v6= 0
*x4= 0.025144388	*L1= 0
*x5= 0.172097011	*L2= 0
*x6= 0.159083219	*L3= 0
*w1= 0.484063908	*L4= 0
*w2= 0.872260711	*L5= 0
*w3= 1	*L6= 0
*w4= 0.974855612	*L7= 0
*w5= 0.827902989	*L8= 0.000283025
*w6= 0.840916781	*z1= 0
*w7= 0	*z2= 0
*w8= 0	*z3= 0
*v1= 0	*z4= 0
*v2= 0	*z5= 0
*v3= 0.00064926	*z6= 0

Lineārās programmēšanas uzdevuma risināšanas programma

Const konstante = 500

Const epsilon = 10^{-14}

Dim n, m, i, ii, tt, j, k, kk1, kk2, imax1, imax2, jmax1, jmax2, nosaukum, garums As Long

Dim FailaNr, h, p, negativie_b, nenegativie_b, pozitivie_aij, negativie_dj As Long

Dim kluda, kluda1, disks, texts As String

Dim fails1, fails2 As Boolean

Dim mainiga As Double

Dim A_matrix() As Double

Dim D_matrix() As Double

Dim b_vector() As Double

Dim c_vector() As Double

Dim etap(1 To 2) As Integer

Dim liela() As Double

Dim liela2() As Double

Dim nosaukums() As String

Dim nosaukums2() As String

Dim salidzina() As Long

Dim baze() As String

Dim baze2() As String

Dim txts() As String

Dim paligs() As Long

Dim bpalig() As Double

Dim palig() As Double

Dim abd

Sub Simpleksa_metode(imax, jmax, kk)

Dim elementsj, elementsi, starpiba, min_j As Double

Dim counts As Long

Dim nosacijums As Boolean

Dim reizinajums, dalijumm, dalijumm1, zimes As Double

```

Dim dalijums() As Double
Dim i0() As Long
Dim j0() As Long
ReDim dalijums(imax, jmax + 1)
ReDim salidzina(jmax)
kk = kk - 1
Do
    kk = kk + 1
    ReDim Preserve liela(imax, jmax, kk + 1)
    ReDim Preserve baze(imax, kk + 1)
    ReDim Preserve i0(kk + 1)
    ReDim Preserve j0(kk + 1)
    terxsts = ""
    min_j = 0
    j0(kk) = 1
    negativie_dj = 0
    For j = 1 To jmax
        elementsj = liela(imax, j, kk)
        'MsgBox "kk=" & kk2 & vbCrLf & "min_j=" & Str(min_j) & vbCrLf & elementsj
        If (elementsj < 0) And (j <> jmax) Then
            negativie_dj = negativie_dj + 1
            If negativie_dj = 1 Then
                min_j = elementsj
                j0(kk) = j
            End If
        End If
    Next j

    If (elementsj < min_j) And (j <> jmax) Then
        min_j = elementsj
        j0(kk) = j
    End If
Next j

trets = ""
If negativie_dj > 0 Then
    For i = 1 To imax

```

```

elements_i = liela(i, j0(kk), kk)
If elements_i > 0 Then
    For j = 1 To jmax
        dalijums(i, j) = liela(i, j, kk) / liela(i, j0(kk), kk)
    Next j
End If
For j = 1 To jmax
    trets = trets + Str(Format(dalijums(i, j), "###0.00")) + Chr(9)
Next j
trets = trets + Chr(13)
Next i

pozitivie_aj = 0
For i = 1 To imax
    elements_i = liela(i, j0(kk), kk)
    If (elements_i > 0) And (i < imax) Then
        pozitivie_aj = pozitivie_aj + 1
        If pozitivie_aj = 1 Then
            i0(kk) = i
        Else
            counts = jmax
            starpiba = dalijums(i, counts) - dalijums(i0(kk), counts)
            Do While (starpiba = 0)
                counts = counts - 1
                starpiba = dalijums(i, counts) - dalijums(i0(kk), counts)
            Loop
            If (starpiba < 0) Then i0(kk) = i
        End If
    End If
    baze(i, kk + 1) = baze(i, kk)
    'MsgBox "kk=" & kk2 & vbCrLf & "min_i=" & Str(dalijums(i0(kk),jmax) &
vbCrLf & elements_i
Next i
End If
nosacijums = (pozitivie_aj > 0) And (negativie_dj > 0)

```

```

'MsgBox "kk=" & kk & vbCrLf & "i0=" & i0(kk) & vbCrLf & "j0=" & j0(kk)
If nosacijums Then
    baze(i0(kk), kk + 1) = nosaukums(j0(kk) + 1)
    For j = 1 To jmax
        liela(i0(kk), j, kk + 1) = liela(i0(kk), j, kk) / liela(i0(kk), j0(kk), kk)
    Next j
    For i = 1 To imax
        For j = 1 To jmax
            reizinajums = -liela(i0(kk), j, kk + 1) * liela(i, j0(kk), kk)
            If (i <> i0(kk)) Then
                liela(i, j, kk + 1) = liela(i, j, kk) + reizinajums
                'MsgBox liela(i, j, kk) & " - " & liela(i0(kk), j, k + 1) & " * " & liela(i, j0(kk),
kk) & "=" & liela(i, j, kk + 1)
            End If
        Next j
    Next i
End If
Loop Until (pozitivie_aij = 0) Or (negativie_dj = 0) Or (kk = 100)
If nosacijums Then kk = kk - 1
End Sub

Sub Ievade()
    Dim a, b, c, d, ff, ai, aj, bi, bj, ci, cj, di, dj, indeks As Long
    Dim matrica_liela(1 To konstante, 1 To konstante, 1 To 4) As Double
    Dim indeksi(1 To konstante, 1 To 4) As Integer
    Dim burts1, burts2, str, str1, textins As String
    Dim nosacijums1, nosacijums2, nav_D_rindas, nav_A_rindas As Boolean
    Dim kordinate(1 To 4) As Integer
    Dim vect(1 To 4) As String
    Dim jauns As Double

    vect(1) = "Transponets vektors b ir"
    vect(2) = "Transponets vektors c ir"
    vect(3) = "Matrica A ir"
    vect(4) = "Matrica D ir"

```

```

ff = 1
e = 1
Open adrese For Input As #FailaNr
Do Until EOF(FailaNr)
    Line Input #FailaNr, str$
    str$ = LTrim(RTrim(str$))
    For i = 1 To Len(str$)
        If (Mid(str$, i, 1) = "(") And (i < Len(str$) - 1) Then str$ = Mid(str$, i + 1,
Len(str$) - i)
    Next i
    str1$ = str$
    If str$ <> "" Then
        If (Mid(str$, Len(str$), 1) = ".") Then str$ = Mid(str$, 1, Len(str$) - 1)
        If (Mid(str$, Len(str$), 1) = ")") Then str$ = Mid(str$, 1, Len(str$) - 1)
    End If
    str$ = LTrim(RTrim(str$))
    For i = 1 To 4
        If vect(i) = str$ Then
            kordinate(ff) = i
        End If
    Next i
    textins = str$ + " @"
    If Len(textins) > 2 Then
        a = 1
        b = 0
        c = 1
        d = 1
        If Not (vect(kordinate(ff)) = str$) Then
            Do Until Mid(textins, a, 1) = "@"
                burts1 = Mid(textins, a, 1)
                burts2 = Mid(textins, a + 1, 1)
                nosacijums1 = ((burts1 = " ") And (burts2 <> " ")) Or ((burts1 = " ") And (burts2
<> Chr(9)))
                nosacijums2 = (burts1 = Chr(9)) And (burts2 <> " ") Or ((burts1 = Chr(9)) And
(burts2 <> Chr(9)))
            End Do
        End If
    End If

```

```

b = b + 1
vars = RTrim(LTrim(Mid(textins, c, b)))
If (nosacijums1 Or nosacijums2) And (vars <> "") Then
    matrica_liela(e, d, kordinate(ff)) = vars
    indeksi(e, kordinate(ff)) = d
    Select Case kordinate(ff)
        Case 1
            bi = e
            If (bj < d) Then bj = d
        Case 2
            ci = e
            If (cj < d) Then cj = d
        Case 3
            ai = e
            If (aj < d) Then aj = d
        Case 4
            di = e
            If (dj < d) Then dj = d
    End Select
    d = d + 1
    c = a + 1
    b = 0
End If
a = a + 1
Loop
If (strs <> "") Then
    If (Mid(strs1, Len(strs1), 1) = ".") Then
        ff = ff + 1
        e = 1
    Else
        e = e + 1
    End If
End If
End If
End If
End If

```

```

Loop
Close #FailaNr
nav_A_rindas = False
nav_D_rindas = False
For i = 1 To ai
    If aj <> indeksi(i, 3) Then nav_A_rindas = True
Next i
For i = 1 To di
    If dj <> indeksi(i, 4) Then nav_D_rindas = True
Next i

```

'bi ir transponētā vektora b rindu skaits, kam ir jābūt vienādam ar 1.

'bj ir transponētā vektora b kolonu skaits.

'ci ir transponētā vektora c rindu skaits, kam ir jābūt vienādam ar 1.

'cj ir transponētā vektora c kolonu skaits.

'ai ir matricas A rindu skaits.

'aj ir matricas A kolonu skaits.

'di ir matricas D rindu skaits.

'dj ir matricas D kolonnu skaits.

'nav_A_rindas parāda, vai visās matricas A rindās ir vienāds elementu skaits

'nav_D_rindas parāda, vai visās matricas D rindās ir vienāds elementu skaits

```

If (bi <> 1) Then kluda = kluda + "Vektoram b transponētam ir jābūt 1 rindai!" &
vbCrLf

```

```

If (ci <> 1) Then kluda = kluda + "Vektoram c transponētam ir jābūt 1 rindai!" &
vbCrLf

```

```

If (aj <> cj) Then kluda = kluda + "Matricas A kolonnu skaits nav vienāds ar vektora c
elementu skaitu!" & vbCrLf

```

```

If (ai <> bj) Then kluda = kluda + "Matricas A rindu skaits nav vienāds ar vektora b
elementu skaitu!" & vbCrLf

```

```

If nav_A_rindas Then kluda = kluda + "Visās matricas A rindās elementu skaits nav
vienāds!"

```

```

If nav_D_rindas Then kluda = kluda + "Visās matricas D rindās elementu skaits nav
vienāds!"

```

```
If (di <> dj) Then kluda = kluda + "Rindu skaits matrica D nav vienāds ar kolonnu  
skaitu!" & vbCrLf
```

```
'If (dj <> cj) Then kluda = kluda + "Matricas D rindu skaits nav vienāds ar vektora c  
elementu skaitu!" & vbCrLf
```

```
If Len(kluda) = 0 Then
```

```
    n = cj
```

```
    m = bj
```

```
    ReDim A_matrix(m, n)
```

```
    ReDim D_matrix(n, n)
```

```
    ReDim b_vector(m)
```

```
    ReDim c_vector(n)
```

```
    ReDim paligs(m + 1)
```

```
    ReDim bpalig(m + 1)
```

```
    negativie_b = 0
```

```
    nenegativie_b = 0
```

```
    For j = 1 To bj
```

```
        b_vector(j) = matrica_liela(bi, j, 1)
```

```
        paligs(j) = 0
```

```
        If b_vector(j) < 0 Then
```

```
            negativie_b = negativie_b + 1
```

```
            paligs(j) = negativie_b
```

```
            bpalig(j) = -1
```

```
        End If
```

```
    Next j
```

```
    For j = 1 To bj
```

```
        If b_vector(j) >= 0 Then
```

```
            nenegativie_b = nenegativie_b + 1
```

```
            paligs(j) = negativie_b + nenegativie_b
```

```
            bpalig(j) = 1
```

```
        End If
```

```
    Next j
```

```
    paligs(m + 1) = m + 1
```

```
    For j = 1 To cj
```

```
        c_vector(j) = matrica_liela(ci, j, 2)
```

```

Next j
texts = ""
For i = 1 To ai
    For j = 1 To aj
        A_matrix(i, j) = matrica_liela(i, j, 3)
    Next j
Next i
For i = 1 To di
    For j = 1 To dj
        D_matrix(i, j) = matrica_liela(i, j, 4)
    Next j
Next i
End If
End Sub

```

```

Private Sub izvade(imaxx, jmaxx, kmaxx, etaps, pedejais As Double, etap_text As
String)

```

```

    Dim Response, parbaude, palignez As Boolean
    Dim a, garumm As Integer
    Dim xuy, cc, dd As Long
    Dim iteracijaa, datii As String
    Dim nosykum As String
    Dim nezinam() As Long
    Dim yyy() As Integer
    Dim yyy1() As Integer
    ReDim nezinam(jmaxx)
    ReDim yyy(jmaxx)
    ReDim yyy1(jmaxx)

```

```

    If etaps = 1 Then
        palignez = True
    Else
        palignez = palignezinamie.Value
        For j = 1 To m + n

```

```

        If j < n + 1 Then
            yyy(j + 1) = m + j
            nezinam(j + 1) = 0
        Else
            yyy(j + 1) = j - n
            nezinam(j + 1) = 1
        End If
        yyy1(j + 1) = yyy(j + 1)
    Next j
End If
For j = 1 To jmaxx - 1
    For i = 1 To imaxx
        If nosaukums(j + 1) = baze(i, kmaxx) Then salidzina(j) = i
    Next i
Next j
txts(1) = Chr(13)
texts = ""
For j = 1 To jmaxx + 1
    txts(1) = txts(1) + nosaukums(j) + Chr(9)
Next j
For k = 1 To kmaxx
    txts(k + 1) = txts(1) + Chr(13)
    For i = 1 To imaxx
        txts(k + 1) = txts(k + 1) + baze(i, k) + Chr(9)
        For j = 1 To jmaxx
            txts(k + 1) = txts(k + 1) + Str(liela(i, j, k))
            If j <> jmax Then txts(k + 1) = txts(k + 1) + Chr(9)
        Next j
        txts(k + 1) = txts(k + 1) + Chr(13)
    Next i
    If Not (visii.Value) And (k > 1) And (k < kmaxx) Then txts(k + 1) = ""
    texts = texts + txts(k + 1)
Next k

cc = -1

```

```

iteracijaa = "Iterāciju" + Chr(9) + "skaits ir" + Chr(9) + Str(kmaxx)
txts(kmaxx + 2) = ""

If (pozitivie_aij = 0) And (negativie_dj > 0) Then kluda1 = "Uzdevumam nav
atrisinājuma, jo mērķa funkcija ir neierobežota no augšas." + Chr(13)

If (etaps = 1) And (pedejais < 0) Then kluda1 = "Uzdevumam nav atrisinājuma, jo
sākotnējam uzdevumam plānu kopa ir tukša." + Chr(13)

parbaude = (negativie_dj = 0) And (Len(kluda1) = 0) And (etap(etaps) = etaps)
txts(kmaxx + 2) = txts(kmaxx + 2) + kluda1

If Not (parbaude) And (iteraciju_skaits.Value) Then txts(kmaxx + 2) = txts(kmaxx + 2)
+ iteracijaa + Chr(13) + Chr(13)

If parbaude Then
    txts(kmaxx + 2) = "Uzdevumam ir atrisinājums" + Chr(13)

    If iteraciju_skaits.Value Then txts(kmaxx + 2) = txts(kmaxx + 2) + iteracijaa +
Chr(13)

    txts(kmaxx + 2) = txts(kmaxx + 2) + Chr(13)

    For j = 1 To jmaxx - 1
        datii = ""
        nosykum = "*" + nosaukums(j + 1) + "=" + Chr(9)

        If (palignez) And (salidzina(j) > 0) Then datii = nosykum +
LTrim(Str(liela(salidzina(j), jmaxx, kmaxx)))

        If (palignez) And Not (salidzina(j) > 0) Then datii = nosykum + "0"
        garumm = Len(nosaukums(j)) - 1

        If Not (palignez) And (nezinam(j) = 0) Then
            If salidzina(j) > 0 Then datii = nosykum + LTrim(Str(liela(salidzina(j), jmaxx,
kmaxx)))

            If salidzina(j) = 0 Then datii = nosykum + "0"

            cc = cc + 1
        End If

        txts(kmaxx + 2) = txts(kmaxx + 2) + datii

        If Len(datii) > 0 Then txts(kmaxx + 2) = txts(kmaxx + 2) + Chr(13)

    Next j

    txts(kmaxx + 3) = ""

    If (etaps = 2) And (Dualais.Value) Then
        txts(kmaxx + 3) = "Duālajam uzdevumam ir atrisinājums" + Chr(13) + Chr(13)

        For i = 2 To m + n + 1

```

```

For j = 2 To m + n + 1
    If yyy(i) < yyy(j) Then
        tt = yyy(i)
        yyy(i) = yyy(j)
        yyy(j) = tt
    End If
Next j
Next i
txts(kmaxx + 3) = txts(kmaxx + 3) + "*g=" + Chr(9) + LTrim(Str(liela(imaxx,
jmaxx, kmaxx)))
txts(kmaxx + 3) = txts(kmaxx + 3) + Chr(13)
For j = 2 To jmaxx - 1
    tt = 0
    For i = 2 To m + n + 1
        If yyy(j) = yyy1(i) Then tt = i
    Next i
    nosykum = "*y" + LTrim(RTrim(Str(j - 1))) + "=" + Chr(9)
    If (palignez) Then txts(kmaxx + 3) = txts(kmaxx + 3) + nosykum +
LTrim(Str(liela(imaxx, tt, kmaxx)))
    If Not (palignez) And (nezinam(tt) = 1) Then txts(kmaxx + 3) = txts(kmaxx + 3)
+ nosykum + LTrim(Str(liela(imaxx, tt, kmaxx)))
    txts(kmaxx + 3) = txts(kmaxx + 3) + Chr(13)
Next j
End If
End If

If (etap(etaps) = etaps) Then
    texts = texts + Chr(13) + txts(kmaxx + 2) + Chr(13) + txts(kmaxx + 3)
    If (Len(kluda1) = 0) And (atrisinajuma.Value) Then
        Response = MsgBox(txts(kmaxx + 2), vbOKOnly, etap_text)
        If (Dualais.Value) And (etaps = 2) Then Response = MsgBox(txts(kmaxx +
3), vbOKOnly, "Duālais uzdevums")
    End If
Else
    texts = "Nevajag" + Chr(13) + "Uzdevumam"

```

```

End If
If Len(kluda1) > 0 Then Response = MsgBox(kluda1, vbOKOnly + vbCritical,
etap_text)
End Sub

```

```

Private Sub abi_etapi_Click()
    etap(1) = 1
    etap(2) = 2
    palignezinamie.Value = True
    Dualais.Value = True
End Sub

```

```

Private Sub Aprekinat_Click()
    negativie_b = 0
    tur = 0
    kluda = ""
    kluda1 = ""
    garums = Len(adrese)
    Do
        garums = garums - 1
    Loop Until (Mid(adrese, garums, 1) = "\")
    disks = Mid(adrese, 1, garums)
    FailaNr = FreeFile
    Set abd = CreateObject("Scripting.FileSystemObject")
    If Not (abd.FolderExists(disks)) Then GoTo beigaas
    If Not (abd.FileExists(adrese)) Then GoTo beigaas1
    Call Ievade
    Open disks + "1. etaps.txt" For Output As #1
        Write #1, "Nevajag" + Chr(13) + "Uzdevumam"
    Close #1

    Open disks + "2. etaps.txt" For Output As #1
        Write #1, "Nevajag" + Chr(13) + "Uzdevumam"
    Close #1

```

If (Len(kluda) > 0) Then Response = MsgBox(kluda, vbOKOnly + vbCritical, "Kļūdu ziņojums")

'Simpleksa metodes 1 etaps

If (Len(kluda) = 0) And (negativie_b > 0) Then

jmax = 1 + 1 + m + n + negativie_b 'Funkcijas kolonna + matricas A kolonnu skaits + jaunu mainigo nevienadību skaits + vektora b kolonna

imax = m + 1 'Matricas A rindu skaits + vektora -c rinda

kk = 1

tt = n + m + 1

'jmax - negativie_b - 1

'MsgBox "m=" & m & " n=" & n & vbCrLf & "i=" & imax & " j=" & jmax

ReDim liela(imax, jmax, kk)

ReDim nosaukums(jmax + 1)

ReDim nosaukums1(jmax - negativie_b + 1)

ReDim baze(imax, 1)

For i = 1 To imax

ii = paligs(i)

For j = 1 To jmax

If bpalig(i) < 0 Then

If (i < imax) And (j > 1) And (j < n + 2) Then

liela(ii, j, 1) = -A_matrix(i, j - 1)

'MsgBox i & " " & j & " " & " aa=" & -A_matrix(ii, j - 1)

End If

If (j = jmax) And (i < m + 1) Then liela(ii, j, 1) = -b_vector(i)

If (i < imax) And (j > n + 1) And (j < jmax) And (j = ii + n + 1) Then liela(ii, j, 1) = -1

If (i < imax) And (j > n + 1) And (j < jmax) And (j = ii + tt) Then liela(ii, j, 1) = 1

Else

If (i = imax) And (j = 1) Then liela(i, j, 1) = 1

If (i < imax) And (j > 1) And (j < n + 2) Then liela(ii, j, 1) = A_matrix(i, j - 1)

If (j = jmax) And (i < m + 1) Then liela(ii, j, 1) = b_vector(i)

If (i < imax) And (j > n + 1) And (j < jmax) And (j = ii + n + 1) Then liela(ii, j, 1) = 1

End If

```

Next j
baze(i, 1) = "x" + LTrim(RTrim(Str(n + i)))
Next i
liela(imax, 1, 1) = 1
texts = ""
baze(imax, 1) = "h"
nosaukums(1) = "bāze"
nosaukums(2) = "h"
nosaukums1(1) = nosaukums(1)
nosaukums1(2) = "f"
For j = 2 To jmax - 1
    If j + negativie_b < jmax Then
        nosaukums(j + 1) = "x" & LTrim(RTrim(Str(j - 1)))
        nosaukums1(j + 1) = "x" & LTrim(RTrim(Str(j - 1)))
    Else
        nosaukums(j + 1) = "u" & LTrim(RTrim(Str(j + 1 +
negativie_b - jmax)))
    End If
Next j
nosaukums(jmax + 1) = "b"
nosaukums1(jmax - negativie_b + 1) = "b"
For i = 1 To negativie_b
    baze(i, 1) = "u" & LTrim(RTrim(Str(i)))
Next i
For j = 2 To jmax
    liela(imax, j, 1) = 0
    ii = 0
    For i = 1 To imax
        If (nosaukums(j + 1) = baze(i, 1)) And (Mid(baze(i, 1), 1, 1) = "u") Then ii =
1
    Next i
    If ii = 0 Then
        For i = 1 To negativie_b
            liela(imax, j, 1) = liela(imax, j, 1) - liela(i, j, 1)
        Next i

```

```

        End If
    Next j

    Call Simpleksa_metode(imax, jmax, kk)

    ReDim txts(kk + 3)
    Call izvade(imax, jmax, kk, 1, liela(imax, jmax, kk), "1. etaps")

    Open disks + "1. etaps.txt" For Output As #1
    Write #1, txts
    Close #1
End If

If (Len(kluda) = 0) And (Len(kluda1) = 0) And (negativie_b > 0) Then
    jmax = jmax - negativie_b
    ReDim baze1(imax - 1)
    ReDim nosaukums(jmax + 1)
    ReDim liela1(imax - 1, jmax)
    ReDim palig(imax)
    ReDim paligs(jmax)
    For i = 1 To jmax + 1
        nosaukums(i) = nosaukums1(i)
    Next i
    For i = 1 To imax - 1
        baze1(i) = baze(i, kk)
        For j = 1 To jmax
            liela1(i, j) = liela(i, j, kk)
            If j = jmax Then liela1(i, j) = liela(i, j + negativie_b, kk)
        Next j
    Next i
    ReDim baze(imax, 1)
    ReDim liela(imax, jmax, 1)
    ii = 0
    For i = 1 To imax - 1
        baze(i, 1) = baze1(i)

```

```

For j = 1 To jmax
    liela(i, j, 1) = liela1(i, j)
    If baze1(i) = nosaukums(j + 1) Then
        paligs(j) = i
        If (j < n + 2) And (i > ii) Then
            ii = i
            palig(i) = c_vector(j - 1)
        End If
    End If
Next j
Next i
For j = 2 To jmax
    If paligs(j) = 0 Then
        For i = 1 To ii
            mainiga = palig(i) * liela(i, j, 1)
            liela(imax, j, 1) = liela(imax, j, 1) + mainiga
            'MsgBox mainiga & "=" & palig(i) & "*" & liela(i, j, 1) & vbCrLf & palig(i)
            & "*" & liela(i, j, 1) & "=" & mainiga & vbCrLf & "liela(" & imax & "," & j & ",1)=" &
            liela(imax, j, 1)
        Next i
        If j - 2 < n Then If c_vector(j - 1) <> 0 Then liela(imax, j, 1) = liela(imax, j, 1) -
        c_vector(j - 1)
        End If
    Next j
    baze(imax, 1) = "f"
    liela(imax, 1, 1) = 1
    kk = 1

    Call Simpleksa_metode(imax, jmax, kk)
    ReDim txts(kk + 3)
    Call izvade(imax, jmax, kk, 2, liela(imax, jmax, kk), "2. etaps")

    Open disks + "2. etaps.txt" For Output As #1
    Write #1, txts
    Close #1

```

End If

If (Len(kluda) = 0) And (negativie_b = 0) Then

jmax = 1 + n + m + 1 'Funkcijas kolonna + matricas A kolonnu skaits + jaunu
mainigo nevienadibu skaits + vektora b kolonna

imax = m + 1 'Matricas A rindu skaits + vektora -c rinda

kk = 1

'MsgBox "m=" & m & " n=" & n & vbCrLf & "i=" & imax & " j=" & jmax

ReDim liela(imax, jmax, kk)

ReDim nosaukums(jmax + 1)

ReDim baze(imax, 1)

For i = 1 To imax

For j = 1 To jmax

If (i = imax) And (j = 1) Then liela(i, j, 1) = 1

If (i < imax) And (j > 1) And (j < n + 2) Then liela(i, j, 1) = A_matrix(i, j - 1)

If (i = imax) And (j > 1) And (j < n + 2) Then liela(i, j, 1) = -(c_vector(j - 1))

If (j = jmax) And (i < m + 1) Then liela(i, j, 1) = b_vector(i)

If (i < imax) And (j > n + 1) And (j < jmax) And (j = i + n + 1) Then liela(i, j, 1)

= 1

Next j

baze(i, 1) = "x" + LTrim(RTrim(Str(n + i)))

Next i

baze(imax, 1) = "f"

nosaukums(1) = "bāze"

nosaukums(2) = "f"

For j = 2 To jmax - 1

nosaukums(j + 1) = "x" & LTrim(RTrim(Str(j - 1)))

Next j

nosaukums(jmax + 1) = "b"

Call Simpleksa_metode(imax, jmax, kk)

ReDim txts(kk + 3)

Call izvade(imax, jmax, kk, 2, liela(imax, jmax, kk), "2. etaps")

Open disks + "2. etaps.txt" For Output As #1

```

        Write #1, texts
        Close #1
    End If

    beigaas:
    Set abd = CreateObject("Scripting.FileSystemObject")
    If Not (abd.FolderExists(disks)) Then
        Response = MsgBox("Nav korekti ievadīta faila atrašanās adrese!", vbOKOnly
+ vbCritical + vbDefaultButton2, "Kļūdu ziņojums")
    Else
        beigaas1:
        If Not (abd.FileExists(adrese)) Then Response = MsgBox("Nav korekti ievadīts faila
nosaukums!", vbOKOnly + vbCritical + vbDefaultButton2, "Kļūdu ziņojums")
    End If
    End Sub

    Private Sub Atvert_Click()
        garums = Len(adrese)
        Do
            garums = garums - 1
        Loop Until (Mid(adrese, garums, 1) = "\")
        disks = Mid(adrese, 1, garums)
        Set abd = CreateObject("Scripting.FileSystemObject")
        If Not (abd.FolderExists(disks)) Then GoTo beigas_atv
        If Not (abd.FileExists(adrese)) Then GoTo beigas_atv1
        Workbooks.add

        Sheets("Sheet1").Name = "1. etaps"
        Sheets("Sheet2").Name = "2. etaps"
        Sheets("Sheet3").Name = "Dzest"

        Application.DisplayAlerts = False
        Worksheets("Dzest").Delete
        Application.DisplayAlerts = True
    
```

```

Sheets("1. etaps").Select
With ActiveSheet.QueryTables.add(Connection:="text;" + disks + "1. etaps.txt", _
    Destination:=Range("A1"))
    .Name = "2. etaps"
    .FieldNames = True
    .RowNumbers = False
    .FillAdjacentFormulas = False
    .PreserveFormatting = True
    .RefreshOnFileOpen = False
    .RefreshStyle = xlInsertDeleteCells
    .SavePassword = False
    .SaveData = True
    .AdjustColumnWidth = True
    .RefreshPeriod = 0
    .TextFilePromptOnRefresh = False
    .TextFilePlatform = xlWindows
    .TextFileStartRow = 1
    .TextFileParseType = xlDelimited
    .TextFileTextQualifier = xlTextQualifierDoubleQuote
    .TextFileConsecutiveDelimiter = False
    .TextFileTabDelimiter = True
    .TextFileSemicolonDelimiter = False
    .TextFileCommaDelimiter = False
    .TextFileSpaceDelimiter = False
    .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
    .TextFileDecimalSeparator = "."
    .Refresh BackgroundQuery:=False
End With

```

```

Columns("A:A").Select
With Selection
    .HorizontalAlignment = xlRight
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0

```

```

        .AddIndent = False
        .ShrinkToFit = False
        .MergeCells = False
    End With
    Columns("B:IV").Select
    ActiveWindow.SmallScroll Down:=11
    Selection.NumberFormat = "#"" ""???/???"

    With Selection
        .HorizontalAlignment = xlLeft
        .VerticalAlignment = xlCenter
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .MergeCells = False
    End With
    ActiveWindow.SmallScroll Down:=-11
    Cells.Find(What:="Uzdevumam", After:=ActiveCell, LookIn:=xlFormulas, _
        LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
        MatchCase:=False).Activate
    ActiveCell.Rows("1:1").EntireRow.Select
    Selection.Font.Bold = True

    With Selection
        .HorizontalAlignment = xlLeft
        .VerticalAlignment = xlCenter
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .MergeCells = True
    End With
    Cells.Select

```

With Selection.Font

.Name = "Times New Roman Baltic"

.Size = 11

.Strikethrough = False

.Superscript = False

.Subscript = False

.OutlineFont = False

.Shadow = False

.Underline = xlUnderlineStyleNone

.ColorIndex = xlAutomatic

End With

ActiveCell.Cells.EntireColumn.AutoFit

If ActiveCell.FormulaR1C1 = "Nevajag" Then

Application.DisplayAlerts = False

Worksheets("1. etaps").Delete

Application.DisplayAlerts = True

End If

Sheets("2. etaps").Select

With ActiveSheet.QueryTables.add(Connection:="text;" + disks + "2. etaps.txt", _

Destination:=Range("A1"))

.Name = "2. etaps"

.FieldNames = True

.RowNumbers = False

.FillAdjacentFormulas = False

.PreserveFormatting = True

.RefreshOnFileOpen = False

.RefreshStyle = xlInsertDeleteCells

.SavePassword = False

.SaveData = True

.AdjustColumnWidth = True

.RefreshPeriod = 0

.TextFilePromptOnRefresh = False

.TextFilePlatform = xlWindows

```

.TextFileStartRow = 1
.TextFileParseType = xlDelimited
.TextFileTextQualifier = xlTextQualifierDoubleQuote
.TextFileConsecutiveDelimiter = False
.TextFileTabDelimiter = True
.TextFileSemicolonDelimiter = False
.TextFileCommaDelimiter = False
.TextFileSpaceDelimiter = False
.TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
.TextFileDecimalSeparator = "."
.Refresh BackgroundQuery:=False
End With

```

```
Columns("A:A").Select
```

```
With Selection
```

```

.HorizontalAlignment = xlRight
.VerticalAlignment = xlCenter
.WrapText = False
.Orientation = 0
.AddIndent = False
.ShrinkToFit = False
.MergeCells = False

```

```
End With
```

```
Columns("B:IV").Select
```

```
ActiveWindow.SmallScroll Down:=11
```

```
Selection.NumberFormat = "#" "" "??"/???"
```

```
With Selection
```

```

.HorizontalAlignment = xlLeft
.VerticalAlignment = xlCenter
.WrapText = False
.Orientation = 0
.AddIndent = False
.IndentLevel = 0
.ShrinkToFit = False
.MergeCells = False

```

```

End With
ActiveWindow.SmallScroll Down:=-11
Cells.Find(What:="Uzdevumam", After:=ActiveCell, LookIn:=xlFormulas, _
    LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
    MatchCase:=False).Activate
ActiveCell.Rows("1:1").EntireRow.Select
Selection.Font.Bold = True
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .MergeCells = True
End With
Cells.Select
With Selection.Font
    .Name = "Times New Roman Baltic"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
ActiveCell.Cells.EntireColumn.AutoFit

Cells.FindNext(After:=ActiveCell).Activate
Cells.FindNext(After:=ActiveCell).Activate
ActiveCell.Rows("1:1").EntireRow.Select
Selection.Font.Bold = True

```

With Selection

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.MergeCells = True

End With

Cells.Select

With Selection.Font

.Name = "Times New Roman Baltic"

.Size = 11

.Strikethrough = False

.Superscript = False

.Subscript = False

.OutlineFont = False

.Shadow = False

.Underline = xlUnderlineStyleNone

.ColorIndex = xlAutomatic

End With

ActiveCell.Cells.EntireColumn.AutoFit

If ActiveCell.FormulaR1C1 = "Nevajag" Then

Application.DisplayAlerts = False

Worksheets("2. etaps").Delete

Application.DisplayAlerts = True

End If

beigas_atv:

Set abd = CreateObject("Scripting.FileSystemObject")

If Not (abd.FolderExists(disks)) Then

Response = MsgBox("Nav korekti ievadīta faila atrašanās adrese!", vbOKOnly
+ vbCritical + vbDefaultButton2, "Kļūdu ziņojums")

Else

```
beigas_atv1:  
    If Not (abd.FileExists(adrese)) Then Response = MsgBox("Nav korekti ievadīts faila  
nosaukums!", vbOKOnly + vbCritical + vbDefaultButton2, "Kļūdu ziņojums")  
End If  
End Sub
```

```
Private Sub Iziet_Click()  
Application.Quit  
End Sub
```

```
Private Sub dualais_Click()  
If (pirmais_etaps.Value) Then Dualais.Value = False  
End Sub
```

```
Private Sub otrais_etaps_Click()  
    etap(1) = 0  
    etap(2) = 2  
End Sub
```

```
Private Sub palignezinamie_Click()  
If (pirmais_etaps.Value) Then palignezinamie.Value = True  
End Sub
```

```
Private Sub pirmais_etaps_Click()  
    etap(1) = 1  
    etap(2) = 0  
    Dualais.Value = False  
    palignezinamie = False  
End Sub
```

```
Private Sub pirmais_pedejais_Click()  
    visii.Value = Not (pirmais_pedejais.Value)  
End Sub
```

```

Private Sub Radit_Click()
    If Radit.Caption = "Rādīt opcijas" Then
        opcijas.Visible = True
        Simplekss.Height = 282.75
        Radit.Caption = "Slēpt opcijas"
    Else
        opcijas.Visible = False
        Simplekss.Height = 77
        Radit.Caption = "Rādīt opcijas"
    End If
End Sub

```

```

Private Sub Radit_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    Radit_Click
End Sub

```

```

Private Sub visii_Click()
    pirmais_pedejais.Value = Not (visii.Value)
End Sub

```

```

Private Sub Userform_Initialize()
    etap(1) = 1
    etap(2) = 2
    Simplekss.Height = 77
    adrese.AddItem "G:\Diplomdarbs\program\LPU1.txt"
    adrese.AddItem "E:\Diplomdarbs\program\LPU1.txt"
    adrese.AddItem "C:\My
Documents\Macibas\LU\Diplomdarbs\Mans\program\LPU1.txt"
    adrese.AddItem "D:\Mani darbi\LU\Diplomdarbs\Mans\program\LPU1.txt"
End Sub

```

Kvadrātiskās programmēšanas uzdevuma risināšanas programma

Const konstante = 500

Const epsilon = 10^{-14}

Dim n, m, mm, i, ii, tt, j, k, vs, kk1, kk2, imax, jmax, imax1, imax2, jmax1, jmax2,
nosaukum, garums As Long

Dim FailaNr, h, p, negativie_b, pozitivie_aij, negativie_dj, vienadibu_skaits As Long

Dim kluda, disks, texts, textss, texts1, nosauku As String

Dim fails1, fails2, Response As Boolean

Dim mainiga As Double

Dim A_matrix() As Double

Dim D_matrix() As Double

Dim b_vector() As Double

Dim c_vector() As Double

Dim e_vector() As Double

Dim atlase_vector() As Double

Dim etap(1 To 2) As Integer

Dim liela() As Double

Dim liela1() As Double

Dim nosaukums() As String

Dim salidzina() As Long

Dim baze() As String

Dim baze1() As String

Dim txts() As String

Dim paligs() As Long

Dim palig() As Double

Dim abd

Sub Simpleksa_metode(imax, jmax, kk)

Dim elementsj, elementsi, starpiba, min_j As Double

Dim counts As Long

Dim nosacijums, xlmw_izmainas As Boolean

Dim reizinajums, dalijumm, dalijumm1, zimes As Double

```

Dim dalijums() As Double
Dim kollonnu_izmainas() As Integer
Dim i0() As Long
Dim j0() As Long
ReDim dalijums(imax, jmax + 1)
ReDim salidzina(jmax)

Dim nosauk1 As String
Dim nosaukcip, bazecip As Integer
kk = kk - 1
Do
    kk = kk + 1
    ReDim Preserve liela(imax, jmax, kk + 1)
    ReDim Preserve baze(imax, kk + 1)
    ReDim Preserve kollonnu_izmainas(jmax)
    For j = 1 To jmax
        kollonnu_izmainas(j) = 1
    Next j
parbaude:
    ReDim Preserve i0(kk + 1)
    ReDim Preserve j0(kk + 1)
    terxts = ""
    min_j = 0
    rindu_izmainas = 0
    j0(kk) = 1
    negativie_dj = 0
    For j = 1 To jmax
        elementsj = liela(imax, j, kk)
        'MsgBox "kk=" & kk2 & vbCrLf & "min_j=" & Str(min_j) & vbCrLf & elementsj
        If (elementsj < 0) And (j <> jmax) And (kollonnu_izmainas(j) = 1) Then
            negativie_dj = negativie_dj + 1
            If negativie_dj = 1 Then
                min_j = elementsj
                j0(kk) = j
            End If
        End If
    Next j

```

```

End If
If (elementsj < min_j) And (j <> jmax) And (kollonnu_izmainas(j) = 1) Then
    min_j = elementsj
    j0(kk) = j
End If
Next j
pozitivie_aij = 0
trets = ""
If negativie_dj > 0 Then
    For i = 1 To imax
        elementsi = liela(i, j0(kk), kk)
        If (elementsi > 0) Then
            For j = 1 To jmax
                dalijums(i, j) = liela(i, j, kk) / liela(i, j0(kk), kk)
            Next j
        End If
        For j = 1 To jmax
            trets = trets + Str(Format(dalijums(i, j), "###0.00")) + Chr(9)
        Next j
        trets = trets + Chr(13)
        baze(i, kk + 1) = baze(i, kk)
    Next i
    i = 0
    For i = 1 To imax
        nosauk1 = Mid(nosaukums(j0(kk) + 1), 1, 1)
        bazecip = Val(Mid(baze(i, kk), 2, Len(baze(i, kk)) - 1))
        nosaukcip = Val(Mid(nosaukums(j0(kk) + 1), 2, Len(nosaukums(j0(kk) + 1)) - 1))
        If (nosauk1 = "x") And (Mid(baze(i, kk), 1, 1) = "v") And (bazecip = nosaukcip)
Then rindu_izmainas = i
        If (nosauk1 = "v") And (Mid(baze(i, kk), 1, 1) = "x") And (bazecip = nosaukcip)
Then rindu_izmainas = i
        If (nosauk1 = "w") And (Mid(baze(i, kk), 1, 1) = "L") And (bazecip = nosaukcip)
Then rindu_izmainas = i
        If (nosauk1 = "L") And (Mid(baze(i, kk), 1, 1) = "w") And (bazecip = nosaukcip)
Then rindu_izmainas = i

```

```

elements_i = liela(i, j0(kk), kk)
If (elements_i > 0) And (i < imax) Then
    pozitivie_aj = pozitivie_aj + 1
    If pozitivie_aj = 1 Then
        i0(kk) = i
    Else
        counts = jmax
        starpiba = dalijums(i, counts) - dalijums(i0(kk), counts)
        Do While (starpiba = 0)
            counts = counts - 1
            starpiba = dalijums(i, counts) - dalijums(i0(kk), counts)
        Loop
        If (starpiba < 0) Or (Mid(baze(i, kk), 1, 1) = "u") Then i0(kk) = i
    End If
End If

'MsgBox "kk=" & kk2 & vbCrLf & "min_i=" & Str(dalijums(i0(kk),jmax) &
vbCrLf & elements_i
Next i
End If

If (rindu_izmainas <> i0(kk)) And (rindu_izmainas > 0) Then
    kollonnu_izmainas(j0(kk)) = 0
    GoTo parbaude
End If

nosacijums = (pozitivie_aj > 0) And (negativie_dj > 0)
'MsgBox "kk=" & kk & vbCrLf & "i0=" & i0(kk) & vbCrLf & "j0=" & j0(kk)
If nosacijums Then
    baze(i0(kk), kk + 1) = nosaukums(j0(kk) + 1)
    For j = 1 To jmax
        liela(i0(kk), j, kk + 1) = liela(i0(kk), j, kk) / liela(i0(kk), j0(kk), kk)
    Next j
    For i = 1 To imax
        For j = 1 To jmax
            reizinajums = -liela(i0(kk), j, kk + 1) * liela(i, j0(kk), kk)
            If (i <> i0(kk)) Then

```

```

        liela(i, j, kk + 1) = liela(i, j, kk) + reizinajums
        If Abs(liela(i, j, kk + 1)) < epsilon Then liela(i, j, kk + 1) = 0
        'MsgBox liela(i, j, kk) & " - " & liela(i0(kk), j, k + 1) & " * " & liela(i, j0(kk),
kk) & "=" & liela(i, j, kk + 1)
    End If
Next j
Next i
End If
Loop Until (pozitivie_aij = 0) Or (negativie_dj = 0) Or (kk = 100)
If nosacijums Then kk = kk - 1
End Sub

Sub Ievade()
Dim a, b, c, d, ff, ai, aj, bi, bj, ci, cj, di, dj, indeks As Long
Dim kd, ud, md, zd, ad1 As Long
Dim matrica_liela(1 To konstante, 1 To konstante, 1 To 4) As Double
Dim indeksi(1 To konstante, 1 To 4) As Integer
Dim burts1, burts2, str, str1, textins As String
Dim nosacijums1, nosacijums2, nav_D_rindas, nav_A_rindasm, pozitivi_definita As
Boolean
Dim kordinate(1 To 4) As Integer
Dim vect(1 To 5) As String
Dim jauns As Double
Dim D1_matrix()
Dim determ()
Dim determ2()

vect(1) = "Transponets vektors b ir"
vect(2) = "Transponets vektors c ir"
vect(3) = "Matrica A ir"
vect(4) = "Matrica D ir"
vect(5) = "Sistemas vienadojumu skaits ir"
ff = 1
e = 1
Open adrese For Input As #FailaNr

```

```

Do Until EOF(FailaNr)
    Line Input #FailaNr, str$
    str$ = LTrim(RTrim(str$))
    For i = 1 To Len(str$)
        If (Mid(str$, i, 1) = "(" And (i < Len(str$) - 1) Then str$ = Mid(str$, i + 1,
Len(str$) - i)
    Next i
    str$1 = str$
    If str$ <> "" Then
        If (Mid(str$, Len(str$), 1) = ".") Then str$ = Mid(str$, 1, Len(str$) - 1)
        If (Mid(str$, Len(str$), 1) = ")") Then str$ = Mid(str$, 1, Len(str$) - 1)
    End If
    str$ = LTrim(RTrim(str$))
    If vect(5) = Mid(str$, 1, 30) Then vienadibu_skait$ = Val(LTrim(LTrim(Mid(str$, 31,
Len(str$) - 30))))
    For i = 1 To 4
        If vect(i) = str$ Then
            kordinate(ff) = i
        End If
    Next i
    textins = str$ + " @"
    If (Len(textins) > 2) And (vect(5) <> Mid(str$, 1, 30)) Then
        a = 1
        b = 0
        c = 1
        d = 1
        If Not (vect(kordinate(ff)) = str$) Then
            Do Until Mid(textins, a, 1) = "@"
                burts1 = Mid(textins, a, 1)
                burts2 = Mid(textins, a + 1, 1)
                nosacijums1 = ((burts1 = " ") And (burts2 <> " ")) Or ((burts1 = " ") And (burts2
<> Chr(9)))
                nosacijums2 = (burts1 = Chr(9)) And (burts2 <> " ") Or ((burts1 = Chr(9)) And
(burts2 <> Chr(9)))
                b = b + 1
            Loop
        End If
    End If

```

```

vars = RTrim(LTrim(Mid(textins, c, b)))
If (nosacijums1 Or nosacijums2) And (vars <> "") Then
    matrica_liela(e, d, kordinate(ff)) = Val(vars)
    indeksi(e, kordinate(ff)) = d
    Select Case kordinate(ff)
        Case 1
            bi = e
            If (bj < d) Then bj = d
        Case 2
            ci = e
            If (cj < d) Then cj = d
        Case 3
            ai = e
            If (aj < d) Then aj = d
        Case 4
            di = e
            If (dj < d) Then dj = d
    End Select
    d = d + 1
    c = a + 1
    b = 0
End If
a = a + 1
Loop
If (strs <> "") Then
    If (Mid(strs1, Len(strs1), 1) = ".") Then
        ff = ff + 1
        e = 1
    Else
        e = e + 1
    End If
End If
End If
End If
Loop

```

Close #FailaNr

nav_A_rindas = False

nav_D_rindas = False

For i = 1 To ai

 If aj <> indeksi(i, 3) Then nav_A_rindas = True

Next i

For i = 1 To di

 If dj <> indeksi(i, 4) Then nav_D_rindas = True

Next i

'bi ir transponētā vektora b rindu skaits, kam ir jābūt vienādam ar 1.

'bj ir transponētā vektora b kolumnu skaits.

'ci ir transponētā vektora c rindu skaits, kam ir jābūt vienādam ar 1.

'cj ir transponētā vektora c kolumnu skaits.

'ai ir matricas A rindu skaits.

'aj ir matricas A kolumnu skaits.

'di ir matricas D rindu skaits.

'dj ir matricas D kolumnu skaits.

'nav_A_rindas parāda, vai visās matricas A rindās ir vienāds elementu skaits

'nav_D_rindas parāda, vai visās matricas D rindās ir vienāds elementu skaits

 If (bi <> 1) Then kluda = kluda + "Vektoram b transponētam ir jābūt 1 rindai!" &
vbCrLf

 If (ci <> 1) Then kluda = kluda + "Vektoram c transponētam ir jābūt 1 rindai!" &
vbCrLf

 If (aj <> cj) Then kluda = kluda + "Matricas A kolonnu skaits nav vienāds ar vektora c
elementu skaitu!" & vbCrLf

 If (ai <> bj) Then kluda = kluda + "Matricas A rindu skaits nav vienāds ar vektora b
elementu skaitu!" & vbCrLf

 If nav_A_rindas Then kluda = kluda + "Visās matricas A rindās elementu skaits nav
vienāds!"

 If nav_D_rindas Then kluda = kluda + "Visās matricas D rindās elementu skaits nav
vienāds!"

 If (di <> dj) Then kluda = kluda + "Rindu skaits matrica D nav vienāds ar kolonnu
skaitu!" & vbCrLf

```
If (aj <> dj) Then kluda = kluda + "Matricas A kolumnu skaits nav vienāds ar matricas D  
kolonnu skaitu!" & vbCrLf
```

```
If vienadibu_skaits <> CInt(vienadibu_skaits) Then kluda = kluda + "Vienādību skaits  
nav vesels skaitlis!" & vbCrLf
```

```
If vienadibu_skaits > bj Then kluda = kluda + "Norādītais vienādību skaits pārsniedz  
patieso vienādību skaitu!" & vbCrLf
```

```
If vienadibu_skaits < 0 Then kluda = kluda + "Norādītais vienādību skaits nedrīkst būt  
negatīvs!" & vbCrLf
```

```
vs = vienadibu_skaits
```

```
kludu_parbaude:
```

```
If Len(kluda) = 0 Then
```

```
    n = dj
```

```
    m = bj
```

```
    mm = m + vienadibu_skaits
```

```
    ReDim A_matrix(mm, n)
```

```
    ReDim D_matrix(n, n)
```

```
    ReDim D1_matrix(n, n)
```

```
    ReDim e_vector(n)
```

```
    ReDim b_vector(mm)
```

```
    ReDim c_vector(n)
```

```
    negativie_b = 0
```

```
    For j = 1 To bj
```

```
        b_vector(j) = matrica_liela(bi, j, 1)
```

```
        If b_vector(j) < 0 Then
```

```
            negativie_b = negativie_b + 1
```

```
        End If
```

```
        If (kk <> 0) And (j > m - vs - 1) Then
```

```
            b_vector(j) = -matrica_liela(bi, j, 1)
```

```
        End If
```

```
    Next j
```

```
    If (negativie_b > 0) Then
```

```
        kluda = kluda + "Vektors b nevar saturēt negatīvus skaitļus!" & vbCrLf
```

```
        GoTo kludu_parbaude
```

```
    End If
```

```

texts = ""
For i = 1 To m
  For j = 1 To n
    A_matrix(i, j) = matrica_liela(i, j, 3)
    If (vs <> 0) And (i + vs + 1 > m) Then A_matrix(i + vs, j) = -matrica_liela(i,
j, 3)

    If A_matrix(i, j) < 0 Then
      kluda = kluda + "Matricas A visi elementi nav nenegatīvi!"
      GoTo kludu_parbaude
    End If
  Next j
Next i
For j = 1 To n
  For i = 1 To n
    D_matrix(i, j) = matrica_liela(i, j, 4)
    D1_matrix(i, j) = matrica_liela(i, j, 4)
  Next i
  c_vector(j) = 0
  e_vector(j) = 1
Next j

ReDim determ(n)
ud = 0
det = 1
ad1 = 1
Do
  ud = ud + 1
  kd = ud
  While (D1_matrix(kd, ud) = 0) And (kd <= n)
    kd = kd + 1
    If kd >= n Then
      det = 0
      GoTo iz
    End If
  Wend

```

```

If kd <> ud Then
    For md = ud To n
        zd = D1_matrix(ud, md)
        D1_matrix(ud, md) = D1_matrix(kd, md)
        D1_matrix(kd, md) = z
        ad1 = -ad1
    Next md
End If
determ(ud) = ad1 * D1_matrix(ud, ud)
For j = n To ud Step -1
    D1_matrix(ud, j) = D1_matrix(ud, j) / D1_matrix(ud, ud)
Next j
If ud < n Then
    For i = ud + 1 To n
        For j = ud + 1 To n
            D1_matrix(i, j) = D1_matrix(i, j) - D1_matrix(i, ud) * D1_matrix(ud, j)
        Next j
    Next i
End If
Loop Until ud = n
iz:
pozitivi_definita = True
ReDim determ2(n)
For i = 1 To n
    If i = 1 Then
        determ2(i) = determ(i)
    Else
        determ2(i) = determ(i) * determ2(i - 1)
    End If
    If determ2(i) <= 0 Then pozitivi_definita = False
Next i
If Not (pozitivi_definita) Then kluda = kluda + "Matrica D nav pozitiv
pusdefinita!"
End If
End Sub

```

```

Private Sub izvade(imaxx, jmaxx, kmaxx, etaps, pedejaiss As Double, etap_text As
String)
    Dim Response, parbaude, palignez As Boolean
    Dim a, garumm As Integer
    Dim xuy, cc, dd As Long
    Dim iteracijaa, datii As String
    Dim nosykum As String
    Dim nezinam() As Long
    ReDim nezinam(jmaxx)

    If etaps = 2 Then
        For j = 1 To jmax
            If (Mid(nosaukums(j + 1), 1, 1) = "x") Then
                nezinam(j) = 0
            Else
                nezinam(j) = 1
            End If
        Next j
    End If

    For j = 1 To jmaxx - 1
        For i = 1 To imaxx
            If nosaukums(j + 1) = baze(i, kmaxx) Then salidzina(j) = i
        Next i
    Next j

    txts(1) = Chr(13)
    texts = ""

    For j = 1 To jmaxx + 1
        txts(1) = txts(1) + nosaukums(j) + Chr(9)
    Next j

    For k = 1 To kmaxx
        txts(k + 1) = txts(1) + Chr(13)
        For i = 1 To imaxx

```

```

txts(k + 1) = txts(k + 1) + baze(i, k) + Chr(9)
For j = 1 To jmaxx
    txts(k + 1) = txts(k + 1) + Str(liela(i, j, k))
    If j <> jmax Then txts(k + 1) = txts(k + 1) + Chr(9)
Next j
txts(k + 1) = txts(k + 1) + Chr(13)
Next i
If Not (visii.Value) And (k > 1) And (k < kmaxx) Then txts(k + 1) = ""
texts = texts + txts(k + 1)
Next k

cc = -1
palignez = palignezinamie.Value
iteracijaa = "Iterāciju" + Chr(9) + "skaits ir" + Chr(9) + Str(kmaxx)
txts(kmaxx + 2) = ""
If (pozitivie_aij = 0) And (negativie_dj > 0) Then kluda1 = "Uzdevumam nav
atrisinājuma, jo mērķa funkcija ir neierobežota no augšas." + Chr(13)
If (etaps = 1) And (pedejais < 0) Then kluda1 = "Uzdevumam nav atrisinājuma, jo
sākotnējam uzdevumam plānu kopa ir tukša." + Chr(13)
parbaude = (negativie_dj = 0) And (Len(kluda1) = 0) And (etap(etaps) = etaps)
txts(kmaxx + 2) = txts(kmaxx + 2) + kluda1
If Not (parbaude) And (iteraciju_skaits.Value) Then txts(kmaxx + 2) = txts(kmaxx + 2)
+ iteracijaa + Chr(13) + Chr(13)
If parbaude Then
    txts(kmaxx + 2) = "Uzdevumam ir atrisinājums" + Chr(13)
    If iteraciju_skaits.Value Then txts(kmaxx + 2) = txts(kmaxx + 2) + iteracijaa +
Chr(13)
    txts(kmaxx + 2) = txts(kmaxx + 2) + Chr(13)
    For j = 1 To jmaxx - 1
        datii = ""
        nosykum = "*" + nosaukums(j + 1) + "=" + Chr(9)
        If (palignez) And (salidzina(j) > 0) Then datii = nosykum +
LTrim(Str(liela(salidzina(j), jmaxx, kmaxx)))
        If (palignez) And Not (salidzina(j) > 0) Then datii = nosykum + "0"
        garumm = Len(nosaukums(j)) - 1

```

```

    If Not (palignez) And (nezinam(j) = 0) Then
        If salidzina(j) > 0 Then datii = nosykum + LTrim(Str(liela(salidzina(j), jmaxx,
kmaxx)))
        If salidzina(j) = 0 Then datii = nosykum + "0"
        cc = cc + 1
    End If
    txts(kmaxx + 2) = txts(kmaxx + 2) + datii
    If Len(datii) > 0 Then txts(kmaxx + 2) = txts(kmaxx + 2) + Chr(13)
Next j
End If

```

```

If (etap(etaps) = etaps) Then
    If (Len(kluda1) = 0) And (atrisinajuma.Value) Then Response =
MsgBox(txts(kmaxx + 2), vbOKOnly, etap_text)
    texts = texts + Chr(13) + txts(kmaxx + 2) + Chr(13)
Else
    texts = "Nevajag" + Chr(13) + "Uzdevumam"
End If
If Len(kluda1) > 0 Then Response = MsgBox(kluda1, vbOKOnly + vbCritical,
etap_text)
End Sub

```

```

Private Sub abi_etapi_Click()
    etap(1) = 1
    etap(2) = 2
    palignezinamie.Value = True
End Sub

```

```

Private Sub Aprekinat_Click()
    negativie_b = 0
    tur = 0
    kluda = ""
    garums = Len(adrese)
Do
    garums = garums - 1

```

```

Loop Until (Mid(adrese, garums, 1) = "\")
disks = Mid(adrese, 1, garums)
FailaNr = freefile
Set abd = CreateObject("Scripting.FileSystemObject")
If Not (abd.FolderExists(disks)) Then GoTo beigaas
If Not (abd.FileExists(adrese)) Then GoTo beigaas1
Call Ievade
Open disks + "1. etaps.txt" For Output As #1
    Write #1, "Nevajag" + Chr(13) + "Uzdevumam"
Close #1
Open disks + "2. etaps.txt" For Output As #1
    Write #1, "Nevajag" + Chr(13) + "Uzdevumam"
Close #1
'MsgBox "Okey " & vienadibu_skaits
If (Len(kluda) > 0) Then Response = MsgBox(kluda, vbOKOnly + vbCritical, "Kļūdu ziņojums")
If (Len(kluda) = 0) And (vs > 0) Then
    jmax = 1 + 3 * n + 2 * mm + 1 + vienadibu_skaits
    jmax1 = 1 + n + mm      'Funkcijas kolonna + matricas A kolonnu skaits + jaunu
mainigo nevienadibu skaits + vektora b kolonna
    imax = mm + n + 1
    imax1 = mm + 1         'Matricas A rindu skaits + vektora -c rinda
    kk = 1
    ReDim liela(imax, jmax, kk)
    ReDim nosaukums(jmax + 1)
    ReDim nosaukums1(jmax - vs + 1)
    ReDim baze(imax, 1)
    ReDim atlase_vector(imax)
    baze(imax, 1) = "h"
    nosaukums(1) = "bāze"
    nosaukums(2) = "h"
    nosaukums1(1) = nosaukums(1)
    nosaukums1(2) = "f"
    liela(imax, 1, 1) = 1
    For i = 1 To n + mm

```

```

For j = 2 To n + 1
    If (i < mm - vs + 1) Then liela(i, j, 1) = A_matrix(i, j - 1)
    If (i > mm - vs) And (i < mm + 1) Then liela(i, j, 1) = -A_matrix(i, j - 1)
    If (i > mm) Then liela(i, j, 1) = 2 * D_matrix(i - mm, j - 1)
    If (i = 1) Then nosaukums(j + 1) = "x" & LTrim(RTrim(Str(j - 1)))
    If (i = 1) Then nosaukums1(j + 1) = "x" & LTrim(RTrim(Str(j - 1)))
Next j
If (i < mm + 1) Then liela(i, jmax, 1) = b_vector(i)
Next i
For i = 1 To mm
    For j = 1 To mm - vs
        If (j = i) Then liela(i, j + n + 1, 1) = 1
    Next j
Next i

If vs > 0 Then
    For i = 1 To vs
        For j = 1 To vs
            If (j = i) Then liela(i + mm - vs, j + n + mm - vs + 1, 1) = -1
            If (j = i) Then liela(i + mm - vs, j + jmax - vs - 1, 1) = 1
        Next j
        liela(i + mm - vs, jmax, 1) = liela(i + m - vs, jmax, 1)
        liela(imax, i + n + mm - vs + 1, 1) = 1
    Next i
End If

For i = 1 To n
    For j = 1 To n
        If (j = i) Then liela(i + mm, j + mm + n + 1, 1) = -1
        If (j = i) Then liela(i + mm, j + 2 * (mm + n) + 1, 1) = e_vector(j)
    Next j
    atlase_vector(i + mm) = e_vector(i)
    liela(i + mm, jmax, 1) = c_vector(i)
Next i

```

```

For i = 1 To n
    For j = 1 To mm
        liela(i + mm, j + 2 * n + mm + 1, 1) = A_matrix(j, i)
    Next j
Next i

If vs > 0 Then
    For j = 2 To m
        For i = mm - vs + 1 To mm
            liela(imax, j, 1) = liela(imax, j, 1) + A_matrix(i, j - 1)
            If j = 2 Then liela(imax, jmax, 1) = liela(imax, jmax, 1) - liela(i, jmax, 1)
        Next i
    Next j
End If

For i = 1 To mm + n
    If i < m + 1 Then
        baze(i, 1) = "w" + LTrim(RTrim(Str(i)))
    ElseIf (i > m) And (i < m + vs + 1) Then
        baze(i, 1) = "u" + LTrim(RTrim(Str(i - m)))
    Else
        baze(i, 1) = "z" + LTrim(RTrim(Str(i - mm)))
    End If
Next i

For j = 1 To n
    liela(imax, j + 2 * (n + mm) + 1, 1) = 0
Next j

For j = 1 To mm
    nosaukums(j + n + 2) = "w" & LTrim(RTrim(Str(j)))
    nosaukums1(j + n + 2) = "w" & LTrim(RTrim(Str(j)))
Next j
For j = 1 To n

```

```

        nosaukums(j + n + mm + 2) = "v" & LTrim(RTrim(Str(j)))
        nosaukums1(j + n + mm + 2) = "v" & LTrim(RTrim(Str(j)))
    Next j
    For j = 1 To mm
        nosaukums(j + 2 * n + mm + 2) = "L" & LTrim(RTrim(Str(j)))
        nosaukums1(j + 2 * n + mm + 2) = "L" & LTrim(RTrim(Str(j)))
    Next j
    For j = 1 To n
        nosaukums(j + 2 * (n + mm + 1)) = "z" & LTrim(RTrim(Str(j)))
        nosaukums1(j + 2 * (n + mm + 1)) = "z" & LTrim(RTrim(Str(j)))

    Next j
    If vs > 0 Then
        For j = 1 To vs
            nosaukums(j + jmax - vs) = "u" & LTrim(RTrim(Str(j)))
        Next j
    End If
    nosaukums(jmax + 1) = "b"
    nosaukums1(jmax - vs + 1) = "b"
    nosauku = Chr(13)
    For j = 1 To jmax + 1
        nosauku = nosauku + nosaukums(j) + Chr(9)
    Next j
    textss = nosauku + Chr(13) + textss + Chr(13)

    Call Simpleksa_metode(imax, jmax, kk)
    ReDim txts(kk + 3)
    Call izvade(imax, jmax, kk, 1, liela(imax, jmax, kk), "1.etaps")

    Open disks + "1. etaps.txt" For Output As #1
    Write #1, txts
    Close #1
End If

If (Len(kluda) = 0) And (Len(kluda1) = 0) And (vs > 0) Then

```

```

jmax = jmax - vs
ReDim baze1(imax - 1)
ReDim nosaukums(jmax + 1)
ReDim liela1(imax - 1, jmax)
ReDim palig(imax)
ReDim paligs(jmax)
For i = 1 To jmax + 1
    nosaukums(i) = nosaukums1(i)
Next i
For i = 1 To imax - 1
    baze1(i) = baze(i, kk)
    For j = 1 To jmax
        liela1(i, j) = liela(i, j, kk)
        If j = jmax Then liela1(i, j) = liela(i, j + vs, kk)
    Next j
Next i
ReDim baze(imax, 1)
ReDim liela(imax, jmax, 1)
ii = 0
For i = 1 To imax - 1
    baze(i, 1) = baze1(i)
    For j = 1 To jmax
        liela(i, j, 1) = liela1(i, j)
        If (baze1(i) = nosaukums(j + 1)) And (Mid(baze1(i), 1, 1) = "z") Then
            paligs(j) = i
            If (i > ii) Then
                ii = i
                palig(i) = -1
            End If
        End If
    Next j
Next i
For j = 2 To jmax
    If paligs(j) = 0 Then
        For i = 1 To ii

```

```

    If palig(i) <> 0 Then
        mainiga = palig(i) * liela(i, j, 1)
        liela(imax, j, 1) = liela(imax, j, 1) + mainiga
        'MsgBox mainiga & "=" & palig(i) & "*" & liela(i, j, 1) & vbCrLf &
palig(i) & "*" & liela(i, j, 1) & "=" & mainiga & vbCrLf & "liela(" & imax & "," & j & ",1)="
& liela(imax, j, 1)
    End If
Next i
    If (j + n + 1 > jmax) And (j < jmax) Then liela(imax, j, 1) = liela(imax, j, 1) + 1
    End If
Next j
baze(imax, 1) = "f"
liela(imax, 1, 1) = 1
kk = 1

Call Simpleksa_metode(imax, jmax, kk)
ReDim txts(kk + 3)
Call izvade(imax, jmax, kk, 2, liela(imax, jmax, kk), "2.etaps")

Open disks + "2. etaps.txt" For Output As #1
Write #1, txts
Close #1
End If
beigaas:
Set abd = CreateObject("Scripting.FileSystemObject")
If Not (abd.FolderExists(disks)) Then
    Response = MsgBox("Nav korekti ievadīta faila atrašanās adrese!", vbOKOnly
+ vbCritical + vbDefaultButton2, "Kļūdu ziņojums")
Else
    beigaas1:
    If Not (abd.FileExists(adrese)) Then Response = MsgBox("Nav korekti ievadīts faila
nosaukums!", vbOKOnly + vbCritical + vbDefaultButton2, "Kļūdu ziņojums")
End If
End Sub

```

```

Private Sub Atvert_Click()
garums = Len(adrese)
Do
    garums = garums - 1
Loop Until (Mid(adrese, garums, 1) = "\")
disks = Mid(adrese, 1, garums)
Set abd = CreateObject("Scripting.FileSystemObject")
If Not (abd.FolderExists(disks)) Then GoTo beigas_atv
If Not (abd.FileExists(adrese)) Then GoTo beigas_atv1
Workbooks.Add

Sheets("Sheet1").Name = "1. etaps"
Sheets("Sheet2").Name = "2. etaps"
Sheets("Sheet3").Name = "Dzest"

Application.DisplayAlerts = False
Worksheets("Dzest").Delete
Application.DisplayAlerts = True

Sheets("1. etaps").Select
With ActiveSheet.QueryTables.Add(Connection:="text;" + disks + "1. etaps.txt", _
    Destination:=Range("A1"))
    .Name = "1. etaps"
    .FieldNames = True
    .RowNumbers = False
    .FillAdjacentFormulas = False
    .PreserveFormatting = True
    .RefreshOnFileOpen = False
    .RefreshStyle = xlInsertDeleteCells
    .SavePassword = False
    .SaveData = True
    .AdjustColumnWidth = True
    .RefreshPeriod = 0
    .TextFilePromptOnRefresh = False
    .TextFilePlatform = xlWindows

```

```

.TextFileStartRow = 1
.TextFileParseType = xlDelimited
.TextFileTextQualifier = xlTextQualifierDoubleQuote
.TextFileConsecutiveDelimiter = False
.TextFileTabDelimiter = True
.TextFileSemicolonDelimiter = False
.TextFileCommaDelimiter = False
.TextFileSpaceDelimiter = False
.TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
.TextFileDecimalSeparator = "."
.Refresh BackgroundQuery:=False
End With

```

```
Columns("A:A").Select
```

```
With Selection
```

```

.HorizontalAlignment = xlRight
.VerticalAlignment = xlCenter
.WrapText = False
.Orientation = 0
.AddIndent = False
.ShrinkToFit = False
.MergeCells = False

```

```
End With
```

```
Columns("B:IV").Select
```

```
ActiveWindow.SmallScroll Down:=11
```

```
'Selection.NumberFormat = "#" "" "??/??"
```

```
Selection.NumberFormat = "General"
```

```
With Selection
```

```

.HorizontalAlignment = xlLeft
.VerticalAlignment = xlCenter
.WrapText = False
.Orientation = 0
.AddIndent = False
.IndentLevel = 0
.ShrinkToFit = False

```

```

.MergeCells = False
End With
ActiveWindow.SmallScroll Down:=-11
Cells.Find(What:="Uzdevumam", After:=ActiveCell, LookIn:=xlFormulas, _
    LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
    MatchCase:=False).Activate
ActiveCell.Rows("1:1").EntireRow.Select
Selection.Font.Bold = True
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .MergeCells = True
End With
Cells.Select
With Selection.Font
    .Name = "Times New Roman Baltic"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
ActiveCell.Cells.EntireColumn.AutoFit

Cells.FindNext(After:=ActiveCell).Activate
Cells.FindNext(After:=ActiveCell).Activate
ActiveCell.Rows("1:1").EntireRow.Select

```

Selection.Font.Bold = True

With Selection

.HorizontalAlignment = xlLeft

.VerticalAlignment = xlCenter

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.MergeCells = True

End With

Cells.Select

With Selection.Font

.Name = "Times New Roman Baltic"

.Size = 11

.Strikethrough = False

.Superscript = False

.Subscript = False

.OutlineFont = False

.Shadow = False

.Underline = xlUnderlineStyleNone

.ColorIndex = xlAutomatic

End With

ActiveCell.Cells.EntireColumn.AutoFit

If ActiveCell.FormulaR1C1 = "Nevajag" Then

Application.DisplayAlerts = False

Worksheets("1. etaps").Delete

Application.DisplayAlerts = True

End If

Sheets("2. etaps").Select

With ActiveSheet.QueryTables.Add(Connection:="text;" + disks + "2. etaps.txt", _

Destination:=Range("A1"))

.Name = "2. etaps"

.FieldNames = True

```

.RowNumbers = False
.FillAdjacentFormulas = False
.PreserveFormatting = True
.RefreshOnFileOpen = False
.RefreshStyle = xlInsertDeleteCells
.SavePassword = False
.SaveData = True
.AdjustColumnWidth = True
.RefreshPeriod = 0
.TextFilePromptOnRefresh = False
.TextFilePlatform = xlWindows
.TextFileStartRow = 1
.TextFileParseType = xlDelimited
.TextFileTextQualifier = xlTextQualifierDoubleQuote
.TextFileConsecutiveDelimiter = False
.TextFileTabDelimiter = True
.TextFileSemicolonDelimiter = False
.TextFileCommaDelimiter = False
.TextFileSpaceDelimiter = False
.TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
.TextFileDecimalSeparator = "."
.Refresh BackgroundQuery:=False
End With

Columns("A:A").Select
With Selection
    .HorizontalAlignment = xlRight
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .ShrinkToFit = False
    .MergeCells = False
End With
Columns("B:IV").Select

```

```

ActiveWindow.SmallScroll Down:=11
Selection.NumberFormat = "General"
'Selection.NumberFormat = "#" "" "??/??"
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .MergeCells = False
End With
ActiveWindow.SmallScroll Down:=-11
Cells.Find(What:="Uzdevumam", After:=ActiveCell, LookIn:=xlFormulas, _
    LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, _
    MatchCase:=False).Activate
ActiveCell.Rows("1:1").EntireRow.Select
Selection.Font.Bold = True
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .MergeCells = True
End With
Cells.Select
With Selection.Font
    .Name = "Times New Roman Baltic"
    .Size = 11
    .Strikethrough = False
    .Superscript = False

```

```

.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.ColorIndex = xlAutomatic
End With
ActiveCell.Cells.EntireColumn.AutoFit

Cells.FindNext(After:=ActiveCell).Activate
Cells.FindNext(After:=ActiveCell).Activate
ActiveCell.Rows("1:1").EntireRow.Select
Selection.Font.Bold = True
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .MergeCells = True
End With
Cells.Select
With Selection.Font
    .Name = "Times New Roman Baltic"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
ActiveCell.Cells.EntireColumn.AutoFit

```

```

If ActiveCell.FormulaR1C1 = "Nevajag" Then
    Application.DisplayAlerts = False
    Worksheets("2. etaps").Delete
    Application.DisplayAlerts = True

End If

beigas_atv:
Set abd = CreateObject("Scripting.FileSystemObject")
If Not (abd.FolderExists(disks)) Then
    Response = MsgBox("Nav korekti ievadīta faila atrašanās adrese!", vbOKOnly
+ vbCritical + vbDefaultButton2, "Kļūdu ziņojums")
Else
    beigas_atv1:
    If Not (abd.FileExists(adrese)) Then Response = MsgBox("Nav korekti ievadīts faila
nosaukums!", vbOKOnly + vbCritical + vbDefaultButton2, "Kļūdu ziņojums")
End If
End Sub

```

```

Private Sub Iziet_Click()
Application.Quit
End Sub

```

```

Private Sub otrais_etaps_Click()
    etap(1) = 0
    etap(2) = 2
End Sub

```

```

Private Sub palignezinamie_Click()
If (pirmais_etaps.Value) Then palignezinamie.Value = True
End Sub

```

```

Private Sub pirmais_etaps_Click()
    etap(1) = 1
    etap(2) = 0

```

```
palignezinamie = False
```

```
End Sub
```

```
Private Sub pirmais_pedejais_Click()
```

```
visii.Value = Not (pirmais_pedejais.Value)
```

```
End Sub
```

```
Private Sub Radit_Click()
```

```
If Radit.Caption = "Rādīt opcijas" Then
```

```
opcijas.Visible = True
```

```
Simplekss.Height = 282.25
```

```
Radit.Caption = "Slēpt opcijas"
```

```
Else
```

```
opcijas.Visible = False
```

```
Simplekss.Height = 77
```

```
Radit.Caption = "Rādīt opcijas"
```

```
End If
```

```
End Sub
```

```
Private Sub Radit_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
```

```
Radit_Click
```

```
End Sub
```

```
Private Sub Userform_Initialize()
```

```
etap(1) = 1
```

```
etap(2) = 2
```

```
Simplekss.Height = 77
```

```
adrese.AddItem "G:\Diplomdarbs\program\LPU1.txt"
```

```
adrese.AddItem "E:\Diplomdarbs\program\LPU1.txt"
```

```
adrese.AddItem
```

"C:\My

Documents\Macibas\LU\Diplomdarbs\Mans\program\LPU1.txt"

```
adrese.AddItem "D:\Mani darbi\LU\Diplomdarbs\Mans\program\LPU1.txt"
```

```
End Sub
```

Diplomdarbs izpildīts
Latvijas Universitātes
Fizikas un matemātikas fakultātes
Matemātiskās analīzes katedrā

Fizikas un matemātikas fakultātes

Matemātika-statistika profesionālās studiju programmas students

(st. Apl. Nr. Mast010011) _____ Juris Breidaks

Darbs izpildīts: _____ Dr.math.Svetlana Asmuss

Diplomdarbs iesniegts katedrā 2006. gada _____ jūnijā.

Diplomdarbs aizstāvēts matemātika-statistika profesionālās studiju programmas
gala pārbaudījumu komisijas sēdē 2006.gada _____ jūnijā, novērtēts ar
atzīmi _____.

Protokols Nr. _____.

Pārbaudījumu komisijas sekretārs: _____.