TEODORAS MEDAISKIS

OPTIMALŪS SPRENDIMAI ĮMONĖJE

TIESINIO PROGRAMAVIMO MODELIAI IR METODAI

Mokymo priemonė

Vilnius

2011

TURINYS

Pratarmė

Matematinių metodų taikymas ekonominių procesų analizei šiandien įprastas dalykas. Matematiniai metodai pritaikomi makro ar mikroekonominiuose tyrimuose, jie padeda racionaliau organizuoti verslą, planuoti investicijas ir t.t.

Daugelį ekonominės ir verslo veiklos sričių praktiškai iškyla būtinybė surasti geriausią iš galimų įgyvendinti veiklos variantų. Tarp įvairių matematinių metodų, taikomų ekonominėje analizėje, šiam uždaviniui spręsti labiausiai pritaikyti optimizavimo metodai, nes būtent jų pagalba ieškoma geriausio iš sprendimų apibrėžtoje galimybių aibėje. Todėl optimizavimo uždavinių sprendimo teorija – svarbus įvairaus lygio ūkinės veiklos optimizavimo pagalbininkas.

Šioje mokymo priemonėje pristatomi ir nagrinėjami paprasčiausi, iš optimizavimo metodų praktiškai dažniausiai naudojami tiesinio programavimo modeliai ir metodai, skirti optimaliems sprendimams surasti bei pagrįsti. Siekiama parodyti potencialias jų taikymo galimybes, leidžiančias ne tik rasti geriausią sprendimą, bet ir atlikti išsamesnę ūkinės veiklos analizę. Pateikiami atitinkami pavyzdžiai. Kita vertus, nemaža dėmesio skiriama matematiniam tiesinio programavimo modelių ir metodų aparatui. Tuo siekiama, kad skaitytojas išmoktų šiuos metodus ne tik taikyti, bet ir valdyti – mokėtų juos modifikuoti ir adaptuoti kintančioms bei nestandartinėms ūkinės veiklos reikmėms.

Mokymo priemonė skirta pirmiausia bakalauro lygio ekonomikos studentams. Tačiau ji gali būti sėkmingai panaudota ir vadybos studijose bei visur, kur mokoma matematinių ūkinės veiklos optimizavimo bei valdymo metodų, operacijų tyrimo, verslo matematikos ir pan. Taip pat ši mokymo priemonė gali pasitarnauti kaip įvadinė medžiaga verslininkams, norintiems išbandyti ir pritaikyti šiuolaikinius ūkinės veiklos optimizavimo bei analizės metodus.

Mokymo priemonei skaityti užtenka paprasčiausių elementarios matematikos ir tiesinės algebros žinių. Tiesinio programavimo teorija pateikiama paprasčiausiu galimu būdu, siekiant kuo didesnio aiškumo ir prieinamumo. Daug dėmesio skiriama nagrinėjamų matematinių objektų bei teiginių ekonominiam interpretavimui bei įprasminimui.

Formulės, lentelės ir brėžiniai numeruojami kiekviename paragrafe iš naujo. Todėl nuoroda į, pavyzdžiui, (6) formulę yra nuoroda į atitinkamą einamojo paragrafo formulę, o (7.6) – nuoroda į §7 (6) formulę. Įrodymų ir pavyzdžių pabaiga žymima ženklu $\prec$.

§ 1. ĮVADAS. OPTIMALŪS SPRENDIMAI. UŽDAVINIAI, MODELIAI IR METODAI

1°. Matematinių modelių vieta ūkinės veiklos analizėje ir valdyme. Kalbėsime apie *organizacines sistemas*, kurias sudaro tikslingai veikiantys ūkinę veiklą vykdantys organizuoti žmonių kolektyvai, naudojantys įvairias technines priemones. Mus domins "mikro" lygio organizacinės sistemos - įmonės, jų padaliniai, kelių įmonių sąveika ir pan. Nagrinėsime formalius metodus, kurie padėtų nagrinėjamose organizacinėse sistemose priimti geriausius *sprendimus,* atitinkančius jų tikslą. Minėti metodai remiasi nagrinėjamų sistemų *matematinių modelių sudarymu ir sprendimu*. Kadangi tiesioginis eksperimentavimas su ūkinę veiklą vykdančiais subjektais praktiškai neįmanomas arba gali būti labai nuostolingas, išvadas apie jų savybes, reakcijas į galimus pokyčius, taip pat ir rekomendacijas dėl geriausių sprendimų gauname naudodami jų atvaizdus – matematinius modelius.

Suformuluotas matematinis modelis tampa atitinkamo *uždavinio* pagrindu: naudojantis modeliu, sprendžiama, ko ir kiek gaminti, pirkti, parduoti, kaip geriausia organizuoti transporto srautus ir t.t. Tokio uždavinio sprendinys ir yra geriausio ūkinio sprendimo pasiūlymas.

Matematinių modelių ir metodų taikymą ūkinės veiklos analizei bei sprendimų priėmimui paskatino revoliuciniai procesai gamyboje, nuolatinis jos sudėtingėjimas ir vis greitesnis atsinaujinimas. Tokiomis sąlygomis priimti teisingus sprendimus, remiantis vien tradiciniais metodais, darosi vis sunkiau, o klaidos ima vis brangiau kainuoti. XX a. viduryje atsirado ir specializuota taikomosios veiklos sritis – *operacijų tyrimas*. Šiandien ekonomiškai išvystytose šalyse sunku surasti tokią ūkio veiklą, kurioje operacijų tyrimo metodai nebūtų taikomi.

2°. Dažniausiai naudojamos matematinių modelių (uždavinių) klasės. Nors matematinio modeliavimo taikymo sfera labai plati, o uždaviniai labai įvairūs, vis dėlto, pasinaudojant operacijų tyrimo patirtimi, galima išskirti kelias dažniausiai taikomų uždavinių klases.

<u>Paskirstymo uždaviniai.</u> Tai bene didžiausia uždavinių klasė. Juos sprendžiant, siekiama racionaliai paskirstyti ribotus išteklius įvairiems darbams atlikti. Jei žinomos išteklių apimtys, sprendžiama, kokius darbus naudingiausia atlikti; jei žinomi darbai, sprendžiama, kaip geriausia juos aprūpinti ištekliais; gali būti žinomi ir ištekliai, ir darbai, o reikia surasti geriausią išteklių paskirstymą darbams atlikti. Paskirstymo uždaviniai gali būti naudojami optimaliam gamybos

planui sudaryti, optimaliam aprūpinimui ištekliais suplanuoti, transporto priemonių, įrengimų paskirstymui darbams optimizuoti ir pan.

Pakeitimo ir remonto uždaviniai. Šiuose uždaviniuose nagrinėjami įrengimai arba jų elementai, kurių funkcionavimas, laikui bėgant, keičiasi. Įrengimai dėvisi, mažėja jų patikimumas. Jei įrengimas remontuojamas (keičiamas) pernelyg retai, didėja aptarnavimo išlaidos ar net galimų avarijų tikimybė. Tačiau dažni remontai ar pakeitimai brangiai kainuoja. Todėl reikia nustatyti optimalų tokių įrengimo pakeitimo ar profilaktinio remonto laiką, kad laukiamos aptarnavimo ir remonto išlaidos būtų minimalios.

Atsargų valdymo uždaviniai (Inventory theory). Šiuose uždaviniuose nagrinėjamas optimalios atsargų apimties nustatymas įvairiuose ūkinės veiklos baruose. Jei atsargos per didelės, didėja jų saugojimo išlaidos, ilgam „užšaldomos" lėšos, kurias būtų galima panaudoti racionaliau. Tačiau mažos atsargos gali būti nepakankamos, be to, dažnas jų pildymas gali brangiai kainuoti. Į klausimus, kiek geriausia turėti atsargų, kada jas pildyti ir pan., atsakoma naudojant labai įvairius modelius: determinuotus, stochastinius, su viena ar keliomis atsargų rūšimis ir t.t.

Masinio aptarnavimo (eilių) teorijos uždaviniai (Queueing theory). Šiais uždaviniais modeliuojamos situacijos, kuriose tam tikri pastoviai atsirandantys darbai arba paraiškos turi būti atliekami (apdorojami) keliais įrengimais (aptarnavimo kanalais). Pavyzdžiu, pirkėjų srautą turi aptarnauti pardavėjai, užsakymus remontui – meistrai bei įrengimai ir pan. Kadangi užsakymai ateina netolygiai, susidaro užsakymų eilės arba kitu metu aptarnaujantys neturi darbo. Reikia taip organizuoti aptarnavimą, kad nuostoliai dėl eilių susidarymo ir dėl prastovų būtų kuo mažesni.

Sutvarkymo ir tvarkaraščių teorijos uždaviniai. Šie uždaviniai sudaromi ir sprendžiami, norint nustatyti optimalią tam tikrų darbų atlikimo tvarką. Paprasčiausias pavyzdys – turimas įrengimas vieną po kito atlieka keletą darbų, bet kiekvieną kartą, prieš pradedant naują darbą, jis turi būti naujai suderintas (paruoštas, pertvarkytas). Suderinimas tarp skirtingų darbų trunka nevienodai ar nevienodai kainuoja. Reikia sudaryti tokią darbų atlikimo seką, kad tie derinimai truktų kuo trumpiau ar mažiausiai kainuotų. Sudėtingesniu atveju turime keletą įrengimų ir darbų, kurie atliekami tais įrengimais. Kol atliekamas vienas darbas, kiti laukia. Laukia taip pat kai kurie įrengimai, nes darbus galima atlikti tik tam tikra seka (negalima iš pradžių dažyti, paskui obliuoti). Reikia sudaryti tokią darbų atlikimo seką, kad visi darbai būtų atlikti kuo greičiau (pigiau, kokybiškiau ir t.t.).

Maršruto sudarymo ir tinklinio planavimo uždaviniai artimi sutvarkymo uždaviniams. Šie uždaviniai sprendžiami, kai reikia tarp kelių punktų nutiesti trumpiausias ar pigiausias komunikacijų linijas, suplanuoti transporto srautus. Jei komunikacijų tinklas jau yra susiklostęs, gali būti sprendžiami trumpiausio kelio tarp bet kurių dviejų punktų suradimo uždaviniai, taip pat maksimalaus produktų srauto, kurį galima perduoti iš vieno punkto į kitą visomis komunikacijos priemonėmis, nustatymo uždaviniai. Sudėtingiausiuose iš šių uždavinių nustatomos naujų įmonių išdėstymo vietos, kartu planuojant jų komunikacijas, transporto srautus ir t.t.

Konfliktinių situacijų uždaviniai sprendžiami, kai reikia priimti sprendimą interesų nesutapimo arba prieštaraujančių tikslų sąlygomis. Gali nesutapti ar prieštarauti kelių bendradarbiaujančių įmonių tikslai, tos pačios įmonės padalinių interesai, pagaliau pati įmonė gali būti priversta rinktis tarp prieštaraujančių prioritetų. Nesutampantys interesai ne visada diametraliai priešingi: dažnai atsitinka, kad visiškas priešininko interesų nepaisymas pražūtingas abiems konflikto dalyviams. Todėl nagrinėjami kooperavimo, koalicijų sudarymo ir panašūs uždaviniai.

Ribos, skiriančios vieną uždavinių klasę nuo kitos, nėra griežtos. Ta pati problema, priklausomai nuo požiūrio, gali būti labai įvairiai formuluojama. Be to, dauguma realių uždavinių kompleksiški - jų sudėtinėms dalims spręsti tenka naudoti iš karto kelių skirtingų klasių uždavinius. Daugiau apie operacijų tyrimo uždavinių klases žr. [4, 10].

3°. Matematiniai metodai. Matyt, negalima būtų nurodyti tokios matematikos srities, kuri tiesiogiai ar netiesiogiai negalėtų būti panaudota, modeliuojant ūkines situacijas ir sprendžiant atitinkamus uždavinius. Netgi abstrakčios matematinės problemos dažnai sprendžiamos tam, kad padėtų pamatą taikomosioms matematinėms disciplinoms, o daugelis jų artimai siejasi su minėtais uždaviniais. Paminėsime tik kai kurias matematikos sritis, kurių metodai tiesiogiai naudojami matematinių modelių sudarymui.

Tiesinės algebros ir matematinės analizės metodai gali būti taikomi sprendžiant daugelį įvairių klasių uždavinių, nes jie įgalina operuoti vektoriais, matricomis, lygčių bei nelygybių sistemomis, funkcijomis, diferencijavimo bei integravimo galimybėmis ir t.t. Tikimybių teorijos metodai papildomai atsiranda visur, kur tenka susidurti su atsitiktiniais dydžiais, pvz., atsargų valdymo, įrengimų pakeitimo ir remonto, masinio aptarnavimo bei daugelyje kitų uždavinių. Matematinės statistikos metodai gali būti reikalingi bet kurios klasės uždavinių sprendimui, nes visur būtina įvertinti sudaromų modelių parametrus turimų duomenų pagrindu. Funkcinės analizės metodų gali prireikti tada, kai ieškome funkcijų, geriausiai išreiškiančių nagrinėjamų ūkinių procesų priklausomybes ir dinamiką.

Kombinatorikos metodai dažnai naudojami, sprendžiant sutvarkymo, paskirstymo, maršruto sudarymo ir kitokius uždavinius, kuriuose iš didžiulio variantų skaičiaus reikia išrinkti geriausią. Grafų teorija taip pat pritaikoma maršruto sudarymo, sutvarkymo, tinklų planavimo uždaviniams. Lošimų teorija tinka konfliktinių situacijų uždavinių formulavimui, analizei bei sprendimui.

Tačiau labiausiai su uždaviniais, skirtais geriausio sprendimo paieškai, yra susijęs matematinis programavimas. Pagrindinis matematinio programavimo uždavinys – surasti ekstremalią (minimalią ar maksimalią) tam tikros funkcijos reikšmę nurodytoje aibėje. Labai bendru pavidalu šį uždavinį galima užrašyti taip:

$$\text{(extr) } \varphi(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathcal{X}.$$

Čia $\mathcal{X}$ yra aibė, nusakanti modeliuojamos sistemos galimybių ar potencialių sprendimų visumą. Vadinsime ją *leistinų planų aibe,* o jos elementus *leistinais planais.* Funkcija $\varphi(\boldsymbol{x})$ kiekvienam leistinam planui priskiria skaitinę reikšmę, įvertinančią šį planą siekiamo tikslo požiūriu (tai gali būti šio plano duodamas pelnas, su jo įgyvendinimu susijos išlaidos ir pan.). Todėl ją vadinsime *tikslo funkcija* ir ieškosime jos didžiausios ar mažiausios reikšmės (didžiausio pelno, mažiausių sąnaudų ir pan.).

Matematinis programavimas pats dalinasi į kelias sritis, kurios turi savo specifiškus uždavinius ir jų sprendimo metodus.

Tiesinis programavimas yra populiariausia, geriausiai ištyrinėta ir dažniausiai taikoma matematinio programavimo sritis. Tiesinio programavimo uždaviniuose funkcija $\varphi(\boldsymbol{x})$ yra tiesinė daugelio nežinomųjų funkcija, o aibė $\mathcal{X}$ apibrėžiama tiesinių lygčių ir nelygybių pagalba. Jei uždaviniui formuluoti naudojamos ir netiesinės funkcijos, tai gaunami kur kas sudėtingesni netiesinio programavimo uždaviniai.

Jei tiesinio ar netiesinio programavimo uždavinį reikia išspręsti sveikais skaičiais, turime sveikaskaitinio programavimo uždavinį. Tokie uždaviniai iškyla, kai programuojame nedidelį skaičių nedalių objektų (pavyzdžiui, kiek statyti namų, kiek paleisti autobusų nurodytu maršrutu ir pan.), taip pat kai sprendžiame kombinatorinius uždavinius. Sveikaskaitinį programavimą apibendrina diskretusis programavimas, kai sprendinio ieškoma skaičioje aibėje $\mathcal{X}$.

Artima matematiniam programavimui yra vadinamoji optimalaus valdymo teorija. Jos uždaviniais modeliuojamos nagrinėjamos sistemos būsenų ir priimamų sprendimų (valdymų) sekos. Tokių daugiažingsnių uždavinių sprendimui, be kitų metodų, naudojamas ir dinaminis programavimas. Jo metodai dažnai taikomi atsargų valdymo, įrengimų pakeitimo ir remonto bei kituose uždaviniuose.

Labai dažnai modeliuojamoji sistema priklauso nuo iš anksto nežinomų aplinkybių, kurias numatyti galime tik su tam tikromis tikimybėmis (pvz., paklausos dydis, įrengimo avarijos galimybė). Tokiu atveju, formuluojant matematinio programavimo uždavinį, įtraukiami atsitiktiniai dydžiai, o pats uždavinys virsta stochastinio programavimo uždaviniu.

Įvairūs matematiniai metodai dažniausiai taikomi kompleksiškai. Be to, tas pat uždavinys gali būti sprendžiamas įvairiais metodais. Pavyzdžiui, įrengimų pakeitimo ir remonto uždavinius galima formuluoti kaip maršruto sudarymo uždavinius ir spręsti grafų teorijos metodais, bet galima juos spręsti ir dinaminio programavimo metodais, pasitelkiant reikalingus tikimybių teorijos faktus. Diskrečiojo programavimo uždaviniai gali buti sprendžiami dinaminio programavimo metodais ir atvirkščiai.

4°. Matematinio modelio sudarymo ir uždavinio sprendimo etapai. Norėdami sudaryti konkrečios ūkinės situacijos matematinį modelį ir jo pagrindu suformuluoti atitinkamą uždavinį, turime atlikti eilę veiksmų. Jų seka yra žinoma iš operacijų tyrimo patirties, kuria čia ir pasinaudosime.

Pradedame nuo problemos formulavimo. Šiame etape būtina kuo smulkiau išnagrinėti ekonominę situaciją, kurioje reikia priimti sprendimą. Tyrinėtojas privalo išsiaiškinti ne tik materialius nagrinėjamos situacijos aspektus (kas gaminama, kokiais įrengimais, kiek užimta žmonių ir pan.), bet ir finansinius, juridinius, psichologinius ir visus kitus įmanomus aspektus, nes kiekvienas iš jų gali nulemti sprendimą. Ne veltui rimtoms problemoms nagrinėti sudaromos skirtingų sričių specialistų grupės. Jų sudėtyje gali būti ekonomistai, matematikai, inžinieriai, teisininkai, psichologai ir t.t.

Išsiaiškinus situaciją ir su įmonės vadovais (arba užsakovais) patikslinus užduotį, reikia nuspręsti, į kokius dalinius uždavinius verta padalinti visą sprendžiamą problemą, ar tuos uždavinius galima spręsti atskirai, ar reikia dirbti iš karto su visu kompleksu, nes vienų uždavinių sprendimas gali priklausyti nuo kitų sprendimo rezultatų. Ar yra visi reikalingi duomenys, o jei nėra, tai ar bus galima juos surinkti, ir kiek šis procesas užtruks? Kaip detaliai nagrinėti problemą – kiek ir kokius faktorius laikyti esminiais, o kokius antraeiliais? Kiek ir kokių reikia turėti valdymo kintamųjų (t.y. kintamųjų, atspindinčių sprendimus), o kiek ir kokių būsenos kintamųjų (t.y. tokių, kurie apibūdintų esamą situaciją, priimtų sprendimų pasekmes)? Kiek ir kokių reikia technologinių parametrų, nustatančių ryšius tarp kintamųjų, ir kaip tuos parametrus įvertinti? Atsakant į visus šiuos klausimus, būtina išvengti ir pernelyg didelio nagrinėjimo detalumo, kuris apsunkintų tyrimą, ir pernelyg grubaus supaprastinimo, dėl kurio galima prasilenkti su konkrečia situacija.

Problemos formulavimo etape labai svarbu gerai išnagrinėti ir suprasti atliekamo tyrimo, taigi ir projektuojamo sprendimo *tikslą.* Modelio parinkimas priklausys nuo to, ar nagrinėjamoje sistemoje (ūkinėje situacijoje) keliamas vienas, ar iš karto keli tikslai. Gali būti, kad sprendimą reikia priimti priešingų interesų ir tikslų konflikto sąlygomis. Labai sudėtinga tikslų analizės dalis – jų konkretinimas, t.y. toks formulavimas, kuris leistų palyginti bet kurias galimas realizuoti alternatyvas ir nuspręsti, kuri iš jų nagrinėjamo tikslo požiūriu geresnė. Konkretinti tikslus tyrėjai privalo kartu su įmonių vadovais (tyrimo užsakovais), nes neretai pradiniame etape jie patys nebūna išsiaiškinę, ko iš tikrųjų siekia.

Išsprendus šias ir dar daugelį kitų čia nepaminėtų pirmojo etapo problemų, galima eiti prie antrojo etapo – modelio sudarymo. Bendrų rekomendacijų modelio sudarymui nėra – šis procesas nestandartinis, kiekvieną kart galintis reikalauti vis kitokio priėjimo, išradingumo ir kūrybingumo. Jo negalima vykdyti pagal šabloną. Vis dėlto kai kurių bendrų momentų yra ir šiame etape.

Jei problema ankstesniame etape gerai suformuluota, tai modelio sudarymui pasiruošta: išskirti esminiai modeliuojamos situacijos faktoriai, apsispręsta dėl įvairių tipų kintamųjų skaičiaus ir jų ryšių formalizavimo galimybių, paruošti formaliam užrašymui sistemos tikslai, išaiškinta, ar yra žinoma uždavinių klasė, kuriai būtų galima priskirti nagrinėjamą situaciją.

Pradedant modelio sudarymą, visos minėtos medžiagos pagrindu reikia apsispręsti dėl jo tipo ir nežinomųjų. Modeliai gali būti įvairių tipų: *statiniai* tinka vieną kart, o *dinaminiai* – daug kartų priimamam sprendimui modeliuoti; *determinuoti* pretenduoja į tikslią siejamų objektų išraišką, o *stochastiniai* atsižvelgia į atsitiktinių poveikių galimybes; *optimizaciniai* skirti geriausio iš galimų sprendimo suradimui, *imitaciniai* leidžia tyrėjui nagrinėti įvairius variantus ir pačiam rinktis geriausią, atsižvelgiant į modelyje, gal būt, neatspindėtus faktorius.

Modelio nežinomųjų visuma priklauso nuo problemos formulavimo etape išskirtų esminių faktorių. Belieka juos galutinai formalizuoti, t.y. išreikšti kiekybiškai, pažymėti raidėmis ir susieti matematinėmis priklausomybėmis. Nežinomieji paprastai atspindi ieškomą sprendimą. Jei norime nuspręsti, kiek gaminsime kurių nors produktų, tai nežinomieji reikš gamybos apimtis; jei sprendžiame, kiek pirkti – jie rodys pirkimo apimtis, jei norime racionaliai nustatyti kainas, nežinomieji rodys kainų dydžius ir t.t.

Matematinį modelį sudaro *kintamieji* ir jų *priklausomybės*. Kintamieji paprasčiausiai išreiškia mus dominančius modelyje figūruojančius kintančius dydžius: produkcijos apimtis, pelną, kainą ir t.t. Jie gali būti *egzogeniniai* ir *endogeniniai*. Pirmųjų reikšmės nustatomos už modelio ribų, antrųjų – randamos iš paties modelio.

Priklausomybės susieja kintamuosius *konstantų* ir *parametrų* pagalba. Pačios priklausomybės – dažniausiai lygtys arba nelygybės (sudėtingesniu atveju - priklausomybę aibėms ir pan.) Konstantos – dažniausiai skaitiniai dydžiai. Pavyzdžiui., priklausomybėje $y = 3x_1 + 5x_2$, dydžiai 3 ir 5 yra konstantos. Dažnai konstantų tiksli reikšmė nežinoma (net ir nebūtina žinoti, jei pakanka kokybinių, o ne kiekybinių rezultatų). Tada jos žymimos raidėmis ir vadinamos *parametrinėmis konstantomis* arba tiesiog *parametrais*, pvz., lygtyje $y = \propto x_1 + \beta x_2$, dydžiai α ir β – parametrai.

Priklausomybės gali būti kelių rūšių. Vienos iš jų – *apibrėžimo priklausomybės (definitional)* išreiškia kintamųjų ryšius, išplaukiančius iš jų apibrėžimų, pvz., P = R-C (pelnas = pajamos minus sąnaudos). Kitos – *elgesio priklausomybės (behavioral)* išreiškia iš pačios modeliuojamos situacijos išplaukiančius ekonominius ryšius, pvz., $y = 3x_1 + 5x_2$ (y yra elektros energijos kiekis, reikalingas x_1 vienetų pirmojo produkto ir x_2 vienetų antrojo produkto kiekiams pagaminti). Trečios priklausomybės gali būti *normatyvinės*, pvz., D=S (paklausa privalo būti lygi pasiūlai) arba $x_1 \geq x_2$ (pirmojo produkto reikalaujama gaminti ne mažiau, negu antrojo). Jos išreiškia tam tikrus normatyvinius reikalavimus privalomus tenkinti modelio kintamiesiems.

Kaip jau minėjome, problemos formulavimo etape reali ekonominė situacija neišvengiamai paprastinama – kai kurie faktoriai laikomi neesminiais, kitų poveikis įvertinamas labiau tiesmukiškai, negu yra iš tikrųjų. Antrąkart paprastinama, sudarant modelį – kai kurie kokybiniai poveikiai verčiami kiekybiniais, netiesiniai ryšiai – tiesiniais, atmetami atsitiktinumo elementai ir t.t. Todėl sudaromas modelis remiasi prielaidomis, kurios tyrinėtojui turi būti aiškiai žinomos ir modelio sudarymo etape įsidėmėtos. Tik galiojant šioms prielaidoms, modelio pagrindu gauti sprendimai gali būti laikomi pagrįstais.

Sudarius modelį, pereinama prie jo <u>sprendimo etapo</u>. Šiame etape pirmiausia reikia išsiaiškinti gauto uždavinio sprendimo metodus. Paprastai sudarytas modelis ir jį atitinkantis uždavinys priklauso kuriai nors žinomai uždavinių klasei, vadinasi, žinomi ir jo sprendimo metodai. Tačiau gali atsitikti, kad jo formulavime yra kai kurių nestandartinių elementų. Tokiais atvejais tenka ieškoti būdų, kaip juos „standartizuoti“, t.y. pritaikyti prie jau žinomų sprendimo schemų arba dėl jų modifikuoti pačius žinomus sprendimo metodus.[1]

Jei modelis priklauso žinomai uždavinių klasei, tai paprastai prieinamos ir programinės jo

[1] Kai kurie tokių veiksmų pavyzdžiai parodyti §§ 10, 11

sprendimo priemonės, kurias belieka panaudoti. Pavyzdžiui, daugelį nedidelių tiesinio (ir ne tik tiesinio) programavimo uždavinių galima spręsti netgi paprastomis Microsoft Excel priemonėmis (žr.§ 13) , o didesniems ir sudėtingesniems panaudoti SAS/OR paketą [11,12].

Modelio sprendimo etape ar iki jo turi būti surinkti visi reikalingi duomenys, įvertintos kintamųjų priklausomybės (jei reikia, pasitelkiant įvairius duomenų analizės metodus). Duomenų rinkimas ir įvertinimas – ilgas ir nelengvas darbas, todėl jį galima pradėti ir ankstesniuose etapuose. Renkant duomenis, naudinga iš anksto žinoti, kurie iš jų ypatingai svarbūs, formuluojant ir sprendžiant sudaromą modelį, o kurie mažiau įtakoja jo rezultatus. Kai kurių skaitinių reikšmių gali nereikėti iš viso, užtenka žinoti jų įvertinimus, pokyčio intervalus (teigiamas, mažesnis už vienetą ir pan.) Tai leidžia racionaliau organizuoti duomenų rinkimo darbą.

Modelio sprendimo etapas – anaiptol ne paskutinis darbo su juo etapas (ir tam tikra prasme netgi ne pats svarbiausias). Labai svarbus yra modelio tyrimo etapas. Šiame etape tiriama, ar modelis pakankamai gerai atvaizduoja modeliuojamą situaciją. Modelio sprendiniai turi teisingai (pagal pačios modeliuojamos situacijos logiką) reaguoti į įvairius parametrų ir duomenų pasikeitimus, tikrovėje stebimus reiškinius. Pavyzdžiui, padidėjus gaminamo produkto paklausai, pelnas turėtų bent jau nesumažėti; įdiegus pažangesnę, taupesnę gamybos technologiją – padidėti ir pan. Ypač rūpestingai reikia patikrinti modelio reakciją į „kraštutines“ (esančias ties pačia modelio prielaidų galiojimo riba) parametrų reikšmes. Ir apskritai, tyrimo etape labai svarbu patikrinti modelio jautrumą. Jei, nedaug pasikeitus modelio duomenims bei parametrams, jo sprendinys radikaliai keičiasi, tai, naudojant tokį modelį, bus sunku gauti pagrįstus rezultatus, kadangi ekonominių duomenų ir parametrų įvertinimas dažnai nėra pakankamai tikslus. Jei modelio jautrumo nepavyksta sumažinti, tai būtina ypač kruopščiai įvertinti duomenis ir parametrus, kurių pasikeitimams jis itin jautrus.

Modelio tyrimo etape galima ne tik patikrinti modelį, bet ir gauti kai kuriuos reikšmingus kokybinius bei kiekybinius rezultatus. Pavyzdžiui, galima sužinoti, kaip reikia keisti gamybą, rinkoje pasikeitus kainoms ar paklausai, išaiškinti „siauras“ gamybos vietas, išnagrinėti, kokių išteklių ir technologijų pakeitimas gali duoti didžiausią efektą ir t.t.

Įdiegimo etapas užbaigia uždavinio sprendimą. Ankstesniuose etapuose paruoštos, pagrįstos ir detaliai išnagrinėtos rekomendacijos turi būti pateiktos įmonės vadovams, kad jie priimtų sprendimą dėl įgyvendinimo. Apskritai vadovų dalyvavimas visuose sprendimo parengimo etapuose yra svarbus tam, kad jie žinotų sprendžiamo uždavinio esmę ir pagrindines detales. Kadangi kiekvienas matematinis modelis supaprastina realią situaciją, tai, juo remiantis, gautos rekomendacijos negali automatiškai tapti sprendimais. Galutinį sprendimą priima ir už jį atsako įmonių vadovai ir sprendimo rengėjai. Jei vadovai nepakankamai supranta atlikto tyrimo esmę ir metodus, jie gali nepasitikėti siūlomais sprendimais ir jų nepriimti, ypač, jei diegimo etape bus susidurta su nesklandumais. O tokie nesklandumai, matyt, yra neišvengiamas reiškinys. Į juos reikia žiūrėti kaip į natūralų viso projekto etapą, kuris gali priversti patikslinti problemos formulavimą, atitinkamai modifikuoti modelį ir t.t. Reikia pabrėžti, kad šis darbas yra dėsninga viso projekto dalis, ir modeliu pagrįstų sprendimų įdiegimas „iš pirmo karto“ yra greičiau išimtis, negu taisyklė.

Į išvardintus modelio sudarymo bei atitinkamo uždavinio sprendimo etapus nereikia žiūrėti dogmatiškai. Sprendžiant vienus uždavinius, ne visi etapai gali būti reikalingi, sprendžiant kitus, gali prireikti ir kitokių etapų. Be to, visi išvardinti etapai nebūtinai atliekami tiksliai „iš eilės“, gali prireikti nuolat grįžti į jau praeitus. Pavyzdžiui, sudarant modelį, gali tekti patikslinti problemos formulavimą, ieškant sprendinio – supaprastinti modelį, ištyrus modelį, gali tekti jį modifikuoti, o tam vėl tektų patikslinti problemos formulavimą ir t.t.

§ 2. PAPRASČIAUSI TIESINIO PROGRAMAVIMO UŽDAVINIAI

1^0. Gamybos planavimo uždavinys. Nagrinėkime įmonę, kuri gali gaminti n skirtingų produktų rūšių, naudodama m išteklių rūšių. Produktais gal būti bet kokie gatavi gaminiai, detalės, komponentai, kirti kitoms įmonėms ir t.t. Ištekliais laikome žaliavas, elektros energiją, kitų įmonių pagamintus komponentus, darbo išteklius ir pan.

Laikykime, kad laikotarpiui, kuriam sudarome gamybos planą, žinomos visų m išteklių rūšių apimtys, kurias žymėsime $b_1, b_2, \dots, b_m$. Be to, laikysime žinomomis visų išteklių sąnaudas, reikalingas kiekvieno produkto vienetui pagaminti. Žymėsime šias sąnaudas raidėmis su dviem indeksais: a_i^j reikšmė parodo, kiek i-tojo ištekliaus vienetų reikia j-tojo produkto vienetui pagaminti.[2] Pavyzdžiui, jei konkrečiame uždavinyje trečiuoju išteklium laikoma elektros energija, o pirmuoju produktu – batai, tai $a_3^1 = 70$ reiškia, kad vienai batų porai pagaminti sunaudojama 70 kWh elektros energijos.

Atkreipkime dėmesį, kad, formuluodami uždavinį, mes nepastebimai priėmėme labai svarbią tiesiškumo prielaidą: sąnaudos kiekvieno produkto gamybai laikomos tiesiogiai proporcingomis jo gamybos apimčiai. Praktiškai ši prielaida ne visada teisinga, bet nagrinėjamame uždavinyje laikysime ją priimtina.

Norint suformuluoti optimizacinį gamybos planavimo uždavinį, reikia mokėti įvertinti, kiek įmonei naudinga gaminti kiekvieną produktą. Naudingumo matu gamybos planavimo uždavinyje laikome pelną: sakysime, kad j-tojo produkto vieneto gamyba duoda c^j litų pelno.

Reikia surasti gamybos planą, duodantį maksimalų pelną duotomis sąlygomis.

Pradėsime nuo matematinio modelio sudarymo. Pažymėkime x_1 planuojamą gaminti pirmojo produkto apimtį, x_2 planuojamą gaminti antrojo produkto apimtį ir t.t. Taigi ieškonas planas yra vektorius $(x_1, x_2, \dots, x_n)$. Planas turi būti toks, kad būtų galima jį įgyvendinti su turimais ištekliais. Tokį planą vadinsime *leistinu.* Apskaičiuosime pirmosios rūšies išteklių sąnaudas, būtinas šiam planui įgyvendinti. Pirmojo produkto gaminame x_1 vienetų, vadinasi, jam pagaminti reikia $a_1^1 x_1$ vienetų pirmosios rūšies išteklių; visam antrajam produktui pagaminti reikės $a_1^2 x_2$ vienetų pirmosios rūšies išteklių; trečiajam - $a_1^3 x_3$ vienetų ir t.t. Taigi pirmosios rūšies išteklių iš viso prireiks $a_1^1 x_1 + a_1^2 x_2 + \cdots + a_1^n x_n$ vienetų, ir šis dydis negali būti didesnis už turimą šios

[2] Kiek neįprastas antrojo indekso žymėjimas raidės viršuje vėliau bus patogus matematinio tyrimo tikslais.

rūšies išteklių atsargą b_1 vienetų. Kadangi planas turi būti aprūpintas visais, o ne tik pirmosios rūšies ištekliais, tai analogiškus išteklių apskaičiavimus turime padaryti m kartų. Gausime tokią nelygybių sistemą:

$$\begin{gathered} a_1^1x_1+a_1^2x_2+\dots+a_1^nx_n \leq b_1, \\ a_2^1x_1+a_2^2x_2+\dots+a_2^nx_n \leq b_2, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ a_m^1x_1+a_m^2x_2+\dots+a_m^nx_n \leq b_m. \end{gathered} \tag{1}$$

Kadangi negalima gaminti neigiamos kurio nors produkto apimties, papildysime gautą sistemą dar n nelygybių:

$$x_1 \geq 0,\ x_2 \geq 0, \dots, x_n \geq 0. \tag{2}$$

Taigi jei planas $(x_1, x_2, \dots, x_n)$ tenkina (1) ir (2) apribojimus, tai pagal uždavinio sąlygą jį galima įgyvendinti (kaip susitarėm sakyti, jis yra *leistinas*). Jei apribojimai neprieštarauja vienas kitam, tai leistinų planų dažniausiai galima rasti labai daug. Pavyzdžiui, leistinas planas yra ir toks: $x_1 = x_2 = \dots = x_n = 0$. Kadangi išteklių apimtys neneigiamos, toks planas leistinas ir reiškia, kad įmonė negamina nė vieno produkto (tačiau jai tai tikriausiai nenaudinga). Todėl iš visų leistinų planų turime išrinkti geriausią, t.y. tą, kuris duoda maksimalų pelną. Apskaičiuosime tą pelną.

Kadangi pirmojo produkto įmonė gamina $x_1$vienetų, tai jo iš jo gamybos gaunamas pelnas yra c^1x_1litų; antrojo produkto įmonė gamina x_2 vienetų, ir jo iš jo gamybos gauna c^2x_2litų pelno ir t.t. Taigi planas turi būti toks, kad

$$c^1x_1+c^2x_2+\dots+c^nx_n \text{ (max)} \tag{3}$$

t.y. pelnas būtų maksimalus.

Užrašytas (1) – (3) uždavinys yra vadinamas gamybos planavimo uždaviniu. Nelygybės (1) ir (2) sudaro jo apribojimų sistemą, o (3) sąlyga – tikslo funkciją. Turėdami skaitines parametrų a_i^j , b_i, c^j reikšmes, galime uždavinį išspręsti ir gauti optimalų planą. Sprendimo metodą nagrinėsime vėliau (§6 -§7), o dabar paimkime konkretų gamybos planavimo uždavinio pavyzdį.

<u>1 pavyzdys</u>. Nedidelėje plytinėje dirba 10 žmonių. Plytinė gali gaminti “A”, “B”, “C” ir ”D” tipo plytas. Plytos gaminamos ir parduodamos ne po vieną, o blokais.

Svarbiausių išteklių sąnaudos kiekvieno tipo plytų blokui pagaminti parodytos lentelėje. Jos apačioje, be to, parodytos ir blokų realizavimo kainos.

1 lentelė

	"A"	"B"	"C"	"D"
Vieno darbininko darbo laikas, reikalingas plytų blokui pagaminti (val.)	2	3	4	3
Vandens sąnaudos vieno bloko gamybai (m^3)	1	4	5	1
Bloko degimo laikas (val.)	3	4	2	2
Molio sąnaudos vienam blokui (t)	2	2,5	3	2
Bloko realizavimo kaina (lt.)	90	110	130	80

Per 8 val. darbo dieną kiekvienas darbininkas gali skirti plytų gamybai 7,8 val. (likęs laikas naudojamas pagalbiniams darbams). Taigi plytinės darbo laiko ištekliai per darbo dieną – 78 val.

Už vandens tiekimą plytinė moka pastovų mokestį, nepriklausantį nuo sunaudoto vandens kiekio, jei šis neviršija 425 m^3 per darbo savaitę. Priešingu atveju tektų tiesti papildomą vandens tiekimo vamzdyną, kas yra labai brangu.

Plytos degamos blokais krosnyje. Vienu metu joje telpa 10 blokų. Krosnis veikia 8 val. per dieną ir leidžia pakrauti blokus, nenutraukiant kitų degimo. Kartą per dvi savaites krosnis 1 val. uždaroma profilaktikai.

Molį plytinei tiekia įmonė, kuriai reikia mokėti 20 lt. už toną. Galima užsakyti bet kokį molio kiekį.

Reikia sudaryti pelningiausią plytinės darbo planą dviem savaitėm (dešimčiai darbo dienų).

Pradėsime nuo matematinio šio uždavinio modelio sudarymo. Pirmiausia turime paskirti pagrindinius nežinomuosius. Tegu x_1 - planuojamų pagaminti per 10 darbo dienų „A“ tipo blokų skaičius, x_2 - „B“ tipo, x_3 - „C“ tipo, x_4 - „D“ tipo plytų blokų skaičius. „A“ tipo plytų blokų skaičiui x_1 pagaminti reikės $2x_1$valandų darbininkų darbo laiko; „B“ tipo plytų blokų skaičiui x_2 pagaminti reikės $3x_1$valandų darbininkų darbo laiko ir t.t. (žr. 1 lentelę). Darbo laiko ištekliai per 10 darbo dienų – 780 val. Todėl turi būti patenkinta nelygybė

$$2x_1 + 3x_2 + 4x_3 + 3x_4 \le 780.$$

Analogišką nelygybę užrašysime vandens sąnaudoms. Per dvi darbo savaites turi būti sunaudota ne daugiau kaip 2·425=850 m^3. Todėl turi būti patenkinta dar viena nelygybė:

$$1x_1 + 4x_2 + 5x_3 + 1x_4 \le 850.$$

Degimo krosnis per 10 darbo dienų dirba 79 val. (atmetus profilaktinio remonto valandą). Be to joje vienu metu telpa 10 blokų, todėl iš viso galime skaičiuoti 790 degimo valandų, jei pasirūpinsime, kad krosnis visą laiką būtų visiškai apkrauta. Todėl, pasinaudoję trečiojoje lentelės eilutėje nurodytomis degimo laiko sąnaudomis, galime užrašyti

$$3x_1 + 4x_2 + 2x_3 + 2x_4 \le 790.$$

Apribojimo molio ištekliams nerašome, nes jo galime užsakyti neribotą kiekį. Mūsų planui įgyvendinti reikia $2x_1 + 2{,}5x_2 + 3x_3 + 2x_4$ tonų molio. Kadangi už kiekvieną toną įmonė sumoka 20 litų, išlaidos moliui pirkti sudarys $40x_1 + 50x_2 + 60x_3 + 40x_4$ litų.

Pardavę pagamintas plytas, gausime $90x_1 + 110x_2 + 130x_3 + 80x_4$ litų. Atėmę iš šios sumos išlaidas moliui ir įvairias pastovias išlaidas, nepriklausančias nuo gamybos apimčių (darbininkų darbo užmokestis, išlaidos elektrai ir t.t.), gausime pelną

$$50x_1 + 60x_2 + 70x_3 + 40x_4 - d$$

čia d - pastovios, pagal uždavinio sąlygą nuo gamybos apimčių nepriklausančios gamybos sąnaudos.[3] Ieškodami sprendinio, jas galime atmesti, nes jos neturi įtakos x_1, x_2, x_3, x_4 reikšmėms. Taigi formuluojame tokį uždavinį:

$$50x_1 + 60x_2 + 70x_3 + 40x_4 \text{ (max)}$$

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 + 3x_4 &\le 780, \\ 1x_1 + 4x_2 + 5x_3 + 1x_4 &\le 850, \\ 3x_1 + 4x_2 + 2x_3 + 2x_4 &\le 790. \end{aligned} \tag{4}$$

$$x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0.$$

Suformuluotą uždavinį spręsime §6 po to, kai išnagrinėsime tiesinio programavimo uždavinių savybes ir sprendimo algoritmą. O kol kas panagrinėkim, kaip, formuluojant uždavinį, galima atsižvelgti į kai kurias ūkinėje veikloje galinčias iškilti papildomas sąlygas.

[3] Prielaidas, kad darbo užmokestis, elektros energijos sąnaudos ir kt. nepriklauso nuo gamybos apimčių, priimame tik tam, kad kuo paprasčiau suformuluotume mūsų pavyzdį. Praktiškai šių prielaidų būtų nesunku atsisakyti.

Sakykime, kad dėl kokių nors priežasčių plytinei per 10 darbo dienų būtina pagaminti ne mažiau kaip 300 visų rūšių tipų plytų blokų. Į šią sąlygą galime atsižvelgti, papildydami (4) nelygybes tokiu apribojimu:

$$x_1 + x_2 + x_3 + x_4 \geq 300. \tag{5}$$

Dar vienas svarbus plytinės uždavinio modifikavimas gali būti reikalingas, ištyrus gaminamų plytų paklausą. Sakykim, plytos naudojamos tokiai statybai, kurioje „A“ tipo plytų reikia lygiai du kartus daugiau, negu „C“ tipo plytų. Gaminant plytas kitu santykiu, netenkinančius komplektacijos kiekius būtų labai sunku parduoti. Todėl į uždavinio formulavimą turime įtraukti dar vieną sąlygą $x_1 = 2x_3$. Taigi apribojimų sistemą (4) turime papildyti lygtimi

$$x_1 - 2x_3 = 0. \tag{6}$$

Matome, kad uždavinyje (4) –(6) dabar turime įvairaus tipo apribojimų. Kaip spręsti tokį uždavinį taip pat nagrinėsime §7. $\prec$

Gamybos planavimo uždavinys – tik vienas iš daugelio tiesinio programavimo uždavinių ekonominio interpretavimo būdų. Artimi savo matematiniu pavidalu uždaviniai gali būti visiškai skirtingai interpretuojami. Tuo įsitikinsime, išnagrinėję kitą dažnai minimą ir naudojamą tiesinio programavimo uždavinį.

2^0. Dietos (mišinių) uždavinys. Šiame uždavinyje reikia sudaryti maitinimo planą. Žinoma, kad maitinamasis per maitinimo laikotarpį, kuriam sudaromas planas, turi gauti m maistingųjų medžiagų rūšių (pvz., baltymų, riebalų, vitaminų ir pan.). Būtinus šių medžiagų kiekius žymėsime $b_1, b_2, \dots, b_m$. Maistingosios medžiagos gaunamos nuperkant n maisto produktų rūšių (pvz., bulves, kruopas ir pan.). Žinoma, kiek kokių maistingųjų medžiagų yra kiekvieno produkto vienete. Žymėsime tą dydį a_i^j. Jo reikšmė parodo, kiek i-tosios maistingosios medžiagos yra j-tojo produkto vienete. Be to, žinomos perkamų produktų kainos: pirmojo produkto vienetas kainuoja c^1 litų, antrojo - c^2 ir t.t. Reikia kuo pigiau nupirkti produktų rinkinį, kuriame būtų visos būtinos maistingosios medžiagos.

Sudarysime matematinį šios situacijos modelį. Kaip visada, reikia pradėti nuo nežinomųjų paskyrimo. Tegu x_1 - planuojamas pirkti pirmojo produkto kiekis, x_2 - antrojo produkto kiekis ir t.t. Taigi $(x_1, x_2, \dots, x_n)$ - maisto produktų pirkimo planas. Jis turi būti leistinas, t.y. toks, kad jame nepritrūktų nė vienos būtinos maistingosios medžiagos. Suskaičiuosime pirmosios maistingosios medžiagos kiekį mūsų plane. Pirmajame perkamame produkte (visame jo kiekyje x_1) jos turime $a_1^1 x_1$ vienetų, antrajame - $a_1^2 x_2$ vienetų ir t.t. Taigi iš viso visuose produktuose turime $a_1^1 x_1 + a_1^2 x_2 + \cdots + a_1^n x_n$ pirmosios maistingosios medžiagos. Ši suma negali būti mažesnė už būtiną šios medžiagos kiekį b_1. Panašius skaičiavimus atlikę visoms maistingosioms medžiagoms, gauname tokią apribojimų sistemą:

$$
\begin{aligned}
a_1^1 x_1 + a_1^2 x_2 + \cdots + a_1^n x_n &\geq b_1, \\
a_2^1 x_1 + a_2^2 x_2 + \cdots + a_2^n x_n &\geq b_2, \\
\dots\dots\dots\dots\dots\dots\dots\dots & \\
a_m^1 x_1 + a_m^2 x_2 + \cdots + a_m^n x_n &\geq b_m.
\end{aligned} \tag{7}
$$

$$x_1 \geq 0,\ x_2 \geq 0, \dots, x_n \geq 0. \tag{8}$$

Kadangi negalima pirkti neigiamo kurio nors produkto kiekio, į apribojimų sistemą įtrauktos (8) nelygybės.

Taigi pirkimo planas $(x_1, x_2, \dots, x_n)$ leistinas, jei jis tenkina (7) ir (8) apribojimus. Suprantama, kad realiomis sąlygomis gali atsirasti labai daug leistinų planų. Iš jų mums reikia išrinkti geriausią. Kadangi už x_1 vienetų pirmojo produkto mokame $c^1 x_1$ litų, už x_2 vienetų antrojo produkto - $c^2 x_2$ litų, tai mūsų tikslas parinkti tokį planą, kad

$$c^1 x_1 + c^2 x_2 + \cdots + c^n x_n \ (\text{min}) \tag{9}$$

t.y. visa nupirktų produktų kaina būtų minimali.

Uždavinys (7) – (9) vadinamas dietos uždaviniu. Jo interpretavimo kaip mišinių uždavinio pavyzdį žr. toliau pateikiamame pratime 3^0.3.

3^0. Pratimai. Užrašykite kelių toliau pateikiamų uždavinių matematinius modelius. Atsakymus rasite §6 6^0.

3^0.1. Įmonė gali gaminti keturias produktų rūšis, naudodama trijų rūšių išteklius: žaliavas (kg.), darbo jėgą (val.) ir įrengimų laiką (val.). Turimi išteklių kiekiai ir sąnaudos kiekvieno produkto vieneto gamybai parodytos 2 lentelėje. Taip pat parodytos pajamos, gaunamos už parduotą kiekvieno produkto vienetą bei su to vieneto gamyba susijusios tiesioginės sąnaudos.

Suformuluoti šiai įmonei gamybos planavimo uždavinį.

2 lentelė

	Turimas kiekis	Išteklių sąnaudos produkto vienetui			
		prirmojo	antrojo	trečiojo	ketvirtojo
Žaliavų (kg.)	**450**	**7**	**9**	**12**	**4**
Darbo valandų (val.)	**300**	**8**	**3**	**2**	**5**
Įrengimų valandų (val.)	**300**	**4**	**6**	**9**	**10**
Pajamos (lt.)		**150**	**200**	**250**	**140**
Sąnaudos(lt.)		**48**	**70**	**58**	**40**

3^0.2. Įmonė gali gaminti keturias produktų rūšis, naudodama trijų rūšių išteklius: žaliavas (kg.), darbo jėgą (val.) ir įrengimų laiką (val.). Turimi išteklių kiekiai ir sąnaudos kiekvieno produkto vieneto gamybai parodytos 3 lentelėje. Taip pat parodytos pajamos, gaunamos už parduotą kiekvieno produkto vienetą bei su to vieneto gamyba susijusios tiesioginės sąnaudos. Suformuluoti šiai įmonei gamybos planavimo uždavinį.

3 lentelė

	Turimas kiekis	Išteklių sąnaudos produkto vienetui			
		pirmojo	antrojo	trečiojo	ketvirtojo
Žaliavų (kg.)	**1700**	**6**	**4**	**5**	**2**
Darbo valandų (val.)	**2700**	**8**	**3**	**6**	**7**
Įrengimų valandų (val.)	**3000**	**9**	**1**	**5**	**6**
Pajamos (tūkst. lt.)		**100**	**50**	**70**	**60**
Sąnaudos(tūkst. lt.)		**46**	**25**	**30**	**29**

3^0.3. Chemijos pramonės įmonėje ruošiamasi pradėti naujų mineralinių trąšų gamybą. Trašos bus gaminamos, sumaišaint keturių rūšių komponentus: A, B, C, D. 4 lentelėje pateikiami duomenys apie kalio, fosforo ir azoto kiekį gramais kiekviename kilograme atitinkamo komponento.

4 lentelė

Komponentas	Kalio (g./kg.)	Fosforo (g./kg.)	Azoto (g./kg.)
A	**50**	**40**	**50**
B	**30**	**10**	**40**
C	**80**	**20**	**50**
D	**10**	**20**	**30**

Žinoma, kad viename kilograme pagamintų trąšų turi būti ne mažiau kaip 190 g kalio, 110 g fosforo ir 150 g azoto. A komponento kg kaina 3,3 lito, B komponento – 1 litas, C komponento 2,6 lito, o D komponento 1,8 lito. Reikia apskaičiuoti, kokia proporcija turi būti sumaišomi komponentai, kad kilograme pagamintų trąšų būtų reikiamas kalio, fosforo bei azoto kiekis, o jo savikaina mažiausia.

§3. GEOMETRINIS TIESINIO PROGRAMAVIMO UŽDAVINIŲ INTERPRETAVIMAS

Prieš pradedant detaliai tyrineti tiesinio programavimo uždavinius, naudinga susipažinti su jų geometriniu interpretavimu. Be abejo, toks interpretavimas nepakeičia teorinio nagrinėjimo, bet jis leidžia pailiustruoti pagrindines tiesinio programavimo sąvokas, uždavinių struktūrą ir jų sprendimo problemas.

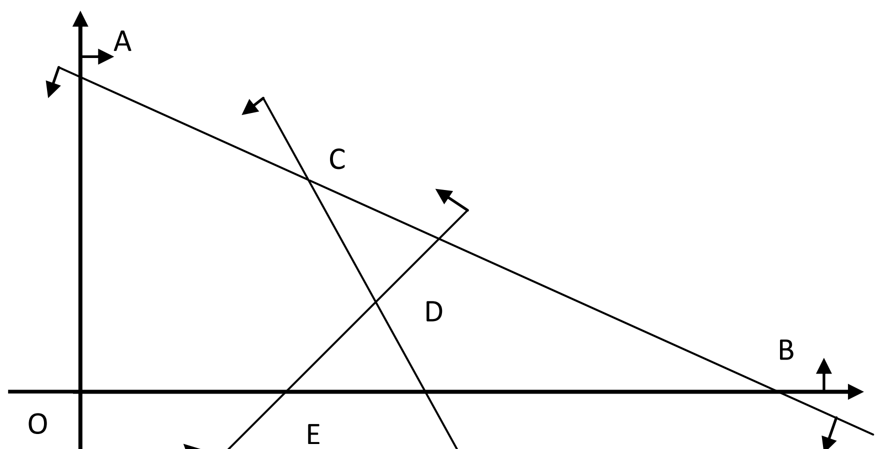
Geometriškai interpretuosime tiesinio programavimo uždavinius su dviem nežinomaisiais, nes tik tokį uždavinį galima atvaizduoti plokščiame dvimačiame brėžinyje. Nagrinėsime uždavinį

$$\begin{gathered} c^1x_1+c^2x_2 \text{ (max)} \\ a_1^1x_1+a_1^2x_2 \le b_1, \\ a_2^1x_1+a_2^2x_2 \le b_2, \\ \dots\dots\dots\dots\dots\dots \\ a_m^1x_1+a_m^2x_2 \le b_m. \\ x_1 \ge 0,\ x_2 \ge 0. \end{gathered} \tag{1}$$

Tai uždavinys su dviem nežinomaisiais ir m apribojimų-nelygybių.

Atvaizduosime plokštumoje visus taškus, kurių koordinatės x_1 ir x_2 tenkina uždavinio apribojimus. Pradėsime nuo apribojimų- nelygybių Imkime kurią nors iš jų, sakykim, pirmąją, ir iš pradžių užrašykime kaip lygybę: $a_1^1x_1+a_1^2x_2 = b_1$. Kaip žinome, tai yra tiesės lygtis. Nubrėžkime šią tiesę 1 brėžinyje, sakykim, kaip tiesę AB.

1 brėžinys



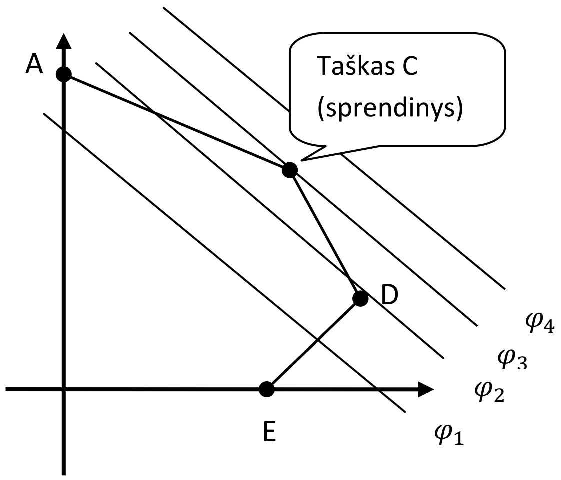
Aibė taškų *(x_1, x_2),* kurių koordinatės tenkina nelygybę $a_1^1 x_1 + a_1^2 x_2 \leq b_1$, priklausys vienai plokštumos, perskirtos šia tiese, pusei. Kuriai būtent pusei, lengva sužinoti, patikrinus, ar taškas (0,0) yra ieškomoje plokštumos pusėje. Jei $a_1^1 \cdot 0 + a_1^2 \cdot 0 \leq b_1$, t.y. $0 \leq b_1$, tai reikalinga pusplokštumė yra ta, kurioje yra taškas (0,0). Ją brėžinyje pažymime rodyklytėmis. Analogiškai atžymime ir kitus apribojimus atitinkančias pusplokštumes. Jų brėžinyje atvaizdavome tris.

Belieka atvaizduoti neneigiamumo apribojimus. Akivaizdu, kad visi taškai, kurių $x_1 \geq 0$, yra dešinėje nuo ordinačių ašies, o visi taškai, kurių $x_2 \geq 0$, yra viršuje nuo abscisių ašies, t.y. pirmame brėžinio kvadrante.

Kadangi ieškomi taškai turi tenkinti visus apribojimus, jie priklauso visų nubrėžtų pusplokštumių sankirtai ir sudaro daugiakampį (briaunainį) OACDE.

Atvaizduosime brežinyje (1) uždavinio kriterinę dalį – *tikslo funkciją* $\varphi = c^1 x_1 + c^2 x_2$. Šios funkcijos grafikas yra plokštuma trimatėje erdvėje *(x_1, x_2, φ).* Kadangi trimačio brėžinio nedarome, vaizduosime ne patį funkcijos grafiką, o jos *lygio linijas.* Šios linijos jungia taškus, kuriose tikslo funkcijos reikšmė ta pati. Pavyzdžiui, tiesė $c^1 x_1 + c^2 x_2 = 1$ jungia visus taškus, kuriuose tikslo funkcijos reikšmė lygi vienetui. Visos lygio linijos yra viena kitai lygiagrečios.

2 brėžinys



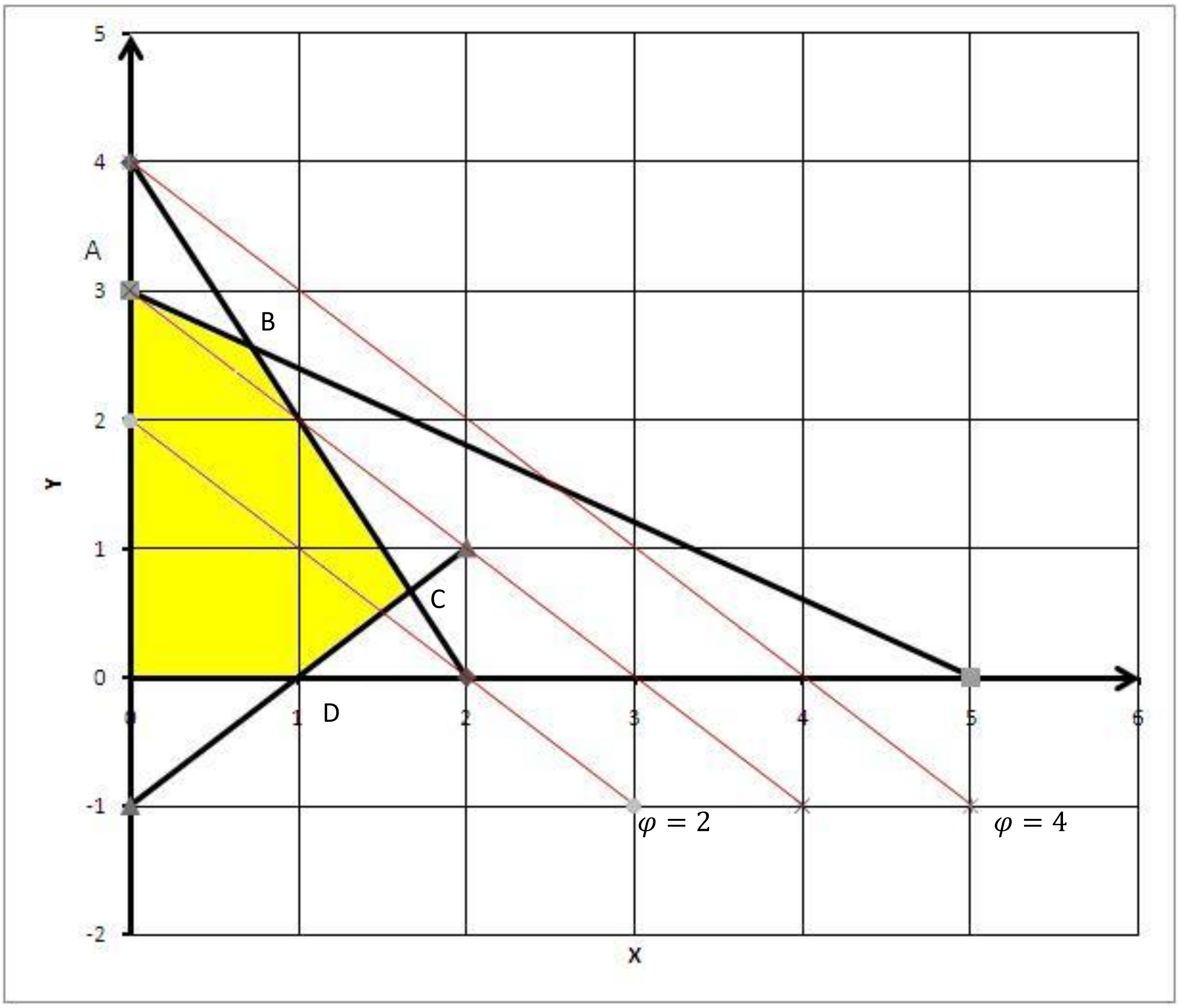
Brėžinyje atvaizdavome keturias lygio linijas su vis didesne tikslo funkcijos reikšme: $\varphi_1 < \varphi_2 < \varphi_3 < \varphi_4$. Matome, kad turime daug leistinų taškų, kuriuose tikslo funkcijos reikšmė yra φ_1 , taip pat daug taškų, kuriuose ši reikšmė lygi φ_2. Tačiau yra tik vienintelis taškas C, kuriame tikslo funkcijos reikšmė lygi φ_3 , ir nėra nė vieno leistino taško, kuriame jos reikšmė būtų dar didesnė - φ_4. Taigi C - taškas su didžiausia tikslo funkcijos reikšme, todėl jis yra uždavinio sprendinys.

Pavyzdys. Spręskime uždavinį

$$x_1 + x_2 \text{ (max)}$$
$$2x_1 + x_2 \leq 4,$$
$$3x_1 + 5x_2 \leq 15,$$
$$x_1 - x_2 \leq 1,$$
$$x_1 \geq 0,\ x_2 \geq 0.$$

Apribojimų tiesėms nubrėžti patogu naudotis tokiu metodu. Imkime tiesę $3x_1 + 5x_2 = 15$. Lengvai randame du jos taškus (0,3) ir (5,0). Atžymėję šiuos du taškus brėžinyje, nubrėžiame per juos tiesę (žr. 3 brežinį).

3 brėžinys



Kadangi 3·0 + 5·0 < 15, taškas (0,0) priklauso ieškomai pusplokštumei, taigi ji yra nubrėžtos tiesės „apačioje“. Analogiškai nubrėžiame ir kitas apribojimų tieses. Atkreipkime dėmesį, kad viena iš leistinų pusplokštumių bus „aukščiau“ tiesės $x_1 - x_2 = 1$, nes 0 – 0 < 1.

Išvedę kelias lygio linijas ($\varphi = 2$, $\varphi = 3$, $\varphi = 4$), matome, kad maksimalią reikšmę daugiakampyje OABCD tikslo funkcija įgyja taške B. Iš brėžinio aišku, kad ji didesnė už 3, bet neviršija 4. Patikslinti šią reikšmę galime, apskaičiavę optimalaus plano – taško B koordinates.Šis taškas priklauso tiesėms $2x_1 + x_2 = 4$, $3x_1 + 5x_2 = 15$. Išsprendę šią lygčių sistemą, gauname, kad $x_1^* = \frac{5}{7}, x_2^* = 2\frac{4}{7}\ \varphi^* = 3\frac{2}{7}$.◄

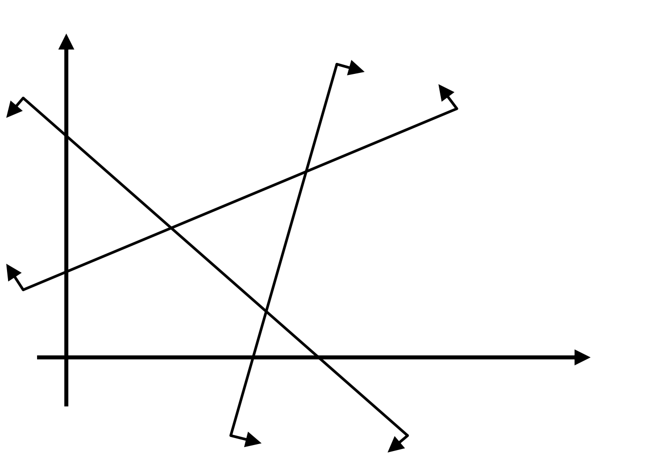
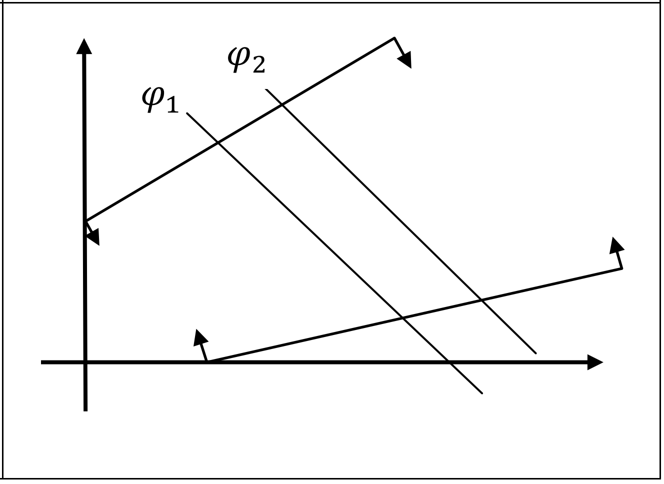
Praktiškai sprendžiant dviejų nežinomųjų uždavinį grafiniu būdu, nebūtina brėžti daug lygio linijų. Užtenka nubrėžti tik vieną iš jų ir „stumti“ ją lygiagrečiai sau pačiai lygio didėjimo kryptimi (t.y. vektoriaus (c^1, c^2) kryptimi). Lygio liniją „stumiame“ tol, kol ji dar kerta ar bent liečia leistinų planų daugiakampį. Paskutinis susikirtimo taškas (arba aibė taškų) yra optimalus planas (optimalūs planai).

Geometrinis (grafinis) tiesinio programavimo uždavinių interpretavimas, kaip matome, leidžia išspręsti paprasčiausius tiesinio programavimo uždavinius su dviem nežinomaisiais. Tačiau kur kas reikšmingesnis jis tuo, kad leidžia kai ką spėti apie tiesinio programavimo uždavinių savybes ir šiuos spėjimus vaizdžiai iliustruoti.

Pirma, nagrinėdami brėžinius, mes matome, kad taškai, kurių koordinatės tenkina uždavinio apribojimus (leistini planai), sudaro daugiabriaunes aibes.

Antra, sprendinį (optimalų planą) visada gauname tokių aibių viršūnėse (kraštutiniuose taškuose: C 2 brėž., B 3 brėž.)

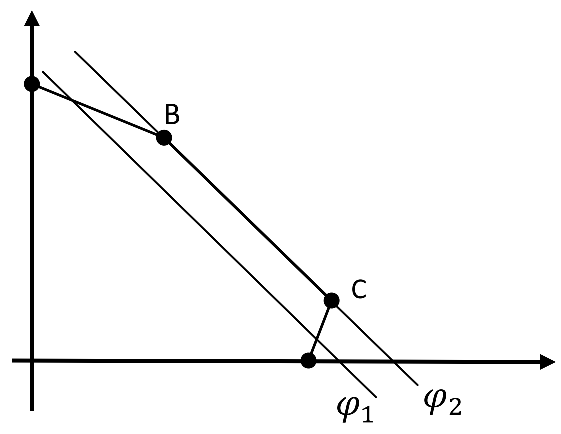
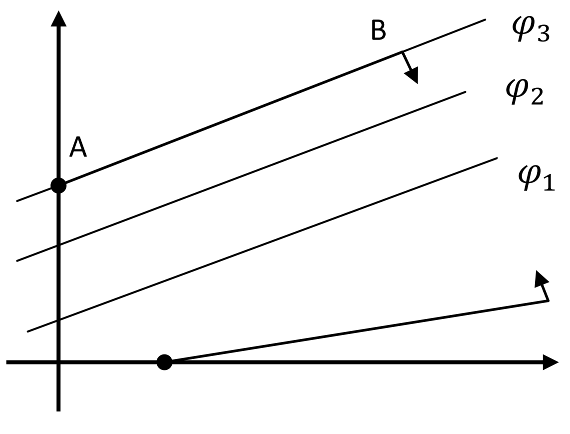
Geometrinis interpretavimas leidžia kai ką spėti ir apie sprendinio egzistavimą bei vienatį.

4a brėžinys	4b brėžinys
	

4a brėžinyje matome, kad nėra nė vieno taško, kuris priklausytų visoms trims nubrėžtoms pusplokštumėms. Tai reiškia, kad uždavinio apribojimai prieštaringi. Matyt, jį formuluodami, padarėme klaidų – reikalaujame nesuderinamų dalykų, kurių visų kartu neįmanoma įgyvendinti. 4b brėžinyje atvaizduota situacija, kurioje tikslo funkcijos reikšmė gali pasiekti kaip norima didelę reikšmę: kiek „bestumtume" lygio liniją funkcijos didėjimo kryptimi, ji neišeis už leistinų planų aibės ribų. Matyt, formuluodami uždavinį, neesminiu palaikėme kažkokį esminį apribojimą, dėl kurio iš tikrųjų tikslo funkcijos reikšmė negalėtų didėti iki begalybės. Taigi iš šių brėžinių galime padaryti tokį trečią spėjimą.

Tiesinio programavimo uždavinys gali būti neišsprendžiamas dėl dviejų priežasčių: apribojimų prieštaringumo arba tikslo funkcijos neaprėžtumo.

Jei tiesinio programavimo uždavinys vis dėlto išsprendžiamas, tai, be jau nagrinėtų 2 ir 3 brėžiniuose situacijų, kai jo sprendinys (optimalus planas) vienintelis, galime gauti ir tokias, kurias atvaizduosime 5a ir 5b brėžiniuose.

5a brėžinys	5b brėžinys
	

5a brėžinyje atvaizduotas uždavinys, kuriame „stumdami" tikslo funkcijos lygio liniją, matome, kad jos kraštutinė padėtis sutampa su visa atkarpa BC. Tai reiškia, kad maksimali tikslo funkcijos reikšmė įgyjama iš karto visuose atkarpos BC taškuose. Taigi taškai B ir C, taip pat ir visi taškai juos jungiančioje atkarpoje yra optimalūs planai (sprendiniai). 5b brėžinyje atvaizduotas uždavinys, kurio leistinų planų aibė neaprėžta. Tačiau tikslo funkcijos lygio linijos ir didėjimo kryptis šiuo atveju tokios, kad kraštutinė lygio linijos padėtis sutampa su visu spinduliu, išeinančiu iš taško A. Todėl taškas A ir visi spindulio AB taškai yra optimalūs planai (sprendiniai). Jų aibė neaprėžta.

Geometrinis tiesinio programavimo uždavinių interpretavimas leido mums vaizdžiai pamatyti pagrindines jų savybes ir sprendinių struktūrą. Toliau šiuos dalykus analizuosime griežtai.

§ 4. TIESINIO PROGRAMAVIMO MATEMATINIO APARATO ELEMENTAI

1^0. Kai kurios tiesinės algebros sąvokos. Šiame paragrafe mes prisiminsime kai kurias tiesinės algebros žinias, būtinas tiesinio programavimo uždavinių analizei ir sprendimui. Taip pat susitarsime dėl žymėjimų sistemos, naudojamos tolesniame dėstyme. Paragrafe pateikiami matematiniai faktai neįrodinėjami. Įrodymus galima rasti specialioje literatūroje.

Sutvarkytą n skaičių rinkinį vadinsime n-mačiu *vektoriumi.* Pačius skaičius vadinsime *vektoriaus koordinatėmis* ir žymėsime $x_1, x_2, \ldots, x_n$. Kad būtų lengviau atskirti vektorių nuo skaliaro (skaičiaus), žymėsime vektorių paryškinta raide, pvz., $\boldsymbol{x}$, $\boldsymbol{b}$ ir pan. Du vektorius laikysime lygiais, jei lygios jų atitinkamos koordinatės.

Apibrėšime vektorių sudėties ir daugybos iš skaliaro veiksmus. Vektorių $\boldsymbol{x}$ ir $\boldsymbol{y}$ suma laikysime vektorių $\boldsymbol{x} + \boldsymbol{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$. Skaliaro α ir vektoriaus $\boldsymbol{x}$ sandauga vadinsime vektorių $\alpha\boldsymbol{x} = (\alpha x_1, \alpha x_2, \ldots, \alpha x_n)$. Visų n-mačių vektorių aibę su šitaip apibrėžtais sudėties ir daugybos iš skaliaro veiksmais vadiname n-mate tiesine erdve ir žymime R^n. Taigi, rašydami $\boldsymbol{z} \in R^n$, norime pasakyti, kad $\boldsymbol{z}$ yra n-matis vektorius.

Susitarkime $\mathbf{0}$ žymėti nulinį R^n vektorių: $\mathbf{0} = (0,0, \ldots, 0)$.

Vektorių nelygybę $\boldsymbol{x} \leq \boldsymbol{y}$ suprasime kaip n nelygybių: $x_1 \leq y_1, x_2 \leq y_2, \ldots, x_n \leq y_n$.

Matrica vadinsime skaičių lentelę, kuri bendru atveju turi m eilučių ir n stulpelių. Matricas žymėsime didžiosiomis raidėmis, o jų elementus – mažosiomis. Galime užrašyti, pavyzdžiui, tokią matricą

$$A = \begin{pmatrix} a_1^1 & a_1^2 \ldots & a_1^n \\ a_2^1 & a_2^2 & \ldots a_2^n \\ \ldots & \ldots & \ldots \\ a_m^1 & a_m^2 & \ldots a_m^n \end{pmatrix}$$

Esant reikalui, matricą žymėsime ir trumpiau: A = (a_i^j), $i = \overline{1, m}$, $j = \overline{1, n}$ (žymėjimas $i = \overline{1, m}$ reiškia, kad indeksas i įgyja reikšmes 1,2,... m).

Jei m = n, matrica vadinama *kvadratine.*

Matricas, kaip ir vektorius, galima dauginti iš skaliaro bei sudėti. Daugindami matricą iš skaliaro, iš jo dauginame visus matricos elementus. Sudėti galima tik vienodų dimensijų matricas.

Šiuo atveju abiejų matricų elementus sudedame panariui ir vėl gauname tos pačios dimensijos matricą.

Svarbus veiksmas su matricomis – jų daugyba. Matricas A ir B galima sudauginti, jei matrica A turi tiek pat stulpelių, kiek matrica B eilučių. Sakykim, matrica A turi m eilučių ir k stulpelių (sakome, kad jos dimensija $m \times k$), o matrica B turi k eilučių ir n stulpelių (jos dimensija $k \times n$). Tada AB yra matrica, kuri turi m eilučių ir n stulpelių (jos dimensija $m \times n$). Atkreipkime dėmesį, kad atvirkštine tvarka (BA) šių matricų nebus galima sudauginti, nebent m būtų lygus n.

Į vektorius daugeliu atveju patogu žiūrėti kaip į matricas, turinčias vieną eilutę arba vieną stulpelį. Iš tikrųjų, $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ gali būti suprantamas ne tik kaip vektorius, bet ir kaip $1 \times n$ dimensijos matrica. Jei jo koordinates surašysime stulpeliu, tai turėsime n eilučių ir vieno stulpelio matricą, kurios dimensija $m \times 1$. Tokį vektorių vadinsime *vektoriumi-stulpeliu* ir susitarsime, kad toliau, jei nebus specialiai pasakyta kitaip, visus vektorius $\boldsymbol{x}, \boldsymbol{y}$ ir t.t. laikysime vektoriais-stulpeliais. Išimtį padarysime tik raide $\boldsymbol{c}$ žymimam vektoriui $\boldsymbol{c} = (c^1, c^2, \dots, c^n)$, kurį laikysime vektoriumi-eilute (jo, kaip matricos, dimensija $1 \times n$).

Vektoriai-stulpeliai, taip pat ir vektoriai-eilutės gali būti dauginami iš matricų. Sakykim, matricą A $(m \times n)$ galima padauginti iš vektoriaus stulpelio $\boldsymbol{x}$ $(n \times 1)$. Sandauga bus vektorius-stulpelis $A\boldsymbol{x}$ $(m \times 1)$. Parodysim, kaip apskaičiuojama ši sandauga.

$$\begin{pmatrix} a_1^1 & a_1^2 \dots & a_1^n \\ a_2^1 & a_2^2 & \dots a_2^n \\ \dots & \dots \dots & \dots \dots \\ a_m^1 & a_m^2 & \dots a_m^n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} a_1^1 x_1 + a_1^2 x_2 + \dots + a_1^n x_n \\ a_2^1 x_1 + a_2^2 x_2 + \dots + a_2^n x_n \\ \dots \dots \dots \dots \dots \dots \dots \\ a_m^1 x_1 + a_m^2 x_2 + \dots + a_m^n x_n \end{pmatrix}$$

Atkreipkime dėmesį, kad sandauga $A\boldsymbol{x}$ yra ne kas kita, kaip (2.1) nelygybių kairioji pusė. Jei šių nelygybių dešiniąją pusę užrašysim kaip vektorių-stulpelį $\boldsymbol{b}$ $(m \times 1)$, visą (2.1) nelygybių sis–temą galėsime kompaktiškai užrašyti pavidalu $A\boldsymbol{x} \le \boldsymbol{b}$**.** Nelygybes $x_1 \ge 0,\ x_2 \ge 0, \dots, x_n \ge 0$ užrašysim pavidalu $\boldsymbol{x} \ge \boldsymbol{0}$**,** o sąlygą (max) $c^1 x_1 + c^2 x_2 + \dots + c^n x_n$ užrašysim vektoriaus – eilutės $\boldsymbol{c}$ pagalba kaip (max) $\boldsymbol{cx}$**.** (Atkreipkime dėmesį, kad vektorių-eilutę $(1 \times n)$ ir vektorių-stulpelį $(n \times 1)$ sudauginti pagal matricų daugybos taisykles galima – rezultatas bus skaliarinis (1×1) dydis – mūsų uždavinio tikslo funkcijos reikšmė.)

Taigi §2 suformuluotą gamybos planavimo uždavinį galime kompaktiškai užrašyti taip:

$$\text{(max) } \boldsymbol{cx}, \quad A\boldsymbol{x} \le \boldsymbol{b}, \quad \boldsymbol{x} \ge \boldsymbol{0}.$$

Labai panašiai galime užrašyti ir dietos uždavinį:

$$\text{(min)}\ \boldsymbol{cx},\quad A\boldsymbol{x} \geq \boldsymbol{b},\quad \boldsymbol{x} \geq \mathbf{0}.$$

Tolesnei analizei mums reikės atskirai nagrinėti matricos A eilutes ir stulpelius. Šios matricos i-tąją eilutę žymėsime $\boldsymbol{a_i}$, t.y. $\boldsymbol{a_i} = (a_i^1, a_i^2, \dots, a_i^n)$. Analogiškai šios matricos j-tąjį stulpelį žymėsim $\boldsymbol{a}^j$. Atkreipkime dėmesį, kad, taip pažymėję, galime užrašyti

$$A\boldsymbol{x} = \begin{pmatrix} \boldsymbol{a_1x} \\ \boldsymbol{a_2x} \\ \dots \\ \boldsymbol{a_mx} \end{pmatrix}$$

Panašiai galime užrašyti ir vektoriaus eilutės bei matricos sandaugą. Imkim (1×m) vektorių eilutę $\boldsymbol{p} = (p^1, p^2, \dots, p^m)$. Tada galime užrašyti, kad

$$\boldsymbol{p}A = (\boldsymbol{pa}^1, \boldsymbol{pa}^2, \dots, \boldsymbol{pa}^n).$$

§ 5. TIESINIO PROGRAMAVIMO UŽDAVINIŲ TIPAI IR PAGRINDINĖS SAVYBĖS

Iš anksčiau nagrinėtų pavyzdžių, mes matome, kad tiesinio programavimo uždaviniai gali būti formuluojami gana įvairiai. Jie gali būti sprendžiami, ieškant maksimumo (gamybos planavimo uždavinys) ar minimumo (dietos uždavinys); apribojimai gali būti nelygybės tipo „≥" arba „≤", bet gali būti ir lygybės; nežinomųjų reikšmės paprastai turi būti neneigiamos, bet yra uždavinių, kuriuose kai kurie nežinomieji aprėžti iš viršaus ar apačios kitais dydžiais, o gal ir iš viso neaprėžti. Todėl bendriausiu atveju tiesinio programavimo uždavinį galima užrašyti taip:

$$\text{(extr)}\ c^1x_1+c^2x_2+\cdots+c^nx_n$$

$$a_i^1x_1+a_i^2x_2+\cdots+a_i^nx_n \begin{Bmatrix} \le \\ = \\ \ge \end{Bmatrix} b_i\,,\ i=\overline{1,m} \tag{1}$$

$$\propto_j \le x_j \le \beta_j\,,\ j=\overline{1,n}$$

Pirmojoje eilutėje užrašyta sąlyga vadinama uždavinio *tikslo funkcija*, o likusioji dalis – *apribojimų sistema.*

Užrašius uždavinį tokiu bendru pavidalu, būtų labai sunku formuluoti kokias nors jo sprendimo taisykles. Kiekvieną kartą reikėtų daryti išlygas dėl to, minimumo ar maksimumo ieškome, į kurią pusę nukreiptos nelygybės ir t.t. Todėl suformuluosime porą labiau apibrėžtų uždavinių, kuriuos ir spręsime, o paskui bandysime gautas sprendimo taisykles pritaikyti kitokių tiesinio programavimo uždavinių sprendimui.

Standartiniu pavadinsime tokį tiesinio programavimo uždavinį:

$$\text{(max)}\ \boldsymbol{cx},\ \ A\boldsymbol{x}\le\boldsymbol{b},\ \ \boldsymbol{x}\ge\boldsymbol{0}. \tag{2}$$

Kaip matome, gamybos planavimo uždavinys yra standartinis tiesinio programavimo uždavinys.

Kaip jau susitarėme, šio uždavinio *leistinu planu* vadiname n-matį vektorių-stulpelį $\boldsymbol{x}$, tenkinantį apribojimus $A\boldsymbol{x}\le\boldsymbol{b},\ \ \boldsymbol{x}\ge\boldsymbol{0}$. *Optimaliu planu* arba *sprendiniu* vadiname tokį leistiną planą $\boldsymbol{x}^*$, kuriam galioja nelygybė $\boldsymbol{cx}^*\ge\boldsymbol{cx}$ visiems leistiniems $\boldsymbol{x}$.

Kanoniniu pavadinsime tokį tiesinio programavimo uždavinį:

$$\text{(max)}\ \boldsymbol{cx},\ \ A\boldsymbol{x}=\boldsymbol{b},\ \ \boldsymbol{x}\ge\boldsymbol{0}. \tag{3}$$

Abiem atvejais čia A yra $m \times n$ apribojimų matrica, $\boldsymbol{b}$ – $m \times 1$ apribojimų vektorius-stulpelis, $\boldsymbol{c}$ – $1 \times n$ tikslo funkcijos koeficientų vektorius-eilutė, $\boldsymbol{x}$ – $n \times 1$ nežinomųjų vektorius-stulpelis.

Tiesinio programavimo uždavinio sprendimo taisyklės paprastai formuluojamos kanoniniam uždaviniui. Tačiau, nagrinėdami praktinius pavyzdžius, anksčiau susidūrėme su standartiniu uždaviniu. Todėl parodysime, kaip standartinį uždavinį paversti specialaus pavidalo kanoniniu uždaviniu, kuris bus labai patogus sprendimo taisyklėms išsiaiškinti.

Papildykime (2) uždavinį vektorium-stulpeliu $\boldsymbol{y}$ $(m \times 1)$ papildomų nežinomųjų. Gausime tokį uždavinį:

$$\text{(max) } \boldsymbol{cx}, \quad A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0}, \quad \boldsymbol{y} \geq \boldsymbol{0}. \tag{4}$$

Įrodysime, kad (2) ir (4) uždaviniai ekvivalentiški tokia prasme: jei $\boldsymbol{x}$ yra leistinas (2) uždavinio planas, tai galima rasti tokį $\boldsymbol{y}$, kad $\boldsymbol{x}$ ir $\boldsymbol{y}$ tenkintų (4) apribojimus. Ir atvirkščiai: jei $\boldsymbol{x}$ ir $\boldsymbol{y}$ tenkina (4) apribojimus, tai $\boldsymbol{x}$ yra leistinas (2) uždavinio planas.

Iš tikrųjų, jei $\boldsymbol{x}$ yra leistinas (2) uždavinio planas, tai $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$. Pažymėję $\boldsymbol{y} = \boldsymbol{b} - A\boldsymbol{x}$, matome, kad $\boldsymbol{y} \geq \boldsymbol{0}$. Taigi $\boldsymbol{x}$ ir $\boldsymbol{y}$ neneigiami ir tenkina (4) apribojimus. Atvirkščiai, jei $\boldsymbol{x}$ ir $\boldsymbol{y}$ tenkina (4) apribojimus, tai $A\boldsymbol{x} = \boldsymbol{b} - \boldsymbol{y} \leq \boldsymbol{b}$. Taigi, $\boldsymbol{x}$ tenkina (2) apribojimus ir yra leistinas (2) uždavinio planas. Vadinasi, (max) $\boldsymbol{cx}$ abiejų uždavinių atveju ieškome toje pačioje aibėje, todėl ir optimalus $\boldsymbol{x}$ bus tas pats. ∢

Jei į (2) uždavinį žiūrime kaip į gamybos planavimo uždavinį, tai vektorius $\boldsymbol{y}$ žymi nepanaudotas planui $\boldsymbol{x}$ įgyvendinti išteklių apimtis. Kaip jau aiškinomės, $A\boldsymbol{x}$ yra planui $\boldsymbol{x}$ įgyvendinti reikalingos išteklių apimtys, $\boldsymbol{y}$ - perteklinės (nepanaudotos) apimtys, taigi $A\boldsymbol{x} + \boldsymbol{y}$ - visos turimos išteklių apimtys, ir jos pagal sąlygą lygios $\boldsymbol{b}$. Be to, $\boldsymbol{y} \geq \boldsymbol{0}$, nes „neigiamas perteklius" reikštų paprasčiausią išteklių trūkumą, o to uždavinio sąlyga neleidžia. Pavyzdžiui, §2 nagrinėtame plytinės uždavinyje y_1 žymėtų nepanaudotas darbo valandas, y_2 - perteklinį vandens kiekį, o y_3 - nepanaudotas krosnies darbo valandas.

§ 6. SIMPLEKSINIS METODAS

1^0. Simpleksinio metodo esmė. Šiame paragrafe spręsime tiesinio programavimo uždavinį su m apribojimų-nelygybių ir n neneigiamų nežinomųjų:

$$\text{(max)}\ \varphi = \boldsymbol{cx},\quad A\boldsymbol{x} \le \boldsymbol{b},\quad \boldsymbol{x} \ge \boldsymbol{0} \tag{1}$$

(čia, kaip ir anksčiau $A - m \times n$ matrica, $\boldsymbol{b} - m \times 1$ vektorius stulpelis, $\boldsymbol{c} - 1 \times n$ vektorius eilutė, o $\boldsymbol{x} - n \times 1$ nežinomųjų vektorius stulpelis. Tolesnio dėstymo patogumui tikslo funkcijos reikšmę pažymime raide φ.

Ankstesniuose paragrafuose mes suformulavome šį uždavinį ir išsiaiškinome, kokioms problemoms spręsti jis gali būti panaudotas. Dviejų nežinomųjų atveju netgi išmokome jį spręsti. Dabar pabandykime spręsti bendrą n-matį atvejį.

Naudosimės vadinamuoju *simpleksiniu metodu.* Pirmą kartą pagrindžiant metodą, buvo naudojami tam tikri matematiniai objektai – simpleksai. Todėl ir metodas pavadintas simpleksiniu. Dabar apsieiname be simpleksų, bet pavadinimas liko. Metodo esmė paprasta – iš pradžių surandame kokį nors (1) uždavinio leistiną planą $\boldsymbol{x}$ (tegu ir neoptimalų). Paskui įvertiname visus tam tikra prasme jam „kaimyninius“ planus. Jei tarp jų nėra geresnio, t.y. su didesne tikslo funkcijos reikšme, turimas planas optimalus. Jei geresnis planas atsiranda, pereiname prie jo ir visą procesą kartojame iš naujo. Taip per baigtinį žingsnių skaičių pasiekiamas optimalus planas (arba įrodoma, kad jo nėra). Tačiau peržiūrėti visų bent kiek sudėtingesnio uždavinio planų praktiškai neįmanoma, kadangi šių planų be galo daug. Užtenka peržiūrėti tik vadinamus atraminius planus. Geometrine prasme atraminiai planai yra leistinų planų aibės viršūnės (taškai A, C, D E §2 2 brėž.). Jų taip pat labai daug, tačiau jų skaičius baigtinis, be to, simpleksinis metodas sukonstruotas taip, kad užtemka peržiūrėti tik nedidelę dalį atraminių planų, kad jų tarpe surastume optimalų planą – uždavinio sprendinį.

Pereikim prie (1) uždavinio sprendimo. Kad būtų paprasčiau, laikykime, kad $\boldsymbol{b} \ge \boldsymbol{0}$ (vėliau šios sąlygos galėsime atsisakyti).

2^0. Ekvivalentūs užrašymai ir simpleksinės lentelės. Uždavinio apribojimai užrašyti kaip nelygybės. Tai nepatogu, todėl pasinaudokime ekvivalenčia šio uždavinio forma (žr. §5), kurią gauname įvedę m papildomų neneigiamų nežinomųjų y_i, kurie sudaro– $m \times 1$ papildomų nežinomųjų vektorių stulpelį $\boldsymbol{y}$. Gauname ekvivalentišką uždavinį

$$\text{(max)}\ \varphi = \boldsymbol{cx},\quad A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b},\quad \boldsymbol{x} \ge \boldsymbol{0}\ \boldsymbol{y} \ge \boldsymbol{0}. \tag{2}$$

Išskleistu pavidalu šis uždavinys atrodo taip:

$$(\max)\ \varphi = c^1x_1 + c^2x_2 + \cdots + c^nx_n.$$

$$\begin{aligned} a_1^1x_1 + a_1^2x_2 + \cdots + a_1^nx_n + y_1 &= b_1, \\ a_2^1x_1 + a_2^2x_2 + \cdots + a_2^nx_n + y_2 &= b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_m^1x_1 + a_m^2x_2 + \cdots + a_m^nx_n + y_m &= b_m. \end{aligned} \qquad (3)$$

$$x_1 \geq 0,\ x_2 \geq 0, \ldots, x_n \geq 0;\ y_1 \geq 0,\ y_2 \geq 0, \ldots, y_m \geq 0.$$

Perrašykime šio uždavinio apribojimus ir tikslo funkciją kiek kitokiu pavidalu

$$\begin{aligned} \boldsymbol{y} &= A(-\boldsymbol{x}) + \boldsymbol{b}, \\ \varphi &= (-\boldsymbol{c})(-\boldsymbol{x}) + 0. \end{aligned} \qquad (4)$$

Neneigiamumo apribojimus laikome savaime suprantamais ir toliau neperrašome, bet nepamirškime, kad jie, be abejo, lieka galioti.

Išskleistu pavidalu (4) lygtys atrodo taip:

$$\begin{aligned} y_1 &= a_1^1(-x_1) + a_1^2(-x_2) + \cdots + a_1^k(-x_k) + \cdots + a_1^n(-x_n) + b_1, \\ y_2 &= a_2^1(-x_1) + a_2^2(-x_2) + \cdots + a_2^k(-x_k) + \cdots + a_2^n(-x_n) + b_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_r &= a_r^1(-x_1) + a_r^2(-x_2) + \cdots + a_r^k(-x_k) + \cdots + a_r^n(-x_n) + b_r, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m &= a_m^1(-x_1) + a_m^2(-x_2) + \cdots + a_m^k(-x_k) + \cdots + a_m^n(-x_n) + b_m. \\ \varphi &= -c^1(-x_1) - c^2(-x_2) - \cdots - c^k(-x_k) + \cdots - c^n(-x_n) + 0. \end{aligned} \qquad (5)$$

Dėl priežasčių, kurias netrukus išsiaiškinsim, (5) lygtyse išskleistu pavidalu parodome visą r-tąją lygtį ir k-tąjį nežinomąjį.

Simpleksinio metodo žingsniai atliekami transformuojant (5) lygtis. Norėdami sutrumpinti užrašymą, šias lygtis pavaizduojame dar dar vienu būdu – specialios lentelės pavidalu (žr. 1 lentelę). Šią lentelę vadinsime *simpleksine lentele* ir laikysime ją ekvivalentiška (5) lygčių sistemai. Taigi (4), (5) ir 1 lentelė yra tas pats, tik kitaip užrašyta.

Atkreipkime dėmesį, kad (4) lygtys leidžia mums lengvai pamatyti leistiną (2) uždavinio planą. Paėmę $\boldsymbol{x} = \boldsymbol{0}$, gauname, kad $\boldsymbol{y} = \boldsymbol{b}$, $\varphi = 0$. Gamybos planavimo uždavinyje toks planas reiškia, kad joks produktas negaminamas, nepanaudoti lieka visi ištekliai, o pelnas lygus nuliui.

1 lentelė

	$-x_1$	$-x_2$	…	$-x_k$	…	$-x_n$	
y_1	a_1^1	a_1^2	…	a_1^k	…	a_1^n	b_1
y_2	a_2^1	a_2^2	…	a_2^k	…	a_2^n	b_2
…	…	…	…	…	…	…	…
y_r	a_r^1	a_r^2	…	a_r^k	…	a_r^n	b_r
…	…	…	…	…	…	…	…
y_m	a_m^1	a_m^2	…	a_m^k	…	a_m^n	b_m
φ	$-c^1$	$-c^2$	…	$-c^k$	…	$-c^n$	0

1 lentelė yra (4) lygčių ekvivalentas, taigi joje šį planą taip pat galime perskaityti. Laikome, kad visų nežinomųjų, esančių viršutinėje lentelės eilutėje, t.y. $x_1,\ x_2, \dots, x_n$ reikšmės yra lygios nuliui (juos vadinsime *nebaziniais* nežinomaisiais). Lentelės kairiajame stulpelyje esančių nežinomųjų (juos vadinsime *baziniais*) reikšmės užrašytos dešiniajame jos stulpelyje, t.y. $y_1 = b_1,\ y_2 = b_2, \dots, y_m = b_m$, taip pat ir $\varphi = 0$. Taigi lentelėje fiksuotas pradinis planas, kuris, tikriausiai, neoptimalus: nieko negaminame, išteklių nenaudojame, pelno nėra. Bandykime rasti geresnį planą. Tuo tikslu transformuosime (5) lygčių sistemą ir ją atitinkančią 1 lentelę.

3⁰. Transformacijos. (5) lygčių sistemoje sukeiskime nežinomuosius x_k ir $\boldsymbol{y_r}$ vietomis: x_k nukelkime į kairę, o y_r į dešinę. Pradėkime nuo r-tosios lygties. Laikykim, kad $a_r^k \neq 0$. Tada

$$x_k = \frac{a_r^1}{a_r^k}(-x_1) + \frac{a_r^2}{a_r^k}(-x_2) + \dots + \frac{1}{a_r^k}(-y_r) + \dots + \frac{a_r^n}{a_r^k}(-x_n) + \frac{b_r}{a_r^k} \qquad (6)$$

Gautąją x_k išraišką įstatysime į visas likusias (5) lygtis. Imkim *i*-tąją lygtį ir atlikim veiksmus:

$$y_i = a_i^1(-x_1) + a_i^2(-x_2) + \dots + a_i^k(-x_k) + \dots + a_i^n(-x_n) + b_i =$$

$$= a_i^1(-x_1) + a_i^2(-x_2) + \dots + (-a_i^k)\left[\frac{a_r^1}{a_r^k}(-x_1) + \frac{a_r^2}{a_r^k}(-x_2) + \dots + \frac{1}{a_r^k}(-y_r) + ..\right.$$

$$\left. \cdot + \frac{a_r^n}{a_r^k}(-x_n) + \frac{b_r}{a_r^k}\right] + \dots + a_i^n(-x_n) + b_i$$

Sutvarkę panašius narius, gausim

$$y_i = \left(a_i^1 - \frac{a_i^k\, a_r^1}{a_r^k}\right)(-x_1) + \left(a_i^2 - \frac{a_i^k\, a_r^2}{a_r^k}\right)(-x_2) + \cdots + \left(-\frac{a_i^k}{a_r^k}\right)\left(-y_r\right) + \cdots + \left(a_i^n - \frac{a_i^k\, a_r^n}{a_r^k}\right)(-x_n) + (b_i - \frac{a_i^k\, b_r}{a_r^k}) \tag{7}$$

Atlikę analogiškus veiksmus su paskutiniąja (5) sistemos lygtimi, gausime

$$\varphi = \left(-c^1 - \frac{(-c^k)\, a_r^1}{a_r^k}\right)(-x_1) + \left(-c^2 - \frac{(-c^k)\, a_r^2}{a_r^k}\right)(-x_2) + \cdots + \left(-\frac{-c^k}{a_r^k}\right)(-y_r) + \cdots + \left(-c^n - \frac{(-c^k)\, a_r^n}{a_r^k}\right)(-x_n) + (0 - \frac{(-c^k)\, b_r}{a_r^k}) \tag{8}$$

Taigi, sukeitus x_k ir y_r vietomis, kitaip tariant, įvedus x_k į bazę, o y_r iš jos išvedus, (5) lygčių sistema virto (6)-(8) sistema. Ją kur kas patogiau užrašyti mums jau žinomos simpleksinės lentelės pavidalu (žr. 2 lentelę)

2 lentelė

	$-x_1$	$-x_2$	$\cdots$	$-y_r$	$\cdots$	$-x_n$	
y_1	$a_1^1 - \frac{a_1^k\, a_r^1}{a_r^k}$	$a_1^2 - \frac{a_1^k\, a_r^2}{a_r^k}$	$\cdots$	$-\frac{a_1^k}{a_r^k}$	$\cdots$	$a_1^n - \frac{a_1^k\, a_r^n}{a_r^k}$	$b_1 - \frac{a_1^k\, b_r}{a_r^k}$
y_2	$a_2^1 - \frac{a_2^k\, a_r^2}{a_r^k}$	$a_2^2 - \frac{a_2^k\, a_r^2}{a_r^k}$	$\cdots$	$-\frac{a_2^k}{a_r^k}$	$\cdots$	$a_2^n - \frac{a_2^k\, a_r^n}{a_r^k}$	$b_2 - \frac{a_2^k\, b_r}{a_r^k}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_k	$\frac{a_r^1}{a_r^k}$	$\frac{a_r^2}{a_r^k}$	$\cdots$	$\frac{1}{a_r^k}$	$\cdots$	$\frac{a_r^n}{a_r^k}$	$\frac{b_r}{a_r^k}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
y_m	$a_m^1 - \frac{a_m^k\, a_r^1}{a_r^k}$	$a_m^2 - \frac{a_m^k\, a_r^2}{a_r^k}$	$\cdots$	$-\frac{a_m^k}{a_r^k}$	$\cdots$	$a_m^n - \frac{a_m^k\, a_r^n}{a_r^k}$	$b_m - \frac{a_m^k\, b_r}{a_r^k}$
φ	$-c^1 - \frac{(-c^k)\, a_r^1}{a_r^k}$	$-c^2 - \frac{(-c^k)\, a_r^2}{a_r^k}$	$\cdots$	$-\frac{-c^k}{a_r^k}$	$\cdots$	$-c^n - \frac{(-c^k)\, a_r^n}{a_r^k}$	$0 - \frac{(-c^k)\, b_r}{a_r^k}$

Kaip ir anksčiau, priimkime, kad nebazinių (lentelės viršuje išvardintų) nežinomųjų reikšmės lygios nuliui. Įstatę šias reikšmes į (6)-(8) lygtis, gauname, kad bazinių (likusių nežinomųjų, išvardintų kairiajame lentelės stulpelyje), reikšmės yra lygios lentelės dešiniajame stulpelyje apskaičiuotiems dydžiams, o tikslo funkcijos reikšmė lygi dešiniajame apatiniame lentelės langelyje užrašytam dydžiui.

4^0. Simpleksinio metodo žingsniai. Matome, kad įvedę x_k į bazę, o y_r iš jos išvedę, t.y. transformavę 1 lentelę į 2 lentelę, gauname naują mūsų uždavinio planą su kitomis nežinomųjų ir tikslo funkcijos reikšmėmis. Pasinaudosime tuo, kad 2 lentelėje gautas planas būtų leistinas ir geresnis už tą, nuo kurio pradėjome 1 lentelėje. Kad to pasiektume, turime tinkamai parinkti įvedamą į bazę ir iš jos išvedamą nežinomuosius. Kaip tai padaryti?

Pirmiausia pažymėtina, kad keičiamų vietomis nežinomųjų numeriai k ir r turi būti parinkti taip, kad būtų $a_r^k > 0$. Priešingu atveju gautume, kad naujoji x_k reikšmė b_r/a_r^k (žr. 2 lentelę) būtų neigiama ar iš viso neegzistuotų (prisiminkim, kad $b_r \geq 0$ dėl 1^0 padarytos prielaidos). Neneigiamos turi būti ir visos kitos naujojo plano nežinomųjų reikšmės, (jos užrašytos 2 lentelės dešiniajame stulpelyje). Kitaip tariant, turi galioti

$$b_i - \frac{a_i^k\, b_r}{a_r^k} \geq 0,\ \ i = \overline{1,m}. \tag{9}$$

Ši sąlyga yra savaime patenkinta tiems numeriams i, kuriems $a_i^k \leq 0$. Tuo lengvai įsitikiname, atsiminę, kad $b_r/a_r^k \geq 0$, taigi pagal (9) formules prie neneigiamo b_i nagrinėjamu atveju faktiškai pridedamas neneigiamas dydis. Taigi (9) sąlygą galime perrašyti taip:

$$b_i - \frac{a_i^k\, b_r}{a_r^k} \geq 0, \text{ visiems } i,\ \text{ kuriems } a_i^k > 0\ . \tag{10}$$

Atlikę paprastus veiksmus, šią sąlygą perrašome taip:

$$\frac{b_r}{a_r^k} \leq \frac{b_i}{a_i^k} \quad \text{visiems } i,\ \text{kuriems } a_i^k > 0\ . \tag{11}$$

Ši sąlyga reiškia, kad naujasis planas bus leistinas tik tuo atveju, kai įvedamo į bazę nebazinio nežinomojo numeriui k išvedamo bazinio nežinomojo numeris r bus parinktas taip, kad santykis b_r/a_r^k būtų mažiausias iš santykių b_i/a_i^k, apskaičiuotų tiems baziniams numeriams i, kuriems $a_i^k > 0$.

Taigi, jei nuspręsime, kurį nebazinį nežinomąjį įvesti į bazę, tai išvedamo iš bazės nežinomojo numerį turėsime parinkti pagal šią taisyklę, kitaip gautas naujasis planas bus neleistinas (kai kurie nežinomieji įgis neigiamas reikšmes). Tačiau kurį gi nežinomąjį įvesti į bazę?

Iš (8) formulių, taip pat ir iš 2 lentelės matome, kad įvedus x_k į bazę, o y_r iš jos išvedus, tikslo funkcijos reikšmė pasikeičia dydžiu $-\frac{(-c^k)\, b_r}{a_r^k}$. Kadangi, kaip jau nusprendėme anksčiau, $b_r/a_r^k \geq 0$, tai tikslo funkcijos pokyčio ženklą nulemia dydis $-(-c^k)$. Jei norime, kad tikslo funkcijos reikšmė padidėtų (ar bent nesumažėtų), šis dydis turi būti teigiamas, vadinasi, $(-c^k)$ turi būti neigiamas. Dydžiai $(-c^1)$, $(-c^2), \ldots$, $(-c^n)$ surašyti apatinėje 1 lentelės eilutėje, taigi joje turime surasti neigiamą dydį, ir jį atitinkantis nežinomojo numeris k kaip tik ir bus to nežinomojo, kurį turime išvesti iš bazės, numeris. O parinkti vietoj jo į bazę įvedamo nežinomojo numerį r jau mokame.

Tačiau gali atsitikti taip, kad 1 lentelės apatinėje eilutėje visi dydžiai bus neneigiami. Tada iš (8) formulių ir anksčiau išdėstytų samprotavimų aišku, kad jokiu bazinio ir nebazinio nežinomųjų sukeitimu tikslo funkcijos nepadidinsime. Beliks pripažinti, kad lentelėje užrašytas planas jau optimalus.[4]

Gautus rezultatus užrašysime kelių simpleksinių taisyklių pavidalu.

<u>1 taisyklė.</u> Jei einamosios simpleksinės lentelės apatinėje eilutėje visi elementai neneigiami, lentelėje esantis atraminis planas optimalus. Jei šioje eilutėje yra neigiamų elementų, taikome 2 taisyklę.

<u>2 taisyklė</u>. Einamosios simpleksinės lentelės apatinėje eilutėje surandame neigiamą elementą. Jį atitinkantį stulpelį (sakykim, k-tąjį) pavadiname *vedančiuoju* stulpeliu. Pereinam prie 3 taisyklės.

<u>3 taisyklė.</u> Peržiūrim dešiniojo einamosios lentelės stulpelio ir vedančiojo stulpelio elementų santykius b_i/a_i^k su teigiamais vardikliais. Jei nė vieno tokio santykio nėra, uždavinio tikslo funkcija neaprėžta, ir jis dėl to neišsprendžiamas. Jei tokių santykių yra, išrenkam mažiausią iš jų, sakykim b_r/a_r^k. Lentelės eilutę, kurioje yra tas santykis (sakykim, r-tąją), pavadiname *vedančiąja* eilute, o vedančiosios eilutės ir vedančiojo stulpelio sankirtoje esantį elementą – *vedančiuoju elementu.* Pereiname prie 4 taisyklės.

[4] Šį faktą, tiesą sakant, reikėtų įrodyti griežtai. Iš pateiktų samprotavimų aišku tik tiek, kad vienu bazinio ir nebazinio nežinomųjų sukeitimu nieko nelaimėsime (t.y. tarp turimo atraminio plano „kaimyninių" atraminių planų nėra geresnio). Tačiau gal yra geresnių „nekaimyninių" ar ir visai neatraminių planu? Galima įrodyti, kad taip nėra, tačiau čia šio įrodymo nenagrinėsime.

4 taisyklė. Sudarome naują simpleksinę lentelę, kurioje nežinomasis su numeriu k bus įvestas į bazę, o nežinomasis su numeriu r išvestas iš bazės. Tuo tikslu naujoje lentelėje apskaičiuojame šiuos dydžius (žr. 2 lentelę):

a) vietoje vedančiojo elemento įrašom jam atvirkštinį dydį;
b) likusius vedančiosios eilutės elementus padalinam iš vedančiojo elemento;
c) likusius vedančiojo stulpelio elementus padalinam iš vedančiojo elemento ir pakeičiam ženklą priešingu;
d) likusius naujos lentelės elementus apskaičiuojame taip: iš kiekvieno ankstesnės lentelės elemento atimame jo eilutės vedančiojo stulpelio elemento bei jo stulpelio vedančiosios eilutės elemento sandaugą, padalintą iš vedančiojo elemento.

Naujai gautą lentelę laikome einamąja lentele ir pereiname prie 1 taisyklės.

Prieš pereidami prie konkretaus pavyzdžio sprendimo, padarykime keletą pastabų.

Formuluodami 3 taisyklę, mes tvirtinome, kad jei vedančiajame stulpelyje nebus nė vieno teigiamo elemento, tai mūsų uždavinio tikslo funkcija bus neaprėžta. Iš tikrųjų, pažiūrėję į (5) lygtis (kurių ekvivalentas yra 1 lentelė), matome, kad x_k tokiu atveju galėtume neribotai didinti, o visos kitų nežinomųjų reikšmės išliktų neneigiamos ir tik didėtų. Tuo tarpu tikslo funkcijos reikšmė dėl $(-c^k)$ neigiamumo taip pat be galo didėtų: $(-c^k)(-x_k) \to \infty$.

Gali atsitikti, kad apatinėje simpleksinės lentelės eilutėje visi elementai bus neneigiami, bet kai kurie iš jų bus lygūs nuliui. Tai, kaip įrodėme, netrukdo esantį lentelėje planą pripažinti optimaliu. Tačiau nulių buvimas rodo, kad turimas planas ne vienintelis optimalus. Iš tikrųjų, sakykim, $(-c^k) = 0$. Tada k-tąjį stulpelį laikydami vedančiuoju, atlikime simpleksinį žingsnį pagal 3 ir 4 taisykles. Gausime naują atraminį planą, kurio tikslo funkcijos reikšmė „padidės" dydžiu $(-c^k)b_r/a_r^k = 0$, kitaip tariant, liks ta pati. Taip gausime kitą optimalų atraminį planą. Tokią galimybę numatėme dar §3, nagrinėdami dvimačių uždavinių savybes.

Simpleksiniu metodu spręsdami tiesinio programavimo uždavinį, pagal aukščiau pagrįstas taisykles vieną po kitos skaičiuojame simpleksines lenteles, kol randame tokią, kurioje esantis atraminis planas bus optimalus (apatinės eilutės elementai neneigiami), arba paaiškės, kad tikslo funkcija neaprėžta (visas vedantysis stulpelis turės tik neteigiamus elementus). Tačiau ar turime pagrindo manyti, kad reikės suskaičiuoti baigtinį lentelių skaičių? Gal skaičiuosime tas lenteles vieną po kitos, ir vis negausime optimalaus plano? Kitaip tariant, gal simpleksinių žingsnių skaičius bus begalinis? Norėdami atsakyti į šį klausimą, atkreipkime dėmesį, kad kiekvienas simpleksinis žingsnis sukeičia vieną bazinį nežinomąjį su nebaziniu. Mūsų atveju turime m+n nežinomųjų, iš kurių m baziniai. Tačiau m bazinių nežinomųjų iš bendro m+n nežinomųjų skaičiaus galima parinkti tik baigtiniu skaičiumi skirtingų būdų. Vadinasi, konkrečiame uždavinyje galim turėti tik baigtinį skirtingų simpleksinių lentelių skaičių. Pereidami nuo vienos lentelės prie kitos, tikslo funkcijos reikšmės nemažiname. Jei kuriame žingsnyje tikslo funkcijos reikšmę griežtai padidiname, tai toliau prie jau praeitų lentelių nebegrįžtame (kitaip tikslo funkcijos reikšmė sumažėtų). Todėl žingsnis po žingsnio tarsi kopiame laipteliais (lentelėmis) aukštyn, o tų laiptelių skaičius, kaip matėme, baigtinis. Todėl ir simpleksinio metodo žingsnių skaičius baigtinis. „Užsiciklinti“, t.y. pakliūti į begalinį save kartojančių žingsnių ratą simpleksinis algoritmas gali tik vienu atveju: jei kuriame nors žingsnyje tikslo funkcijos reikšmė nepadidėja (dėl to, kad einamoji reikšmė $b_r = 0$). Jei ši reikšmė nepadidėja kitame ir dar kitame žingsnyje, tai gali būti, kad, apsukę ratą, grįšime prie tos pačios simpleksinės lentelės, ir vėl pradėsime tą patį ratą sukti iš pradžių (įrodyta, kad tokį ratą gali sudaryti ne mažiau kaip 4 žingsniai). Tačiau „užsiciklinimo“ pavojus tik teorinis – šiuolaikiniai tiesinio programavimo uždavinių sprendimo algoritmai nesunkiai jo išvengia.

5^0. Pavyzdys. Plytinės uždavinio sprendimas. Išspręsime plytinės uždavinį, suformuluotą §2 ir užrašytą priklausomybėmis (2.4). Pradėti turime nuo pradinės simpleksinės lentelės sudarymo. Kaip matome iš 1 lentelės, tai labai paprasta: viršuje su minuso ženklais surašome pagrindinius nežinomuosius (jų plytinės uždavinyje keturi: $x_1, x_{2'}, x_3, x_4$), lentelės kairėje surašome papildomus nežinomuosius y_1, y_2, y_3 (jų yra trys, nes uždavinys turi tiek apribojimų), į pačią lentelę surašome matricos A elementus, dešiniajame stulpelyje įrašome vektoriaus $\boldsymbol{b}$ elementus, o lentelės apačioje – tikslo funkcijos koeficientus su priešingais ženklais. Taip gauname pradinę simpleksinę 3a lentelę

3a lentelė

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	
y_1	2,000	3,000	4,000	3,000	780,00
y_2	1,000	4,000	5,000	1,000	850,00
y_3	3,000	4,000	2,000	2,000	790,00
	-50,00	-60,00	-70,00	-40,00	0,00

Šioje lentelėje užfiksuotas toks atraminis planas: $x_1 = x_2 = x_3 = x_4 = 0$ (nebazinių nežinomųjų reikšmės); $y_1 = 780$, $y_2 = 850$, $y_3 = 790$ (bazinių nežinomųjų reikšmės). Kitaip tariant, nė vienos rūšies plytų plytinė negamina, o visi ištekliai lieka nepanaudoti. Pelnas šiuo atveju lygus nuliui.

Pagal 1 taisyklę šis planas neoptimalus, nes apatinėje 3a lentelės eilutėje yra neigiamų elementų. Ieškosime geresnio atraminio plano. Pagal 2 taisyklę apatinėje šios lentelės eilutėje surandame neigiamą elementą. Galime rinktis bet kurį iš keturių, nes visi jie neigiami. Paprastai parenkamas mažiausias, šiuo atveju tai -70. Jį atitinkantį nežinomojo x_3 stulpelį pažymime kaip vedantįjį, o patį nežinomąjį x_3 įvesime į bazę (kitaip tariant, pradėsime gaminti „C“ tipo plytų blokus).

Pagal 3 taisyklę peržiūrėsime dešiniojo stulpelio ir vedančiojo stulpelio teigiamų elementų santykius 780:4, 850:5, 790:2. Mažiausias iš jų antrasis, taigi jį atitinkanti nežinomojo y_2 eilutė tampa vedančiąja, o vedančiuoju elementu $a_2^3 = 5$.

Pagal 4 taisyklę nežinomąjį x_3 įvesime į bazę, nežinomasis y_2 iš jos išvesime. Parodysime visus veiksmus, atliekamus pagal 4 taisyklę:

3b lentelė

	$-x_1$	$-x_2$	$-y_2$	$-x_4$	
y_1	2-4·1/5	3-4·4/5	-4/5	3-4·1/5	780-4·850/5
x_3	1/5	4/5	1/5	1/5	850/5
y_3	3-2·1/5	4-2·4/5	-2/5	2-2·1/5	790-2·850/5
	-50-1·(-70)/5	-60-4·(-70)/5	70/5	-40-1·(-70)/5	0-850·(-70)/5

Apskaičiavę šiuos dydžius, gauname 3c lentelę

3c lentelė

	$-x_1$	$-x_2$	$-y_2$	$-x_4$	
y_1	1,2000	-0,2000	-0,8000	2,2000	100,00
x_3	0,200	0,800	0,200	0,200	170,00
y_3	2,600	2,400	-0,400	1,600	450,00
	-36,000	-4,000	14,000	-26,000	11900,00

Šioje lentelėje fiksuotas toks atraminis planas: $x_1 = x_2 = y_2 = x_4 = 0$; $y_1 = 100$, $x_3 = 170$, $y_3 = 450$. Taigi gaminamos tik „C“ tipo plytos, ir jų gaminama $x_3 = 170$ blokų. Vandens ištekliai sunaudojami visiškai ($y_2 = 0$), darbo jėgos išteklių lieka nepanaudota $y_1 = 100$ val., o krosnies darbo valandų lieka nepanaudota y_3 = 450 val. Pelnas padidėjo iki 11900 litų. Tačiau šis planas pagal 1 taisyklę dar ne optimalus: apatinėje lentelės eilutėje yra neigiamų dydžių. Pavyzdžiui, pirmame stulpelyje esantis dydis -36 rodo, kad „A“ plytų bloko gamyba padidintų pelną 36 litais. Todėl pagal 2 taisyklę renkamės pirmąjį stulpelį vedančiuoju ir pagal 3 taisyklę peržiūrėkime dešiniojo ir vedančiojo stulpelio elementų santykius 100:1,2, 170:0,2, 450:2,6. Mažiausias iš jų pirmasis, taigi pirmoji eilutė vedančioji, o 1,2 – vedantysis elementas. Pagal 4 taisyklę atliekame simpleksinį žingsnį ir gauname 3d lentelę.

3d lentelė

	$-y_1$	$-x_2$	$-y_2$	$-x_4$	
x_1	0,8333	-0,1667	-0,6667	1,8333	83,33
x_3	-0,1667	0,8333	0,3333	-0,1667	153,33
y_3	-2,1667	2,8333	1,3333	-3,1667	233,33
	30,00	-10,00	-10,00	40,00	14900,00

Šioje lentelėje fiksuotas toks atraminis planas: $y_1 = x_2 = y_2 = x_4 = 0$; $x_1 = 83{,}33$ x_3 = 153,33, y_3 = 233,33. Taigi gaminamos „A“ ir „C“ tipo plytos, ir jų gaminama atitinkamai 83,33 ir 153,33 blokų. Darbo laiko ir vandens ištekliai sunaudojami visiškai ($y_1 = y_2 = 0$), krosnies darbo valandų lieka nepanaudota $y_3 = 233{,}33$ val. Pelnas padidėjo iki 14900 litų. Tačiau pagal 1 taisyklę šis planas dar ne optimalus: apatinėje lentelės eilutėje yra neigiamų dydžių. Pagal 2 taisyklę, vedančiuoju galime rinktis bet kurį iš juos atitinkančių stulpelių. Pasirenkame trečiąjį stulpelį ir pagal 3 taisyklę peržiūrimedešiniojo ir vedančiojo stulpelio teigiamų elementų santykius 153,33 : 0,33, 233,33 : 1,333. (Pirmoje eilutėje esantį santykį 83,33 : (-0,1667) ignoruojame, nes jo vardiklis neigiamas!) Mažiausias yra trečiojoje eilutėje esantis santykis, taigi ši eilutė vedančioji, o 1,333 – vedantysis elementas. Pagal 4 taisyklę atliekame simpleksinį žingsnį ir gauname 3e lentelę.

3e lentelė

	$-y_1$	$-x_2$	$-y_3$	$-x_4$	
x_1	-0,2500	1,2500	0,5000	0,2500	200,00
x_3	0,3750	0,1250	-0,2500	0,6250	95,00
y_2	-1,6250	2,1250	0,7500	-2,3750	175,00
	13,75	11,25	7,50	16,25	16650,00

Šioje lentelėje fiksuotas toks atraminis planas: $y_1 = x_2 = y_3 = x_4 = 0$; $x_1 = 200$, x_3 = 95, y_2 = 175. Taigi gaminamos „A“ ir „C“ tipo plytos, ir jų gaminama atitinkamai 200 ir 95 blokai. Darbo laiko ir krosnies laiko ištekliai sunaudojami visiškai ($y_1 = y_3 = 0$), vandens lieka nepanaudota $y_2 = 175$ m^3. Pelnas padidėjo iki 16650 litų. Kadangi apatinėje eilutėje visi dydžiai teigiami, pagal 1 taisyklę šis planas optimalus. Jis ir vienintelis optimalus, nes apatinėje eilutėje nėra nulių. Uždavinį išsprendėme. $\prec$

6^0. Pratimai. Išspręskite uždavinius, suformuluotus §2 3^0.

6^0.1 Rasti optimalų planą gamybos planavimo uždavinyje

$$\begin{gathered}
\textbf{(max) } 102x_1 + 130x_2 + 192x_3 + 100x_4 \\
7x_1 + 9x_2 + 12x_3 + 4x_4 \le 450, \\
8x_1 + 3x_2 + 2x_3 + 5x_4 \le 300, \\
4x_1 + 6x_2 + 9x_3 + 10x_4 \le 300. \\
x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 \ge 0.
\end{gathered}$$

6^0.2 Rasti optimalų planą gamybos planavimo uždavinyje

$$\begin{gathered}
\textbf{(max) } 54x_1 + 25x_2 + 40x_3 + 31x_4 \\
6x_1 + 4x_2 + 5x_3 + 2x_4 \le 1700, \\
8x_1 + 3x_2 + 6x_3 + 7x_4\ 2700, \\
9x_1 + 1x_2 + 5x_3 + 6x_4 \le 3000 \\
x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 \ge 0.
\end{gathered}$$

Abiejų uždavinių sprendimą simpleksinėmis lentelėmis rasite §9 5^0.

§ 7. ĮVAIRIŲ TIPŲ TIESINIO PROGRAMAVIMO UŽDAVINIŲ SPRENDIMAS

1^0. Neigiami apribojimų vektoriaus elementai. Praeitame paragrafe išsprendėme tiesinio programavimo uždavinį su m apribojimų-lygybių ir n nežinomųjų:

$$(\max)\ \varphi = \boldsymbol{cx},\quad A\boldsymbol{x} \le \boldsymbol{b},\quad \boldsymbol{x} \ge \boldsymbol{0} \tag{1}$$

(čia, kaip ir anksčiau A – $m \times n$ matrica, $\boldsymbol{b}$ – $m \times 1$ vektorius stulpelis, $\boldsymbol{c}$ – $1 \times n$ vektorius eilutė, o $\boldsymbol{x}$ – $n \times 1$ nežinomųjų vektorius stulpelis.

Šį uždavinį užrašėme ekvivalenčia forma (žr. §5), kurią gavome įvedę m papildomų neneigiamų nežinomųjų y_i, kurie sudaro $m \times 1$ papildomų nežinomųjų vektorių stulpelį $\boldsymbol{y}$.

$$(\max)\ \varphi = \boldsymbol{cx},\quad A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b},\quad \boldsymbol{x} \ge \boldsymbol{0}\ \boldsymbol{y} \ge \boldsymbol{0}. \tag{2}$$

Praeitame paragrafe pradėjome šį uždavinį spręsti nuo akivaizdaus plano $\boldsymbol{x} = \boldsymbol{0},$ $\boldsymbol{y} = \boldsymbol{b}$, $\varphi = 0$. Tačiau šis planas buvo leistinas tik tuo atveju, kai $\boldsymbol{b} \ge \boldsymbol{0}.$ Tačiau ką daryti, jei ši sąlyga nepatenkinta, t.y. vektorius $\boldsymbol{b}$ turi neigiamų koordinačių? Bandysime ir šiuo atveju remtis simpleksiniais žingsniais.

Planą $\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{y} = \boldsymbol{b}$ pavadinsime *pseudoplanu* (jis tenkina (2) apribojimus-lygybes, bet netenkina neneigiamumo sąlygų). Bandysime simpleksiniais žingsniais eiti nuo vieno pseudoplano prie kito, kol gausime leistiną planą, o tada, kaip jau mokame, eisime nuo vieno leistino plano prie kito, kol gausime optimalų.

Perrašykime mūsų uždavinį, kaip ir anksčiau, į simpleksinę lentelę (§6 1 lentelę). Joje bus atspindėtas pseudoplanas, kuriame bent viena iš papildomų nežinomųjų $\boldsymbol{y}$ reikšmių bus neigiama, sakykim, $y_t = b_t < 0.$ Atlikime simpleksinį žingsnį, siekdami, kad naujame pseudoplane ar plane y_t reikšmė padidėtų, o gal ir iš viso nebūtų neigiama.

Pradėkime nuo to, kad t-tojoje lentelės eilutėje suraskime neigiamą elementą $a_t^k < 0.$ Jei tokio elemento nerastume, uždavinio sprendimą turėtume nutraukti ir pripažinti, kad jis neišsprendžiamas dėl apribojimų prieštaringumo. Iš tikrųjų, t-toji lentelės eilutė atitinka lygtį

$$y_t = a_t^1(-x_1) + a_t^2(-x_2) + \cdots + a_t^k(-x_k) + \cdots + a_t^n(-x_n) + b_t.$$

Jei šioje lygtyje visi $a_t^j \ge 0, j = \overline{1,n},$ tai kokius neneigiamus x_j beimtume, dėl b_t neigiamumo y_t negalės įgyti neneigiamos reikšmės. Vadinasi, jokio leistino plano šiame uždavinyje nėra iš viso. Jei taip nėra, tai $a_t^k < 0$ pavyks surasti.

Laikykime *k*-tąjį stulpelį vedančiuoju (kitaip tariant, įveskime x_k į bazę). Jei vedančiuoju elementu parinktume patį a_t^k, tai y_t išeitų iš bazės, jo reikšmė pasidarytų lygi nuliui, o naujoji x_k reikšmė būtų $b_t/a_t^k > 0$ (žr. §6 taisykles). Taip atsikratytume vieno neigiamo pseudoplano elemento, ko ir siekiame. Tačiau toks žingsnis būtų skubotas. Mat, galėtų atsitikti, kad kitų nežinomųjų reikšmės, buvusios neneigiamos, taptų neigiamomis, ir vietoje žingsnio į priekį padarytume žingsnį atgal.

Todėl laikykime *k*-tąjį stulpelį vedančiuoju, bet vedančiąją eilutę rinkime taip, kad buvusios neneigiamos nežinomųjų reikšmės neneigiamomis ir išliktų. Pritaikydami §6 transformacijos taisykles (2 lentelė), šį reikalavimą galime užrašyti taip:

$$b_i - \frac{a_i^k\, b_r}{a_r^k} \geq 0, \text{ visiems } i \text{, kuriems } b_i \geq 0 \tag{3}$$

Vedančiąją r-tąją eilutę galime rinkti įvairiai. Paprastai ją renkame taip, kad galiotų $b_r/a_r^k \geq 0$. Priešingai, negu §6 nagrinėtu atveju, dabar mus domina ir atvejai, kai $b_r \leq 0$, $a_r^k < 0$. Bent vienas teigiamas santykis b_r/a_r^k tikrai yra (kai $r = t$). Ar šiuo atveju tikrai bus patenkinta (3) sąlyga? Iš (3) formulių matome, kad jei $a_i^k \leq 0$, tai b_i reikšmė nesumažėja, taigi, buvusi neneigiama, neneigiama ir lieka. Jei $a_i^k > 0$, tai, pertvarkę (3), matome, jog turi galioti $b_i/a_i^k \geq b_r/a_r^k$. Kitaip tariant, b_r/a_r^k turi būti mažiausias iš neneigiamų santykių b_i/a_i^k. Tada vietoje y_r į bazę įeis x_k su reikšme $b_r/a_r^k \geq 0$, o naujoji y_t reikšmė bus

$$y_t = b_t - \frac{a_t^k\, b_r}{a_r^k} \geq b_t \tag{4}$$

Jei taip atsitiktų, kad $r = t$, tai vedančiuoju elementu taptų pats, a_t^k, ką jau aptarėme. Taigi, atlikus simpleksinį žingsnį, neigiama y_t reikšmė išnyksta arba bent jau padidėja, likdama neigiama. Pastaruoju atveju atliekame dar vieną žingsnį ir t.t. Kadangi t eilutėje esančios koordinatės reikšmė didėja, tai prie buvusio pseudoplano nebegrįžtame, ir per baigtinį žingsnių skaičių arba atsikratome neigiamos pseudoplano koordinatės reikšmės arba paaiškėja uždavinio apribojimų nesuderinamumas (tam tikras „užsiciklinimo“ pavojus išlieka ir šiuo atveju – žr. §6 4^0 aptarimą). Taip panaikiname neigiamą pseudoplano reikšmę t eilutėje. Jei neigiamų reikšmių lieka kitose eilutėse, su jomis atliekame analogiškus veiksmus, kol mūsų pseudoplanas virsta leistinu planu. Tada pagal §6 suformuluotas taisykles ieškome optimalaus plano.

5 taisyklė. Peržiūrim dešiniojo einamosios lentelės stulpelio elementus. Jei jų tarpe yra neigiamų, pasirenkam kurį nors, sakykim, esantį t - toje eilutėje. Peržiūrim lentelės elementus a_t^j, esančius šioje eilutėje. Jei tarp jų nėra nė vieno neigiamo, uždavinys neišsprendžiamas, nes neturi jokio leistino plano. Priešingu atveju randame $a_t^k < 0$ ir k-tąjį stulpelį laikome vedančiuoju. Peržiūrime dešiniojo ir vedančiojo stulpelio elementų teigiamus santykius b_i/a_i^k, taip pat santykius, kur $b_i = 0,\ a_i^k > 0$ (jei tokių yra). Eilutę, kurioje yra mažiausias iš jų (sakykim, r-tąją) laikome vedančiąja, o elementą a_r^k – vedančiuoju elementu. Atliekame simpleksinį žingsnį pagal 4 taisyklę. Jei po šio žingsnio neigiama reikšmė dešiniojo stulpelio t - toje eilutėje neišnyko, šioje taisyklėje nurodytus veiksmus kartojame. Taip darome tol, kol dešiniajame lentelės stulpelyje nelieka neigiamų elementų. Pereiname prie 1 taisyklės.

2^0. Pavyzdys. Plytinės uždavinio su papildoma sąlyga - nelygybe sprendimas. Spręsime plytinės uždavinį, suformuluotą §2 ir užrašytą priklausomybėmis (2.4) su papildoma sąlyga (2.5). Primename, kad pagal šią sąlygą plytinei per 10 darbo dienų būtina pagaminti ne mažiau kaip 300 visų rūšių tipų plytų blokų, t.y.

$$x_1 + x_2 + x_3 + x_4 \geq 300.$$

Kadangi sprendimo taisykles suformulavome nelygybių "≤" atvejui, paversime šią nelygybę ekvivalentiška, padauginę ją iš -1:

$$-x_1 - x_2 - x_3 - x_4 \leq -300. \quad (5)$$

Kaip jau darėme §6, užpildome pradinę simpleksinę lentelę, atsižvelgdami į (5) sąlygą.

1a lentelė

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	
y_1	2,000	3,000	4,000	3,000	780,00
y_2	1,000	4,000	5,000	1,000	850,00
y_3	3,000	4,000	2,000	2,000	790,00
y_4	-1,000	-1,000	-1,000	-1,000	-300,00
	-50,000	-60,000	-70,000	-40,000	0,00

Šioje lentelėje užfiksuotas toks pseudoplanas: $x_1 = x_2 = x_3 = x_4 = 0$ (nebazinių nežinomųjų reikšmės); $y_1 = 780$, y_2 = 850, y_3 = 790, y_4 = -300 (bazinių nežinomųjų reikšmės). Kadangi dešiniajame lentelės stulpelyje yra neigiamas elementas, taikome 5 taisyklę.

Peržiūrime ketvirtąją lentelės eilutę, ieškodami joje neigiamų elementų. Jų yra bent keturi – visi lygūs -1. Renkamės bet kurį iš jų, sakykim, pirmąjį, todėl pirmąjį stulpelį laikome vedančiuoju. Peržiūrim dešiniojo ir vedančiojo stulpelio elementų santykius 780:2, 850:1, 790:3, (-300):(-1). Mažiausias iš jų trečiasis, taigi jį atitinkanti nežinomojo y_3 eilutė tampa vedančiąja, o vedančiuoju elementu $a_3^1 = 3$. Atliekame simpleksinį žingsnį. Gauname

1b lentelė

	$-y_3$	$-x_2$	$-x_3$	$-x_4$	
y_1	-0,6667	0,3333	2,6667	1,6667	253,33
y_2	-0,333	2,667	4,333	0,333	586,67
x_1	0,333	1,333	0,667	0,667	263,33
y_4	0,333	0,333	-0,333	-0,333	-36,67
	16,667	6,667	-36,667	-6,667	13166,67

Šioje lentelėje y_4 reikšmė -36,67 tebėra neigiama. Todėl ir vėl ketvirtojoje eilutėje turime ieškoti neigiamų elementų. Jie yra stulpeliuose, atitinkančiuose nežinomuosius x_3 ir x_4. Vedančiuoju galime rinktis bet kurį iš šių stulpelių. Pasirinkime x_4 atitinkantį stulpelį ir laikykime jį vedančiuoju. Peržiūrime santykius 253,33:1,6667 = 152; 586,67:0,333 = 1760; 263,33:0,667 = 395; (-36,67):(-0,333) = 110. Mažiausias iš jų paskutinis, taigi y_4 atitinkanti eilutė tampa vedančiąja. Vadinasi, nežinomasis y_4 bus išvestas iš bazės, ir naujame plane neigiamos reikšmės nebeliks. Atlikę simpleksinį žingsnį, gauname lentelę

1c lentelė

	$-y_3$	$-x_2$	$-x_3$	$-y_4$	
y_1	1,0000	2,0000	1,0000	5,0000	70,0
y_2	0,000	3,000	4,000	1,000	550,0
x_1	1,000	2,000	0,000	2,000	190,0
x_4	-1,000	-1,000	1,000	-3,000	110,0
	10,000	0,000	-30,000	-20,000	13900,0

Šioje lentelėje jau turime leistiną planą (nebe pseudoplaną) $y_3 = x_2 = x_3 = y_4 = 0$ (nebazinių nežinomųjų reikšmės); $y_1 = 70$, y_2 = 550, x_1 = 190, x_4 = 110 (bazinių nežinomųjų reikšmės). Tikslo funkcijos reikšmė 13900. Iš 1 taisyklės žinome, kad ši reikšmė neoptimali, nes apatinėje lentelės eilutėje yra neigiamų elementų. Todėl atliekame įprastą simpleksinį žingsnį. Vedančiuoju pasirinksim nežinomąjį x_3 atitinkantį stulpelį. Peržiūrėsim dešiniojo ir atitinkančio x_3 stulpelių elementų santykius 70:1 = 70; 550:4 = 137,5; 110:1 = 110. Mažiausias pirmasis, vadinasi, atitinkanti eilutė bus vedančioji. Atliekame simpleksinį žingsnį. Gauname tokią lentelę

1d lentelė

	$-y_3$	$-x_2$	$-y_1$	$-y_4$	
x_3	1,00	2,00	1,00	5,00	70,0
y_2	-4,00	-5,00	-4,00	-19,00	270,0
x_1	1,00	2,00	0,00	2,00	190,0
x_4	-2,00	-3,00	-1,00	-8,00	40,0
	40,00	60,00	30,00	130,00	16000,00

Šioje lentelėje pagaliau matome optimalų planą: $y_3 = x_2 = y_1 = y_4 = 0$; $x_3 = 70$, $x_1 = 190$, $x_4 = 40$ $y_2 = 270$. Taigi gaminamos „A“ , „C“ ir „D“ tipo plytos, ir jų gaminama atitinkamai 190, 70 ir 40 blokų. Darbo laiko ir krosnies laiko ištekliai sunaudojami visiškai ($y_1 = y_3 = 0$), vandens lieka nepanaudota $y_2 = 270$ m^3. Pelnas lygus 16000 litų ir, lyginant su ankstesniu §6 išspręstu variantu, 650 litų mažesnis. $\prec$

3^0. Apribojimai - lygybės. Iki šiol sprendėme standartinius tiesinio programavimo uždavinius (jų apribojimai yra " ≤ " tipo nelygybės). Tačiau praktikoje gali iškilti ir uždavinių, kurių apribojimai yra griežtos lygybės. Sakykim, (1) uždavinio t – toji apribojimų eilutė yra

$$a_t^1 x_1 + a_t^2 x_2 + \dots + a_t^k x_k + \dots + a_t^n x_n = b_t. \tag{6}$$

Laikykime, kad $b_t \geq 0$. Jei konkrečiame uždavinyje būtų priešingai, visą (6) lygybę, nemažindami nagrinėjimo bendrumo, padaugintume iš -1. Laikykim, kad toks veiksmas, jei reikia, jau atliktas.

Norėdami pritaikyti anksčiau nagrinėtą sprendimo būdą, kaip ir anksčiau, įveskime papildomą nežinomąjį y_t . Gausime

$$a_t^1 x_1 + a_t^2 x_2 + \dots + a_t^k x_k + \dots + a_t^n x_n + y_t = b_t,$$

$$y_t = a_t^1(-x_1) + a_t^2(-x_2) + \dots + a_t^k(-x_k) + \dots + a_t^n(-x_n) + b_t. \tag{7}$$

Čia y_t yra *dirbtinis nežinomasis*, kurio reikšmė leistiname plane turi būti lygi nuliui, kitaip (6) lygybė nebus patenkinta. Jį įvedėme tam, kad lengvai rastume pradinį planą simpleksiniams žingsniams pradėti. Iš tikrųjų, kaip ir anksčiau, laikykime, kad pradedam nuo plano $\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{y} = \boldsymbol{b}$ (įskaitant ir $y_t = b_t$). Šis planas nėra leistinas, jei $y_t = b_t > 0$. Todėl simpleksiniais žingsniais bandysim dirbtinį nežinomąjį y_t išvesti iš bazės, kad jo reikšme taptų lygia nuliui, ir turėtume leistiną planą.

Lygtis (7) yra viena iš simpleksinės lentelės eilučių. Peržiūrime koeficientus a_t^j $j = \overline{1, n}$. Jei jų tarpe nėra nė vieno teigiamo, tai $b_t > 0$ atveju uždavinys neturi nė vieno leistino plano. Iš tikrųjų, iš (7) lygčių matome, kad, kokius neneigiamus x_j berinktume, y_t reikšmė išliks teigiama, o leistiname plane ji privalo būti lygia nuliui. Tokiu atveju uždavinio sprendimą nutrauksime dėl jo apribojimų nesuderinamumo.

Jei bent vienas teigiamas $a_t^k > 0$ yra, k-tąjį stulpelį laikome vedančiuoju. Jei vedančiąja imtume t – tąją eilutę, tai y_t išeitų iš bazės, o x_k įeitų su reikšme $b_t / a_t^k > 0$ (žr. §6 taisykles). Taip iš karto pasiektume tikslą, kad y_t reikšmė būtų nulis. Tačiau neatsižvelgę į kitus nežinomuosius, galime vietoj buvusių jų neneigiamų reikšmių gauti neigiamas. Todėl vedančiąją eilutę rinksime atsargiau. Kaip ir anksčiau, peržiūrėsime dešiniojo ir vedančiojo stulpelio santykius b_i / a_i^k, kuriems $b_i \geq 0$, $a_i^k > 0$ (bent vienas toks santykis yra, būtent, kai $i = t$). Iš jų išrinksime mažiausią b_r / a_r^k , o r-tąją eilutę laikysime vedančiąja. Nesunku įsitikinti, kad, atlikus simpleksinį žingsnį, neneigiamų nežinomųjų reikšmės išliks neneigiamos, o y_t reikšmė sumažės. Iš tikrųjų, kadangi $b_r / a_r^k \leq b_i / a_i^k$, tai naujoji i-tojo nežinomojo reikšmė $b_i - b_r a_i^k / a_r^k \geq 0$. Naujoji y_t reikšmė sumažėja: $b_t - b_r a_t^k / a_r^k \leq b_t$. Kartodami veiksmus su dirbtinio nežinomojo eilute, žingsnis po žingsnio mažiname dirbtinio nežinomojo reikšmę, kol jį pašaliname iš bazės. Galime išbraukti ir visą šį nežinomojo stulpelį. Tada tęsiame veiksmus pagal jau žinomas taisykles, bet dirbtiniam nežinomajam nebeleidžiame grįžti į bazę (jei jo stulpelio neišbraukėme).

<u>6 taisyklė.</u> Peržiūrim dešiniojo einamosios lentelės stulpelio elementus. Jei jų tarpe yra dirbtinių nežinomųjų reikšmių, pasirenkam kurią nors, sakykim, t - tąją eilutę, atitinkančią dirbtinį nežinomąjį. Peržiūrim lentelės elementus a_t^j, esančius šioje eilutėje. Jei tarp jų nėra nė vieno teigiamo, uždavinys neišsprendžiamas, nes neturi jokio leistino plano. Priešingu atveju randame $a_t^k > 0$ ir k-tąjį stulpelį laikome vedančiuoju. Peržiūrime dešiniojo ir vedančiojo stulpelio

elementų teigiamus santykius b_i/a_i^k, kur $b_i \geq 0,\ a_i^k > 0$. Eilutę, kurioje yra mažiausias iš jų (sakykim, r-tąją) laikome vedančiąja, o elementą a_r^k – vedančiuoju elementu. Atliekame simpleksinį žingsnį pagal 4 taisyklę. Jei po šio žingsnio y_t dar neišėjo iš bazės ($r \neq t$), šioje taisyklėje nurodytus veiksmus kartojame. Taip darome tol, kol dirbtiniai nežinomieji pašalinami iš bazės. Pereiname prie 5 taisyklės (jei turime pseudoplaną) arba 1 taisyklės, jei turime leistiną planą.

4^0. Pavyzdys. Plytinės uždavinio su papildoma sąlyga - lygybe sprendimas. Dar kartą spręsime plytinės uždavinį, suformuluotą §2 ir užrašytą priklausomybėmis (2.4). Tačiau dabar pareikalausime, kad "A" ir "D" plytų blokų būtų pagaminta 230 vienetų. Tai reiškia, kad į mūsų uždavinį turime įtraukti sąlygą

$$x_1 + x_4 = 230. \tag{8}$$

Kaip jau darėme §6, užpildome pradinę simpleksinę lentelę, atsižvelgdami į (8) sąlygą.

2a lentelė

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	
y_1	2,000	3,000	4,000	3,000	780,00
y_2	1,000	4,000	5,000	1,000	850,00
y_3	3,000	4,000	2,000	2,000	790,00
y_4	1,000	0,000	0,000	1,000	230,00
	-50,000	-60,000	-70,000	-40,000	0,00

Šioje lentelėje užfiksuotas toks planas su dirbtiniu nežinomuoju y_4 : $x_1 = x_2 = x_3 = x_4 = 0$ (nebazinių nežinomųjų reikšmės); $y_1 = 780$, $y_2 = 850$, $y_3 = 790$, $y_4 = 230$ (bazinių nežinomųjų, tarp jų ir dirbtinio, reikšmės). Kadangi dirbtinis nežinomasis yra bazinis, taikome 6 taisyklę.

Peržiūrim ketvirtosios eilutės elementus, ieškodami teigiamo. Kadangi tokių elementų yra du, renkames bet kurį, sakykim, pirmąjį. Tada nežinomąji x_1 atitinkantis stulpelis tampa vedančiuoju. Peržiūrim dešiniojo ir vedančiojo stulpelio elementų santykius: 780:2 = 390; 850:1 = 850; 790:3 = 263,3 ; 230:1 = 230. Mažiausias iš jų paskutinis, taigi jį atitinkanti dirbtinio nežinomojo y_4 eilutė tampa vedančiąja, o vedančiuoju elementu $a_4^1 = 1$. Atliekame simpleksinį žingsnį. Gauname lentelę

2b lentelė

	$-y_4$	$-x_2$	$-x_3$	$-x_4$	
y_1	-2,000	3,000	4,000	1,000	320,00
y_2	-1,000	4,000	5,000	0,000	620,00
y_3	-3,000	4,000	2,000	-1,000	100,00
x_1	1,000	0,000	0,000	1,000	230,00
	50,000	-60,000	-70,000	10,000	11500,00

Šioje lentelėje dirbtinis nežinomasis jau išvestas iš bazės, todėl turime leistiną planą. Dirbtinį nežinomąjį atitinkantį stulpelį galima būtų išbraukti iš lentelės, nes y_4 jau nebegali grįžti į bazę. Likusioje lentelės dalyje taikome įprastas simpleksines taisykles. Apatinėje eilutėje turime neigiamų elementų, todėl planas dar ne optimalus. Vedančiuoju pasirenkame nežinomąjį x_3 atitinkantį stulpelį. Peržiūrime dešiniojo ir vedančiojo stulpelio elementų santykius: 320:4 = 80; 620:5 = 124; 100:2 = 50. Mažiausias trečiasis, todėl vedančiąja laikome trečiąją eilutę. Atlikę simpleksinį žingsnį, gauname tokią lentelę:

2c lentelė

	$-y_4$	$-x_2$	$-y_3$	$-x_4$	
y_1	4,000	-5,000	-2,000	3,000	120,00
y_2	6,500	-6,000	-2,500	2,500	370,00
x_3	-1,500	2,000	0,500	-0,500	50,00
x_1	1,000	0,000	0,000	1,000	230,00
	-55,000	80,000	35,000	-25,000	15000,00

Šioje lentelėje turimas planas dar ne optimalus, kadangi apatinėje eilutėje dar yra neigiamas elementas -25 (į neigiamą dydį –55 nebekreipiame dėmesio, nes y_4 į bazę grįžti negali). Vedančiuoju laikome nežinomąjį x_4 atitinkantį stulpelį ir peržiūrime santykius 120:3 = 40; 370:2,5 = 148; 230:1 = 230. Mažiausias iš jų pirmasis, taigi pirmoji eilutė ir bus vedančioji. Atlikę simpleksinį žingsnį, gauname lentelę

2d lentelė

	$-y_4$	$-x_2$	$-y_3$	$-y_1$	
x_4	1,333	-1,667	-0,667	0,333	40,0
y_2	3,167	-1,833	-0,833	-0,833	270,0
x_3	-0,833	1,167	0,167	0,167	70,0
x_1	-0,333	1,667	0,667	-0,333	190,0
	-21,667	38,333	18,333	8,333	16000,0

Šioje lentelė jau turime optimalų planą: apatinės eilutės elementai neneigiami, o į dirbtinio nežinomojo stulpelį nekreipiame dėmesio. Taigi "A" tipo plytų gaminame x_1 = 190 blokų, "C" tipo plytų gaminame x_3 = 70 blokų, "D" tipo plytų gaminame x_4 = 40 blokų, "B" tipo plytų negaminame. Lieka nepanaudota y_2 = 270 m^3 vandens išteklių. ∢

5^0. Pratimas. Išspręskite simpleksinėmis lentelėmis mišinių uždavinį, suformuluotą §2 3^0.

$$\textbf{(min) } 33x_1 + 10x_2 + 26x_3 + 18x_4$$

$$50x_1 + 30x_2 + 80x_3 + 10x_4 \geq 190,$$

$$40x_1 + 10x_2 + 20x_3 + 20x_4 \geq 110,$$

$$50x_1 + 40x_2 + 50x_3 + 30x_4 \geq 150.$$

$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0.$$

§ 8. DUALŪS TIESINIO PROGRAMAVIMO UŽDAVINIAI

Nagrinėdami tiesinio programavimo priemonėmis modeliuojamus uždavinius, mes iki šiol labiau domėjomės jų materialiu aspektu – sprendėme, kiek ir kokių produktų gaminti, užteks, ar ne turimų išteklių ir pan. Dabar bandysime išnagrinėti kai kuriuos šių uždavinių vertinius aspektus.

1°. Dualus gamybos planavimo uždavinys. Prisiminkime gamybos planavimo uždavinio, išnagrinėto §2, sąlygą. Nagrinėjome įmonę, kuri gali gaminti n skirtingų produktų rūšių, naudodama m išteklių rūšių. Visų m išteklių rūšių apimtis laikome žinomomis ir žymime $b_1, b_2, \dots, b_m$. Be to, laikome žinomomis visų išteklių sąnaudas, reikalingas kiekvieno produkto vienetui pagaminti: a_i^j reikšmė parodo, kiek i-tojo ištekliaus vienetų reikia j-tojo produkto vienetui pagaminti. Naudingumo matu gamybos planavimo uždavinyje laikome pelną: sakysime, kad j-tojo produkto vieneto gamyba duoda c^j litų pelno, o kiekvieno produkto pelningumas žinomas. Matematinis šio uždavinio kaip tiesinio programavimo uždavinio formulavimas išskleistu pavidalu užrašytas priklausomybėmis (2.1) – (2.3), o matriciniu pavidalu

$$\text{(max)}\ \boldsymbol{cx},\quad A\boldsymbol{x} \le \boldsymbol{b},\quad \boldsymbol{x} \ge \boldsymbol{0}. \tag{1}$$

Bandykime apskaičiuoti, kiek įmonei verti jos naudojami ištekliai. Remsimės tokiu sumetimu: kiekvieną iš išteklių įmonė vertina tiek, už kokią mažiausią kainą be nuostolių sutiktų jo atsisakyti. Šiek tiek užbėgdami į priekį, galime pasakyti, kad tai bus ta pati kaina, už kurią papildomą išteklių vienetą dar bus verta pirkti.

Pažymėkime minėtą i-tosios išteklių rūšies įvertinimą (kainą) p^i $(i = \overline{1,m})$ ir vadinkime jį *dualia kaina.* Visų dualių kainų vektorių-eilutę žymėsime $\boldsymbol{p} = (p^1, p^2, \dots, p^m)$.

Įvertinkime dualią kainą visų išteklių, reikalingų j-tojo produkto vienetui pagaminti. Pirmosios rūšies išteklių šiam tikslui reikia a_1^j vienetų taigi jų duali kaina $p^1 a_1^j$, antrosios rūšies išteklių reikia a_2^j vienetų taigi jų duali kaina $p^2 a_2^j$ ir t.t. Taigi visų išteklių, reikalingų j-tojo produkto vienetui pagaminti, duali kaina yra lygi $\boldsymbol{pa}^j = p^1 a_1^j + p^2 a_2^j + \cdots p^m a_m^j$. Kadangi, panaudojus šiuos išteklius, pagaminamas vienetas j-tojo produkto, kurio pelningumas yra c^j, tai atsisakyti šių išteklių įmonei verta už sumą, ne mažesnę, negu c^j. Taigi dualios kainos turi būti pasiūlytos tokios, kad galiotų nelygybės

$$p^1 a_1^j + p^2 a_2^j + \cdots p^m a_m^j \geq c^j \quad (j = \overline{1,n}\,), \tag{2}$$

arba, užrašius tą patį trumpiau,

$$\boldsymbol{p}\boldsymbol{a}^j \geq c^j \quad (j = \overline{1,n}\,), \tag{3}$$

arba visai trumpai

$$\boldsymbol{p}A \geq \boldsymbol{c}. \tag{4}$$

Taigi dualios kainos $\boldsymbol{p}$ turi tenkinti (4) apribojimus, be to, jos pagal ekonominę prasmę turi būti neneigiamos: $\boldsymbol{p} \geq \boldsymbol{0}$.

Atkreipkime dėmesį, kad (4) nelygybės leidžia $\boldsymbol{p}$ neribotai didinti (jei bent vienas $\boldsymbol{p}$, tenkinantis (4), egzistuoja). Tai reiškia, kad įmonė gali atsisakyti išteklių ir už dvigubai, ir už trigubai ir dar didesnes kainas. Tačiau tikrą išteklių įvertinimą parodys minimalios kainos, už kurias vis dar verta atsisakyti išteklių. Jų vertė šiomis kainomis yra $\boldsymbol{p}\boldsymbol{b} = p^1 b_1 + p^2 b_2 + \cdots + p^m b_m$. Taigi dualus gamybos planavimo uždavinys formuluojamas taip:

$$\textbf{(min)}\, \boldsymbol{p}\boldsymbol{b} \quad \boldsymbol{p}A \geq \boldsymbol{c}, \ \boldsymbol{p} \geq \boldsymbol{0}. \tag{5}$$

Uždavinį (1) vadinsime *tiesioginiu,* o (5) – *dualiu* tiesinio programavimo uždaviniu, $\boldsymbol{x}$ - *tiesioginiais*, $\boldsymbol{p}$ - *dualiais* nežinomaisiais. Šie pavadinimai remiasi matematiniu dualių uždavinių formulavimu. Ekonominiuose taikymuose $\boldsymbol{x}$ dažnai vadinamas planu, $\boldsymbol{p}$ - dualiomis arba „šešėlinėmis“ kainomis (angl. *„shadow prices“).*

Kaip ir tiesioginio uždavinio atveju, dualaus uždavinio (5) *leistinu planu* vadinsime m-matį vektorių-eilutę $\boldsymbol{p}$, tenkinantį apribojimus $\boldsymbol{p}A \geq \boldsymbol{c}$, $\boldsymbol{p} \geq \boldsymbol{0}$. Šio uždavinio *optimaliu planu* arba *sprendiniu* vadinsime leistiną planą $\boldsymbol{p}^*$, kuris tenka nelygybę $\boldsymbol{p}^*\boldsymbol{b} \leq \boldsymbol{p}\boldsymbol{b}$ visiems leistiniems $\boldsymbol{p}$.

Pavyzdys. Nagrinėkime plytinės uždavinį, suformuluotą §2. Bandykime apskaičiuoti, kiek plytinei verta viena darbininko darbo valanda (žymėsime šį dydį p^1), vienas kubinis metras vandens (p^2) ir viena krosnies darbo valanda (p^3). Įvertinsime išteklius, reikalingus vienam "A" tipo plytų blokui pagaminti. Pagal §2 1 lentelės duomenis, matome, kad darbininkų darbo laikas bus įvertintas dydžiu $2p^1$, vandens sąnaudos – dydžiu $1p^2$, o krosnies darbo valandos – dydžiu

$3p^3$. Taigi visi ištekliai bus įvertinti suma $2p^1 + 1p^2 + 3p^3$, ir ši suma turės būti ne mažesnė, kaip "A" blokų gamybos pelningumas, t.y. 50 litų. Todėl užrašome nelygybę

$$2p^1 + 1p^2 + 3p^3 \geq 50.$$

Analogiškas nelygybes užrašę ir kitų tipų plytų blokams bei atsižvelgę į neneigiamumo apribojimus, gausime tokią dualaus plytinės uždavinio apribojimų sistemą:

$$\begin{aligned} &2p^1 + 1p^2 + 3p^3 \geq 50, \\ &3p^1 + 4p^2 + 4p^3 \geq 60, \\ &4p^1 + 5p^2 + 2p^3 \geq 70, \\ &3p^1 + 1p^2 + 2p^3 \geq 40. \\ &p^1 \geq 0, p^2 \geq 0, p^3 \geq 0. \end{aligned} \tag{6}$$

Tačiau mums svarbu rasti ne bet kokias kainas, kuriomis ištekliai būtų įvertinti ne mažiau, negu iš jų gaunamas pelnas, o *mažiausias* tokias kainas. Visi per nagrinėjamą laikotarpį plytinės gamyboje sunaudoti ištekliai šiomis kainomis verti $780p^1 + 850p^2 + 790p^3$, ir šį dydį turime minimizuoti. Taigi užrašome sąlygą

$$\text{(min)} \; 780p^1 + 850p^2 + 790p^3. \tag{7}$$

Gautas (6) – (7) uždavinys yra dualus §2 (4) priklausomybėmis suformuluotam tiesioginiam uždaviniui. Jį išsprendę, galėsime objektyviai įvertinti plytinės naudojamų išteklių naudingumą. ⋖

2°. Dualių uždavinių savybės. Prieš pradėdami spręsti dualų uždavinį, išsiaiškinsime svarbiausias jo savybes, kurios mums padės atskleisti dualių nežinomųjų prasmę ir praktinę reikšmę.

<u>1 savybė</u>. Tegu $\boldsymbol{x}$ - tiesioginio, o $\boldsymbol{p}$ - dualaus uždavinio leistini planai. Tada $\boldsymbol{cx} \leq \boldsymbol{pb}$.

Įrodymas. Kadangi $\boldsymbol{x}$ - leistinas, tai pagal (1) užrašome $A\boldsymbol{x} \leq \boldsymbol{b}$. Padauginę šią nelygybę iš kairės iš leistino (5) plano $\boldsymbol{p} \geq \boldsymbol{0}$, gauname $\boldsymbol{p}A\boldsymbol{x} \leq \boldsymbol{pb}$. Kadangi $\boldsymbol{p}$ - leistinas, tai pagal (5) užrašome $\boldsymbol{p}A \geq \boldsymbol{c}$. Padauginę šią nelygybę iš dešinės iš leistino (1) plano $\boldsymbol{x} \geq \boldsymbol{0}$, gauname $\boldsymbol{p}A\boldsymbol{x} \geq \boldsymbol{cx}$. Sulyginę gautas nelygybes, matome, kad $\boldsymbol{cx} \leq \boldsymbol{p}A\boldsymbol{x} \leq \boldsymbol{pb}$. ⋖

Gautoji nelygybė parodo, kad gaunamas pelnas ne didesnis, negu verti ištekliai, kurie panaudoti jam gauti; kita vertus, objektyviai įvertinti ištekliai visada verti ne mažiau, negu pelnas, kurį iš jų galima gauti.

2 savybė. Tegu $\boldsymbol{x}$ - tiesioginio, o $\boldsymbol{p}$ - dualaus uždavinio leistini planai, ir bent vienas jų neoptimalus. Tada $\boldsymbol{cx} < \boldsymbol{pb}$.

Įrodymas. Tegu $\boldsymbol{x}$ – neoptimalus. Tai reiškia, kad egzistuoja geresnis leistinas (1) uždavinio planas, sakykim, $\widetilde{\boldsymbol{x}}$, toks, kad $\boldsymbol{c}\widetilde{\boldsymbol{x}} > cx$**.** Tačiau dėl 1 savybės $\boldsymbol{c}\widetilde{\boldsymbol{x}} \leq \boldsymbol{pb}$**,** todėl $\boldsymbol{cx} < pb$**.** Analogiškai samprotautume ir tuo atveju, jei $\boldsymbol{p}$ būtų neoptimalus. ≺

Pagal įrodytąją nelygybę, jei gamybos planas neoptimalus, tai jo duodamas pelnas mažesnis, nei verti ištekliai, sunaudoti jam įgyventi. Ir atvirkščiai – jei ištekliai įvertinti neoptimaliai, tai jie pervertinti – t.y. įvertinti daugiau, nei, juos optimaliai panaudojus, galima gauti pelno.

3 savybė. Tegu $\boldsymbol{x}^*$ - tiesioginio, o $\boldsymbol{p}^*$ - dualaus uždavinio leistini planai, ir $\boldsymbol{cx}^* = \boldsymbol{p}^*\boldsymbol{b}$. Tada jie yra kiekvienas savo uždavinio optimalūs planai.

Įrodymas. Imkim bet kokį leistiną tiesioginio uždavinio planą $\boldsymbol{x}$**.** Pagal 1 savybę jam ir dualaus uždavinio planui $\boldsymbol{p}^*$ galime užrašyti nelygybę $\boldsymbol{cx} \leq \boldsymbol{p}^*\boldsymbol{b}$. Tačiau $\boldsymbol{p}^*\boldsymbol{b} = \boldsymbol{cx}^*$, taigi $\boldsymbol{cx} \leq \boldsymbol{cx}^*$, o tai ir reiškia, kad bet kokio leistino tiesioginio uždavinio plano tikslo funkcijos reikšmė ne didesnė, nei $\boldsymbol{cx}^*$. Taigi $\boldsymbol{x}^*$optimalus.

Analogiškai samprotaujame ir dėl dualaus uždavinio. Imkim bet kokį leistiną dualaus uždavinio planą $\boldsymbol{p}$**.** Pagal 1 savybę jam ir planui $\boldsymbol{x}^*$ galime užrašyti nelygybę $\boldsymbol{cx}^* \leq \boldsymbol{pb}$. Tačiau $\boldsymbol{cx}^* = \boldsymbol{p}^*\boldsymbol{b}$, taigi $\boldsymbol{p}^*\boldsymbol{b} \leq \boldsymbol{pb}$, o tai ir reiškia, kad bet kokio leistino dualaus uždavinio plano tikslo funkcijos reikšmė ne mažesnė, nei $\boldsymbol{p}^*\boldsymbol{b}$. Taigi $\boldsymbol{p}^*$optimalus. ≺

4 savybė. Jei vieno iš dualių uždavinių tikslo funkcija neaprėžta, tai kitas neturi leistinų planų.

Įrodymas. Pagal 1 savybę žinome, kad leistinus tiesioginio ir dualaus uždavinio planus sieja nelygybė $\boldsymbol{cx} \leq \boldsymbol{pb}$. Jei $\boldsymbol{cx}$ galėtų būti neaprėžtai didelis, tai neatsirastų jokio $\boldsymbol{p}$, kad $\boldsymbol{pb}$ vis tiek viršytų kaip norima didelį dydį. Ir atvirkščiai, jei $\boldsymbol{pb}$ galėtų neaprėžtai mažėti, tai neatsirastų jokio $\boldsymbol{x}$, kad $\boldsymbol{cx}$ vis tiek būtų mažesnis už į minus begalybę artėjantį dydį. ≺

Dvi paskutiniosios savybės verčia susimąstyti ties svarbiais klausimais: ar išsprendžiamas (turi optimalų planą) vienas iš dualių uždavinių, jei žinome, kad kitas išsprendžiamas arba žinome, kad jis neišsprendžiamas? Jei išsprendžiami abu uždaviniai, tai ar būtinai jų optimalius planus sieja lygybė $\boldsymbol{cx}^* = \boldsymbol{p}^*\boldsymbol{b}$? Pagal 3 savybę žinome, kad ši lygybė pakankama planų optimalumui. Bet ar ji būtina? Gal optimaliems planams galioja nelygybė $\boldsymbol{cx}^* < \boldsymbol{p}^*\boldsymbol{b}$? 2 savybė mums sako tik tiek, kad ši nelygybė galioja neoptimaliems planams, bet gal ji galioja ir optimaliems? Į šiuos klausimus atsako labai svarbi tiesinio programavimo teorijai teorema.

Pirmoji dualumo teorema. Jei išsprendžiamas vienas iš dualių tiesinio programavimo uždavinių, tai išsprendžiamas ir kitas. Optimalios tikslo funkcijų reikšmės abiejuose uždaviniuose sutampa.

Šios teoremos įrodymas gerokai sudėtingesnis už mūsų nagrinėtus savybių įrodymus ir remiasi fundamentaliomis atskyrimo teoremomis. Teoremos įrodymą galima rasti [3],[4]. Iš teoremos išplaukia labai svarbi išvada, kurią pateiksime kaip 5 dualių uždavinių savybę.

5 savybė. Leistini tiesioginio uždavinio $\boldsymbol{x}^*$ ir dualaus uždavinio $\boldsymbol{p}^*$ planai optimalūs tada ir tik tada, kai galioja lygybė $\boldsymbol{cx}^* = \boldsymbol{p}^*\boldsymbol{b}$.

Įrodymas. Lygybės pakankamumas įrodytas 3 savybėje, o būtinumas išplaukia iš pirmosios dualumo teoremos. ≺

Taigi gamybos planavimo uždavinyje dabar galime tvirtinti, kad jo planas optimalus tik tada, kai šio plano duodamas pelnas lygus sunaudotų išteklių vertingumui; ištekliai yra verti lygiai tiek, kokį pelną iš jų galima gauti, optimaliai sutvarkius gamybą.

Pasinaudoję pirmąja dualumo teorema, galime padaryti išvadą ir dėl dualių uždavinių išsprendžiamumo.

6 savybė. Dualūs uždaviniai gali būti

a) abu išsprendžiami;

b) abu neišsprendžiami, nes vieno tikslo funkcija neaprėžta, o kitas neturi leistinų planų;

c) abu neišsprendžiami, nes nė vienas neturi leistinų planų.

Įrodymas išplaukia iš pirmosios dualumo teoremos ir 4 savybės. Atvejo c) galimybei pagrįsti literatūroje pateikiami atitinkamos uždavinių poros pavyzdžiai [2, p.132].

3°. Antroji dualumo teorema. Dualių kainų prasmei suprasti labai svarbi dar viena teorema, žinoma kaip antroji dualumo arba pusiausvyros teorema.

Antroji dualumo teorema. Tegu $\boldsymbol{x}^*$ - tiesioginio, o $\boldsymbol{p}^*$ - dualaus uždavinio leistini planai. Jie yra optimalūs kiekvienas savo uždavinio planai tada ir tik tada, kai patenkintos šios sąlygos:

jei $\boldsymbol{a}_i\boldsymbol{x}^* < b_i$, tai $p^{*i} = 0$; jei $p^{*i} > 0$, tai $\boldsymbol{a}_i\boldsymbol{x}^* = b_i$ $(i = \overline{1,m})$; (8)

jei $\boldsymbol{p}^*\boldsymbol{a}^j > c^j$, tai $x_j^* = 0$; jei $x_j^* > 0$ tai $\boldsymbol{p}^*\boldsymbol{a}^j = c^j$ $(j = \overline{1,n})$. (9)

Įrodymas. Tegu $\boldsymbol{x}^*$ - tiesioginio, o $\boldsymbol{p}^*$ - dualaus uždavinio optimalūs planai. Įrodysim, kad jie tenkina (8) ir (9) sąlygas.

Kadangi $\boldsymbol{x}^*$ - leistinas, tai $A\boldsymbol{x}^* \le \boldsymbol{b}$. Padauginam šią nelygybę iš kairės iš $\boldsymbol{p}^* \ge \boldsymbol{0}$. Gausim nelygybę $\boldsymbol{p}^*A\boldsymbol{x}^* \le \boldsymbol{p}^*\boldsymbol{b}$. Kadangi $\boldsymbol{p}^*$ - leistinas, tai $\boldsymbol{p}^*A \ge \boldsymbol{c}$. Padauginam šią nelygybę iš dešinės iš $\boldsymbol{x}^* \ge \boldsymbol{0}$. Gausim nelygybę $\boldsymbol{p}^*A\boldsymbol{x}^* \ge \boldsymbol{c}\boldsymbol{x}^*$. Kadangi $\boldsymbol{x}^*$ ir $\boldsymbol{p}^*$ optimalūs, tai pagal 5 savybę $\boldsymbol{c}\boldsymbol{x}^* = \boldsymbol{p}^*\boldsymbol{b}$. Sulyginę gautas nelygybes ir šią lygybę, galime užrašyti

$$\boldsymbol{c}\boldsymbol{x}^* = \boldsymbol{p}^*A\boldsymbol{x}^* = \boldsymbol{p}^*\boldsymbol{b}. \tag{10}$$

Užrašysime kairiąją iš šių lygybių $(\boldsymbol{p}^*A-\boldsymbol{c})\boldsymbol{x}^* = 0$ išskleistu pavidalu (žr. §4 žymėjimus):

$$(\boldsymbol{p}^*\boldsymbol{a}^1 - c^1)x_1^* + \cdots + (\boldsymbol{p}^*\boldsymbol{a}^j - c^j)x_j^* + \cdots + (\boldsymbol{p}^*\boldsymbol{a}^n - c^n)x_n^* = 0 \tag{11}$$

Atkreipkime dėmesį, kad visi šios sumos dėmenys neneigiami, nes $(\boldsymbol{p}^*\boldsymbol{a}^j - c^j) \ge 0$ ir $x_j^* \ge 0$ $(j = \overline{1,n})$. Neneigiamų dėmenų suma gali būti lygi nuliui tik tuo atveju, jei visi šie dėmenys – nuliai (jei bent vienas būtų griežtai teigiamas, visa suma irgi būtų teigiama). Taigi matome, kad sandaugoje $(\boldsymbol{p}^*\boldsymbol{a}^j - c^j)x_j^* = 0$ arba $(\boldsymbol{p}^*\boldsymbol{a}^j - c^j) = 0$ arba $x_j^* = 0$ arba abu dauginamieji - nuliai. O tai ir reiškia, kad galioja (9) sąlygos.

Dabar išskleistu pavidalu užrašysime dešiniąją iš (10) lygybių $\boldsymbol{p}^*(A\boldsymbol{x}^* - \boldsymbol{b}) = 0$.

$$p^{*1}(\boldsymbol{a}_1\boldsymbol{x}^* - b_1) + \cdots + p^{*i}(\boldsymbol{a}_i\boldsymbol{x}^* - b_i) + \cdots + p^{*m}(\boldsymbol{a}_m\boldsymbol{x}^* - b_m) = 0 \tag{12}$$

Visi šios sumos dėmenys neteigiami, nes $(\boldsymbol{a}_i\boldsymbol{x}^* - b_i) \le 0$, o $p^{*i} \ge 0$ $(i = \overline{1,m})$. Neteigiamų dėmenų suma gali būti lygi nuliui tik tuo atveju, jei visi šie dėmenys – nuliai (jei bent vienas būtų griežtai neigiamas, visa suma irgi būtų neigiama). Taigi matome, kad sandaugoje $p^{*i}(\boldsymbol{a}_i\boldsymbol{x}^* - b_i) = 0$ arba $(\boldsymbol{a}_i\boldsymbol{x}^* - b_i) = \boldsymbol{0}$, arba $p^{*i} = 0$, arba abu dauginamieji – nuliai. O tai ir reiškia, kad galioja (8) sąlygos.

Taigi sąlygos (8) ir (9) yra būtinos planų $\boldsymbol{x}^*$ ir $\boldsymbol{p}^*$ optimalumui. Dabar įrodysime, kad jos ir pakankamos.

Iš tikrųjų, jei galioja (8) sąlygos, tai galioja ir (12), nes dėl (8) visi šios sumos dėmenys lygūs nuliui. Taigi $\boldsymbol{p}^*(A\boldsymbol{x}^* - \boldsymbol{b}) = 0$ arba $\boldsymbol{p}^*A\boldsymbol{x}^* = \boldsymbol{p}^*\boldsymbol{b}$. Jei galioja (9) sąlygos, tai galioja ir (11), nes ir šios sumos visi dėmenys dėl (9) lygūs nuliui. Taigi $(\boldsymbol{p}^*A - \boldsymbol{c})\boldsymbol{x}^* = 0$ arba $\boldsymbol{p}^*A\boldsymbol{x}^* = \boldsymbol{c}\boldsymbol{x}^*$. Vadinasi, $\boldsymbol{c}\boldsymbol{x}^* = \boldsymbol{p}^*\boldsymbol{b}$, o tai, kaip žinome, pakankama planų $\boldsymbol{x}^*$ ir $\boldsymbol{p}^*$ optimalumo sąlyga. ⊲

Antroji dualumo teorema padeda mums dar geriau suprasti dualių nežinomųjų (dualių kainų) prasmę. Sąlygos (8) reiškia, kad išteklių, kurių viso turimo kiekio optimaliame plane nepanaudojame ($\boldsymbol{a}_i\boldsymbol{x}^* < b_i$), duali kaina lygi nuliui ($p^{*i} = 0$). Suprantama, už papildomą šių išteklių vienetą įmonei neverta nieko mokėti, nes jo nepavyks panaudoti gamyboje – tos rūšies išteklių ir taip turima per daug. Ir atvirkščiai, jei kurių nors išteklių duali kaina teigiama, ($p^{*i} > 0$), tai tokie ištekliai gamyboje sunaudojami visiškai ($\boldsymbol{a}_i\boldsymbol{x}^* = b_i$). Kaip vėliau matysime, papildomas tokių išteklių vienetas padidina maksimalų pelną dydžiu p^{*i}.

Sąlygos (9) reiškia, kad optimaliame plane negaminamas toks produktas ($x_j^* = 0$), kurio vienetui pagaminti reikalingų išteklių vertė (dualiomis kainomis) didesnė už šio produkto duodamą pelningumą ($\boldsymbol{p}^*\boldsymbol{a}^j > c^j$). Gamindami tokį produktą, tik neracionaliai naudotume išteklius. Kita vertus, jei jau produktas gaminamas ($x_j^* > 0$), tai jo gamybai sunaudoti ištekliai dualiomis kainomis verti lygiai tiek, kokį pelningumą užtikrina tas produktas ($\boldsymbol{p}^*\boldsymbol{a}^j = c^j$).

Taigi dualios kainos dar kartą atskleidžia savo prasmę kaip išteklių naudingumo konkrečios gamybos požiūriu įvertinimai. Jei optimalus tiesioginio uždavinio planas $\boldsymbol{x}^*$ turi m neneigiamų koordinačių, tai galima įrodyti, kad egzistuoja maksimalios tikslo funkcijos reikšmės $\varphi^* = \boldsymbol{c}\boldsymbol{x}^*$ išvestinė pagal kiekvieną iš b_i. Kadangi $\varphi^* = \boldsymbol{c}\boldsymbol{x}^* = \boldsymbol{p}^*\boldsymbol{b}$, tai nesunku matyti, kad ji lygi p^{*i}:

$$\frac{d\varphi^*}{db_i} = p^{*i}. \tag{13}$$

Tai reiškia, kad papildomas i-tųjų išteklių vienetas gali padidinti maksimalų pelną dydžiu p^{*i}, taigi p^{*i} ir yra ta kaina, kurią įmonei verta mokėti už papildomą šių išteklių vienetą.

4°.Dualūs nežinomieji simpleksinėje lentelėje. Kaip išsiaiškinome praeitame punkte, dualios kainos gali suteikti labai vertingos informacijos, planuojant įmonės gamybą. Todėl pageidautina kartu su tiesioginio uždavinio sprendiniu surasti ir dualaus uždavinio sprendinį (dualias kainas). Be abejo, galima spręsti (5) uždavinį kaip tiesinio programavimo uždavinį, ir gauti reikalingas reikšmes. Tačiau to daryti nebūtina. Dualių kainų reikšmes galima perskaityti tiesioginio uždavinio sprendimo baigiamojoje simpleksinėje lentelėje.

Atkreipkime dėmesį į baigiamosios simpleksinės lentelės apatinę eilutę. Joje fiksuota tikslo funkcijos reikšmės išraiška per nebazinius nežinomuosius. Parodysim tik vieną šios išraiškos narį, laikydami, kad y_r optimaliame plane – nebazinis nežinomasis.

$$\varphi = \cdots + \Delta^r(-y_r) + \ \dots + \varphi^* \tag{14}$$

Dydis Δ^r yra neneigiamas dydis, esantis baigiamosios simpleksinės lentelės nežinomąjį y_r atitinkančio stulpelio apatinėje eilutėje. Kaip žinome iš simpleksinio metodo, nebazinių nežinomųjų, tarp jų ir y_r, reikšmės lygios nuliui, todėl ir $\varphi = \varphi^*$.

Tačiau kas atsitiks, jei r-tųjų išteklių gausim dar vieną vienetą, ir jį panaudosim gamyboje? Kitaip tariant, jei leisim šių išteklių pertekliui y_r įgyti reikšmę -1? Įstatę šią reikšmę į (14), matysime, kad tikslo funkcijos reikšmė padidės dydžiu Δ^r. Kitaip tariant, r-jo ištekliaus vienetas bus vertas Δ^r litų, nes tiek padidins mūsų pelną. Tačiau tai, kaip jau išsiaiškinome, ir yra duali šio ištekliaus kaina. Todėl galime spėti, kad Δ^r ir yra ieškomas p^{*r} dydis. Šį faktą galime griežtai matematiškai įrodyti (žr. [3], [4]). Čia to nedarysime, o apsiribosime mūsų jau nagrinėto plytinės uždavinio konkrečiu pavyzdžiu.

Primename šio uždavinio sprendimo baigiamąją simpleksinę lentelę.

1 lentelė

	$-y_1\ (p^1)$	$-x_2$	$-y_3\ (p^3)$	$-x_4$	
x_1	-0,2500	1,2500	0,5000	0,2500	200,00
x_3	0,3750	0,1250	-0,2500	0,6250	95,00
$y_2\ (p^2)$	-1,6250	2,1250	0,7500	-2,3750	175,00
	13,75	11,25	**7,50**	16,25	16650,00

Ši lentelė nuo 6.3e lentelės skiriasi tik tuo, kad prie nežinomųjų y_i pavadinimų patogumo dėlei dar prirašyti atitinkami dualių nežinomųjų p^i pavadinimai. Kaip jau sakėme, dualias kainas randame apatinėje lentelės eilutėje tuose stulpeliuose, kurie atitinka nebazinius nežinomuosius y_1 ir y_3. Todėl $p^{*1} =$ 13,75, o $p^{*3} =$ 7,50. Lentelės viršuje nerandame nežinomojo y_2, kadangi jis bazinis, ir jo reikšmė 175. Tai reiškia, kad antrųjų išteklių (vandens) turime perteklių, ir pagal antrąją dualumo teoremą tokių išteklių duali kaina bus lygi nuliui: $p^{*2} = 0.$ Taigi matome, kad mūsų plytinėje viena darbininko darbo valanda verta 13,75 litų, viena krosnies darbo valanda verta

7,5 litų, o už papildomą kubinį metrą vandens mokėti neverta, nes viso jau turimo vandens optimaliame plane panaudoti nereikia.

5^0 Pratimai. Užrašykite ir interpretuokite §2 3^0 ir §6 6^0 suformuluotiems gamybos planavimo uždaviniams dualius uždavinius.

§ 9. POSTOPTIMIZACINĖ (JAUTRUMO) ANALIZĖ

Šiame paragrafe ištirsime gamybos planavimo uždavinio (matematiniu požiūriu – standartinio tiesinio programavimo uždavinio) optimalaus plano priklausomybę nuo įvairių parametrų: išteklių, kainų, sąnaudų ir pan. Ištirti optimalų planą yra ne mažiau svarbu, negu jį rasti, nes toks tyrimas parodo, kaip įmonei naudingiausia reaguoti į rinkoje vykstančius pokyčius. Remiantis optimalaus plano analize, galima nustatyti kainas, kurias verta mokėti už papildomus išteklius arba kurių būtina reikalauti už papildomos produkcijos gamybą, patikrinti naujų produktų gamybos naudingumą, naujų technologinių sprendimų ekonominį efektyvumą ir apskritai kur kas geriau suprasti įmonės situaciją. Kadangi minėtas tyrimas atliekamas po to, kai surastas optimalus planas, jis dažnai vadinamas *postoptimizacine analize*. Kartais naudojamas ir *jautrumo analizės* pavadinimas, kai keliamas uždavinys išsiaiškinti, kaip jautriai optimalus planas reaguoja į uždavinio parametrų pokyčius.

Postoptimizacinės analizės veiksmus pagrįsime, remdamiesi mūsų išnagrinėtu plytinės uždavinio pavyzdžiu (žr. formulavimą §2 ir sprendimą §6). Suprantama, tokiu būdu negalima griežtai pagrįsti toliau atliekamų veiksmų, tačiau mes to ir nesiekiame. Matematiškai griežtus postoptimizacinės analizės veiksmų pagrindimus galima rasti [4] arba [2].

1°.Išteklių pokyčio poveikis optimaliam planui. Išnagrinėkime, kaip pasikeis plytinės uždavinio optimalus planas, jei vietoj pradinės pirmųjų išteklių (darbo valandų) apimties $b_1 = 780$ turėsime $b_1{+}\Delta b_1 = 780{+}\Delta b_1$. Kadangi optimaliame plane šie ištekliai panaudojami visiškai, jų „perteklius“ y_1 lygus nuliui, o pats nežinomasis y_1 yra nebazinis.

Prisiminsime šio uždavinio sprendimo baigiamąją simpleksinę lentelę

6.3e lentelė

	$-y_1$	$-x_2$	$-y_3$	$-x_4$	
x_1	-0,2500	1,2500	0,5000	0,2500	200,00
x_3	0,3750	0,1250	-0,2500	0,6250	95,00
y_2	-1,6250	2,1250	0,7500	-2,3750	175,00
φ	13,75	11,25	7,50	16,25	16650,00

Ši lentelė, kaip žinome, yra tokių lygčių kompaktiškas užrašymas:

$$x_1 = -0{,}250(-y_1) + 1{,}250(-x_2) + 0{,}50(-y_3) + 0{,}250(-x_4) + 200.$$

$$x_3 = 0{,}375(-y_1) + 0{,}125(-x_2) - 0{,}25(-y_3) + 0{,}625(-x_4) + 95. \qquad (1)$$

$$y_2 = -1{,}625(-y_1) + 2{,}125(-x_2) + 0{,}75(-y_3) - 2{,}375(-x_4) + 175.$$

$$\varphi = 13{,}75(-y_1) + 11{,}25(-x_2) + 7{,}50(-y_3) + 16{,}25(-x_4) + 16650.$$

Pagal §6 įrodytas taisykles laikome, kad $y_1 = x_2 = y_3 = x_4 = 0$, todėl $x_1 = 200$, $x_3 = 95$, $y_2 = 175$, $\varphi = 16650$, ir tai yra optimalus planas dėl apatinės lentelės eilutės koeficientų neneigiamumo.

Tačiau dabar pirmųjų išteklių turime Δb_1 vienetų daugiau. Norėdami juos visus panaudoti, turime leisti jų „pertekliaus" nežinomajam y_1 įgyti reikšmę $-\Delta b_1$ (neigiamas „perteklius" ir yra „perviršis", lyginant su pardine sąlyga). Įstatę $y_1 = -\Delta b_1$ reikšmę į (1) lygtis ir toliau laikydami, kad $x_2 = y_3 = x_4 = 0$, gauname

$$x_1 = -0{,}250(\Delta b_1) + 200.$$

$$x_3 = 0{,}375(\Delta b_1) + 95.$$

$$y_2 = -1{,}625(\Delta b_1) + 175.$$

$$\varphi = 13{,}75(\Delta b_1) + 16650.$$

Tą patį užrašysime kiek kitaip:

$$\begin{pmatrix} x_1 \\ x_3 \\ y_2 \\ \varphi \end{pmatrix} = \begin{pmatrix} 200 \\ 95 \\ 175 \\ 16650 \end{pmatrix} + \Delta b_1 \begin{pmatrix} -0{,}250 \\ 0{,}375 \\ -1{,}625 \\ 13{,}750 \end{pmatrix} \qquad (2)$$

Pavyzdžiui, jei $\Delta b_1 = 80$, tai „A" tipo plytų blokų gaminsime $x_1 = 200$-0,25·80 = 180 vienetų, „C" tipo - $x_3 = 95$+0,375·80 = 125 vienetus, nepanaudotų vandens išteklių liks $y_2 =$ 175-1,625·80 = 45 m^3, o pelnas sudarys $\varphi =$ 16650+13,75·80 = 17750 litų. Taigi už 80 darbo valandų, t.y. už papildomo darbininko darbą 10 dienų po 8 val. per dieną plytinė gali mokėti 1100 litų.

Analogiškai nagrinėjame ir atvejį, kai $\Delta b_1 <$ 0, t.y. darbo valandų skaičius sumažėja. Pavyzdžiui, jei $\Delta b_1 =$ -80, t.y. plytinė turės vienu darbininku mažiau, tai „A“ tipo plytų blokų gamyba padidės: $x_1 =$ 200-0,25·(-80) = 220, „C“ tipo plytų blokų gamyba sumažės: $x_3 =$ 95+0,375·(-80) = 65, liks nepanaudota $y_2 =$ 175-1,625·(-80) = 305 m^3 vandens, o pelnas sumažės ir sudarys $\varphi =$ 16650+13,75·(-80) = 15550 litų.

Tačiau Δb_1 negalime be galo didinti ar mažinti. Kaip galima įrodyti, optimalaus plano bazė nesikeis, kol, skaičiuodami pagal (2) formules, negausime neigiamų nežinomųjų reikšmių. Todėl Δb_1 kitimo ribas rasime iš priklausomybių, reikalaujančių nežinomųjų neneigiamumo:

200,00- 0,250$\Delta b_1 \geq 0$, todėl $\Delta b_1 \leq$ 200/0,250 = 800;

95,00+ 0,3750$\Delta b_1 \geq 0$, todėl $\Delta b_1 \geq$ -95/0,375 = -253,33;

175,00- 1,6250$\Delta b_1 \geq 0$, todėl $\Delta b_1 \leq$ 175/1,625 = 107,69.

Taigi (2) priklausomybės galioja, jei

$$\Delta b_1 \in [-253,33;\ 107,69].$$

Atsiminę, kad $b_1 = 780$, galime sakyti, kad (2) plano ir pelno pokyčių apskaičiavimo taisyklės galioja, kol

$$b_1 \in [526,67;\ 887,69].$$

Tokiu pat būdu nagrinėjame ir trečiųjų išteklių (krosnies darbo valandų) pokyčius. Šiuo atveju y_3 – taip pat nebazinis nežinomasis, todėl dabar pokyčiams apskaičiuoti imame y_3 atitinkantį stulpelį:

$$\begin{pmatrix} x_1 \\ x_3 \\ y_2 \\ \varphi \end{pmatrix} = \begin{pmatrix} 200 \\ 95 \\ 175 \\ 16650 \end{pmatrix} + \Delta b_3 \begin{pmatrix} 0,50 \\ -0,25 \\ 0,75 \\ 7,50 \end{pmatrix}$$

Atlikę analogiškus ankstesniems įvertinimus, įsitikiname, kad šios priklausomybės galioja, kai

$$\Delta b_3 \in [-233,33;\ 380].$$

Antrųjų (vandens) išteklių kitimą analizuojame kitaip, nes $y_2 = 175$ - bazinis nežinomasis. Optimaliame plane nepanaudojame 175 m^3 vandens. Jei vandens turėsime dar daugiau, tai optimalus planas nepasikeis, tik padidės nepanaudoto vandens perteklius; tas pat bus, jei vandens perteklių mažinsime, kol jo dar užteks. Taigi, kai

$$\Delta b_2 \in [-175; \infty).$$

turimas planas lieka optimalus.

2°.Pelningumo pokyčių poveikis optimaliam planui. Planuojant gamybą, labai svarbu žinoti, kaip reikia keisti gaminamų produktų apimtis, jei dėl įvairių priežasčių jų gamybos pelningumas didėja ar mažėja. Postoptimizacinės analizės metodais galima atsakyti į šį klausimą. Galima įrodyti [4], kad kitaip, nei išteklių pokyčio atveju, pelningumui kintant tam tikrame intervale, gamybos apimtys nesikeičia. Pokyčio intervalui nustatyti vėl naudojame baigiamąją simpleksinę lentelę. Parodysim, kaip tai daroma.

Tęskime plytinės uždavinio analizę. „A“ tipo plytų blokų vieneto gamybos pelningumas $c^1 = 50$. Intervalą, kuriame gali keistis šių plytų gamybos pelningumas, kad visų produktų gamybos planas liktų nepakitęs, nustatysime iš baigiamosios simpleksinės lentelės apatinės eilutės ir nežinomąjį x_1 („A“ tipo plytų blokus) atitinkančios eilutės elementų santykių. Išrašysime atskirai x_1 ir apatinę (φ)eilutes:

x_1	-0,2500	1,2500	0,5000	0,2500
φ	13,75	11,25	7,50	16,25

Pelningumo c^1 pokyčio Δc^1 ribas nustatysime pareikalavę, kad Δc^1 būtų toks, kad galiotų

$$(13{,}75;\ 11{,}25;\ 7{,}50;\ 16{,}25) + \Delta c^1(-0{,}25;\ 1{,}25;\ 0{,}5;\ 0{,}25) \geq \boldsymbol{0}.$$

Tą patį parašysim kiek kitaip:

$13{,}75 - 0{,}25\Delta c^1 \geq 0$, todėl $\Delta c^1 \leq 13{,}75/0{,}25 = 55$;

$11{,}25 + 1{,}25\Delta c^1 \geq 0$, todėl $\Delta c^1 \geq -11{,}25/1{,}25 = -9$;

$7{,}50 + 0{,}50\Delta c^1 \geq 0$, todėl $\Delta c^1 \geq -7{,}5/0{,}5 = -15$;

$16{,}25 + 0{,}25\Delta c^1 \geq 0$, todėl $\Delta c^1 \geq -16{,}25/0{,}25 = -65$.

Taigi Δc^1 iš viršaus ribojamas dydžiu 55, o iš apačios dydžiu -9 : $\Delta c^1 \in [-9; 55]$. Kitaip tariant, kol „A“ tipo plytų blokų gamybos pelningumas nesumažėja daugiau kaip 9 litais už vienetą arba nepadidėja daugiau, negu 55 litais už vienetą, optimalus gamybos planas nepasikeičia. Kadangi c^1 reikšmė pagal sąlygą yra 50, galime pasakyti, kad optimalus gamybos planas nepasikeičia, kol „A“ blokų gamybos pelningumas yra tarp 41 ir 105 litų už vienetą.

Analogiškai galime atlikti ir „C“ tipo plytų blokų pelningumo c^3 pokyčių analizę. Pelningumo kitimo intervalą nustatysime iš baigiamosios simpleksinės lentelės apatinės eilutės ir nežinomąjį x_3 („C“ tipo plytų blokus) atitinkančios eilutės elementų santykių:

x_3	0,3750	0,1250	-0,2500	0,6250
φ	13,75	11,25	7,50	16,25

Vėl skaičiuojame neneigiamumo sąlygas:

$$(13{,}75;\ 11{,}25;\ 7{,}50;\ 16{,}25) + \Delta c^3(0{,}375;\ 0{,}125;\ -0{,}25;\ 0{,}625) \geq \boldsymbol{0},$$

arba

$$13{,}75 + 0{,}375\Delta c^3 \geq 0, \qquad \Delta c^3 \geq -13{,}75/0{,}375 = -36{,}67;$$

$$11{,}25 + 0{,}125\Delta c^3 \geq 0, \qquad \Delta c^3 \geq -11{,}25/0{,}125 = -90;$$

$$7{,}50 - 0{,}25\Delta c^3 \geq 0, \qquad \Delta c^3 \leq 7{,}5/0{,}25 = 30;$$

$$16{,}25 + 0{,}625\Delta c^3 \geq 0, \qquad \Delta c^3 \geq -16{,}25/0{,}625 = -26.$$

Taigi $\Delta c^3 \in [-26; 30]$. Kitaip tariant, kol „C“ tipo plytų blokų gamybos pelningumas nesumažėja daugiau kaip 26 litais už vienetą arba nepadidėja daugiau, negu 30 litų už vienetą, optimalus gamybos planas nepasikeičia. Atsiminę, kad c^3 reikšmė yra 70, galime pasakyti, kad optimalus gamybos planas nepasikeičia, kol „C“ blokų gamybos pelningumas yra tarp 44 ir 100 litų už vienetą.

Kitaip analizuojame „B“ ir „D“ tipo plytų blokų gamybos pelningumo pokyčius. Kadangi šios plytos negaminamos ($x_2 = x_4 = 0$), tai iš (2) lygčių matome, kad „B“ plytų bloko vieneto gamyba $(x_2 = 1)$ duotų 11,25 litų nuostolio, o „D“ plytų bloko vieneto gamyba $(x_4 = 1)$ duotų 7,5 litų nuostolio. Šie skaičiai aiškiai matomi ir baigiamosios simpleksinės lentelės stulpeliuose, atitinkančiuose nežinomuosius x_2 ir x_4 :

	y_1	x_2	y_3	x_4
φ	13,75	11,25	7,50	16,25

Todėl galime tvirtinti, kad „B“ tipo plytas apsimokės gaminti, jei jų pelningumas padidės 11,25 litų už vienetą ir pasieks 60+11,25=71,25 litų, o „D“ tipo plytas apsimokės gaminti jei jų pelningumas padidės 16,25 litų už vienetą ir pasieks 40+16,25=56,25 litų. Kol toks pelningumas nepasiektas, lieka galioti senas optimalus planas.

3°. „Priverstinių“ gamybos pokyčių poveikis optimaliam planui. Kartais praktikoje gali atsirasti aplinkybės, kai reikia pagaminti kurį nors produktą, nors optimaliame plane jis negaminamas. Arba tenka gaminti kurio nors iš gaminamų produktų ne tiek, kiek optimalu, o kažkokį kitą kiekį. Ir šiuo atveju postoptimizacinės analizės metodais galima greitai įvertinti atsirandančias pasekmes – gamybos plano ir pelno pokyčius.

Tęskime plytinės uždavinio analizę. Sakykim, gautas užsakymas pagaminti tam tikrą kiekį optimaliame plane negaminamo produkto, sakykim „B“ tipo plytų blokų. Optimaliame plane $x_2 = 0$, bet prašoma „B“ plytų pagaminti $\bar{x}_2$ vienetų. Įstatę $x_2 = \bar{x}_2$ reikšmę į (1) lygtis ir toliau laikydami, kad $y_1 = y_3 = x_4 = 0$, gauname

$$x_1 = 1{,}250(-\bar{x}_2) + 200.$$

$$x_3 = 0{,}125(-\bar{x}_2) + 95.$$

$$y_2 = 2{,}125(-\bar{x}_2) + 175.$$

$$\varphi = 11{,}25(-\bar{x}_2) + 16650.$$

Tą patį užrašysime kiek kitaip:

$$\begin{pmatrix} x_1 \\ x_3 \\ y_2 \\ \varphi \end{pmatrix} = \begin{pmatrix} 200 \\ 95 \\ 175 \\ 16650 \end{pmatrix} - \bar{x}_2 \begin{pmatrix} 1{,}250 \\ 0{,}125 \\ 2{,}125 \\ 11{,}25 \end{pmatrix} \tag{3}$$

Taigi, jei gautas užsakymas pagaminti $\bar{x}_2$ = 8 „B“ tipo plytų blokus, tai nesunkai apskaičiuojame, kad, perskirstę išteklius, gaminsime x_1 = 200-8·1,25= 190 „A“ tipo plytų blokų, x_3 = 94 „C“ tipo plytų blokų. Liks nepanaudota y_2 = 158 m^3 vandens išteklių, o pelnas sumažės nuo 16650 litų iki 16560 litų, taigi iš viso 90 litų (kadangi planas dabar bus nebe optimalus). Todėl įmonė už kiekvieną specialiai gaminamą „B“ tipo plytų bloką turi reikalauti mažiausiai 11,25 litų kompensacijos, pavyzdžiui pakelti šio bloko realizavimo kainą iki 71,25 litų (2^0 gavome tą pačią išvadą).

Kaip ir anksčiau, (3) skaičiavimo taisyklės galioja, kol pagal jas nustatomi gamybos apimčių ir išteklių pertekliaus dydžiai lieka neneigiami (nežinomųjų reikšmės lieka neneigiamos). Belieka rasti, kokiame $\bar{x}_2$ kitimo intervale tai teisinga. Remdamiesi neneigiamumo reikalavimu, gauname:

200,00 - 1,250$\bar{x}_2 \geq 0$ $\qquad$ $\bar{x}_2 \leq$ 200/1,250 = 160,

95,00- 0,125$\bar{x}_2 \geq 0$ $\qquad$ $\bar{x}_2 \leq$ 95/0,125 = 760,

175,00- 2,125$\bar{x}_2 \geq 0$ $\qquad$ $\bar{x}_2 \leq$ 175/2,125 = 82,35.

Kadangi neigiamos $\bar{x}_2$ reikšmės plytinės uždavinyje beprasmiškos, galime parašyti, kad atlikta analizė galioja intervale

$$\bar{x}_2 \in [0; 82{,}35].$$

Jei būtų pareikalauta gaminti dar daugiau, negu 82,35 „B“ tipo plytų blokų, mūsų taikytas gamybos ir pelno pokyčių įvertinimas nebegaliotų.

Panašiai analizuojame ir „D“ tipo plytų blokų papildomą gamybą. Šiuo atveju pokyčiams apskaičiuoti naudojame baigiamosios simpleksinės lentelės stulpelį, atitinkantį nežinomąjį x_4 :

$$\begin{pmatrix} x_1 \\ x_3 \\ y_2 \\ \varphi \end{pmatrix} = \begin{pmatrix} 200 \\ 95 \\ 175 \\ 16650 \end{pmatrix} - \bar{x}_4 \begin{pmatrix} 0{,}250 \\ 0{,}625 \\ -2{,}375 \\ 16{,}25 \end{pmatrix} \tag{4}$$

Analogiškai ankstesniam atvejui nustatome, kad (4) formulės galioja, kol

$$\bar{x}_4 \in [0; 152].$$

Sudėtingiau analizuojami „priverstiniai“ bazinių nežinomųjų x_1 ir x_3 reikšmių pokyčiai. Čia jų nenagrinėsime. Šio atvejo taisykles ir pavyzdžius galima rasti [4].

4°. Papildomo produkto gamybos įvertinimas. Postoptimizacinės analizės metodais galima atlikti dar vieną užduotį – įvertinti, ar apsimoka gaminti naują iki šiol negamintą produktą. Plytinės uždavinio pavyzdžiu paaiškinsim, kaip tai daroma. Tarkim, plytinė svarsto „E“ tipo plytų blokų gamybą. Vienam tokiam blokui pagaminti reikia 2 val. darbo laiko, 3 m^3 vandens ir 3 krosnies darbo valandų. Apskaičiuosime, kiek dualiomis kainomis verti šie ištekliai. Atsižvelgę, kad darbininko darbo valanda verta $p^{*1} =$ 13,75, krosnies darbo valanda $p^{*3} =$ 7,50, o vandens turime perteklių ($p^{*2} =$ 0), apskaičiuojame, kad ištekliai, reikalingi vienam „E“ tipo plytų blokui pagaminti, verti 2·13,75+3·0+3·7,5 = 50 litų. Vadinasi, „E“ tipo plytų blokų pelningumas c^5 turi būti ne mažesnis kaip 50 litų, kitaip gamindami šias plytas tik „gadintume“ išteklius ir sumažintume pelną. Jei pelningumas didesnis, iš naujo išsprendę uždavinį optimaliame plane gausime, kad „E“ tipo plytos bus gaminamos, o kitų plytų gamyba bus sumažinta ar nutraukta.

5°. Pratimai.

5.1°. Jums pateikiamas §2 3^0 suformuluoto uždavinio sprendimas simpleksinėmis lentelėmis. Atlikite sprendinio postoptimizacinę analizę.

$$\begin{aligned}
&\textbf{(max) } 102x_1 + 130x_2 + 192x_3 + 100x_4\\
&7x_1 + 9x_2 + 12x_3 + 4x_4 \le 450,\\
&8x_1 + 3x_2 + 2x_3 + 5x_4 \le 300,\\
&4x_1 + 6x_2 + 9x_3 + 10x_4 \le 300.\\
&x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 \ge 0.
\end{aligned}$$

Sprendimas.

	-X1	-X2	-X3	-X4	
Y1	7,00	9,00	12,00	4,00	450,00
Y2	8,00	3,00	2,00	5,00	300,00
Y3	4,00	6,00	9,00	10,00	300,00
	-102,00	-130,00	-192,00	-100,00	0,00

	-X1	-X2	-Y3	-X4	
Y1	1,67	1,00	-1,33	-9,33	50,00
Y2	7,11	1,67	-0,22	2,78	233,33
X3	0,44	0,67	0,11	1,11	33,33
	-16,67	-2,00	21,33	113,33	6400,00

	-Y1	-X2	-Y3	-X4	
X1	0,6000	0,6000	-0,8000	-5,6000	30,00
Y2	-4,2667	-2,6000	5,4667	42,6000	20,00
X3	-0,2667	0,4000	0,4667	3,6000	20,00
	10,00	8,00	8,00	20,00	6900,00

Postoptimizacinės analizės atsakymai.

Pirmų išteklių duali kaina 10,00; pokyčio intervalas, kuriame išlieka ta pati bazė [-50; 4,7].
Antrų išteklių duali kaina 0,00; pokyčio intervalas, kuriame išlieka ta pati bazė [-20; ∞].
Trečių išteklių duali kaina 8,00; pokyčio intervalas, kuriame išlieka ta pati bazė [-3,7;37,5].

Pirmo produkto apsimoka gaminti 30 vienetų, kol jo pelningumas intervale [88,67; 105,57] lt./vnt.
Antrą produktą apsimokės gaminti, jei jo pelningumas pakils iki 138 lt. / vnt.
Trečio produkto apsimoka gaminti 20 vienetų, kol jo pelningumas [186,4; 229,5] lt./vnt.
Ketvirtą produktą apsimokės gaminti, jei jo pelningumas pakils iki 120 lt. / vnt.

Antro produkto „priverstinės“ gamybos pokyčio intervalas, kuriame išlieka ta pati bazė [0; 50]
Ketvirto produkto „priverstinės“ gamybos pokyčio intervalas, kuriame išlieka ta pati bazė [0; 0,47]

5.2°. Jums pateikiamas §2 3^0 suformuluoto uždavinio sprendimas simpleksinėmis lentelėmis. Atlikite sprendinio postoptimizacinę analizę.

$$\begin{aligned}
&\textbf{(max) } 54x_1 + 25x_2 + 40x_3 + 31x_4 \\
&\quad 6x_1 + 4x_2 + 5x_3 + 2x_4 \le 1700, \\
&\quad 8x_1 + 3x_2 + 6x_3 + 7x_4 \ 2700, \\
&\quad 9x_1 + 1x_2 + 5x_3 + 6x_4 \le 3000 \\
&\quad x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0,\ x_4 \ge 0.
\end{aligned}$$

Sprendimas

	-X1	-X2	-X3	-X4	
Y1	6,00	4,00	5,00	2,00	1700,0
Y2	8,00	3,00	6,00	7,00	2700,0
Y3	9,00	1,00	5,00	6,00	3000,0
	-54,00	-25,00	-40,00	-31,00	0,00

	-Y1	-X2	-X3	-X4	
X1	0,17	0,67	0,83	0,33	283,33
Y2	-1,33	-2,33	-0,67	4,33	433,33
Y3	-1,50	-5,00	-2,50	3,00	450,00
	9,00	11,00	5,00	-13,00	15300,00

	-Y1	-X2	-X3	-Y2	
X1	0,2692	0,8462	0,8846	-0,0769	250,00
X4	-0,3077	-0,5385	-0,1538	0,2308	100,00
Y3	-0,5769	-3,3846	-2,0385	-0,6923	150,00
	5,00	4,00	3,00	3,00	16600,00

Postoptimizacinės analizės atsakymai.

Pirmų išteklių duali kaina 5,00; pokyčio intervalas, kuriame išlieka ta pati bazė [-928,6; 260,0].
Antrų išteklių duali kaina 3,00; pokyčio intervalas, kuriame išlieka ta pati bazė [-433,3; 216,7].
Trečių išteklių duali kaina 0,00; pokyčio intervalas, kuriame išlieka ta pati bazė [-150; ∞].

Pirmo produkto apsimoka gaminti 250 vienetų, kol jo pelningumas intervale [50,6; 93,0] lt./vnt.
Antrą produktą apsimokės gaminti, jei jo pelningumas pakils iki 29 lt. / vnt.
Trečią produktą apsimokės gaminti, jei jo pelningumas pakils iki 43 lt. / vnt.
Ketvirto produkto apsimoka gaminti 100 vienetų, kol jo pelningumas [18,0; 38,4] lt./vnt.

Antro produkto „priverstinės“ gamybos pokyčio intervalas, kuriame išlieka ta pati bazė [0; 295,5]
Trečio produkto „priverstinės“ gamybos pokyčio intervalas, kuriame išlieka ta pati bazė [0; 282,6]

§ 10. TRANSPORTO UŽDAVINYS

1°.Uždavinio formulavimas. Transporto uždavinyje nagrinėjame m tiekėjų, turinčių tam tikro produkto atsargas $a_1, a_2, \ldots, a_m$ vienetų, ir n gavėjų, kurių poreikiai šiam produktui lygūs $b_1, b_2, \ldots, b_n$ vienetų. „Tiekėjais" gali būti įmonės, gaminančios kitoms įmonėms arba prekybai tiekiamą produkciją, prekybos bazės, sandėliai ir t.t. „Gavėjais" gali būti visi, kas naudoja arba perdirba tiekiamą produktą – įmonės, parduotuvės, atskiri užsakovai ir t.t. „Produktas" taip pat gali būti suprantamas labai įvairiai, bet paprasčiausiame transporto uždavinyje jis yra vienintelis.

Žinoma, kad produkto vieneto pervežimas iš i-tojo tiekėjo j-tajam gavėjui kainuoja c_{ij} litų ($i = \overline{1,m}, j = \overline{1,n}$). Reikia sudaryti tokį pervežimų planą, kad visos tiekėjų atsargos būtų išvežtos, gavėjų poreikiai patenkinti, o bendra visų pervežimų kaina būtų minimali.

Matematinio transporto uždavinio modelio sudarymą pradėsime, kaip įprasta, nuo nežinomųjų paskyrimo. Tarkime, x_{ij} $(i = \overline{1,m}, j = \overline{1,n})$ - produkto apimtis, pervežama iš i-tojo tiekėjo j-tajam gavėjui. Apskaičiuosime, kad iš pirmojo tiekėjo visiems gavėjams bus išvežta $x_{11} + x_{12} + \cdots + x_{1n} = \sum_{j=1}^{n} x_{1j}$ produkto vienetų. Ši apimtis pagal sąlygą turi būti lygi pirmojo tiekėjo turimai produkto atsargai a_1. Panašiai samprotaudami apie likusius tiekėjus, matome, kad pervežimų apimtys x_{ij} turi tenkinti m lygčių:

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}). \tag{1}$$

Bet kuris gavėjas, sakykim j-tasis, gauna iš visų tiekėjų $x_{1j} + x_{2j} + \cdots + x_{mj} = \sum_{i=1}^{m} x_{ij}$ produkto vienetų. Ši suma pagal sąlygą turi būti lygi jo poreikiui b_j. Kadangi turi būti patenkinti visų gavėjų poreikiai, turi galioti dar n lygčių

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}). \tag{2}$$

Pervežimų apimtys negali būti neigiamos (negalima „grąžinti" produkto iš gavėjo tiekėjui:

$$x_{ij} \geq 0 \quad (i = \overline{1,m}, j = \overline{1,n}). \tag{3}$$

Apskaičiuosime bendrą visų pervežimų kainą. Produkto vieneto pervežimas iš i-tojo tiekėjo j-tajam gavėjui kainuoja c_{ij} litų, o pervežama x_{ij} vienetų, taigi vieno pervežimo (vienos pervežimų grandies) kaina yra $c_{ij}x_{ij}$ litų. Sumuodami pervežimų kainas iš visų tiekėjų visiems gavėjams, gauname bendrą visų pervežimų kainą, kuri turi būti minimali:

$$(\min) \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4}$$

Uždavinys (1)-(4) ir yra paprasčiausias transporto uždavinys: reikia rasti tokia pervežimų apimtis x_{ij}, kurios tenkintų apribojimus (1)-(3) ir suteiktų minimalią reikšmę (4) tikslo funkcijai. Panašiai kaip bendro tiesinio programavimo uždavinio atveju, dydžius x_{ij} $(i = \overline{1,m}, j = \overline{1,n})$, tenkinančius apribojimus (1)-(3), vadinsime transporto uždavinio *leistinu planu,* o leistiną planą, kuriame (4) tikslo funkcija įgyja minimalią reikšmę – *optimaliu planu* arba *sprendiniu.*

Prieš nagrinėdami šio uždavinio savybes, sprendimo būdus ir taikymus, susitarsime dėl kai kurių jo parametrų. Laikysime, kad $a_i > 0$ $(i = \overline{1,m})$, $b_j > 0$ $(j = \overline{1,n})$, t.y. uždavinyje nėra tiekėjų, turinčių nulines arba neigiamas atsargas, taip pat nėra ir gavėjų su nuliniais ar neigiamais poreikiais. Tokių „tiekėjų“ arba „gavėjų“ egzistavimas uždavinio prasmės požiūriu būtų beprasmis. Taip pat laikysime, kad $m \geq 2, n \geq 2$, nes uždavinys su vieninteliu tiekėju ar gavėju nevertas dėmesio – jo sprendimas trivialus. Taip pat laikysime, kad visi $c_{ij} \geq 0$, nes neigiamos pervežimų kainos neturėtų prasmės.

2°.Transporto uždavinio savybės. Atkreipkime dėmesį, kad transporto uždavinys, nors jo užrašymas aiškiai skiriasi nuo tiesinio programavimo uždavinio užrašymo, nagrinėto ankstesniuose paragrafuose, vis dėlto yra tiesinio programavimo uždavinys. Iš tikrųjų, visi jo apribojimai tiesiniai, tikslo funkcija tiesinė taip pat, taigi jam galioja visos mums jau žinomos tiesinio programavimo uždavinių savybės.

Transporto uždavinį galima užrašyti ir įprastu tiesinio programavimo uždaviniams pavidalu

$$(\min)\ \boldsymbol{cx},\ \ T\boldsymbol{x} = \boldsymbol{d},\ \ \boldsymbol{x} \geq \boldsymbol{0}.$$

Čia T yra $m+n$ eilučių ir mn stulpelių matrica, kurios elementai – tik nuliai ir vienetai:

$$T = \begin{pmatrix}
1 & 1 & \ldots & 1 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 & \ldots & 0 & 0 & \ldots & 0 \\
\ldots & \ldots & \ddots & \ldots & \ldots & \ldots & \ddots & \ldots & \ldots & \ldots & \ldots & \ddots & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & \ldots & 1 & 1 & \ldots & 1 \\
 & & & & & & & & & & & & \\
1 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & \ldots & 1 & 0 & \ldots & 0 \\
0 & 1 & \ldots & 0 & 0 & 1 & \ldots & 0 & \ldots & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 1 & \ldots & 0 & 0 & \ldots & 1
\end{pmatrix},$$

$$\boldsymbol{c} = (c_{11}, c_{12}, \ldots\ c_{1n}, \ldots, c_{m1}, c_{m2}, \ldots\ c_{mn}),$$

$$\boldsymbol{x} = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ \vdots \\ x_{m1} \\ \vdots \\ x_{mn} \end{pmatrix}, \qquad \boldsymbol{d} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Dėl ypatingos apribojimų matricos struktūros transporto uždaviniui galima formuluoti kai kurias tik šiam uždaviniui būdingas savybes. Prieš jas nagrinėdami, pradėsim nuo labai svarbios transporto uždaviniui subalansuotumo sąvokos.

<u>Apibrėžimas</u>. Transporto uždavinys vadinamas subalansuotu, jei patenkinta sąlyga

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \tag{5}$$

Subalansuotumo sąvokos prasmė akivaizdi: uždavinys subalansuotas, jei tiekėjų atsargos lygios gavėjų poreikiams.

<u>1 savybė</u>. Transporto uždavinys turi leistinų planų tada ir tik tada, kai jis subalansuotas.

Įrodymas. Sakykim, leistinas planas x_{ij} $(i = \overline{1,m}, j = \overline{1,n})$, egzistuoja. Kadangi jis tenkina (1) lygybes, sumuodami jas pagal i, gausime

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} a_i. \tag{6}$$

Leistinas planas taip pat tenkina (2) lygybes, kurias susumuosime pagal j

$$\sum_{j=1}^{n} \sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} b_j. \tag{7}$$

Šių (6) ir (7) lygybių kairiosios pusės lygios viena kitai, nes jose sumuojami tie patys dydžiai, tik kita tvarka. Todėl lygios ir dešiniosios (6) ir (7) pusės, taigi (5) subalansuotumo sąlyga patenkinta. Vadinasi, ji būtina, egzistuojant leistinam planui. Įrodysime, kad ji ir pakankama.

Tarkim, (5) sąlyga galioja. Pažymėkim $v = \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$ ir sudarykim tokį transporto uždavinio planą: $x_{ij} = \frac{a_i b_j}{v} > 0$ $(i = \overline{1,m}, j = \overline{1,n})$. Paprastu sumavimu nesunku įsitikinti, kad šis planas leistinas. Iš tikrųjų,

$$\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} \frac{a_i b_j}{v} = \frac{a_i}{v} \sum_{j=1}^{n} b_j = a_i \quad (i = \overline{1,m}),$$

todėl mūsų sudarytam planui galioja (1) lygybės. Analogiškai įsitikiname, kad jam galioja ir (2) lygybės. Akivaizdu, kad galioja ir (3) nelygybės. Taigi radome bent vieną leistiną planą, vadinasi leistini planai egzistuoja.$\prec$

2 savybė. Subalansuotas transporto uždavinys visada išsprendžiamas.

Įrodymas. Tiesinio programavimo uždavinys gali būti neišsprendžiamas tik dėl dviejų priežasčių: nėra leistinų planų arba tikslo funkcija neaprėžta. Kaip žinome iš 1 savybės, subalansuotas transporto uždavinys turi leistinų planų. Jo tikslo funkcija dėl to, kad visi $c_{ij} \geq 0$, taip pat neneigiama, taigi aprėžta iš apačios nuliu. Vadinasi, nėra nė vienos iš abiejų priežasčių, dėl ko transporto uždavinys būtų neišsprendžiamas. $\prec$

3°.Transporto uždavinio subalansavimas. Kaip matome iš 1 ir 2 savybių, sprendžiamas gali būti tik subalansuotas transporto uždavinys. Tačiau praktikoje gavėjų poreikiai tiksliai sutampa su tiekėjų galimybėmis, matyt, labai retai. Todėl atsargų ir poreikių nesutapimo atveju reikia transporto uždavinį pritaikyti ir šiai situacijai. Laimei, tai labai nesunku padaryti nė kiek nepažeidžiant uždavinio logikos.

Jei tiekėjų atsargos didesnės nei gavėjų poreikiai ($\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$), uždavinį papildome fiktyviu *(n+1)* gavėju, kurio poreikiai b_{n+1} kaip tik sutampa su tiekėjų turimų atsargų pertekliumi: $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$. Suprantama, kad dėl to uždavinys tampa subalansuotu uždaviniu su m tiekėjų ir $n+1$ gavėju. Belieka tik nurodyti pervežimo kainas iš visų tiekėjų naujajam (fiktyviam) gavėjui. Kadangi šie pervežimai taip pat bus fiktyvūs (iš tikrųjų produktas niekur nebus vežamas), tai pervežimų kainos bus nuliai: $c_{i,n+1} = 0 \ (i = \overline{1,m})$. Dabar uždavinį galime spręsti, nes jis subalansuotas. Jei išsprendę gausime teigiamą $x_{i,n+1}$ reikšmę, tai reikš, kad *i*-tojo tiekėjo sandėlyje liko neišvežta $x_{i,n+1}$vienetų produkto. Pavyzdžiui, jei $x_{7,n+1} = 12$, tai reiškia, kad septintojo tiekėjo sandėlyje liko 12 vienetų neišvežto tiekėjams produkto.

Jei tiekėjų atsargos mažesnės nei gavėjų poreikiai ($\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$), uždavinį papildome fiktyviu *(m+1)* tiekėju, kurio atsarga a_{m+1} kaip tik sutampa su tiekėjų turimų atsargų pertekliumi: $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$. Šiuo atveju taip pat gauname subalansuotą uždavinį su $m+1$ tiekėju ir n gavėjų. Belieka tik nurodyti pervežimo kainas iš naujojo (fiktyvaus) tiekėjo visiems gavėjams. Kadangi šie pervežimai taip pat bus fiktyvūs (iš tikrųjų produktas niekur nebus vežamas), tai pervežimų kainos bus nuliai: $c_{m+1,j} = 0 \ (j = \overline{1,n})$. Dabar uždavinį galime išspręsti, nes jis subalansuotas. Jei išsprendę gausime teigiamą $x_{m+1,j}$ reikšmę, tai reikš, kad j-tasis gavėjas negavo $x_{m+1,j}$ vienetų pageidauto produkto. Pavyzdžiui, jei $x_{m+1,5} = 10$, tai reiškia, kad penktasis gavėjas gavo 10 vienetų iš fiktyvaus (nesančio) tiekėjo, taigi gavo 10 vienetų mažiau, negu buvo jo poreikis.

Pavyzdys. Trys smėlio karjerai aprūpina tris statybas smėliu. Karjerai per planavimo laikotarpį gali pateikti atitinkamai 10, 13 ir 42 tonas smėlio, o statyboms reikia atitinkamai 18, 28 ir 8 tonų. Vienos smėlio tonos pervežimo išlaidas iš kiekvieno karjero į kiekvieną statybą nurodo 3×3 matrica

$$\begin{pmatrix} 5 & 4 & 13 \\ 11 & 13 & 17 \\ 12 & 12 & 20 \end{pmatrix}.$$

Matome, kad uždavinys nesubalansuotas: karjerų galimybės didesnės už statybų poreikius: 10+13+42 > 18+28+8. Norėdami uždavinį subalansuoti, papildome jį ketvirta (fiktyvia) statyba, kurios poreikis – 11 tonų. Dabar galime uždavinį surašyti į tokią lentelę:

1 lentelė

5 10	4	13	0	**Karjerai** 10
11 8	13 5	17	0	13
12	12 23	20 8	0 11	42
18 **Statybos**	28	8	11	65

Šioje lentelėje užfiksuota ne tik uždavinio sąlyga, bet ir vienas iš leistinų planų: pirmas karjeras teikia 10 tonų smėlio pirmai statybai, antras karjeras – 8 tonas pirmai ir 5 tonas antrai statybai, trečias karjeras – 23 tonas antrai ir 8 tonas trečiai statybai, o 11 tonų smėlio iš šio karjero lieka nepanaudota (tiekiama fiktyviai statybai).⊲

4°.Apie transporto uždavinio sprendimą. Transporto uždavinys yra tiesinio programavimo uždavinys, todėl sprendžiamas simpleksiniu metodu. Tačiau dėl matricos T (žr. 1^0) struktūros metodo taisyklės paprastesnės, vedantysis elementas visada vienetas, taigi simpleksiniuose žingsniuose lieka tik sudėties ir atimties veiksmai (dalinama visada iš vieneto). Todėl, jei atsargų ir poreikių apimtys užduotos sveikais skaičiais, tai ir rezultatas bus gautas taip pat sveikais skaičiais. Simpleksiniai žingsniai atliekami lentelių, tokių, kaip ankstesniame pavyzdyje parodyta 1 lentelė, pagalba (žr. [5], [2]). Pats metodas vadinamas *potencialų metodu.* Mes jo detaliai nenagrinėsime. Vietoj to geriau panagrinėkime papildomas galimybes, kurias suteikia transporto tipo uždavinių interpretavimas ir analizė.

Transporto uždavinio sprendimas, panaudojant Microsoft Excel skaičiuoklę, parodytas §13.

5°.Transporto uždaviniai su papildomomis sąlygomis.

Uždavinys su nelygiaverčiais gavėjais. Transporto uždavinyje su fiktyviu ($m+1$) tiekėju, kaip jau matėme anksčiau, kai kurie gavėjai gauna produktą iš šio fiktyvaus tiekėjo, kitaip tariant, viso pageidaujamo produkto kiekio jie iš tikrųjų negauna. Paprastai optimaliame plane viso produkto negauna tie gavėjai, kuriems vežti brangiausia. Tačiau gali taip atsitikti, kad kai kurie iš gavėjų yra „privilegijuoti“ – jų poreikiai privalo būti patenkinti. Vadinasi, reikia padaryti taip, kad šie gavėjai gautų produktą iš realių, bet ne iš fiktyvaus tiekėjo. Norint to pasiekti, užtenka vietoj nulinės nustatyti labai didelę pervežimo kainą iš fiktyvaus $m+1$-jo tiekėjo „privilegijuotam“ gavėjui. Sakykim, r-tasis gavėjas yra „privilegijuotas“. Tada vietoj $c_{m+1,r} = 0$ (kaip darėme anksčiau) nustatome $c_{m+1,r} = \infty$.[5] Dabar iš fiktyvaus tiekėjo „vežti“ produktą r-tajam gavėjui tikrai neapsimokės, ir šis gavėjas gaus visą pageidautą produktą iš realių tiekėjų, t.y. gaus jį iš tikrųjų.

Panaši situacija yra ir tada, kai vežėjas patiria nuostolius dėl gavėjams neatvežto produkto. Sakykim, jei produkto vienetas neatvežamas r-tajam gavėjui, vežėjas moka c_r litų baudą už kiekvieną neatvežtą produkto vienetą. Norint į visas vežėjo išlaidas įtraukti ir galimas baudas, užtenka nurodyti produkto vieneto „pervežimo“ iš fiktyvaus tiekėjo r-tajam gavėjui kainą $c_{m+1,r} = c_r$. Taip padarius, uždavinio optimaliame plane bus atsižvelgta ne tik į realių pervežimų išlaidas, bet ir į išlaidas, atsirandančias dėl „pervežimų“ iš fiktyvaus tiekėjo, t.y. į galimas baudas. Tikėtina, kad optimalus planas šiuo atveju bus kitoks, negu anksčiau, kai visi $c_{m+1,j} = 0$.

[5] Spręsdami uždavinį kompiuteriu, paprastai vietoj begalybės paimame kokį nors skaičių, daug didesnį už visas kitas pervežimų kainas.

Uždavinys su nelygiaverčiais tiekėjais. Galime susidurti ir su priešinga situacija, kai tiekėjų atsargos didesnės už gavėjų poreikius, todėl transporto uždavinyje turime fiktyvų ($n+1$) gavėją. Šiuo atveju kai kurių tiekėjų turimos atsargos bus „išvežtos“ fiktyviam gavėjui, taigi iš tikrųjų liks neišvežtos. Kaip ir anksčiau, optimaliame plane tikriausiai liks neišvežtos tų tiekėjų atsargos, iš kurių pervežimai daugiausia kainuoja. Tačiau ir šiuo atveju kai kurie tiekėjai gali būti „privilegijuoti“ ta prasme, kad visą jų turimą produktą būtina išvežti. Sakykim, k-tasis tiekėjas neturi galimybių saugoti likusį neišvežtą produktą, todėl jį visą būtina iš šio tiekėjo išvežti. Tada vietoj $c_{k,n+1} = 0$ (kaip darėme bendru atveju) nustatome $c_{k,n+1} = \infty$. Dabar iš k-tojo tiekėjo „vežti“ produktą fiktyviam gavėjui tikrai neapsimokės, ir visa šio tiekėjo turima atsarga bus išvežta realiems gavėjams, t.y. išvežta iš tikrųjų.

Gali atsitikti ir taip, kad likusio iš tiekėjo neišvežto produkto saugojimas kažkiek kainuoja, ir vežėjas turi tą kainą sumokėti. Arba neišvežtas produktas tiesiog sugenda, ir dėl to atsiranda nuostoliai. Sakykim, jei produkto vienetas lieka pas k-tąjį tiekėją, vežėjui tai sudaro c_k litų nuostolio už kiekvieną neišvežto produkto vieneto saugojimą (ar praradimą). Norint į visas vežėjo išlaidas įtraukti ir šiuos nuostolius, reikia nurodyti produkto vieneto „pervežimo“ iš k-tojo tiekėjo fiktyviam gavėjui kainą $c_{k,n+1} = c_k$. Taip padarius, uždavinio optimaliame plane bus atsižvelgta ne tik į realių pervežimų išlaidas, bet ir į išlaidas, atsirandančias dėl „pervežimų“ fiktyviam gavėjui, t.y. į galimas neišvežto produkto sandėliavimo ir kitokias išlaidas. Tikėtina, kad optimalus planas šiuo atveju bus kitoks, negu anksčiau, kai visi $c_{i,n+1} = 0$.

Uždavinys su privalomais pervežimais. Ši uždavinio versija atsiranda, kai reikalaujama būtinai iš k-tojo tiekėjo r-tajam gavėjui pervežti ne mažiau kaip d vienetų produkto, t.y. reikalaujama, kad $x_{kr} \geq d$. Taip gali atsitikti tada, kai vežamas produktas ne visiškai vienodas, pavyzdžiui, r-tasis gavėjas pageidauja ne mažiau kaip d vienetų būtent k-tojo tiekėjo paruošto produkto. Atsižvelgti į šį reikalavimą labai paprasta. Laikome, kad d vienetų produkto jau pervežta, todėl k-tojo tiekėjo atsargą sumažiname dydžiu d iki $a_k - d$, r-tojo gavėjo poreikį iki $b_r - d$, ir toliau sprendžiame uždavinį kaip įprasta. Gavus sprendinį, beliks prie dydžio x_{kr} pridėti d.

Uždavinys su uždraustais pervežimais. Kai kuriose transporto situacijose gali atsitikti, kad kai kurių tiekėjų produkto negalima vežti arba jis tiesiog netinkamas kai kuriems gavėjams. Atsižvelgti į šį reikalavimą labai paprasta. Sakykim, produkto negalima vežti iš k-tojo tiekėjo r-

tajam gavėjui. Nustatę pervežimo kainą $c_{k,r} = \infty$, optimaliame plane išvengsime teigiamos $x_{k,r}$ reikšmės, t.y. uždrausto pervežimo.

Uždavinys su ribotais pervežimais. Sudėtingiau atsižvelgti į reikalavimą ne uždrausti, o tik apriboti pervežimo apimtį. Sakykim, reikalaujama, kad iš k-tojo tiekėjo r-tajam gavėjui būtų vežama ne daugiau kaip d vienetų produkto. Norėdami atsižvelgti į šį reikalavimą, r-tąjį gavėją dirbtinai suskaidysime į du: $r1$-ąjį ir $r2$-ąjį. $r1$-ojo gavėjo poreikį nustatysime lygų d vienetų, o likusį r-tojo gavėjo poreikį priskirsim $r2$-ajam, nustatydami jį lygų $b_r - d.$ Pervežimo kainas iš visų tiekėjų $r1$-ajam gavėjui paliksime tokias pat, kaip ir r-tajam: $c_{i,r1} = c_{i,r}\ (i = \overline{1,m}).$ Pervežimo kainas iš visų tiekėjų $r2$-ajam gavėjui taip pat paliksime tokias pat, kaip ir r-tajam: $c_{i,r2} = c_{i,r}$ visiems i, išskyrus $c_{k,r2} = \infty.$ Šių veiksmų dėka iš k-tojo tiekėjo $r1$-ajam gavėjui bus galima pervežti ne daugiau kaip d vienetų produkto (nes toks jo poreikis), o $r2$-ajam gavėjui vežti bus uždrausta. Taigi sujungus $r1$-ąjį ir $r2$-ąjį gavėjus atgal į vieną, iš k-tojo tiekėjo jam bus pervežta ne daugiau kaip d vienetų produkto. Kitų tiekėjų požiūriu pervežimų kainos $r1$-ajam ir $r2$-ajam gavėjui vienodos, taigi optimaliam sprendimui r-tojo gavėjo dirbtinis suskaidymas įtakos neturės. Išsprendę uždavinį, apjungsime abu gavėjus į vieną ir gausime sprendinį, tenkinantį suformuluotą pervežimo apimties apribojimą.

6°.Kai kurie kiti uždaviniai, formuluojami transporto uždavinio pagrindu. Transporto uždavinys gali būti pagrindu daugeliui uždavinių, kuriuose iš viso nenagrinėjami jokie pervežimai arba kalbama ne tik apie juos. Išnagrinėsime keletą pavyzdžių.

Gamybinių pajėgumų išdėstymo uždavinys. Yra m miestų, kuriuose galima statyti įmones, gaminsiančias tam tikrą produktą. Miestų pajėgumai yra $a_1,\ a_2,\ \ldots,\ a_m$ - tiek maksimaliai produkto galima pagaminti šiuose miestuose. Žinomi ir šio produkto poreikiai visuose miestuose. Juos žymėsime $b_1,\ b_2,\ \ldots,\ b_m.$ Numatoma, kad, pastačius įmones, produktas bus vežamas iš miestų gamintojų į visus miestus. Žinomos produkto vieneto pervežimų kainos iš kiekvieno miesto į kiekvieną: $c_{ij}\ \ (i = \overline{1,m}, j = \overline{1,n}).$ Reikia nuspręsti, kuriuose miestuose ir kokio dydžio įmones statyti, kad miestų poreikiai būtų patenkinti, o gaminamą produkciją vežioti būtų pigiausia[6].

Laikysime, kad $\sum_{i=1}^{m} a_i > \sum_{j=1}^{m} b_j$, t.y. potencialūs gamybos pajėgumai didesni už planuojamus poreikius. Jei būtų priešingai, uždavinys sprendinio neturėtų – poreikių tiesiog nebūtų galima patenkinti. Jei galiotų lygybė $\sum_{i=1}^{m} a_i = \sum_{j=1}^{m} b_j$, tai uždavinys virstų paprasčiausiu transporto uždaviniu: kiekviename mieste reikėtų statyti maksimalaus galingumo įmonę (kitaip nepatenkintume poreikių), o paskui išspręsti subalansuotą transporto uždavinį, kad optimaliai sureguliuotume pervežimų srautus.

Kadangi laikome, kad potencialūs gamybos galingumai didesni už poreikius, turime spręsti, kuriuos iš jų panaudoti visiškai, kuriuos tik ir dalies, o kurių galima ir nepanaudoti. Gali būti, kad kai kuriuose miestuose įmonių statyti neapsimokės iš viso.

Kaip įprasta transporto uždavinyje, papildysim mūsų uždavinį fiktyviu m+1-ju miestu vartotoju, kurio poreikį nustatysime lygų $b_{m+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{m} b_j$. Pervežimo kainas iš visų miestų-gamintojų nustatysime lygias nuliui $c_{i\,m+1} = 0\ (i = \overline{1,m})$, kadangi faktiškai joks produktas fiktyviojo miesto poreikiams nebus gaminamas ir nebus vežamas. Taip gausime subalansuotą transporto uždavinį su m tiekėjų ir m+1 gavėju. Jį išsprendę, sužinosime optimalius pervežimų srautus $x_{ij}\ (i = \overline{1,m}, j = \overline{1,m+1})$. Beliks apskaičiuoti, kokio galingumo įmonę kuriame mieste statyti. Atsakyti į šį klausimą labai paprasta: i-tajame mieste statomos įmonės galingumas x_i turi būti lygus į visus realius (bet ne į fiktyvų!) miestus išvežamo produkto kiekiui:

$$x_i = \sum_{i=1}^{m} x_{ij} \quad (i = \overline{1,m}).$$

Atsargų valdymo uždavinys. Įmonė planuoja darbą m dienų į priekį. Žinoma, kad tomis dienomis ji gali pagaminti $a_1, a_2, \dots, a_m$ vienetų tam tikro produkto. Šio produkto poreikis, kurį įmonė privalo patenkinti, tomis dienomis yra $b_1, b_2, \dots, b_m$. Poreikiai ir gamybos apimtys gali smarkiai skirtis: kai kuriomis dienomis galima pagaminti daug, bet poreikis mažas; kitomis dienomis gamyba, gal būt, iš viso nevyksta, o poreikį patenkinti būtina. Todėl įmonė ketina gaminti produktą atsargai, kad galėtų jos dėka patenkinti poreikį visomis dienomis. Tačiau produkto atsargos sandėliavimas kainuoja: produkto vieneto sandėliavimo vienos paros bėgyje kainą

[6] Paprasčiausioje uždavinio versijoje laikoma, kad visų įmonių statybos kaina pastovi ir nepriklauso nuo miesto. Sudėtingesnėse uždavinio versijose atsižvelgiama ir į statybos kainas.

žymėsime raide s. Taip pat atsižvelgsime į produkto vieneto gamybos kainą i-tąją dieną. Šią kainą žymėsime g_i (gamybos kainos gali būti nevienodos – kai kuriomis dienomis gali tekti mokėti darbininkams už viršvalandžius ar gali būti kitų priežasčių).

Reikia sudaryti įmonės gamybos planą, kad poreikiai būtų patenkinti, o gamybos ir sandėliavimo išlaidos minimalios.

Laikysime, kad $a_1 \geq b_1,\ a_1+a_2 \geq b_1 + b_1$ ir t.t., kitaip tariant, patenkinta sąlyga

$$\sum_{i=1}^{k} a_i \quad \geq \sum_{i=1}^{k} b_i \quad (k = \overline{1,m}).$$

Ši sąlyga reiškia, kad per bet kurias pirmąsias k dienų galima pagaminti ne mažiau produkto, negu per tas dienas reikia poreikiui patenkinti. Jei ši sąlyga negaliotų, be abejo, uždavinio išspręsti nebūtų galima.

Norėdami užrašyti matematinį šio uždavinio modelį, pradėkime nuo nežinomųjų paskyrimo. Sakykim, x_{ij} yra produkto kiekis, gaminamas i-tąją dieną ir skirtas j-tosios dienos poreikiui $(j \geq i)$. Dydžiai x_{ij} turi tenkinti šias sąlygas:

$$\sum_{j=i}^{m} x_{ij} \leq a_i \quad (i = \overline{1,m}). \tag{8}$$

$$\sum_{i=1}^{j} x_{ij} = b_j \quad (j = \overline{1,m}). \tag{9}$$

$$x_{ij} \geq 0 \ (i = \overline{1,m}, j = \overline{1,m}\ ). \tag{10}$$

Sąlyga (8) reiškia, kad jokią dieną šios ir visų vėlesnių dienų poreikiui patenkinti negali būti pagaminta daugiau, nei tą dieną įmanoma pagaminti; sąlyga (9) reikalauja, kad iki bet kurios dienos imtinai būtų pagaminta tiek produkto, kiek jo reikia tos dienos poreikiui patenkinti.

Dabar suformuluokim uždavinio tikslo funkciją. Atkreipkime dėmesį, kad produkto vieneto gamyba i-tąją dieną j-tosios dienos vartojimui kainuoja g_i litų plius $s(j-i)$ litų sandėliavimo išlaidų. Todėl, pažymėję $c_{ij} = g_i + s(j-i)$, gauname, kad su x_{ij} vienetų gamyba susijusios išlaidos yra $c_{ij}x_{ij}$, o visų išlaidų minimizavimą galima užrašyti taip:

$$(\min) \sum_{i=1}^{m} \sum_{j=i}^{m} c_{ij} x_{ij} \tag{11}$$

Gautas (8)-(11) uždavinys labai panašus į transporto uždavinį, tačiau nuo jo skiriasi nelygybėmis ir nežinomųjų skaičiumi (nėra nežinomųjų x_{ij}, kai $j < i$). Todėl jo sprendimui standartinė transporto uždavinio programa gali netikti. Tačiau tai ne bėda – (8)-(11) uždavinį nesunku paversti ekvivalenčiu transporto uždaviniu. Parodysime, kaip tai padaryti.

Kaip jau įprasta, norėdami uždavinį užrašyti lygybėmis, papildome jį fiktyvia $m+1$-ąja dieną, kurią „sunaudojamas“ visas galimas produkto perteklius. Kaip jau susitarėme anksčiau, tai reiškia, kad perteklinis produktas tiesiog negaminamas. Todėl jo gamybos ir sandėliavimo išlaidos – nuliai: $c_{i,m+1} = 0 \ (i = \overline{1,m})$. Be to, norėdami pereiti prie standartinio transporto uždavinio, papildykime mūsų uždavinį nežinomaisiais x_{ij}, kai $j < i$. Teigiami tokių nežinomųjų dydžiai reikštų, kad kažkokias produkto apimtis pagaminame pavėluotai. Kadangi tai pagal sąlygą uždrausta, nustatykime $c_{ij} = \infty$, kai $j < i$. To dėka, kaip jau aiškinomės uždavinio su uždraustais pervežimais atveju, atitinkamos x_{ij} reikšmės bus lygios nuliui, ir vėlavimo patenkinti poreikį išvengsime. Padarytų pakeitimų dėka uždavinys bus užrašomas taip:

$$\sum_{j=1}^{m+1} x_{ij} = a_i \quad (i = \overline{1,m}). \tag{12}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,m+1}). \tag{13}$$

$$x_{ij} \geq 0 \ (i = \overline{1,m}, j = \overline{1,m+1}\ ). \tag{14}$$

$$(\min) \ \sum_{i=1}^{m} \sum_{j=i}^{m+1} c_{ij} x_{ij}. \tag{15}$$

Matematiniu (ir sprendimo) požiūriu tai įprastas transporto uždavinys su m „tiekėjų“ ir m+1 „gavėju“, nors iš tikrųjų jo interpretavimas visai kitoks.

<u>Paskyrimų uždavinys.</u> Šiame uždavinyje reikia paskirti n kandidatų į n vietų (pavyzdžiui, n įrengimų priskirti n darbams atlikti ir pan.) Žinoma, kad i-tojo kandidato skyrimas į j-tąją vietą kainuoja c_{ij} litų. Reikia taip paskirti kandidatus į vietas, kad bendra išlaidų suma būtų minimali.

Šio uždavinio atveju taip pat pasinaudosime transporto uždavinio galimybėmis. Kaip visada, pradėkime nuo nežinomųjų paskyrimo. Naudokimės nežinomaisiais x_{ij}, kuriems leiskime turėti tik dvi reikšmes: $x_{ij} = 1$, jei i-tasis kandidatas paskirtas į j-tąją vietą $x_{ij} = 0$, jei nepaskirtas (tokie nežinomieji vadinami dichotominiais nežinomaisiais).

Kadangi i-tasis kandidatas gali būti paskirtas tik į vieną vietą, tai apskaičiavę sumą $x_{i1} + x_{i2} + \cdots + x_{in}$, galime gauti tik vienetą, nes tik vienas iš dėmenų bus lygus vienetui, o visi kiti nuliai. Šį faktą visiems kandidatams galime užrašyti kaip n lygčių:

$$\sum_{j=1}^{n} x_{ij} = 1 \ \ (i = \overline{1,n}). \tag{16}$$

Panašiai samprotausime ir apie kiekvieną vietą: suma $x_{1j}+x_{2j}+\cdots+x_{nj}$ taip pat bus lygi vienetui, nes į j-tąją paskyrimo vietą gali būti paskirtas tik vienas kandidatas. Ir šį faktą užrašysime kaip n lygčių visoms paskyrimo vietoms:

$$\sum_{i=1}^{n} x_{ij} = 1 \ \ (j=\overline{1,n}). \tag{17}$$

Belieka nurodyti įprastą sąlygą

$$x_{ij} \geq 0 \ \ \ (i=\overline{1,n}, j=\overline{1,n}). \tag{18}$$

bei tikslo funkciją

$$(\text{min}) \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{19}$$

Uždavinys (16)-(19) savo forma yra įprastas transporto uždavinys, kurį galime spręsti įprastais metodais. Tačiau ar išsprendę uždavinį negausime neleistinų x_{ij} reikšmių? Iš (16)-(19) lygybių aišku, kad $0 \leq x_{ij} \leq 1$, taigi neigiamų ir didesnių už vienetą reikšmių nebus. Tačiau mūsų netenkina ir trupmeninės x_{ij} reikšmės. Ar tikrai mūsų uždavinio optimalus planas bus sveikaskaitinis? Dar 4^0 minėjome, kad, sprendžiant transporto uždavinį potencialų metodu, su pradiniu planu atliekami tik sudėties ir atimties veiksmai. Pradinį planą galime parinkti taip, kad x_{ij} reikšmės būtų tik nuliai ir vienetai. Atliekant sudėties ir atimties veiksmus, tos reikšmės negali pasidaryti trupmeninės, vadinasi gautas optimalus planas bus sveikaskaitinis. (Jei optimalus planas ne vienintelis, gali būti ir optimalių planų su nesveikaskaitinėmis reikšmėmis, tačiau mums jie neįdomūs, nes neatitinka uždavinio formulavimo.)

§ 11. TIESINIO DISKREČIOJO PROGRAMAVIMO UŽDAVINIAI

1°.Diskrečiojo programavimo esmė ir uždavinių klasifikavimas. Iki šiol daugiausia nagrinėjome tokias situacijas, kuriose planuojamų dydžių reikšmės galėjo būti ir sveikos, ir trupmeninės. Tačiau labai dažnai praktikoje tenka susidurti su objektais, kurių dydžiai gali būti matuojami tik sveikais skaičiais arba skaičiais iš tam tikros diskrečios (baigtinių ar suskaičiuojamų reikšmių) aibės. Pavyzdžiui, galima statyti tik baigtinį namų skaičių, maršruto linija gali kursuoti tik sveikas autobusų skaičius, iš detalių komplektuojant gaminį, komplektų skaičius taip pat turi būti tik sveikas skaičius.

Tiesinio programavimo uždavinys, kurio visi arba dalis nežinomųjų gali įgyti reikšmes tik iš baigtinės ar suskaičiuojamos reikšmių aibės, vadinamas *tiesinio diskrečiojo programavimo* uždaviniu. Jo atskiras ir dažniausia pasitaikantis atvejis yra *tiesinio sveikaskaitinio programavimo* uždavinys, kurio nežinomųjų reikšmės privalo būti sveiki skaičiai. Jei tik dalies nežinomųjų reikšmės turi būti sveiki skaičiai, uždavinį vadiname *dalinai sveikaskaitiniu.*

Diskrečiojo programavimo uždavinius galime suskirstyti į šias grupes:

1. Transporto tipo uždaviniai. Šiuose uždaviniuose, kaip matėme § 11 4^0, visada yra optimalių sveikaskaitinių planų, jei pradiniai duomenys – sveiki skaičiai. Taigi šiuos uždavinius galima spręsti įprastais metodais, specialiai nesirūpinant, kad gautas sprendinys būtų sveikaskaitinis. Paskyrimų uždavinys, nagrinėtas § 11 6^0, yra tokio tipo uždavinio pavyzdys.

2. Nedalių objektų programavimo uždaviniai. Tai uždaviniai, kuriuose planuojami objektai, matuojami tik sveikais skaičiais (pavyzdžiui, namai, transporto priemonės, komplektai ir pan.). Optimalūs planai, surasti įprastais metodais, dažnai nėra išreiškiami sveikais skaičiais, todėl reikalinga taikyti specialius sprendimo metodus.

3. Kombinatoriniai uždaviniai. Šiuose uždaviniuose su kiekvienu tam tikros grupės objektų pertvarkiu siejama tam tikra nauda ar sąnaudos. Reikia rasti geriausią iš galimų pertvarkių (geriausią maršrutą per žinomus punktus, geriausią darbų atlikimo tvarką, geriausią įrengimų priskyrimo darbams būdą ir pan.).

4. Uždaviniai su neiškiliomis arba nesusijusiomis leistinų planų aibėmis ir uždaviniai su netolydžiomis tikslo funkcijomis. Šių uždavinių pradiniame formulavime sveikaskaitinių nežinomųjų gali ir nebūti, tačiau jie atsiranda kaip pagalbinė šios grupės uždavinių sprendimo priemonė.

Be abejo, gali būti ir uždavinių, turinčių iš karto kelių grupių bruožus. Paskyrimų uždavinys, pavyzdžiui, yra ir transporto tipo, ir kombinatorinis uždavinys.

Transporto uždavinį jau nagrinėjome, todėl dabar susipažinkime su kai kuriais kitų grupių uždaviniais.

2°. Nedalių objektų programavimo uždaviniai. Praktikoje neretai pasitaiko uždaviniai, kuriuose gamyba planuojama ne atskirais gaminiais, o jų komplektais. Būdingą tokio tipo uždavinio pavyzdį dabar suformuluosime.

<u>Karpymo uždavinys</u>. Turime b brangios medžiagos lapų. Kiekvieną lapą galime karpyti vienu iš žinomų n būdų. Karpydami lapus, gauname detales, kurios yra m skirtingų rūšių. Žinome, kad, kirpdami vieną lapą j-tuoju būdu, gauname a_i^j i-tosios rūšies detalių ($i = \overline{1,m}, j = \overline{1,n}$). Iš detalių sudaromi ir panaudojami komplektai (pavyzdžiui, siuvami drabužiai, iškirptomis plokštėmis dengiamos pastato sienos ir pan.). Todėl svarbus ne pačių pagamintų detalių, o pilnų komplektų skaičius. Žinome, kad į komplektą įeina k_i i-tosios rūšies detalių, taigi vieno komplekto sudėtį nurodo vektorius $\boldsymbol{k} = (k_1, k_2, \dots, k_m)$. Reikia taip sukarpyti lapus, kad gautume kuo didesnį komplektų skaičių.

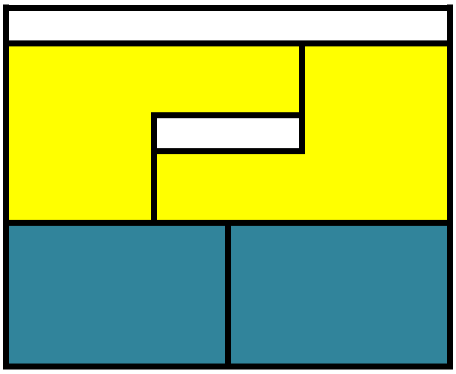
Kaip ir visad, sudarant matematinį nagrinėjamos situacijos modelį, bene svarbiausia yra gerai paskirti nežinomuosius. Kadangi turime nuspręsti, kiek lapų kuriuo būdu karpyti, laikykime, kad j-tuoju būdu karpysime x_j lapų ($j = \overline{1,n}$). Suprantama, kad $x_1 + x_2 + \cdots + x_n \leq b.$, t.y. negalime sukarpyti lapų daugiau, negu turime. Raide z pažymėkime nežinomąjį komplektų skaičių, kurį gausime sukarpę visus lapus.

Suskaičiuokime, kiek gausime i-tosios rūšies detalių. Iš lapų, karpomų pirmuoju būdu, gausime $a_i^1 x_1$ i-tosios rūšies detalių, iš lapų, karpomų antruoju būdu - $a_i^2 x_2$ šios rūšies detalių ir t.t. Taigi iš viso i-tosios rūšies detalių turėsime $a_i^1 x_1 + a_i^1 x_1 + \cdots + a_i^1 x_1 = \boldsymbol{a_i x}$ vienetų. Šis skaičius turėtų būti ne mažesnis už zk_i , nes tiek i-tosios rūšies detalių reikia, norint sudaryti z komplektų. Sąlygas $\boldsymbol{a_i x} \geq zk_i$ ($i = \overline{1,m}$) galime užrašyti matricomis kaip $A\boldsymbol{x} \geq z\boldsymbol{k}$, o visą karpymo uždavinį tokiu būdu:

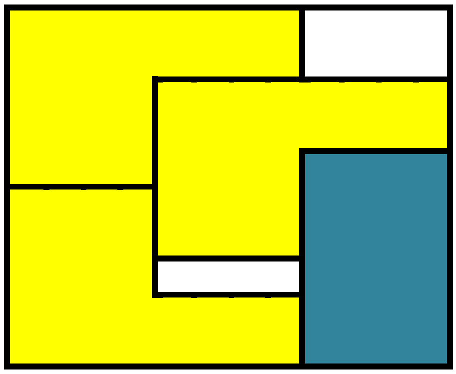
$$
\begin{gathered}
(\max)\ z \\
A\boldsymbol{x} - z\boldsymbol{k} \geq \boldsymbol{0}. \\
x_1 + x_2 + \cdots + x_n \leq b. \\
\boldsymbol{x} \geq \boldsymbol{0},\ z \geq 0 \text{ - sveiki skaičiai}
\end{gathered}
\tag{1}
$$

Gavome tiesinio programavimo uždavinį, kuris nuo nagrinėtų anksčiau skiriasi viena svarbia sąlyga – visos jo nežinomųjų reikšmės turi būti sveiki skaičiai (nei trupmeninio kuriuo nors būdu karpomų lapų skaičiaus, nei trupmeninio komplektų skaičiaus negali būti.)

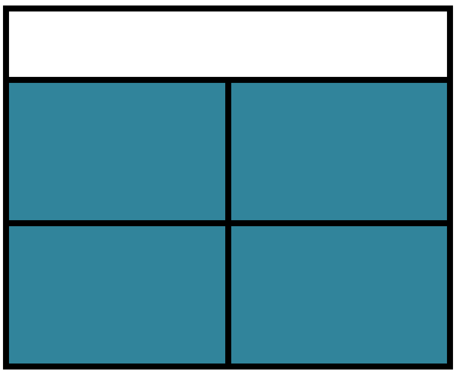
Pavyzdys. Turime 114 stačiakampių brangios medžiagos lapų. Kiekvieną lapą galime karpyti vienu iš trijų brėžiniuose nurodytų būdų. Taip gauname dviejų rūšių detales.



Pirmasis karpymo būdas



Antrasis karpymo būdas



Trečiasis karpymo būdas

Pirmuoju karpymo būdu iš vieno lapo gauname po dvi pirmosios ir antrosios rūšies detales; antruoju – tris pirmosios rūšies ir vieną antrosios rūšies, o trečiuoju – keturias antrosios rūšies detales.

Detalių komplektą sudaro 3 pirmosios rūšies ir 7 antrosios rūšies detalės. Reikia gauti maksimalų komplektų skaičių.

Pažymime x_1 pirmu, x_2 antru, o x_3 – trečiu būdu karpomų lapų skaičių. Apskaičiuojame, kad pirmos rūšies detalių šiuo atveju turėsime $2x_1 + 3x_2$ vienetų, ir šis dydis turi būti ne mažesnis už $3z$, t.y. už z komplektams sudaryti reikalingą pirmosios rūšies detalių skaičių. Analogiškai apskaičiuojame, kad antros rūšies detalių turėsime $2x_1 + x_2 + 4x_3$ vienetų, ir šis dydis turi būti ne mažesnis už $7z$. Taigi turi būti $2x_1 + 3x_2 \geq 3z, 2x_1 + x_2 + 4x_3 \geq 7z$. Pertvarkę narius ir atsižvelgę į turimą lapų skaičių, visą uždavinį užrašome taip:

$$
\begin{gathered}
(\max)\ z \\
2x_1 + 3x_2 \qquad - 3z \geq 0 \\
2x_1 + x_2 + 4x_3 - 7z \ \geq 0 \\
x_1 + x_2 + x_3 \leq 114 \\
x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ z \geq 0 \text{ - sveiki skaičiai}
\end{gathered}
$$

Šio uždavinio sprendimą nagrinėsime § 12. ⋞

3°. Kombinatoriniai uždaviniai. Vienas iš būdingų tokio pobūdžio uždavinių – paskyrimų uždavinys, nagrinėtas §10 6°. Dėl savo specifikos jis formuluojamas ir sprendžiamas kaip transporto uždavinys. Tačiau jei kandidatų ir vietų (darbų ir įrengimų) skaičius nesutampa, uždavinys vadinamas *ekspertų* arba *padengimų* uždaviniu, ir sprendžiamas gerokai sudėtingiau.

Kuprinės uždavinys. Turime b tūrio „kuprinę“, į kurią norėtume sudėti n naudingų „daiktų“. Kiekvienas įdėtas „daiktas“ užima kuprinėje tūrį a^j $(j = \overline{1,n})$. Įdėto daikto naudingumas vertinamas dydžiu $c^j (j = \overline{1,n})$. Kadangi visi „daiktai“ į „kuprinę“ netelpa, reikia nustatyti, kuriuos „daiktus“ įdėti, kad gauta nauda būtų didžiausia.

„Kuprinė“, „daiktai“ ir „nauda“ įvairiose situacijose gali būti įvairiai interpretuojami. Pavyzdžiui, norimus išvežti krovinius reikia sukrauti į ribotą traukinio vagonų tūrį, o kiekvieno krovinio išvežimas apmokamas tam tikra suma (arba nereikia mokėti baudos už neišvežimą laiku). Kitas pavyzdys: reikia pasirinkti, kuriuos iš įvairiai apmokamų ir nevienodai trunkančių darbų verta atlikti per tam tikrą ribotą laiką. Panašiai galime spręsti, į kuriuos iš daugelio galimų projektų verta investuoti ribotą pinigų sumą, žinant laukiamą kiekvieno projekto naudą.

Matematinį kuprinės uždavinio formulavimą užrašysime, naudodami specialius nežinomuosius x_j $(j = \overline{1,n})$, kuriems leisime įgyti tik dvi reikšmes: $x_j = 1$, jei j-tasis „daiktas“ dedamas į „kuprinę“ ir $x_j = 0$, jei j-tasis „daiktas“ nededamas. Tada j-tojo „daikto“ užimtas „kuprinėje“ tūris bus $a^j x_j$, t.y. a^j, jei „daiktas“ įdėtas, ir 0 priešingu atveju. Taigi visų „daiktų“ užimtas tūris bus $a^1x_1 + a^2x_2 + \cdots + a^nx_n = \boldsymbol{ax}$. Analogiškai apskaičiuojame ir įdėtų „daiktų“ naudą: $c^1x_1 + c^2x_2 + \cdots + c^nx_n = \boldsymbol{cx}$. Taigi kuprinės uždavinį galime užrašyti tokiu pavidalu:

$$(\max)\ \boldsymbol{cx}\ ;\ \boldsymbol{ax} \leq b;\ \ 0 \leq x_j \leq 1\ (j = \overline{1,n}) \text{ - sveiki skaičiai.} \qquad (2)$$

Čia panaudoti nežinomieji, įgyjantys tik dvi reikšmes – nulį ir vienetą – vadinami *binariniais* arba *dichotominiais* (kaip minėjome paskyrimų uždavinyje) arba *buliniais* (pagal matematiko G. Boole, nagrinėjusio tokių nežinomųjų sistemas, pavardę). Matematinio programavimo uždaviniai su tokiais nežinomaisiais kartais vadinami *bulinio programavimo* uždaviniais.

Kuprinės uždavinys gali būti papildytas įvairiomis sąlygomis ir apibendrinimais. Pavyzdžiui, sąlyga $x_i \geq x_j$ reiškia, kad, jei įdėtas j-tasis „daiktas", tai būtinai turi būti įdėtas ir i –tasis. Dedamiems į „kuprinę" „daiktams" gali būti keliami ne tik tūrio, bet ir svorio, pakrovimo trukmės bei kitokie ribojimai. Tada kuprinės uždavinys vadinamas daugiamačiu, o (2) formulėse vietoj vieno apribojimo $\boldsymbol{ax} \leq b$ turime visą apribojimų sistemą $A\boldsymbol{x} \leq \boldsymbol{b}$. Dar vienas variantas gaunamas, kai įvairių vienos rūšies daiktų turime ne po vieną, o po kelis. Tokiu atveju nežinomieji x_j gali įgyti ir didesnes už vienetą sveikas reikšmes.

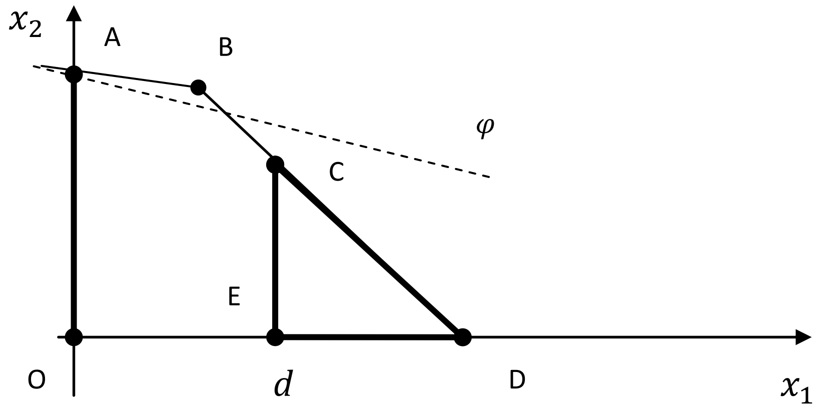
Komivojažieriaus (keliaujančio prekiautojo) uždavinys – vienas iš dažniausiai minimų kombinatorinių uždavinių. Keliaujantis prekiautojas turi aplankyti tam tikrą skaičių miestų, nė į vieną jų neužsukdamas du kartus, ir sugrįžti atgal į pradinį miestą. Žinant kelionės tarp bet kurių dviejų išlaidas ar trukmę, reikia sudaryti pigiausią (greičiausią, trumpiausią) maršrutą. Komivojažieriaus uždavinys gali būti taikomas, kai reikia sudaryti optimalų benzovežio maršrutą, gydytojui kuo greičiau aplankyti ligonius ir t.t. Uždavinys gali būti pritaikytas ir tam atvejui, kai planuojama tvarka, kuria automatas turi atlikti tam tikrą darbų ciklą ir pradėti jį iš pradžios. Automato pertvarkymas nuo vieno darbo prie kito trunka nevienodą laiką ar nevienodai kainuoja, todėl darbų ciklo trukmė ar jo išlaidos priklauso nuo darbų atlikimo tvarkos. Uždavinio sprendimas leidžia rasti optimalią darbų atlikimo seką.

Komivojažieriaus uždavinys gali būti užrašytas tiesinio sveikaskaitinio programavimo uždavinio pavidalu, kurio čia nenagrinėsime.

4°. Uždaviniai su neiškiliomis arba nesusijusiomis leistinų planų aibėmis. Išnagrinėkime keletą atvejų, kai pradiniame tiesinio programavimo uždavinio formulavime gali ir nebūti sveikaskaitinių nežinomųjų, bet yra tam tikrų „nestandartinių" sąlygų, į kurias patogu atsižvelgti, papildomai įvedus dichotominius nežinomuosius.

4.1°. Sakykim, standartiniame gamybos planavimo uždavinyje (§2) reikalaujama kurio nors produkto arba visai negaminti arba gaminti ne mažiau kaip d vienetų. Tokio uždavinio leistinų planų aibė dvimačiu atveju gali būti atvaizduota 1 brėžiniu. Kaip jau išsiaiškinome §2, uždavinio be papildomos sąlygos leistinų planų aibę sudarytų keturkampis OABDO. Tačiau šiuo atveju, kai pirmo produkto reikia pagaminti arba d ir daugiau vienetų, arba visai negaminti, leistinų planų aibė skyla į dvi dalis – atkarpą OA ir trikampį CDE. Tokią aibę vadiname *nesusijusia aibe.* Kaip matome iš 1 brėžinio, be papildomos sąlygos optimalus planas buvo taške B, tačiau dabar jis taške A – jei pirmojo produkto negalima gaminti mažiau kaip d vienetų, tai geriau jo visai negaminti.

1 brėžinys



Sąlygą $arba\, x_j \geq d\ arba\, x_j = 0$ galima užrašyti specialiu būdu, naudojant dirbtinį dichotominį nežinomąjį z_j. Imkim kokį nors labai didelį skaičių M kad sąlyga $x_j \geq M$ būtų patenkinta bet kuriame realią prasmę turinčiame plane (paprastai imamas didžiausias kompiuteriu užrašomas skaičius). Naudodami dichotominį nežinomąjį, užrašysime

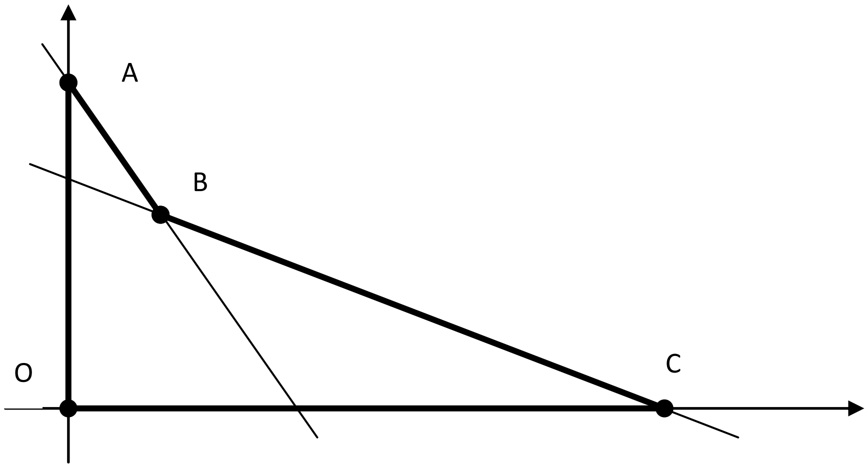
$$x_j - Mz_j \leq 0,\ x_j - dz_j \geq 0;\ 0 \leq z_j \leq 1 - \text{sveiki skaičiai} \qquad (3)$$

Jei, išsprendus uždavinį, $z_j = 0$, tai (3) sąlygos virsta $x_j \leq 0$, $x_j \geq 0$, taigi $x_j = 0$**.** Jei, priešingu atveju, $z_j = 1$, tai (3) sąlygos virsta $x_j \leq M$, $x_j \geq d$. Pirmoji iš šių sąlygų neesminė, nes tenkinama visada, o antroji atspindi papildomą reikalavimą. (3) nelygybės yra tiesinės, ir, įtraukus jas į gamybos planavimo uždavinį, jis lieka tiesinis, bet su vienu sveikaskaitiniu (dichotominiu) nežinomuoju. Be abejo, jei papildomas reikalavimas formuluojamas keliems produktams, tokių dirbtinių dichotominių nežinomųjų gali būti daugiau.

4.2°. Dabar išnagrinėkime kitą standartinio gamybos planavimo uždavinio apibendrinimą. Sakykim, kad iš m gamybos planavimo uždavinio apribojimų $\boldsymbol{a}_i\boldsymbol{x} \leq b_i, (i = \overline{1, m})$ galima nepatenkinti, $k<m,$ t.y. sudaryti planą, nesubalansuotą pagal k iš m apribojimų. Pavyzdžiui, jei turim uždavinį su dviem apribojimais ir reikalaujam, kad būtų patenkintas bent vienas iš jų,

gauname 2 brėžinyje storesniu kontūru apvestą leistinų planų aibę OABCO. Matome, kad šiuo atveju optimalaus sprendinio ieškome neiškiliojoje aibėje.

2 brėžinys



Ir šiuo atveju pasinaudosime dirbtiniais dichotominiais nežinomaisiais $z_i, (i = \overline{1,m})$ bei dideliu skaičiumi $M.$ Standartinio uždavinio apribojimus perrašysime taip:

$$\boldsymbol{a}_i\boldsymbol{x} \le b_i + z_i M, (i = \overline{1,m}),\ \ 0 \le z_i \le 1\ (i = \overline{1,m})\text{ - sveiki skaičiai.} \tag{4.1}$$

$$z_1 + z_2 + \cdots + z_m \le k. \tag{4.2}$$

Jei $z_i = 0$, tai (4.1) virsta įprastu apribojimu $\boldsymbol{a}_i\boldsymbol{x} \le b_i$, taigi i-tasis apribojimas privalo būti patenkintas. Jei $z_i = 1$, tai (4.1) virsta apribojimu $\boldsymbol{a}_i\boldsymbol{x} \le b_i + M$, taigi $\boldsymbol{a}_i\boldsymbol{x}$ reikšmė iš viršaus praktiškai niekuo neribojama. Kitaip tariant, $z_i = 1$ yra tarsi „leidimas“ netekinti i-tojo apribojimo. Nelygybė (4.2) nurodo, kad tokių „leidimų“ gali būti ne daugiau, negu k. Jeigu kai kurie apribojimai privalo besąlygiškai būti patenkinti, tai atitinkamose eilutėse dichotominių nežinomųjų tiesiog neįvedame ir į lygtis jų neįtraukiame. Nelygybes (4) užrašome tik tose eilutėse, kur sutinkame, kad kai kurie iš apribojimų galėtų būti nepatenkinti.

4.3°. Asortimentinis apribojimas. Dabar išnagrinėkime tokį standartinio gamybos planavimo uždavinio variantą, kuriame ribojamas produkcijos asortimentas. Sakykime, iš galimų n produktų rūšių galima gaminti tik $k < n$ rūšių (įmonė didesnės gamybos įvairovės nepajėgi įgyvendinti). Kaip ir anksčiau, pasinaudokime dichotominiais nežinomaisiais $z_j, (j = \overline{1,n})$ bei dideliu skaičiumi $M,$ kurį parinksim taip, kad bet kokia reali x_j reikšmė būtų už jį daug mažesnė. Papildysime mūsų uždavinį sąlygomis

$$0 \le x_j \le z_j M, (j = \overline{1,n}),\ 0 \le z_j \le 1\ (\ i = \overline{1,n})\text{ - sveiki skaičiai.} \tag{5.1}$$

$$z_1 + z_2 + \cdots + z_n \le k. \tag{5.2}$$

Jei $z_i = 0$, tai (5.1) virsta apribojimu $0 \le x_j \le 0$, kitaip tariant, x_j reikšmė „privalomai“ tampa lygia nuliui. Jei $z_i = 1$, tai (5.1) virsta apribojimu $0 \le x_j \le M$, taigi x_j reikšmė iš viršaus praktiškai niekuo neribojama, ir ją nulems kitos uždavinio sąlygos. Todėl $z_i = 1$ yra tarsi „leidimas“ gaminti j-tąjį produktą. Nelygybė (5.2) nurodo, kad tokių „leidimų“ gali būti ne daugiau, negu k.

4.4°. Netolydi tikslo funkcija. Vienas iš svarbių atvejų, kada tenka spręsti uždavinį su netolydžia tikslo funkcija, yra tada, kai reikia atsižvelgti į pastovias ir kintamas kokios nors veikos (gamybos, transportavimo ir pan.) sąnaudas. Sakykim, norint pagaminti x vienetų kurios nors produkcijos, reikia sumokėti cx litų už išteklius (kintamą dalį) ir dar d litų (pastovią dalį) už įrengimų parengimą, žaliavų atgabenimą ir pan. Jei produkcijos negaminame, nei kintamos, nei pastovios dalies mokėti nereikia. Taigi šiuo atveju tikslo funkcija yra tokia:

$$\text{(min) } c(x) = \begin{Bmatrix} cx + d, & kai\ x > 0 \\ 0, & kai\ x = 0 \end{Bmatrix} \tag{6}$$

Ir šiuo atveju galime pasinaudoti dichotominiu nežinomuoju z bei dideliu skaičiumi M. Tikslo funkciją užrašysime taip:

$$\text{(min) } cx + dz, \tag{7}$$

o uždavinio apribojimų sistemą papildysime priklausomybėmis

$$0 \le x \le Mz, 0 \le z \le 1 \text{ - sveikas skaičius.} \tag{8}$$

Jei, išsprendus uždavinį, $z = 0$, tai dėl (8) ir $x = 0$, vadinasi, produkto negaminame, taigi ir išlaidų nėra. Jei vis dėlto $z = 1$, tada produkto gamyba (8) priklausomybėmis neribojama, o pagal (7) įskaičiuojamos jo gamybos išlaidos $cx + d$. (Atvejis $z = 1,\ x = 0$ minimizavimo uždavinyje visada blogesnis už $z = 0,\ x = 0$, ir optimizavimo procedūroje bus automatiškai atmestas.)

Šį būdą atsižvelgti į netolydžią tikslo funkciją galima taikyti ir daugelio nežinomųjų, kurių kiekvienas „įvertinamas“ (6) tikslo funkcija, atveju. Tada kiekvienam nežinomajam priskiriam po dichotominį nežinomąjį ir taikome jau išdėstytą būdą.

§ 12. TIESINIO DISKREČIOJO PROGRAMAVIMO UŽDAVINIŲ SPRENDIMO METODAI

1°. Pagrindinės sprendimo problemos. Susipažinus su diskrečiojo programavimo uždaviniais, gali atrodyti, kad jie sprendžiami lengviau, negu kiti matematinio programavimo uždaviniai, nes jų leistinų planų aibė baigtinė arba suskaičiuojama. Tačiau iš tikrųjų yra atvirkščiai – būtent dėl tokio leistinų planų aibės pobūdžio jie yra vieni sunkiausiai sprendžiamų uždavinių. Todėl, formuluojant optimizavimo uždavinius, patartina, kiek įmanoma, vengti sveikaskaitinių nežinomųjų. Pavyzdžiui, jei gamybos apimtys planuojamos tūkstančiais, šimtais ar net keliomis dešimtimis vienetų, neracionalu jas atitinkančius nežinomuosius laikyti sveikaskaitiniais. Tokiu uždaviniu uždavinį galime spręsti kaip nesveikaskaitinį, o gautą sprendinį suapvalinti. Paklaidos nebus itin didelės.

Visai kas kita, kai planuojamų objektų skaičius nedidelis arba reikia atsižvelgti į aplinkybes, verčiančias naudoti dichotominius nežinomuosius. Šiuo atveju „suapvalintas“ sprendinys gali būti labai toli nuo tikrojo.

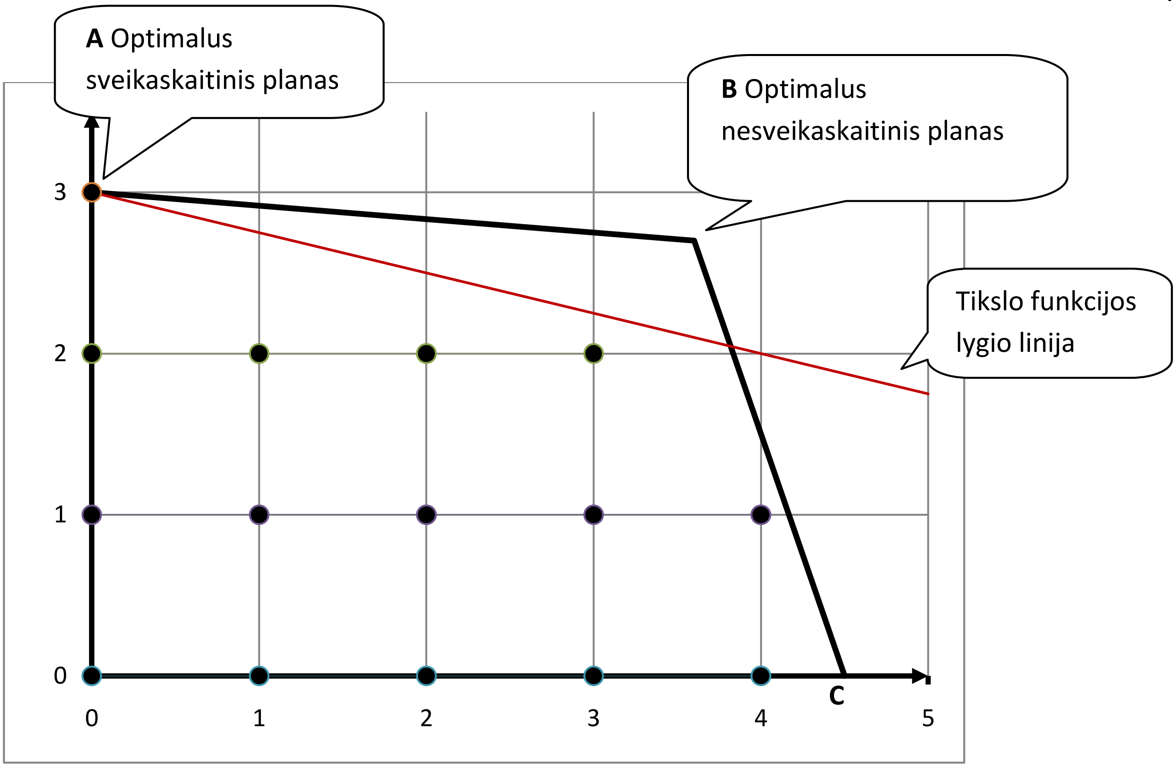
Iliustruosime tai paprastu pavyzdžiu. Imkime uždavinį

$$\begin{gathered}\text{(max) } x_1 + 4x_2 \\ x_1 + 12x_2 \leq 36, \\ 6x_1 + 2x_2 \leq 27. \\ x_1 \geq 0,\ x_2 \geq 0 \text{ - sveiki skaičiai}\end{gathered} \tag{1}$$

Grafinis jo leistinų planų aibės vaizdas parodytas 1 brėžinyje. Leistinų nesveikaskaitinių planų aibę vaizduoja keturkampis OABCO, o paryškinti taškai jo viduje žymi sveikaskaitinius planus. Taip pat parodyta viena iš lygio linijų. Matome, kad optimalus nesveikaskaitinis planas yra taške B. Jo koordinatės (3,6 ; 2,7). Jei šias reikšmes bandysime apvalinti, gausime taškus (4;3), (4;2), (3;3) ir (3;2), iš kurių tik pastarasis leistinas, bet ir tas labai tolimas nuo optimalaus (0;3).

Nepaisant to, apvalinimas kartais naudojamas kaip būdas, leidžiantis gauti kad ir ne optimalų, bet artimą tikslo funkcijos reikšme planą. Nagrinėto pavyzdžio atveju tai būtų taškas (3;2), kuriame tikslo funkcijos reikšmė lygi 11 (optimali reikšmė taške (0;3) yra 12). Tačiau norint

1 brėžinys



surasti tinkamą „suapvalintą“ planą, reikia peržiūrėti visus apvalinamo plano „kaimynus“. Dideliame uždavinyje toks peržiūrėjimas gali būti neracionalus, nes tenka atlikti daug veiksmų, o rezultatas, kaip matėme, nėra patikimas. Pavyzdžiui, uždavinyje su 10 nežinomųjų tektų peržiūrėti 2^{10} „kaimyninių“ planų. Taigi kuo mažesnės laukiamos sveikaskaitinių nežinomųjų reikšmės ir kuo daugiau uždavinyje tų nežinomųjų, tuo labiau apvalinimas netinka netgi kaip apytikris sprendinio suradimo būdas.

Kombinatoriniuose uždaviniuose apvalinimo iš viso negalime pritaikyti. Užtat juose perspektyvus gali atrodyti visų galimų variantų perrinkimo metodas. Tačiau bent kiek sudėtingesniam uždaviniui spręsti šis metodas taip pat netinka dėl gausybės galimų variantų. Pavyzdžiui, paskyrimų uždavinyje (§10 6^0) 20 objektų paskirti į 20 vietų galime $20! \approx 2.4 \cdot 10^{18}$ būdų. Jei peržiūrėtume milijoną variantų per sekundę, tai skaičiavimą baigtume per 77 tūkstančius metų...

Nors perrinkimas ir apvalinimas, kaip matome, nėra tinkami būdai sveikaskaitiniams uždaviniams spręsti, vis dėlto jų pagrindu galima sukurti tam tikrus sprendimo algoritmus, kuriuos trumpai pristatysime.

2°. Šakų ir rėžių metodas. Šis metodas remiasi planų perrinkimo idėja, tačiau perrenkami toli gražu ne visi planai, atmetant neperspektyvias planų „šakas“. Kadangi metodas taikomas ne tik sveikaskaitinio, bet ir apskritai matematinio programavimo uždaviniams spręsti, paaiškinsim jo esmę bendro matematinio programavimo uždavinio atvejui, kurį formuluosime taip:

$$(\max)\ \varphi(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X} \tag{2}$$

Standartinio tiesinio programavimo uždavinio atveju $\varphi(\boldsymbol{x}) = \boldsymbol{cx}$, yra tiesinė tikslo funkcija, o $\boldsymbol{\mathcal{X}} = \{\, \boldsymbol{x} \mid A\boldsymbol{x} \leq \boldsymbol{b},\ \ \boldsymbol{x} \geq \boldsymbol{0} \,\}$ – leistinų planų aibė (žr. §5)

Sprendimą pradedame nuo to, kad kuriuo nors būdu (priklausomai nuo uždavinio specifikos) randame tikslo funkcijos $\varphi(\boldsymbol{x})$ viršutinį rėžį aibėje $\mathcal{X}$, t.y. tokį skaičių, φ, kad $\varphi(\boldsymbol{x}) \leq \varphi$ visiems $\boldsymbol{x} \in \mathcal{X}$.

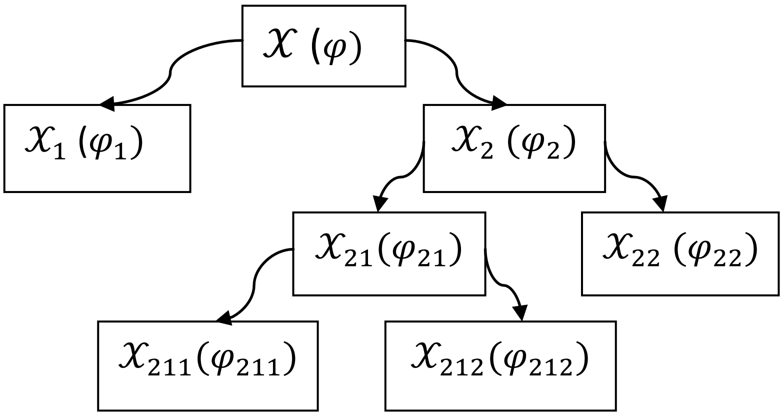
Po to aibę $\mathcal{X}$ kuriuo nors būdu (vėlgi priklausomai nuo uždavinio specifikos) daliname į dvi nesikertančias šakas $\mathcal{X}_1$ ir $\mathcal{X}_2$ taip, kad

$$\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2\,;\ \ \mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset. \tag{2}$$

Kiekvienoje iš šakų vėl randame viršutinį tikslo funkcijos rėžį, t.y. skaičius φ_1 ir φ_2, kad $\varphi(\boldsymbol{x}) \leq \varphi_1$ visiems $\boldsymbol{x} \in \mathcal{X}_1$ ir $\varphi(\boldsymbol{x}) \leq \varphi_2$ visiems $\boldsymbol{x} \in \mathcal{X}_2$. Atkreipkime dėmesį, kad $\varphi_1 \leq \varphi$ ir $\varphi_2 \leq \varphi$, nes dabar rėžius įvertiname aibės $\mathcal{X}$ poaibiuose. (Pavyzdžiui, galime įvertinti, kad studentų auditorijoje nebus aukštesnio kaip 2 metrų ūgio studento; atskirai įvertinę vaikinus ir merginas, gauname, kad nėra aukštesnio už 1,9 metro ūgio vaikino ir 1,8 metro ūgio merginos.)

Imame didesnį iš gautų rėžių (sakykim, $\varphi_2 \geq \varphi_1$, ir iš naujo šakojame gautą aibę (žr. 2 brėžinį).

2 brėžinys



Gautose šakose $\mathcal{X}_{21}$ ir $\mathcal{X}_{22}$ taip pat įvertiname tikslo funkcijos rėžius. Dabar aibė $\mathcal{X}$ jau padalinta į tris toliau nešakotas aibes: $\mathcal{X}_1$, $\mathcal{X}_{21}$ ir $\mathcal{X}_{22}$. Randame didžiausiąjį iš rėžių φ_1, φ_{21} ir φ_{22}. Sakykim, didžiausias yra φ_{21}, todėl $\mathcal{X}_{21}$ vėl padalinam į dvi naujas šakas $\mathcal{X}_{211}$ ir $\mathcal{X}_{212}$ ir vėl randame jų rėžius. Taigi aibė $\mathcal{X}$ dabar padalinta jau į keturias nesikertančias dar nešakotas aibes: $\mathcal{X}_1$, $\mathcal{X}_{211}$, $\mathcal{X}_{212}$ ir $\mathcal{X}_{22}$. Suradę didžiausiąjį iš jų rėžių, šakojimą tęsiame toliau.

Procesas nutrūksta, jei kurioje iš dar nešakotų aibių surandame planą $\boldsymbol{x}^*$, kad $\varphi(\boldsymbol{x}^*)$, sutampa su tiksliu viršutiniu tikslo funkcijos rėžiu šioje aibėje, o visų dar nešakotų aibių rėžiai neviršija reikšmės $\varphi(\boldsymbol{x}^*)$. Planas $\boldsymbol{x}^*$ yra optimalus, nes visose dar nešakotose aibėse, iš kurių susideda $\mathcal{X}$, tikslo funkcijos reikšmė gali būti tik mažesnė.

Pavyzdžiui, jei brėžinyje atvaizduotame šakojime surastume $\boldsymbol{x}^* \in \mathcal{X}_{212}, \varphi(\boldsymbol{x}^*) = \varphi_{212}$, be to, galiotų nelygybės $\varphi_{212} \geq \varphi_{211}, \varphi_{212} \geq \varphi_{22}, \varphi_{212} \geq \varphi_1$, tai $\boldsymbol{x}^*$ būtų optimalus planas.

Jei $\mathcal{X}$ baigtinė aibė (t.y. jos elementų skaičius baigtinis), tai šakų ir rėžių metodu sprendinys bus randamas per baigtinį žingsnių skaičių, nes nešakotose aibėse pagaliau liks tik po vieną elementą (arba sprendinys bus surastas anksčiau). Jei $\mathcal{X}$ nėra baigtinė aibė, sprendinio paieškos procesas gali būti begalinis. Jį nutraukę, tenkinamės artutiniu sprendiniu.

Aprašytoji šakų ir rėžių metodo schema vadinama visiško šakojimo schema. Be jos, naudojama dar ir vienpusio šakojimo schema. Pagal ją kiekviename žingsnyje tolesniam šakojimui aibė parenkama ne iš visų, o tik iš praeitame žingsnyje naujai gautų dar nešakotų aibių. Pavyzdžiui, 2 brėžinyje atvaizduotu atveju iš pradžių lyginame $\varphi_1$ir φ_2. Kadangi $\varphi_2 \geq \varphi_1$,šakojame aibę $\mathcal{X}_2$. Aibę tolesniam šakojimui atrenkame lygindami φ_{21} ir φ_{22} (bet ne φ_1). Kitame žingsnyje lyginame rėžius φ_{211} ir φ_{212} vieną su kitu, bet ne su φ_{22} ir φ_1. Tęsdami vienpusį šakojimą, gauname aibę, kurioje randame elementą su tikslo funkcijos reikšme, lygia viršutiniam rėžiui (blogiausiu atveju, taip atsitinka tik tada, kai aibėje lieka tas vienintelis elementas). Jį atitinkanti tikslo funkcijos reikšmė vadinama rekordu. Iš uždavinio pašalinamos visos dar nešakotos aibės, kurių rėžiai mažesni už rekordą (tokiose aibėse geresnio už rekordą elemento nebus). Iš likusių aibių išrenkama ta, kurios rėžis didžiausias, ir vėl šakojama pagal aprašytą schemą. Jei šiame procese atsiranda aibių, kurių rėžis mažesnis už rekordą, jos nedelsiant pašalinamos iš uždavinio. Tokiu būdu gaunamas naujas rekordas arba įrodoma, kad geresnio už jį elemento nėra.

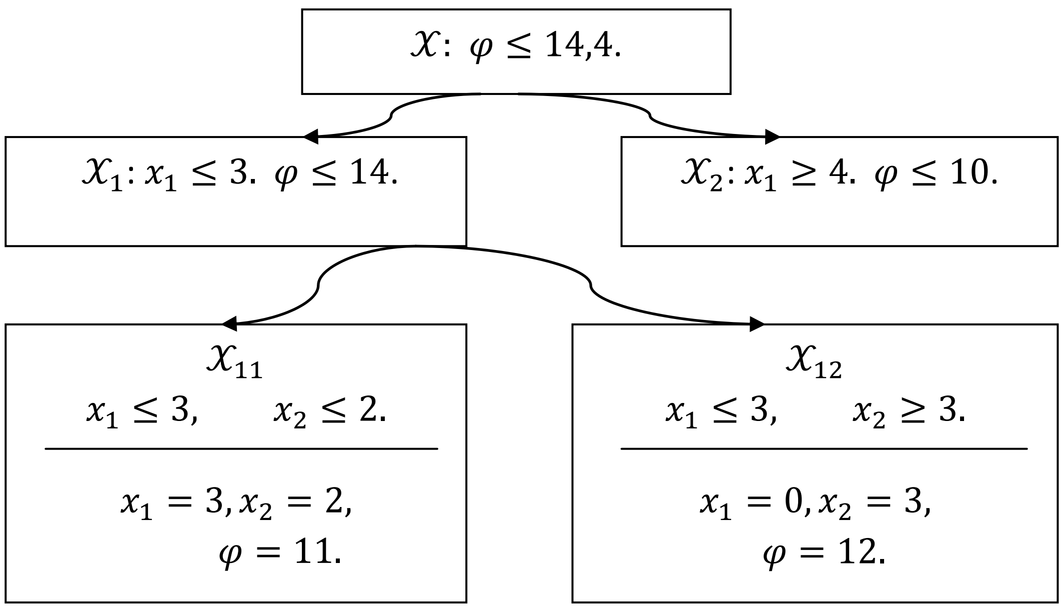
Tiek visiško, tiek ir vienpusio šakojimo atveju aibes galima dalinti ne į dvi, o į daugiau nesikertančių dalių.

Be abejo, jei sprendiniui surasti tenka $\mathcal{X}$ išskaidyti į šakas, turinčias po vieną elementą, tai toks sprendimo būdas tolygus visų jos elementų perrinkimui ir negali būti efektyvus. Todėl šakų ir rėžių metodo efektyvumas priklauso nuo to, kaip išradingai konkrečiame uždavinyje vykdomas šakojimas bei rėžių įvertinimas. Gerai organizavus šį procesą, gana greitai atmetamos neperspektyvios šakos, ir sprendinys randamas palyginti nedidelėje alternatyvų aibėje.

Pavyzdys. Išspręsime šiame paragrafe suformuluotą (1) uždavinį. Jo leistinų planų aibė $\mathcal{X}$ atvaizduota paryškintais taškais 1 brėžinyje. Pirmiausia nustatysime tikslo funkcijos viršutinį rėžį visoje leistinų planų aibėje. Tuo tikslu išspręsime uždavinį, nepaisydami sveikaskaitiškumo sąlygos. Spręsdami simpleksiniu metodu arba grafiškai, gausime tokį planą: $x_1 = 3{,}6,\ x_2 = 2{,}7,\ \varphi = 14{,}4$ (taškas B 1 brėžinyje). Taigi didesnės kaip 14,4 reikšmės tikslo funkcija įgyti negali.

Dabar sugalvosime, kaip aibę šakoti. Kadangi x_1 reikšmė tarp 3 ir 4, o mums reikia sveikos reikšmės, vieną šaką $\mathcal{X}_1$ sudarykime su sąlyga $x_1 \le 3$, o kitą - $\mathcal{X}_2$ su sąlyga $x_1 \ge 4$. Dabar išspręskime jau du uždavinius – pradinį su viena ir su kita papildoma sąlyga. Sveikaskaitiškumo reikalavimą ir vėl ignoruokime. Gausime, kad pirmojo uždavinio sprendinys yra $x_1 = 3,\ x_2 = 2{,}75,\ \varphi = 14$, o antrojo - $x_1 = 4,\ x_2 = 1{,}5,\ \varphi = 10$. Taigi aibėje $\mathcal{X}_1$ nebus plano su didesne reikšme, kaip 14, o aibėje $\mathcal{X}_2$ – su didesne reikšme, kaip 10. Vadinasi, tolesniam šakojimui renkamės aibę $\mathcal{X}_1$. Kadangi , $x_2 = 2{,}75$, šakas sudarome pagal sąlygas $x_2 \le 2$ (šaka $\mathcal{X}_{11}$) ir $x_2 \ge 3$ (šaka $\mathcal{X}_{12}$). Vėl sprendžiame du uždavinius – pradinį su papildomom sąlygom $x_1 \le 3, x_2 \le 2$ ir pradinį su papildomom sąlygom $x_1 \le 3, x_2 \ge 3$. Nors ir vėl nekreipiame dėmesio į sveikaskaitiškumo reikalavimą, abiem atvejais gauname sprendinius sveikais skaičiais.

3 brėžinys

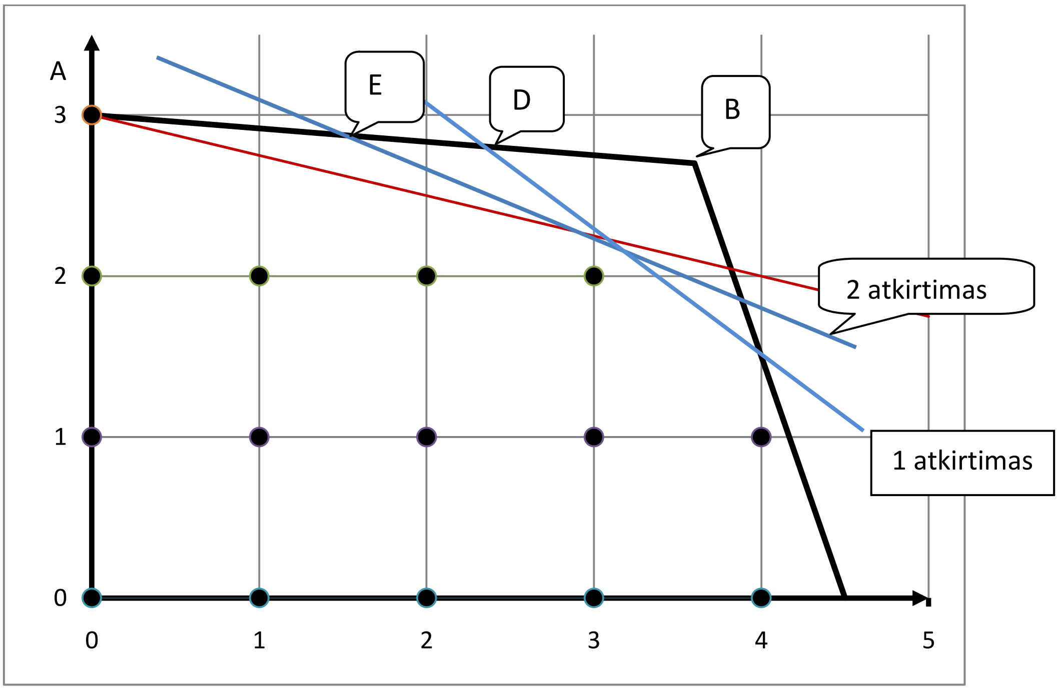


Matome, kad abejose šakose turime sprendinius sveikais skaičiais. Gautas šakoje $\mathcal{X}_{12}$ yra geriausias, nes abejose dar nešakotose aibėse - nei šakoje $\mathcal{X}_{11}$, nei šakoje $\mathcal{X}_{2}$ geresnio plano būti negali. ≺

3°. Atkirtimo metodai. Atkirtimo metodų esmė – atkirsti nuo uždavinio leistinų planų aibės gautą nesveiskaskaitinį optimalų planą taip, kad visi sveikaskaitiniai planai liktų leistinų planų aibėje. Tada vėl ieškomas naujas optimalus planus. Jei jis nesveikaskaitinis, atkertamas ir jis, vėl ieškoma optimalaus plano ir t.t., kol randamas optimalus sveikaskaitinis planas. Šią idėja dar 1954 metais pasiūlė vienas iš tiesinio programavimo pradininkų G.Dantzigas, o 1958 metais realizavo vengrų matematikas R.Gomory.

Iliustruosime atkirtimo algoritmą grafiškai. Grįžkime prie 1^0 nagrinėto uždavinio ir jo brėžinio.

4 brėžinys



Kaip jau žinome, taškas B yra optimalus nesveikaskaitinis planas. Atkirskime jį nuo leistinų planų aibės, išsaugodami joje visus sveikaskaitinius taškus. Tuo tikslu brėžiame 1 atkirtimą. Vėl sprendžiame optimizavimo uždavinį. Nesunku įsitikinti, kad dabar optimalus planas bus taške D. Tačiau ir D nėra sveikaskaitinis planas. Todėl konstruojame 2 atkirtimą, ir vėl sprendžiame optimizavimo uždavinį. Gauname tašką E, kurį ir vėl teks atkirsti ir t.t. Konstruojant atkirtimus Gomory nurodytu būdu, sprendinys (jei jis yra) gaunamas per baigtinį žingsnių skaičių. Mūsų brėžinio atveju tai taškas A, į kurį taškų B, D, E seka akivaizdžiai artėja.

§ 13. TIESINIO PROGRAMAVIMO UŽDAVINIŲ SPRENDIMAS MICROSOFT EXCEL SKAIČIUOKLĖS PAGALBA

1°.Bendro tiesinio programavimo uždavinio sprendimas. Nedideliems tiesinio programavimo uždaviniams spręsti gerai tinka Microsoft Excel skaičiuoklės sprendimo paieškos priedas **Solver.** Jį rasime valdymo meniu grupėje **Duomenys.**[7]

Prieš kviesdami **Solver,** turime parengti mūsų uždavinį atitinkančią Excel lentelę. Kaip tai daroma, parodyta mūsų jau nagrinėto §2 plytinės uždavinio pavyzdžiu (žr. 1 pavyzdį).

Langeliuose C4:F6 įrašome šio uždavinio apribojimų matricos A koeficientus – išteklių sąnaudas kiekvieno produkto (plytų bloko) vienetui pagaminti. Langeliuose G4:G6 įrašome apribojimų vektorių $\boldsymbol{b}$, t.y. turimus išteklių kiekius, o langeliuose C8:F8 – tikslo funkcijos koeficientus $\boldsymbol{c}$ – produktų gamybos pelningumus. Skaitymo patogumui aplink šių duomenų lauką galime užrašyti išteklių ir produktų pavadinimus (juos rašyti nebūtina – kompiuteris išspręs uždavinį ir be to, tačiau taip patogiau vartotojui).

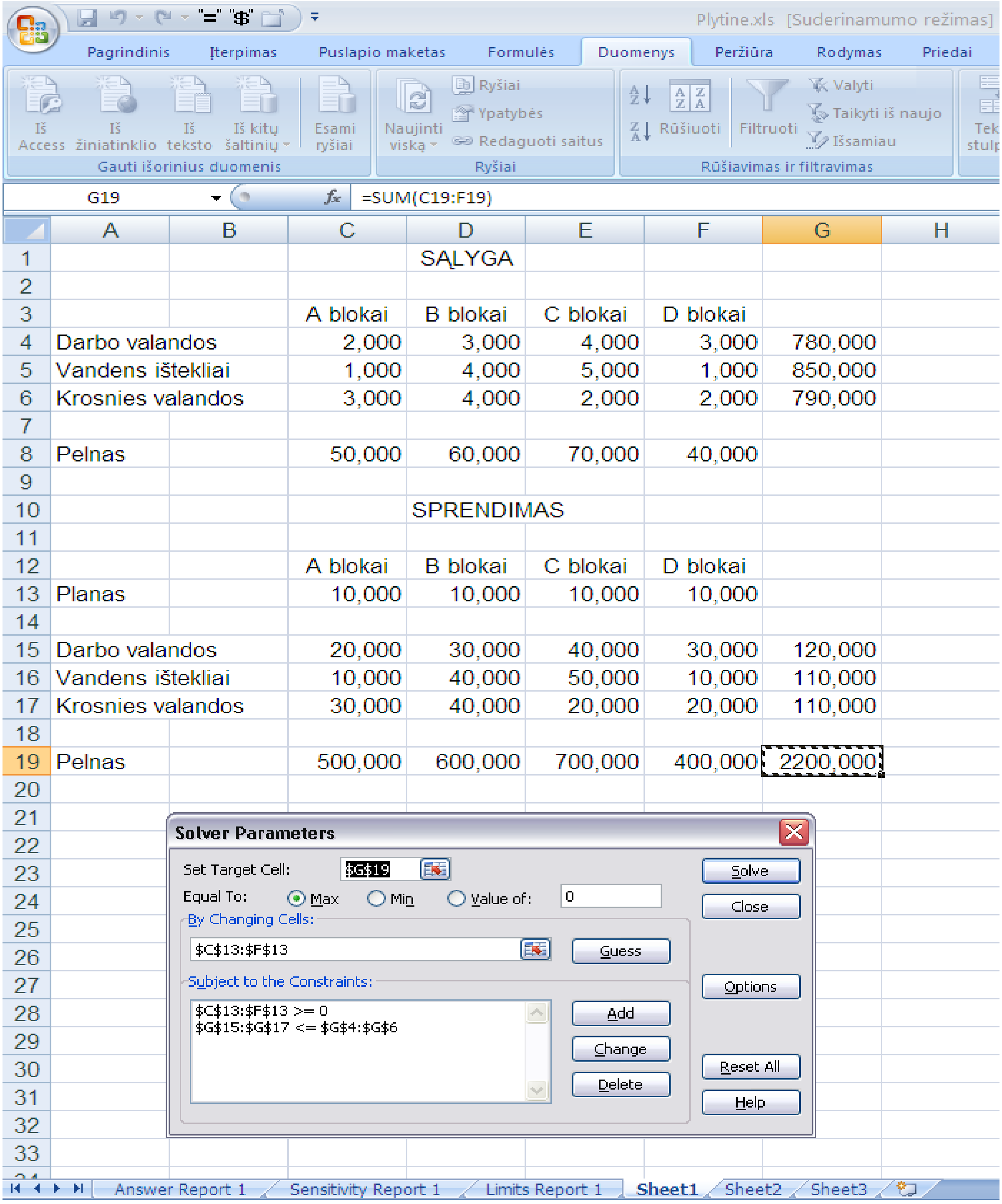
Suformavę sąlygos lauką, pereiname prie sprendimo lauko formavimo. Pirmiausia langeliuose C13:F13 užrašome kokį nors (netgi nebūtinai leistiną) pradinį uždavinio planą $\boldsymbol{x}$. Mūsų pavyzdyje pasirinkta, kad visų plytų blokų gaminame po 10 vienetų $x_1 = x_2 = x_3 = x_4 = 10.$ Toliau lauke C15:F18 skaičiuojame, kiek kokių išteklių reikia pasirinktam planui įgyvendinti. Tuo tikslu visi stulpelio C4:C6 elementai dauginami iš C13 langelyje esančio dydžio, visi stulpelio D4:D6 elementai dauginami iš D13 langelyje esančio dydžio ir t.t. Tą patį padarome ir su pelno apskaičiavimu: langelyje C19 įrašyta C8 ir C13 sandauga, t.y. pelnas, kurį gausim pagaminę 10 „A“ tipo plytų blokų, kurių kiekvieno gamybos pelningumas 50 litų. Analogiškai langelyje D19 įrašyta D8 ir D13 sandauga, rodanti pelną, gaunamą iš antrojo produkto gamybos ir t.t.

Belieka susumuoti, kiek kokių išteklių mūsų pradiniam planui sunaudojama iš viso, ir koks gaunamas pelnas. Tai atlikta langeliuose G15:G17 ir G19. Langelyje G15 įrašyta formulė =SUM(C15:F15), sumuojanti, kiek iš viso mūsų pasirinktam planui įgyvendinti bus panaudota darbo valandų. Analogiškai langelyje G16 įrašyta formulė =SUM(C16:F16), sumuojanti, kiek iš viso mūsų pasirinktam planui įgyvendinti bus panaudota vandens išteklių. Taip pat susumuotas ir gaunamas pelnas: langelyje G19 įrašyta formulė =SUM(C19:F19), sumuojanti gaunamą pelną.

---

[7] Jei **Solver** ten nėra, reikia jį aktyvuoti. Widows XP operacinėje sistemoje paleidžiame veikti **Excel** programą. Paspaudę pagrindinio meniu mygtuką , patenkame į langą **Excel parinktys.** Šiame lange pasirenkame meniu punktą **Priedai** ir aktyvuojame **Sprendimo paieškos** priedą (Sover.xlam)

1 pavyzdys



Dabar viskas parengta uždavinio sprendimui. Iškvietę sprendimo paieškos priedą **Solver**, turėsime jam nurodyti, jog leidžiame kaitalioti langeliuose C13:F13 esančius dydžius (ieškoti geriausio gamybos plano), kad G19 langelyje gautume maksimalią įmanomą reikšmę. Tačiau tai reikia padaryti taip, kad langeliuose G15:G17 atsirandantys dydžiai neviršytų G4:G6 esančių dydžių (nepritrūktų išteklių). Be to, langeliuose C13:F13 esantys dydžiai negali būti neigiami.

Tai padarome tokiu būdu. Iškvietus priedą **Solver**, atidaromas 1 pavyzdyje parodytas langas. Pirmiausia užpildome langelį *Set Target Cell* – jame (paprastai atžymėję pele) nurodome, kuriame lentelės langelyje yra optimizuojamas dydis. Mūsų pavyzdžio atveju tai $G$19 ($ ženklą prideda pats kompiuteris, nes tai absoliutus, o ne santykinis adresas). Toliau turime atžymėti maksimumo ar minimumo ieškome. Mūsų pavyzdžio atveju atžymėtas langelis *Max*. Po to nurodome, kuriuos Excel lentelės elementus leidžiame kaitalioti kaip nežinomuosius, ieškant optimalaus plano. Langelyje *By Changing Cells* nurodome lauką $C$13: $F$13, kur ir patalpinome mūsų nežinomuosius (ir šį veiksmą paprasčiausia padaryti pele). Belieka nurodyti apribojimus. Tam turime užpildyti langelį *Subject to the Constraints.* Paspaudus mygtuką *Add*, atsidaro papildoma lentelė, kurioje vėlgi atžymėję pele nurodome, kad $G$15:$G$17 <= $G$4:$G$6, taip pat, kad $C$13:$F$13 >= 0. Sveikaskaitinio programavimo atveju reikiamai grupei nežinomųjų galima nurodyti „int“ . Patvirtinus šiuos nurodymus, susiformuoja optimizavimo užduotis, kaip ir parodyta 1 pavyzdyje. Belieka paspausti mygtuką *Solve*, ir kompiuteris sprendžia mūsų uždavinį. Plano reikšmės langeliuose C13:F13 pakeičiamos optimaliomis, taip pat ir tikslo funkcijos reikšmė G19, o visi kiti dydžiai atitinkamai perskaičiuojami. Patvirtiname sprendinį arba grįžtame prie pradinio formulavimo (jei norime pakeisti sąlygą).

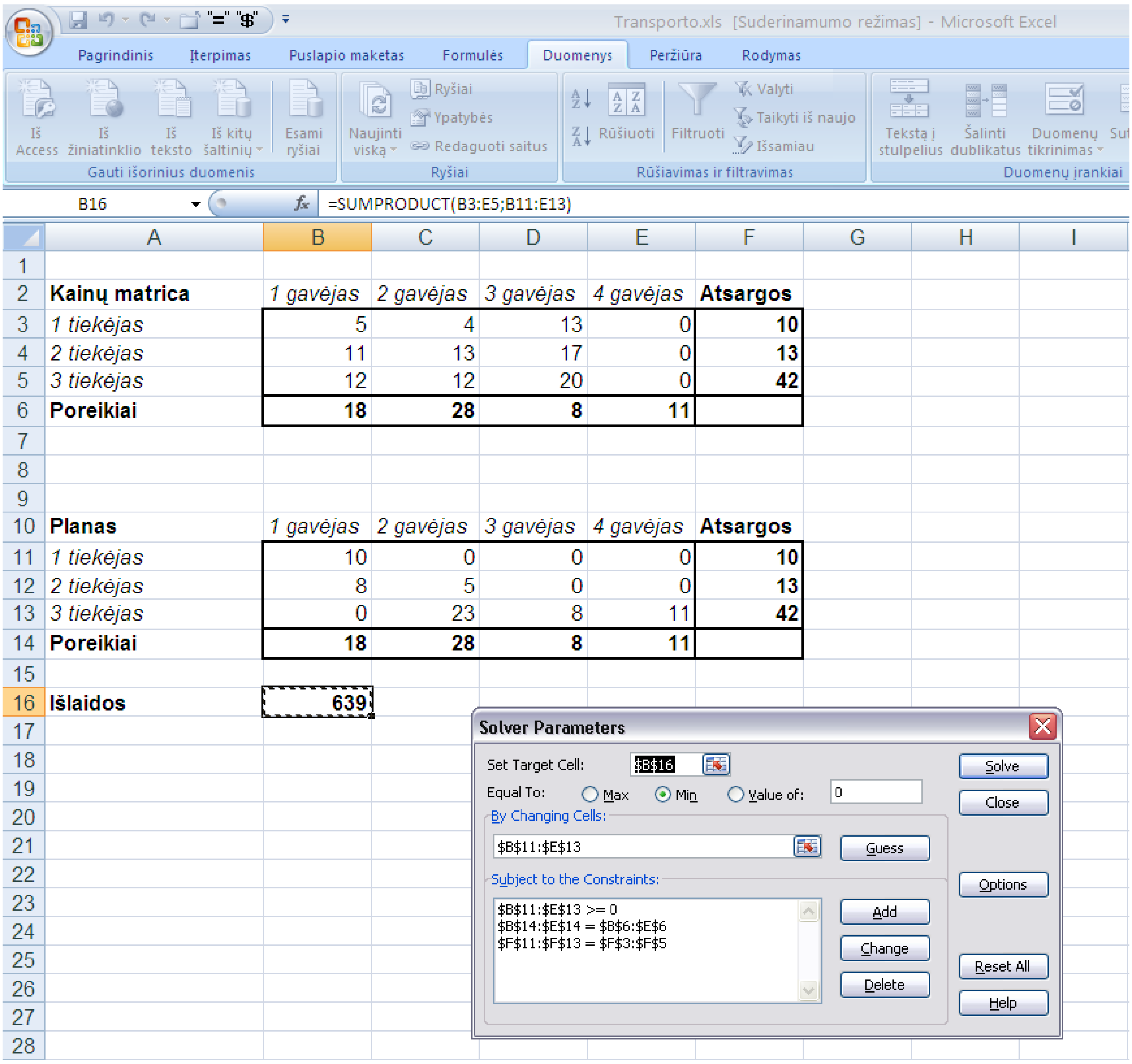
Be sprendinio, priedas **Solver** papildomai pateikia ir nemažai postoptomizacinės analizės elementų. Juos patalpina į darbalapius *Answer Report*, *Sensitivity Report* ir *Limits Report*, kuriuos pats ir suformuoja.

2°.Transporto uždavinio sprendimas. Priedas Solver tinka įvairiems tiesinio ir netgi netiesinio programavimo uždaviniams spręsti. Parodysime, kaip sprendimui pateikti transporto uždavinį. Remsimės §10 3 punkte suformuluotu uždavinio pavyzdžiu.

Kaip ir bendrojo tiesinio programavimo uždavinio atveju, pradedame nuo sąlygos perkėlimo į Excel lentelę (žr. 2 pavyzdį). Turime uždavinį su trimis tiekėjais ir keturiais gavėjais (ketvirtasis – fiktyvus). Langeliuose B3:E5 užrašome pervežimų kainų matricą (c_{ij}), langeliuose F1:F5 tiekėjų turimas atsargas, o langeliuose B6:E11 – gavėjų poreikius.

Toliau pereiname prie sprendimo dalies parengimo. Langeliuose B11:E13 užrašome pradinį pervežimų planą, t.y. dydžius x_{ij}, nuo kurio pradėsime sprendimą. 2 pavyzdyje toks planas sudarytas vadinamuoju „šiaurės vakarų kampo“ metodu. Jei šio metodo nežinome ir kokio nors leistino plano sudaryti nemokame – ne bėda: langelius B11:E13 galime paprasčiausiai užpildyti nuliais. Toliau susumuojame kiekvieno tiekėjo visiems gavėjams tiekiamo produkto apimtis, ir šias sumas įrašome langeliuose F11:F13. Langelyje F11 įrašyta formulė =SUM(B11:E11), langelyje F12 formulė =SUM(B12:E12) ir t.t. Taip pat susumuojame ir kiekvieno gavėjo gaunamas produkto apimtis, atitinkamas sumas įrašome langeliuose B14:E14. Taip langelyje B14 įrašyta formulė =SUM(B11:B13), langelyje C14 įrašyta formulė =SUM(C11:C13) ir t.t. Apskaičiuojame ir tikslo funkcijos reikšmę. Tam pasirenkame langelį B16, į kurį turime įrašyti visų sandaugų $c_{ij}x_{ij}$ sumą. Šiam darbui atlikti gerai tinka Excel funkcija SUMPRODUCT. Langelyje B16 nurodę formulę =SUMPRODUCT(B3:E5;B11:E13), prašome panariui sudauginti lauko B3:E5 elementus su atitinkamais lauko B11:E13 elementais, ir visas sandaugas susumuoti.

2 pavyzdys



Atlikus minėtus parengiamuosius veiksmus, uždavinys jau paruoštas sprendimui. Iškvietę priedą **Solver**, turėsime jam nurodyti, kad, kaitaliojant lauko B11:E13 elementus, reikia minimizuoti langelyje B16 esantį dydį. Tačiau tai reikia padaryti taip, kad langeliuose F11:F13 apskaičiuojami dydžiai būtų lygūs langeliuose F3:F5 esantiems dydžiams (kiekvienas tiekėjas išvežtų gavėjams tiek produkto, kiek jis turi), o langeliuose B14:E14 apskaičiuojami dydžiai būtų lygūs langeliuose B6:E6 esantiems dydžiams (kiekvienas gavėjas gautų tiek produkto, koks yra jo poreikis).

Iškvietus priedą **Solver**, atidaromas 2 pavyzdyje parodytas langas (toks pat kaip ir 1 pavyzdyje). Pirmiausia užpildome langelį *Set Target Cell* – jame (paprastai atžymėję pele) nurodome, kuriame lentelės langelyje yra optimizuojamas dydis. Mūsų pavyzdžio atveju tai $B$16 ($ ženklą prideda pats kompiuteris, nes tai absoliutus, o ne santykinis adresas). Toliau turime atžymėti maksimumo ar minimumo ieškome. Mūsų pavyzdžio atveju atžymėtas langelis *Min*. Po to nurodome, kuriuos Excel lentelės elementus leidžiame kaitalioti kaip nežinomuosius, ieškant optimalaus plano. Langelyje *By Changing Cells* nurodome lauką $B$11: $E$13, kur ir patalpinome mūsų nežinomuosius (ir šį veiksmą paprasčiausia padaryti pele). Belieka nurodyti apribojimus. Tam turime užpildyti langelį *Subject to the Constraints.* Paspaudus mygtuką *Add*, atsidaro papildoma lentelė, kurioje vėlgi atžymėję pele nurodome, kad $F$11:$F$13 = $F$3:$F$5 ir $B$14:$E$14 = $B$6:$E$6, taip pat, kad $B$11:$E$13 >= 0 (nežinomųjų neneigiamumas). Patvirtinus šiuos nurodymus, susiformuoja optimizavimo užduotis, kaip ir parodyta 2 pavyzdyje. Belieka paspausti mygtuką *Solve*, ir kompiuteris sprendžia mūsų uždavinį. Plano reikšmės langeliuose B11:E13 pakeičiamos optimaliomis, taip pat ir tikslo funkcijos reikšmė B16, o visi kiti dydžiai atitinkamai perskaičiuojami. Patvirtiname sprendinį arba grįžtame prie pradinio formulavimo (jei norime pakeisti sąlygą).

Kaip ir 1 pavyzdžio atveju, be sprendinio, priedas **Solver** papildomai pateikia ir nemažai postoptomizacinės analizės elementų.

Čia parodytas sprendimo būdas tinka tik palyginti nedideliems uždaviniams spręsti. Didesniam uždaviniui reiktų pasitelkti specialias tiesinio programavimo uždavinių programas, integruotas į žinomus taikomuosius paketus (pavyzdžiui, [11], [12]).

Rodyklė (nebaigta)

Literatūra

1. **Apynis Antanas.** Optimizavimo metodai. Vilnius, VU leidykla, 2005.
2. **Bubelis Valentinas, Medaiskis Teodoras, Morkeliūnas Algis**. Operacijų tyrimo įvadas. Mokymo priemonė. Vilnius, VU leidykla, 2008.
3. **Čiočys Vaclovas, Jasilionis Rimutis Juozapas**. Matematinis programavimas. Vadovėlis. Vilnius, Mokslas, 1990.
4. **Frederick S. Hillier, Gerald J. Lieberman.** Introduction to Operations Research, McGraw-Hill: Boston MA. 8th. (International) Edition, 2005
5. **Medaiskis Teodoras**. Operacijų tyrimas. Mokymo priemonė. Vilnius, 1988.
6. **Medaiskis Teodoras**. Operacijų tyrimas. Tiesinio bei diskrečiojo programavimo modeliai ir metodai. Mokymo priemonė. Vilnius, 1989.
7. **Puškorius Stasys.** Matematiniai metodai vadyboje. Vilnius, TEV, 2001.

8. **Stungurienė Stanislava.** Verslo matematika. Vilnius, TEV, 2008.
9. **Stungurienė Stanislava.** Operacijų valdymas. Vilnius, TEV, 2010.
10. **Taha Hamdy A.** Operations Research: An Introduction. Nineth edition. Prentice Hall, 2011.
11. **SAS Institute Inc.** SAS/OR ® 9.22 User's Guide: Mathematical Programming. 2010**.**
12. **S.Vakrinienė**. Operacijų tyrimas programine įranga SAS/OR. Mokomoji knyga. Vilnius, Technika, 2003.