

S T U . .  
• • • • •  
F C H P T  
• • • • •

**SLOVENSKÁ TECHNICKÁ UNIVERZITA**

Fakulta chemickej a potravinárskej technológie

Radlinského 9, 812 37 Bratislava

**ODDELENIE INFORMATIZÁCIE A RIADENIA PROCESOV**

Bc. Zuzana BOROŠOVÁ

## **DIPLOMOVÁ PRÁCA**

**VÝUČBOVÝ SYSTÉM PRE PREDMET OPTIMALIZÁCIA**

**Vedúci diplomovej práce**

Doc. Ing. Ján Dvoran, CSc.

Bratislava 2007

Ústav: **Informatizácie, automatizácie a riadenia procesov**

Oddelenie: **Informatizácie a riadenia procesov**

Číslo: 15/UIAM/2007

Vec: **Zadanie diplomovej práce**

Meno a priezvisko študenta: **Bc. Zuzana Borošová**

Meno a priezvisko vedúceho diplomovej práce: **doc. Ing. Ján Dvoran, CSc.**


Názov diplomovej práce:


### **Výučbový systém pre predmet Optimalizácia**

Termín odovzdania diplomovej práce: **19. mája 2007**


Diplomová práca sa odovzdáva v 3 exemplároch vedúcemu ústavu – oddelenia.


Bratislava, 12. februára 2007

  
doc. Ing. Ján Dvoran, CSc.  
vedúci diplomovej práce

  
doc. Dr. Ing. Miroslav Fikar  
riaditeľ ústavu



  
doc. Dr. Ing. Miroslav Fikar  
vedúci oddelenia

  
prof. Ing. Dušan Bakoš, DrSc.  
dekan

***Vyhlasenie :***

Vyhlasujem, že som diplomovú prácu vypracovala samostatne podľa pokynov vedúceho, s použitím literatúry, ktorú uvádzam v osobitnom zozname.

V Bratislave, 18.05.2007

.....

### ***Pod'akovanie***

Chcem sa poďakovať vedúcemu diplomovej práce Doc.Ing. Jánovi Dvoranovi, Csc., za jeho smerovanie v mojej práci a pripomienky, ktoré mi pomohli lepšie zvládnuť danú problematiku.

## **ABSTRAKT**

Úlohou tejto diplomovej práce je spracovanie optimalizačných úloh vo forme učebnej pomôcky pre študentov, ktorí sa počas štúdia stretnú s danou problematikou. Vzhľadom na fakt, že oblasť optimalizácie zahŕňa veľké množstvo úloh, táto diplomová práca pojednáva o metódach lineárneho programovania, v menšej časti o problematike celočíselného programovania.

Každá úloha je najskôr slovne charakterizovaná a potom sformulovaná pomocou matematického modelu ako problém lineárneho programovania. V prípade problémov lineárneho programovania sú uvedené viaceré postupy výpočtov s porovnaním ich výsledkov.

## **ABSTRACT**

The main goal of the thesis is to prepare optimization exercises as a learning aid for students who encounter this topic during their studies. Due to the large scope of the optimization exercises, this thesis focuses primarily on methods of linear programming, less on the integral programming.

Each assignment is at first shortly presented and described and subsequently formulated into a problem of the linear programming using mathematical models. Several alternative computation approaches are offered in cases of linear programming problems and the respective results are compared.

# OBSAH

ÚVOD .....	7
<b>1. LINEÁRNE PROGRAMOVANIE .....</b>	<b>10</b>
1.1 HISTÓRIA VÝVOJA LP .....	10
1.2 TEÓRIA LP .....	11
1.2.1 Základné pojmy .....	12
1.2.2 Všeobecné vlastnosti riešenia úloh LP .....	12
1.2.3 Úloha LP .....	13
1.3 FORMULÁCIA ÚLOHY LP .....	14
1.3.1 Postup pri zostavovaní modelu.....	16
1.3.2 Štandardný tvar úlohy LP.....	16
1.4 DUALITA ÚLOH LP .....	18
1.4.1 Zásady pri formulovaní duálneho modelu úlohy LP .....	20
1.4.2 Vlastnosti duálnych úloh .....	20
1.4.3 Vecná interpretácia duálneho modelu.....	22
1.4.4 Príklady - Riešenie úloh LP pomocou toolboxu programu MATLAB® .....	23
1.5 ÚLOHY O ZMESIACH .....	32
1.5.1 Príklad – úlohy o zmesiach.....	34
<b>2. SIMPLEXOVÁ METÓDA RIEŠENIA ÚLOH LP .....</b>	<b>36</b>
2.1 POUŽITIE SIMPLEXOVEJ METÓDY V PRÍKLADOCH.....	36
2.1.1 Simplexový algoritmus.....	37
2.1.2 Príklad 1 .....	41
2.1.3 Príklad 2 .....	44
2.1.4 Príklad 3 .....	46
2.1.5 Príklad 4 .....	49
2.1.6 Príklad 5 .....	50
2.1.6 Príklad 6 .....	53
2.1.7 Príklad 7 .....	56
<b>3. DOPRAVNÉ ÚLOHY .....</b>	<b>58</b>
3.1 DOPRAVNÁ ÚLOHA A JEJ APLIKÁCIE .....	58
3.1.1 Formulácia dopravnej úlohy .....	58

3.1.2	Vyvážená dopravná úloha .....	59
3.2	KONŠTRUKCIA MODELU.....	60
3.2.1	Zápis modelu .....	61
3.3	VLASTNOSTI DOPRAVNEJ ÚLOHY .....	62
3.3.1	Riešenie vybilancovanej dopravnej úlohy .....	62
3.4	METÓDA SEVEROZÁPADNÉHO ROHU .....	64
3.4.1	Príklad – Metóda severozápadného rohu.....	65
3.5	INDEXOVÁ METÓDA .....	68
3.5.1	Príklad – Indexova metóda.....	68
3.6	VOGELOVA APROXIMAČNÁ METÓDA .....	70
3.6.1	Príklad – Vogelova aproximačná metóda .....	71
3.7	MODIFIKOVANÁ METÓDA (METÓDA RIADKOVÝCH A STĹPCOVÝCH ČÍSEL).....	81
3.8	HABROVA METÓDA ÚZKYCH PROFILOV .....	82
<b>4.</b>	<b>CELOČÍSELNÉ LINEÁRNE PROGRAMOVANIE .....</b>	<b>83</b>
4.1	Príklad - Gomoryho algoritmus rezných nadrovín .....	84
<b>5.</b>	<b>ZÁVER .....</b>	<b>86</b>
<b>6.</b>	<b>LITERATÚRA .....</b>	<b>88</b>
<b>7.</b>	<b>PRÍLOHY .....</b>	<b>90</b>

# ÚVOD

Bez zveličenia možno povedať, že vo všeobecnosti sa činnosť ľudí, vedome alebo nevedome, riadi princípmi zákonov optimálnosti.

Snahou je dosiahnutie maximálnej účinnosti ľudského pôsobenia na prostredie pri zadaných ohraničeniach zdrojov. Vo výrobnom procese je snahou dosiahnuť maximálny zisk pri limitovaných zdrojoch surovín a energie, minimalizovať výrobné náklady, skrátiť priebežnú dobu výroby pri rešpektovaní ohraničujúcich podmienok, vyjadrujúcich požiadavky odberateľov na množstvo a kvalitu výrobných etáp pri rešpektovaní ohraničujúcich podmienok, ktoré môžu byť vyvolané požiadavkami trhu na kvalitu výroby, či na ochranu životného prostredia, na bezpečnosť práce a pod.

V zásade ide o určovanie extrémů funkcie, ktorá sa v optimalizačných úlohách nazýva kritériálna, objektívna alebo účelová funkcia. Táto funkcia vyjadruje cieľ, ktorý sa sleduje pri výbere niektorého z možných variantov riešení a zvyčajne prezentuje niektorý významný ukazovateľ (zisk, náklady, hodnoty produkcie, spotrebu energie a pod.). Často je potrebné vykonať optimálne rozhodnutie vzhľadom na viaceré účelové funkcie s často až diametrálne protichodnými požiadavkami. No aj v takýchto zložitých prípadoch možno hovoriť o istom optimálnom riešení v zmysle vhodne zvoleného kompromisného princípu.

Predložená diplomová práca sa týka jednej časti optimalizačných úloh a tou je problematika lineárneho programovania. Lineárne programovanie má už viac ako polstoročnú históriu a za toto obdobie preniklo do mnohých oblastí vedy, techniky a národného hospodárstva. Táto optimalizačná technika sa stala významným nástrojom zefektívňovania najmä pre pomernú jednoduchosť, či už ide o samotný matematický model alebo o jeho riešenie. Preto sa lineárne programovanie vyučuje na mnohých odboroch vysokoškolského štúdia.

Lineárne programovanie vzniklo z potrieb ekonomickej praxe stanovujúcej isté ciele, ktoré je možné pri rešpektovaní ohraničujúcich podmienok dosiahnuť metódami lineárnej algebry. V súčasnosti je lineárne programovanie veľmi intenzívne využívané pri riešení ekonomických problémov, ale aj problémov z oblastí techniky, poľnohospodárstva a ďalších.

Cieľom tejto diplomovej práce je jej použitie ako študijnej príručky pre výučbový systém pre predmet optimalizácia. Študentom prináša nielen teoretické



základy týkajúce sa danej problematiky, ale aj množstvo príkladov a podrobné postupy na ich riešenie..

Každá úloha je najskôr slovne charakterizovaná a potom sformulovaná pomocou matematického modelu ako problém lineárneho programovania.

Diplomová práca je rozdelená na štyri hlavné kapitoly, rozoberajúce jednotlivé časti týkajúce sa lineárneho programovania:

1. lineárne programovanie
2. simplexová metóda riešenia úloh lineárneho programovania
3. dopravné úlohy
4. celočíselné lineárne programovanie.

Prvá kapitola pojednáva o teórii k danej problematike. Sú tu uvedené základné poznatky o histórii lineárneho programovania, formulácia úloh lineárneho programovania, možnosť ich riešenia a samozrejme časť je venovaná aj dualite úloh lineárneho programovania. Pri príkladoch v tejto kapitole bola ako prostriedok pre riešenie použitá časť programu MATLAB<sup>®</sup>, ktorý má veľmi kvalitne vytvorený toolbox práve na lineárne programovanie. MATLAB<sup>®</sup> predstavuje vysoko výkonný jazyk pre technické výpočty. Ide o interaktívny systém, ktorého základným dátovým typom je pole (dvojmerné, a od verzie 5 – viacmerné) bez nutnosti deklarovať jeho rozmery. Táto vlastnosť, spolu s množstvom zabudovaných funkcií, umožňuje relatívne ľahké riešenie mnohých technických problémov, špeciálne takých, ktoré vedú na vektorovú alebo maticovú formuláciu, v oveľa kratšom čase ako riešenie v klasických programovacích jazykoch. Jednotlivé metódy riešenia pomocou MATLAB<sup>®</sup> sú aplikované na jeden príklad s následným porovnaním výsledkov.

V druhej kapitole je podrobne rozobratá simplexová metóda, ktorá je základnou metódou pri riešení úloh lineárneho programovania. Sú tu uvedené štyri príklady, ktoré majú rôzny tvar obmedzení. Ďalší príklad je riešený interaktívnou metódou [11]. Samozrejmosťou je riešenie prostredníctvom programu MATLAB<sup>®</sup> na posledných troch príkladoch, z ktorých prvé dva majú zadanie slovné, a tretí príklad je na duálny simplexový algoritmus. Pri riešení vlastných úloh lineárneho programovania touto metódou stačí postupovať podľa jednotlivých postupov. Riešenie príkladu s počtom nekonečne veľa riešení sa pre obmedzenie rozsahu predloženej diplomovej práce neuvádza.

Tretia kapitola sa zaoberá dopravným problémom, ktorý je tiež dôležitou súčasťou lineárneho programovania. Je zadáný jeden príklad, na ktorý sa aplikujú jednotlivé metódy dopravných problémov vzhľadom na čas ich vzniku. Prvou metódou bola metóda severozápadného rohu, následne indexová metóda a Vogelova aproximačná metóda. Na modifikovanú metódu a Habrovu metódu úzkych profilov príklad nie je uvedený, sú spomenuté iba z teoretického hľadiska.

Štvrtá kapitola sa venuje celočíselnému lineárnemu programovaniu pričom je uvedený príklad pre Gomoryho algoritmus rezných nadrovín. Vzhľadom na obmedzenie rozsahu predloženej diplomovej práce však nebolo možné rozoberať túto zaujímavú tému hlbšie.

Diplomová práca slúži ako podklad pre e-learningový výučbový systém predmetu Optimalizácia. Ak bol e-learning ešte nedávno považovaný za neznámy pojem, v súčasnosti to neplatí. Možno ho charakterizovať ako najmodernejší spôsob multimediálnej výuky napríklad na báze internetu. Ponúka široké možnosti uplatnenia a vyznačuje sa kreativitou, pričom zabezpečuje odovzdávanie vedomostí najprogresívnejšou formou.

# 1. LINEÁRNE PROGRAMOVANIE

## 1.1 História vývoja LP

Lineárne programovanie (LP) sa už viac ako polstoročie využíva v technickej praxi, národnom hospodárstve i v mnohých oblastiach vedy. Jeho úlohou je určiť minimum lineárnej funkcie  $n$  premenných na množine všetkých nezáporných riešení danej sústavy  $m$  lineárnych rovníc, pričom  $m < n$ .

Lineárne programovanie patrí do disciplín matematického programovania. Pomerne jednoduchý matematický aparát spolu s rozvojom počítačov umožnili, vzhľadom k ostatným disciplínam operačného výskumu, jeho široké uplatnenie v praxi. Vývoj lineárneho programovania bol podmienený [1]:

- vplyvom rozvoja matematických disciplín,
- rozvojom metód plánovania a riadenia národného hospodárstva,
- rozvojom výpočtovej techniky.

Otázkou nájdenia optimálnych riešení rôznych úloh sa ľudia zaoberali už v staroveku, napr. *Euklides* skúmal, ako nájsť najdlhšiu a najkratšiu spojnicu bodu s kružnicou, rovnobežník, ktorý má pri danom obvode najväčšiu plochu apod. Prvé optimalizačné úlohy boli zamerané na geometrické, dynamické alebo fyzikálne problémy. Vývoj matematických základov lineárneho programovania možno pozorovať v práci *Fouriera* (1826) o lineárnych nerovnostiach a úlohe hľadania minimálnej maximálnej odchýlky od stien polyédra. V jeho metóde hľadania riešenia prechodom z jedného vrcholu na druhý vrchol možno vidieť analógiu so simplexovou metódou.

Ďalej možno pripomenúť práce *Gaussa* (1826) a *Jordana* (1904) o eliminačnej metóde sústavy lineárnych rovníc, *Gordana* (1873) a *Farkaša* (1902) o vlastnostiach duálnych sústav, *Minkowského* (1896), ktorý zaviedol pojem bázičného (základného) riešenia rovníc.

V roku 1939 *L.V. Kantorovič* navrhol metódu riešenia istého distribučného problému. Dopravný problém bol formulovaný v roku 1941 *Hitchcockom* a v roku 1944 *John von Neumannom*. *Oscar Morgenstein* formuloval teóriu hier a rozhodovania

a ukázal väzbu na úlohu lineárneho programovania, v roku 1945 formuloval *Stiegler* nutričný problém.

Významným medzníkom v histórii vývoja lineárneho programovania je rok 1947, keď *G. B. Danzig* sformuloval simplexovú metódu. Do tohto roku bol vývoj pozvoľný, praktické aplikácie teórií zatiaľ neboli, čo bolo spôsobené i nedokonalou výpočtovou technikou. Po roku 1947 sa cesta využitia metód lineárneho programovania otvorila jednak tým, že bola definovaná univerzálna metóda riešenia problémov lineárneho programovania, jednak i prudkým rozvojom výpočtovej techniky.

Búrlivý vývoj využitia optimalizačných modelov zaznamenal rok 1947, keď po objavení simplexovej metódy bola ekonomická verejnosť upozornená na veľké možnosti pri formulácii problémov lineárneho programovania. Situáciu dokresľoval i vývoj výpočtovej techniky, keď od roku 1950 sa objavovali prvé programy na riešenie úlohy LP simplexovou metódou.

## 1.2 Teória LP

Lineárne programovanie je druh matematického programovania.

Matematický model sa skladá z:

1. účelovej funkcie
2. ohraničujúcich podmienok (vlastné obmedzenia a podmienky nezápornosti)

Účelová funkcia i ohraničujúce podmienky sú spravidla vyjadrené lineárnymi vzťahmi s konštantnými koeficientmi pri jednotlivých premenných i s konštantnými pravými stranami sústav obmedzení.

### 1.2.1 Základné pojmy

Podľa [2]:

Vektor riešení  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  označuje riešenie optimalizačného problému. Pri určovaní optimálnej výrobnéj štruktúry podniku zložky vektora riešení predstavujú napr. množstvo jednotlivých druhov výrobkov. Optimálna je taká štruktúra výroby, pri ktorej rozhodujúca veličina (napr. zisk, náklady ...) nadobúda extrémne hodnoty (maximálne alebo minimálne). Táto rozhodujúca veličina sa vyjadruje pomocou účelovej funkcie.

Účelová funkcia  $z = f(\mathbf{x})$  je funkcia vektora riešení. Pri hľadaní optimálnej výrobnéj štruktúry predstavuje účelová funkcia napr. závislosť zisku na množstve jednotlivých výrobkov, pričom sa hľadá také množstvá výrobkov, ktoré zaručujú maximálny zisk.

Maximum alebo minimum – extrém – účelovej funkcie sa hľadá za určitých ohraničujúcich podmienok, ktoré ovplyvňujú veľkosť zložiek vektora riešení. Jednou z ohraničujúcich podmienok pri hľadaní optimálnej štruktúry je vzťah medzi disponibilnými zdrojmi nutných k výrobe a spotrebou na jednotlivé výrobky.

Tieto ohraničujúce podmienky sa nazývajú vlastné ohraničujúce podmienky a všeobecne ich možno zapísať ako  $g_i(x) \leq b_i$ ,  $g_i(x) \geq b_i$ ,  $g_i(x) = b_i$ , kde  $g_i(x)$  je daná funkcia vyjadrujúca závislosť spotreby jednotlivého zdroja na vyrobenom množstve výrobkov, a  $b_i$  sú dané konštanty predstavujúce napr. disponibilné množstvo surovín. Ďalšie ohraničujúce podmienky sú podmienky nezápornosti a platia pre všetky zložky vektora riešení.

### 1.2.2 Všeobecné vlastnosti riešenia úloh LP

Úloha LP je tvorená vlastnými ohraničujúcimi podmienkami, podmienkami nezápornosti a účelovou funkciou. Pri riešení úloh LP je sústava ohraničujúcich podmienok upravená na sústavu  $m$  rovníc o  $n$  neznámych ( $n > m$ ). Aby bola sústava  $m$  rovníc riešiteľná, je nutné  $n-m$  neznámych zvoliť. Tieto volené neznáme sú nezákladné a neznáme, podľa ktorých je sústava riešená, sú základné (bázické) neznáme. Pokiaľ sa za  $n-m$  zvolí nula, získaným riešením je **základné (bázické) riešenie**, ktoré obsahuje najviac  $m$  nenulových položiek.

**Nedegenerované riešenie** – počet nenulových položiek základného riešenia sa rovná počtu nezávislých rovníc. Pokiaľ je počet nenulových položiek menší ako počet nezávislých rovníc, jedná sa o **degenerované riešenie**. Množina prípustných riešení úloh LP je množina konvexná, teda konvexná kombinácia prípustných riešení je zase prípustné riešenie.

**Základné prípustné riešenie** - obsahuje najviac  $m$  kladných zložiek a vektory koeficientov týchto neznámych musia byť lineárne nezávislé. Počet základných prípustných riešení sa rovná najviac  $\binom{n}{m} = \frac{n!}{(n-m)!m!}$  - toľkými spôsobmi možno zvoliť  $n-m$  nulových zložiek, aby vznikla sústava  $m$  rovníc.

**Optimálne riešenie úlohy** – prípustné riešenie, ktoré maximalizuje alebo minimalizuje účelovú funkciu.

**Základná veta lineárneho programovania:** Ak má úloha LP optimálne riešenie, má nutne tiež základné optimálne riešenie.

Pokiaľ má úloha LP viac základných optimálnych riešení, potom každá ich konvexná kombinácia predstavuje tiež optimálne riešenie, úloha má nekonečne veľa rovnocenných optimálnych riešení.

### 1.2.3 Úloha LP

Úloha LP:

- a) môže mať jedno optimálne riešenie,
- b) môže mať nekonečne veľa optimálnych riešení,
- c) nemá konečné optimálne riešenie (ak množina prípustných riešení nie je ohraničená)
- d) nemá žiadne riešenie (ak prienikom ohraničení je prázdna množina).

Úloha LP je najjednoduchšia a najviac algoritmicky prepracovaná úlohou matematického programovania. Jej riešenie má početné praktické aplikácie. Medzi typické úlohy vedúce na riešenie úlohy LP patria najmä úlohy z optimalizácie

výrobných plánov, rôzne úlohy optimalizácie delenia materiálu, miešania surovín, optimalizácia dopravných plánov pri zásobovaní a v neposlednej rade i niektoré úlohy teórie hier.

Efektívne algoritmy riešenia úlohy LP sú založené na využití numerických metód riešenia sústav lineárnych algebraických rovníc odvodených od Gaussovej eliminačnej metódy.

### 1.3 Formulácia úlohy LP

Úlohy matematického programovania súvisia s otázkami využívania alebo rozdeľovania obmedzených prostriedkov, potrebných na dosiahnutie určených cieľov. Charakteristickou črtou takýchto úloh je veľké množstvo riešení, vyhovujúcich základným podmienkam. Výber konkrétneho riešenia ako najlepšieho závisí od zamerania, čiže od cieľa danej úlohy, vyjadreného *účelovou funkciou* a nazýva sa *optimálnym riešením* [3].

Celkove neexistuje sústava pravidiel, pomocou ktorých by bolo možné každú praktickú úlohu mechanicky previesť na matematický model a riešiť ju pomocou matematického programovania. Podľa S. I. Gausa zostavenie modelu pre nové zložité úlohy je dlhotrvajúcim procesom. Vždy je treba predovšetkým určiť premenné danej úlohy a vzťahy medzi nimi, príslušné ohraničenia a účelovú funkciu. Model sa musí preskúšať na zjednodušených skutočných alebo ľubovoľne zvolených číslach. Získané riešenie sa potom porovná s predpokladanými výsledkami.

Úzkou podtriedou úloh matematického lineárneho programovania, sú tzv. úlohy lineárneho programovania a ich matematický model spočíva na niektorých predpokladoch, ktoré nie sú v ekonomickej praxi vždy splniteľné. Je to predovšetkým predpoklad linearita vzťahov v účelovej funkcii i v ohraničeniach a predpoklad vzájomnej zastupiteľnosti (substitučnosti) činností.

V prvom rade sa musí zaviesť nasledovná symbolika: Vektory sú považované za stĺpce a označované malými písmenami. Transponovaný vektor k vektoru  $x$  (riadkový vektor) sa označuje  $x^T$ , podobne transponovaná matica k matici  $A$  sa označí  $A^T$ . Podľa toho skalárny súčin vektorov  $c, x$  sa zapisuje  $c^T x$ .

Všeobecnú formuláciu úlohy LP možno zapísať v tvare:

$$z(\mathbf{x}) = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n \quad (1.1)$$

pri ohraničujúcich podmienkach

$$\begin{aligned} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n &\leq b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n &\leq b_2 \\ \dots & \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \dots + a_{mn} \cdot x_n &\leq b_m \end{aligned} \quad (1.2)$$

a za podmienky nezápornosti:

$$x_j \geq 0 \quad j = 1, 2, \dots, n. \quad (1.3)$$

$c_n$  ..... predstavujú cenové koeficienty

$x_n$  ..... úrovne procesov

$a_{mn}$  ..... štruktúrne koeficienty

$b_m$  ..... disponibilné faktory

Lineárna funkcia (1.1) opisuje cieľ úlohy z hľadiska jej ekonomickej interpretácie, a preto sa nazýva *účelová funkcia*, rovnice a nerovnice typu (1.2) a (1.3) predstavujú podmienky *ohraničenia úlohy*. Lineárne vzťahy (1.1 až 1.3) vyjadrujú vlastne všeobecný model úlohy lineárneho programovania. Je to model optimalizačný a obsahuje:

- a) účelovú funkciu,
- b) zdroje,
- c) požiadavky,
- d) podmienky nezápornosti premenných.



### 1.3.1 Postup pri zostavovaní modelu

1. Určenie výsledku výpočtu,
2. výber hľadiska riešenia danej optimalizačnej úlohy , t.j. formulácia účelovej funkcie,
3. vecná a matematická formulácia vlastných ohraničujúcich podmienok.

### 1.3.2 Štandardný tvar úlohy LP

Rôznorodosť foriem zápisu podmienok ohraničení úlohy komplikuje rozpracovanie teórie lineárneho programovania a metód riešenia jednotlivých úloh. Preto je vhodné preskúmať spôsob úpravy ľubovoľnej úlohy LP na úlohu s jediným typom ohraničenia v tvare rovníc. Tento tvar sa nazýva *štandardný* a je vhodný na skúmanie celej teórie a metód riešenia úloh LP.

Základné charakteristiky štandardného tvaru sú:

1. všetky ohraničenia okrem ohraničení nezápornosti premenných  $x_j$  sú v tvare rovníc,
2. koeficient pravej strany každého ohraničenia je nezáporný,
3. všetky premenné sú nezáporné,
4. účelová funkcia je typu „maximalizovať“ alebo „minimalizovať“.

Úlohu LP v štandardnom tvare možno zapísať takto:

- maximalizovať (alebo minimalizovať) (1.1),
- pri ohraničujúcich podmienkach (1.2),
- za podmienok nezápornosti (1.3).

Spôsoby zápisu v skratke:

a) maximalizovať (alebo minimalizovať)

$$z(\mathbf{x}) = \sum_{j=1}^n c_j^T \cdot x_j \quad (1.4)$$

za podmienok

$$\begin{aligned} \sum_{j=1}^n a_{ij} \cdot x_j &= b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n \end{aligned} \quad (1.5)$$

alebo

b) maximalizovať (alebo minimalizovať)

$$z(\mathbf{x}) = \sum_{j=1}^n c_j^T \cdot x_j \quad (1.6)$$

za podmienok

$$\begin{aligned} \sum_{j=1}^n \mathbf{A}_j \cdot x_j &= \mathbf{b}, \\ x_j &\geq 0, \quad j = 1, \dots, n \end{aligned} \quad (1.7)$$

Kde  $\mathbf{A}_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T$ ,  $j = 1, \dots, n$  je  $j$ -ty stĺpcový vektor koeficientov sústavy ohraničení.

Medzi maximalizáciou a minimalizáciou účelovej funkcie (1.1) existujú vecné a formálne vzťahy. Využívajú sa nielen pri formulácii úlohy LP ale aj vo výpočtovom postupe a pri kontrole výsledkov riešení. Platí dôležitá zásada, že

$$\text{maximalizovať } z(\mathbf{x}) = \text{minimalizovať } [-z(\mathbf{x})]$$

a preto jednoduchou úpravou z každej úlohy (1.4 až 1.6), v ktorej sa maximalizuje funkcia  $z(\mathbf{x})$ , možno získať ekvivalentnú úlohu, v ktorej sa minimalizuje funkcia  $-z(\mathbf{x})$ .

Nakoľko všetky konkrétne úlohy LP môžu byť veľmi rozmanité a ohraničujúce podmienky môžu mať povahu rovností alebo nerovností typu  $\leq$  resp.  $\geq$ , potom všetky

takéto úlohy možno upraviť použitím elementárnych transformácií a pomocou koncepcie prídavných (doplňkových) premenných na štandardný tvar.

*Prídavná premenná* (označí sa  $x'_i$  resp.  $u$ ,  $i = 1, 2, \dots$ ) je nezáporná premenná, ktorej prislúcha koeficient účelovej funkcie rovný nule a pomocou prídavnej premennej sa transformuje ohraničenie v tvare nerovnice na rovnice, a to nasledujúcim spôsobom:

- a) doplnková premenná sa pripočíta k ľavej strane nerovnice typu  $\leq$
- b) doplnková premenná sa odpočíta od ľavej strany nerovnice typu  $\geq$ .

## 1.4 Dualita úloh LP

Ku každému matematickému modelu úlohy LP možno jednoznačne priradiť iný matematický model. Toto priradenie sa nazýva *dualitou*. Model, ku ktorému sa priradzuje, sa nazýva *primárny matematický model* a model, ktorý je priradovaný, sa nazýva *duálny matematický model* [3]. Dualita je vzťah vzájomný.

Rozoznávajú sa v podstate dva typy duálne združených úloh:

- a) symetrické duálne úlohy,
- b) nesymetrický duálne úlohy.

V maticovom vyjadrení majú tvar:

- a) *symetrické duálne úlohy*

<p style="text-align: center;">primárna úloha</p> <p>I. <math>\max z = \mathbf{c}^T \mathbf{x}</math></p> <p style="text-align: center;"><math>A\mathbf{x} \leq \mathbf{b}</math></p> <p style="text-align: center;"><math>\mathbf{x} \geq \mathbf{0}</math></p>	<p style="text-align: center;">duálna úloha</p> <p><math>\min f = \mathbf{u}^T \mathbf{b}</math></p> <p style="text-align: center;"><math>A^T \mathbf{u} \geq \mathbf{c}</math></p> <p style="text-align: center;"><math>\mathbf{u} \geq \mathbf{0}</math></p>
--	--

(1.8)

b) *nesymetrické duálne úlohy ( keď premenné  $u_i$  alebo premenné  $x_j$  sú voľné v znamienku )*

primárna úloha

$$\text{I. } \max z = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}$$

duálna úloha

$$\min f = \mathbf{u}^T \mathbf{b} \quad (1.9)$$

$$\mathbf{A}^T \mathbf{u} \geq \mathbf{c}$$

$$\text{II. } \min z = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}$$

$$\max f = \mathbf{u}^T \mathbf{b}$$

$$\mathbf{A}^T \mathbf{u} \leq \mathbf{c} \quad (1.10)$$

$$\text{III. } \max z = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{Ax} \leq \mathbf{b}$$

$$\min f = \mathbf{u}^T \mathbf{b}$$

$$\mathbf{A}^T \mathbf{u} = \mathbf{c} \quad (1.11)$$

$$\mathbf{u} \geq \mathbf{0}$$

$$\text{IV. } \min z = \mathbf{c}^T \mathbf{x}$$

$$\mathbf{Ax} \geq \mathbf{b}$$

$$\max f = \mathbf{u}^T \mathbf{b}$$

$$\mathbf{A}^T \mathbf{u} = \mathbf{c} \quad (1.12)$$

$$\mathbf{u} \geq \mathbf{0}$$

Z hore uvedených definičných vzťahov vyplýva, že duálna úloha k duálnej úlohe je opäť primárna úloha.

Pre duálne úlohy v lineárnom programovaní platí **základná veta o konečnej hodnote**: Ak má jeden zo združených matematických modelov riešenie s konečnou hodnotou účelovej funkcie, potom má i druhý model optimálne riešenie s konečnou hodnotou účelovej funkcie a hodnoty účelových funkcií sú rovnaké. Ak však naopak hodnota účelovej funkcie jedného zo združených matematických modelov môže rásť alebo klesať neobmedzene, potom druhý model nemá vôbec prípustné riešenie.

### 1.4.1 Zásady pri formulovaní duálneho modelu úlohy LP

Skôr ako sa začne formulácia duálneho modelu k primárnemu modelu úlohy lineárneho programovania, treba primárny model upraviť tak, aby všetky vlastné ohraničenia boli buď rovnice alebo nerovnice rovnakého typu [3].

a) *Postup pri úprave nerovnice:*

- pri maximalizačnom modeli sa menia všetky nerovnice na typ  $\leq$  a to tak, že nerovnice typu  $\geq$  sa vynásobia číslom -1,
- pri minimalizačnom modeli sa menia všetky nerovnice na typ  $\geq$ , čiže nerovnice typu  $\leq$  sa prenášobí číslom -1,
- ak sa medzi vlastnými ohraničeniami (ktoré sú nerovnice) vyskytli aj ohraničenia, ktoré sú rovnice, možno každú rovnicu nahradiť dvomi nerovnicami opačného typu, z ktorých jednu potom treba vynásobiť číslom -1. To znamená, že v duálnom modeli namiesto jednej premennej budú dve premenné  $u_i^+$  a  $u_i^-$ , pričom pre obe platí podmienka nezápornosti. Ich rozdiel sa značí  $u_i$ , t.j.  $u_i^+ - u_i^- = u_i$  a to znamená, že premenná  $u_i$  môže byť aj záporná. Je to voľná premenná.

b) *Postup pri úprave na rovnice*

- Pomocou prídavných premenných sa upraví nerovnice na rovnice a k takto upravenému primárnemu modelu sa napíše model duálny.

### 1.4.2 Vlastnosti duálnych úloh

1. Ak v primárnej úlohe sú niektoré nerovnosti opačného smeru, t.j. v úlohách maximalizácie sú nerovnosti

$$\sum_{j=1}^n a_{ij}x_j \geq b_i, \quad (1.13)$$

stačí ich zmeniť na tvar

$$\sum_{j=1}^n -a_{ij}x_j \leq -b_i \quad (1.14)$$

a potom sa k nim vytvoria duálne ohraničenia.

2. Ak v pôvodnej sústave sa vyskytuje rovnica v  $k$ -tom riadku, potom premenná  $u_k$  nebude ohraničená a naopak, ak v primárnej úlohe niektorá premenná  $x_k$  nie je ohraničená, potom v duálnej úlohe v  $k$ -tom riadku je znamienko rovnosti.
3. Ak majú obidve duálne združené úlohy prípustné riešenie, potom majú aj optimálne riešenie.
4. Ak má úloha lineárneho programovania optimálne riešenie, potom má aj duálna úloha k nej optimálne riešenie a optimálne hodnoty obidvoch účelových funkcií sú rovnaké.
5. Ak má jedna z duálne združených úloh viac optimálnych riešení, potom optimálne riešenie druhej úlohy je degenerované.

A tak žiadna úloha LP nie je úlohou samostatnou, ale je spojená s úlohou k nej duálnou. Každá premenná v primárnej úlohe odpovedá jednému ohraničeniu duálnej úlohy a naopak. Je zrejmé, že ku každej úlohe LP možno napísať príslušnú duálnu úlohu.

Duálne združené úlohy sa nemusia riešiť samostatne. Ak sa totiž rieši jedna z úloh simplexovou metódou, rieši sa tým automaticky aj úloha k nej duálna.

Vo vzťahoch medzi prípustnými riešeniami dvojice duálnych úloh LP môže nastať jeden z týchto prípadov [4]:

- a) Primárna aj duálna úloha majú prípustné riešenia  $\mathbf{x}$  a  $\mathbf{u}$ , pričom  $\max \mathbf{c}^T \mathbf{x} = \min \mathbf{u}^T \mathbf{b}$  (alebo  $\min \mathbf{c}^T \mathbf{x} = \max \mathbf{u}^T \mathbf{b}$ ).
- b) Ak maximum účelovej funkcie primárnej (duálnej) úlohy je neohraničené, t.j.  $\max \mathbf{c}^T \mathbf{x} = +\infty$  ( $\max \mathbf{u}^T \mathbf{b} = +\infty$ ), potom duálna (primárna) úloha nemá prípustné riešenie.
- c) Ak minimum účelovej funkcie duálnej (primárnej) úlohy je neohraničené, t.j.  $\min \mathbf{u}^T \mathbf{b} = -\infty$  ( $\min \mathbf{c}^T \mathbf{x} = -\infty$ ), potom primárna (duálna) úloha nemá prípustné riešenie.
- d) Ak jedna z dvojice duálnych úloh nemá prípustné riešenie, potom ho nemá ani druhá úloha.

### 1.4.3 Vecná interpretácia duálneho modelu

Vecný (ekonomický) obsah duálneho modelu možno najlepšie pochopiť na interpretácii primárneho a duálneho modelu vo fyzikálnych jednotkách. Základom tejto interpretácie je určenie fyzikálneho rozmeru duálnych premenných, pretože fyzikálny rozmer konštánt duálneho modelu je ten istý ako pri primárnom modeli, nakoľko duálny model je konštruovaný na základe tých istých konštánt  $a_{ij}$  ako primárny model.

Nech sú dané dve duálne združené úlohy LP. Potom platia tvrdenia:

Rozmery jednotlivých konštánt ľahko možno stanoviť z podmienok primárneho modelu. Vecne:

- koeficient  $a_{ij}$  predstavuje množstvo  $i$ -teho činiteľa, pripadajúceho na jednotku  $j$ -teho procesu,
- premenné  $u_i$  duálneho modelu oceňujú činitele,
- funkcia  $f(\mathbf{u}) = \mathbf{u}^T \mathbf{b}$  udáva celkovú cenu všetkých činiteľov, ktoré sú k dispozícii.

Pre toto relatívne (závislé) ocenenie činiteľov sa často používajú názvy *duálne* alebo *fiktívne*, *zúčtovacie* prípadne *tieňové ceny*.

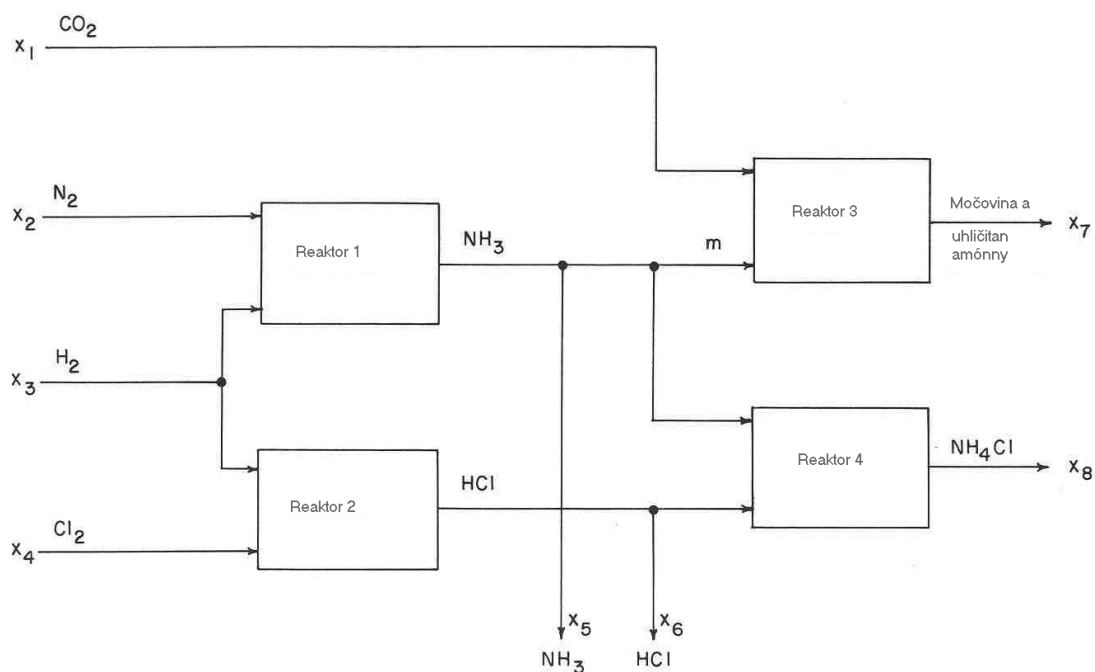
*Poznámka:* Ak je počet ohraničení primárnej úlohy väčší ako počet premenných, bude výhodnejšie riešiť duálnu úlohu.

## 1.4.4 Príklady - Riešenie úloh LP pomocou toolboxu programu MATLAB®

### 1.4.4.1 Slovné príklady

#### Príklad 1 – modifikované z [5].

Chemická továreň, ktorej bloková schéma je zobrazená na obrázku (1.1), produkuje amoniak, kyselinu chlorovodíkovú, močovinu, uhličitan amónny a chlorid amónny z oxidu uhličitého, dusíka, vodíka a chlóru. Premenné  $x$  na obrázku (1.1) udávajú rýchlosť toku v móloch za hodinu.



Obr. 1.1 Bloková schéma chemickej továrne

Ceny surovín predstavujú premenné  $c_1$ ,  $c_2$ ,  $c_3$ , a  $c_4$ , a premenné  $p_5$ ,  $p_6$ ,  $p_7$ , a  $p_8$  predstavujú hodnotu produktov v korunách/mol. Indexy týchto premenných korešpondujú s premennými  $x$ . V treťom reaktore sú pomery mólových tokov  $m=3x_7$  a  $x_1=2x_7$  a v ostatných reaktoroch je prichádzajúce množstvo surovín v stechiometrickom množstve. Kapacita prvého reaktora je menšia alebo rovná ako 2,000 mol/hod NH<sub>3</sub> a kapacita druhého reaktora je menšia alebo rovná ako 1,500 mol/hod HCl.



Úlohy:

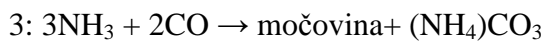
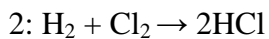
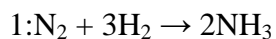
- Vytvorenie maximalizačnej funkcie
- Vytvorenie obmedzujúcich podmienok (rovníc) pre túto továreň

Riešenie:

Maximalizačná funkcia:

$$z = p_5x_5 + p_6x_6 + p_7x_7 + p_8x_8 - c_1x_1 - c_2x_2 - c_3x_3 - c_4x_4$$

Stechiometrické rovnice jednotlivých reaktorov:



Ohraničujúce podmienky:

*Reaktor 3:*

$$\text{Bilancia CO}_2 : x_1 = 2x_7$$

$$x_1 - 2x_7 = 0$$

*Reaktor 1:*

$$\text{Bilancia N}_2 : 2x_2 = 3x_7 + x_8 + x_5$$

$$2x_2 - x_5 - 3x_7 - x_8 = 0$$

*Reaktor 2:*

$$\text{Bilancia Cl}_2 : 2x_4 = x_6 + x_8$$

$$2x_4 - x_6 - x_8 = 0$$

$$\text{Limit HCl v reaktore 2 : } x_6 + x_8 \leq 15000$$

$$\text{Limit NH}_3 \text{ v reaktor 1: } x_5 + x_8 + 3x_7 \leq 2000$$

Bilancia  $H_2$  :  $x_3 = \text{tok } H_2 \text{ do rektora 1} + \text{tok } H_2 \text{ rektora 2}$

tok  $H_2$  do rektora 1 =  $(3x_7 + x_8 + x_5) \frac{3}{2}$

tok  $H_2$  do rektora 2 =  $(x_8 + x_6) \frac{1}{2}$

$x_3 = (3x_7 + x_8 + x_5) \frac{3}{2} + (x_8 + x_6) \frac{1}{2}$

Preskúpaním sa získa:

$$2x_3 - 3x_5 - x_6 - 9x_7 - 4x_8 = 0$$

Zhrnutie ohraničujúcich podmienok:

$$x_1 - 2x_7 = 0$$

$$2x_1 - x_5 - 3x_7 - x_8 = 8$$

$$2x_4 - x_6 - x_8 = 0$$

$$x_6 + x_8 \leq 15000$$

$$x_5 + 3x_7 + x_8 \leq 2000$$

$$2x_3 - 3x_5 - x_6 - 9x_7 - 4x_8 = 0$$

## Príklad 2

Chemický závod vyrába výrobky  $V_1 - V_4$ . Má k dispozícii obmedzené množstvá surovín  $S_1, S_2$  a obmedzenú kapacitu zariadenia Z. Normy spotreby týchto zdrojov sú dané v tabuľke (1.1) (spotreba surovín v tonách a spotreba zariadenia v hodinách na 1 tonu výrobku). Je treba stanoviť program, ktorým závod získa maximálny obrat. Ceny výrobkov sú 4000, 3000, 5000 a 1500 Sk za 1 tonu výrobku

	$V_1$	$V_2$	$V_3$	$V_4$	DM
$S_1$	2	-	3	1	600 t
$S_2$	-	4	4	2	600 t
Z	1	4	2	1	400 hod

Tab. 1.1

Matematický model :  $\max z = 4000x_1 + 3000x_2 + 5000x_3 + 1500x_4$

s ohraničeniami:

$$2x_1 + \quad + 3x_3 + x_4 \leq 600$$

$$4x_2 + 4x_3 + 2x_4 \leq 600$$

$$x_1 + 4x_2 + 2x_3 + x_4 \leq 400$$

Tento príklad bude riešený pomocou funkcie `linprog`, ktorá je súčasťou optimalizačného toolboxu hore uvedeného programu. Zdrojový kód danej funkcie je uvedený v prílohe [6].

Skript pre túto funkciu je uvedený zdrojovým kódom (1.1), kde premenná  $f$  obsahuje hodnoty účelovej funkcie,  $A$  je matica obmedzení,  $b$  sú hodnoty pravej strany obmedzení. Aj v ďalších zdrojových kódoch sa používa rovnaké značenie.

```
F=[-4000 -3000 -5000 -1500];  
A=[2 0 3 1; 0 4 4 2; 1 4 2 1;-1 0 0 0; 0 -1 0 0; 0 0 -1 0; 0 0 0 -1];  
b=[600;600;400;0;0;0;0;0];  
options = optimset('LargeScale','on','Display','final');  
[X,FVAL,EXITFLAG,OUTPUT] =  
LINPROG(f,A,b,[],[],[],[],[],options);  
X  
max_z=4000*X(1)+3000*X(2)+5000*X(3)+1500*X(4)
```

Zdrojový kód 1.1

Výsledok výpočtu je nasledujúci:

```
X =  
    300.0000  
     25.0000  
     0.0000  
     0.0000  
max_z =  
    1.2750e+006
```

Ak chce závod získať maximálny zisk, musí vyrábať výrobok  $V_1$  - 300 ton po 4000 Sk, a výrobok  $V_2$  - 25 ton po 3000 Sk. Ostatné dva výrobky by nemal vyrábať. Získa sa tým maximálny obrat v hodnote 1 275 000 Sk.

Matematicky zapísané:  $x_1 = 300$ ,  $x_2 = 25$ ,  $x_3 = 0$ ,  $x_4 = 0$ , a hodnota účelovej funkcie je  $z = 1\,275\,000$ .

#### 1.4.4.2 Matematicky formulované príklady

Primárna viacrozmerová úloha LP má tvar:

$$\begin{aligned}\max z(x_1, x_2) &= 4x_1 + 5x_2 + x_3 \\ 3x_1 + 2x_2 &\leq 10 \\ x_1 + 4x_2 &\leq 11 \\ 3x_3 + 3x_2 + x_3 &\leq 13 \\ x_i &\geq 0, i = 1, \bar{3}\end{aligned}$$

Úlohy:

- zostavenie duálnej úlohy LP
- riešenie primárnej (duálnej) úlohy LP:
  - pomocou funkcie *linprog*
  - pomocou funkcie *lp*

$$\begin{aligned}\min G(y_1, y_2, y_3) &= 10y_1 + 11y_2 + 13y_3 \\ 3y_1 + y_2 + 3y_3 &\geq 4 \\ 2y_1 + 4y_2 + 3y_3 &\geq 5 \\ y_3 &\geq 1 \\ y &\geq 0, i = 1, \bar{3}\end{aligned}$$

**Funkcia Linprog:**

Skript pre túto funkciu pre primárny model je uvedený zdrojovým kódom (1.2)

```
f=-[4 5 1];
A=[3 2 0; 1 4 0; 3 3 1;-1 0 0; 0 -1 0; 0 0 -1];
b=[10;11;13;0;0;0];
```

```

options = optimset('LargeScale','on','Display','final');
[X,FVAL,EXITFLAG,OUTPUT] =
LINPROG(f,A,b,[],[],[],[],[],options);
X
max_z=4*X(1)+5*X(2)+X(3)

```

Zdrojový kód 1.2

Riešenie je nasledovné:

```

X =
    1.8000
    2.3000
    0.7000
max_z =
    19.4000

```

Zápis pre *duálny* model je nasledovný:

```

f=[10 11 13];
A=[-3 -1 -3;-2 -4 -3;0 0 -1; -1 0 0;0 -1 0;0 0 -1];
b=[-4;-5;-1;0;0;0];
options = optimset('LargeScale','on','Display','final');
[X,FVAL,EXITFLAG,OUTPUT] =
LINPROG(f,A,b,[],[],[],[],[],options);
X
min_f=10*X(1)+11*X(2)+13*X(3)

```

Zdrojový kód 1.3

Riešenie je nasledovné:

```

X =
    0.2000
    0.4000
    1.0000
min_f =
    19.4000

```

### ***Funkcia Lp***

Táto funkcia síce funguje, ale v budúcnosti bude plne nahradená funkciou *linprog*. Zdrojový kód danej funkcie je uvedený v prílohe [6].

Zatiaľ však možno ukázať skript slúžiaci na výpočet príkladu 1.4.4.2.

```
f=-[4 ;5; 1];  
A=[3 2 0; 1 4 0 ;3 3 1];  
b=[10;11;13];  
x=[f,A,b];  
x=lp(f,A,b);  
vlb=[0 0 0];% definuje spodnu hranicu obmedzeni  
vub=[100,100,100];% definiuje hornu hranicu obmedzeni, pricom  
%riesenie je vzdy prave z tohto definovaneho rozsahu  
x=lp(f,A,b,vlb,vub)  
max_z=4*x(1)+5*x(2)+x(3)
```

Zdrojový kód 1.4

Riešenie je nasledovné a zhoduje sa s výsledkami pri použití funkcie *linprog*.

```
x =  
    1.8000  
    2.3000  
    0.7000  
max_z =  
    19.4000
```

### ***Riešenie problému LP geometricky***

Geometrické riešenie problému LP sa využíva pri dvojrozmerných funkciách, pri viacrozmerných sa používajú metódy spomenuté vyššie. Oba typy funkcií sa môžu riešiť simplexovou metódou.

Je zadany príklad:

$$\max z(x_1, x_2) = 10x_1 + 6x_2$$

$$4x_1 + 2x_2 \leq 20$$

$$2x_1 + 3x_2 \leq 18$$

$$x_1 \leq 4$$

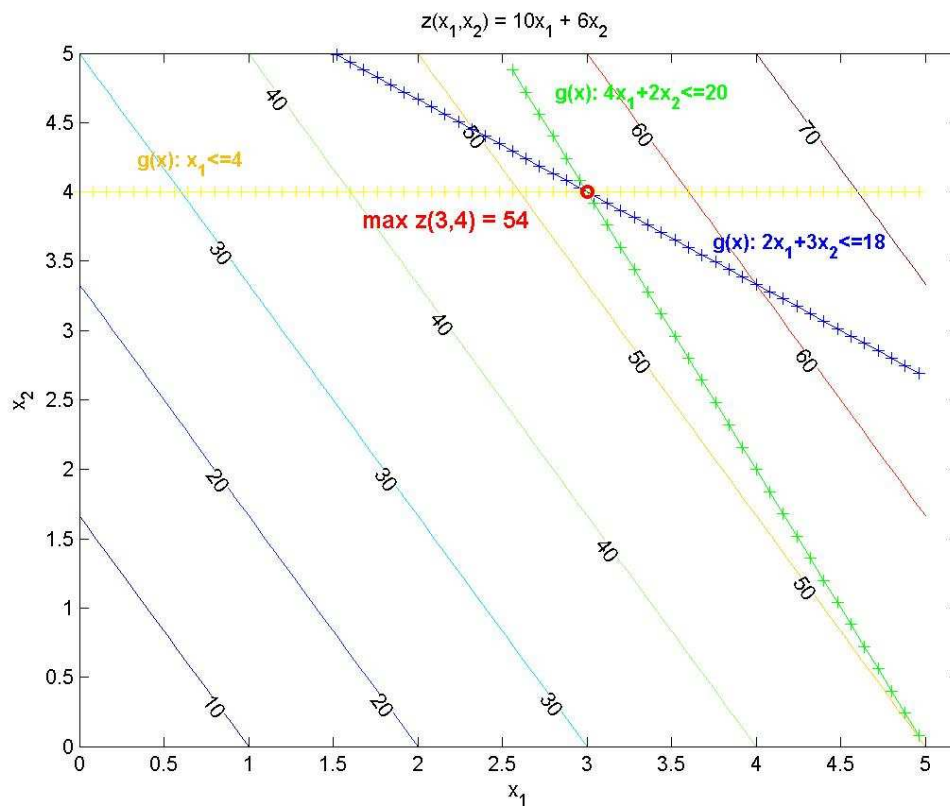
$$x_1 \geq 0, x_2 \geq 0$$

a treba ho riešiť graficky, pričom pomocou simplexovej metódy sa zistili tieto hodnoty základných premenných  $\rightarrow x_1 = 3, x_2 = 4$ . Na grafické zobrazenia dvojrozmernej optimalizačnej úlohy bolo použité programové prostredie MATLAB®.

```
[x1,x2] = meshgrid(0:0.1:5);  
[x1,x2] = meshgrid(0:0.1:5);  
E = 10.*x1 + 6.*x2;  
[p,x]=contour(x1,x2,E);  
clabel(p,x);  
xlabel('x_1');ylabel('x_2');zlabel('E(x_1,x_2)');  
title('E(x_1,x_2) = 10x_1 + 6x_2');  
axis([0 5.2 0 5])  
hold on;  
x1 = [0:0.08:5];  
x2 = (20 - 4.*x1)./2;  
plot(x1,x2,'g-+');  
text(2.75,4.5,'g(x): 4x_1+2x_2<=20','FontSize',10);  
x1 = [0:0.08:5];  
x2 = (18 - 2.*x1)./3;  
plot(x1,x2,'b-+');  
text(3.47,3.7,'g(x): 2x_1+3x_2<=18','FontSize',10);  
x1 = [0:0.08:5];  
x2 = 4.*(x1>=0);  
plot(x1,x2,'r-+');  
text(0.5,4.2,'g(x): x_1<=4','FontSize',10);  
x1 = 3; x2 = 4;  
plot(x1,x2,'co');  
text(x1-2,x2,'max z(3,4) = 54','FontSize',12);
```

Zdrojový kód 1.5

Na základe zdrojového kódu 1.4 program MATLAB<sup>®</sup> vytvoril takýto vrstevnicový graf:



Obr. 1.2 Vrstevnicový graf

Ďalšie spôsoby vytvorenia grafického riešenia úloh LP ukazuje nasledujúci skript.

```
x=0:0.1:5;
y1=max((20-4*x)./2),0)
y2=max((18-2*x)./3),0)
y3=max(4-x,x,0)
plot(x,y1,x,y2,x,y3)
ytop=min([y1;y2;y3])
area(x,ytop)
hold on
[x1,x2] = meshgrid(0:0.1:5);
z=10.*x1 + 6.*x2;
v=18:4:62;
[a,b]=contour(x1,x2,z,v); clabel(a,b)
```



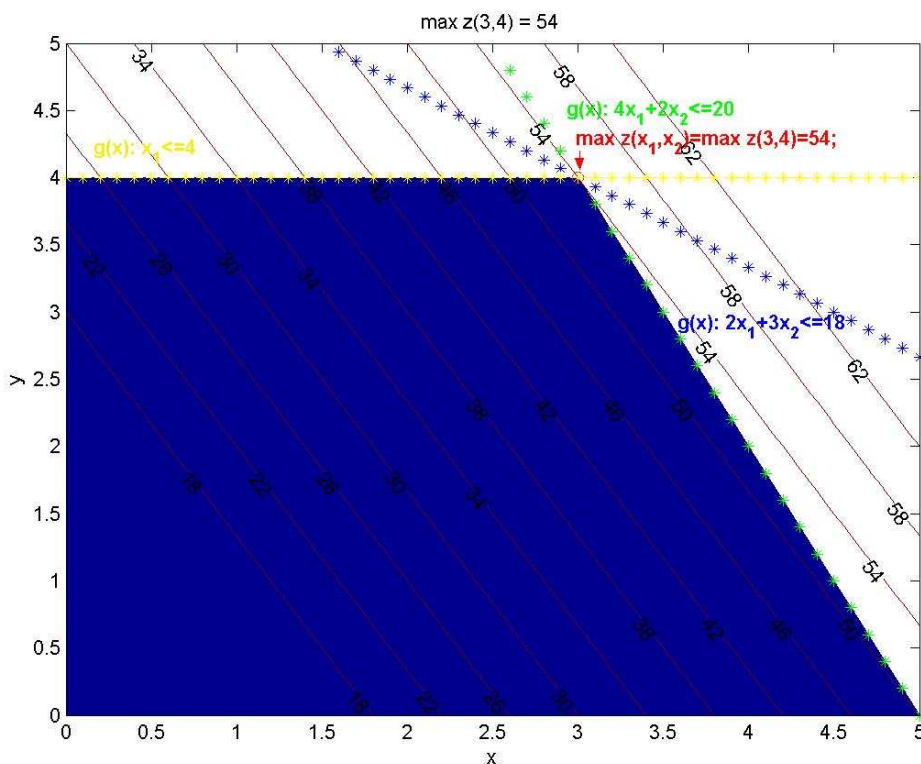
```

plot(3,4,'ro'); text(3,4.1,'max z(x_1,x_2)=max z(3,4)=54;')
xlabel('x'); ylabel('y')
title('max z(3,4) = 54 ')
plot(x,y1,'y*',x,y2,'b*',x,y3,'g*'); axis([0 5 0 5])

```

Zdrojový kód 1.6

Grafické riešenie:



Obr. 1.3

## 1.5 Úlohy o zmesiach

Definícia: Sú k dispozícii určité prostriedky  $P_j$ , z ktorých každý obsahuje účinné zložky  $Z_i$ . Je známa cena  $c_j$  za jednotku prostriedku. Úlohou je pripraviť z daných prostriedkov zmes tak, aby táto obsahovala predpísané množstvá jednotlivých zložiek a aby jej cena bola minimálna [3].

Zavedie sa tabuľka (1.2), označenie a pojmy:

	P <sub>1</sub>	P <sub>2</sub>	...	P <sub>n</sub>	
Z <sub>1</sub>	a <sub>11</sub>	a <sub>12</sub>		a <sub>1n</sub>	b <sub>1</sub>
Z <sub>2</sub>	a <sub>21</sub>	a <sub>22</sub>		a <sub>2n</sub>	b <sub>2</sub>
...					...
Z <sub>3</sub>	a <sub>m1</sub>	a <sub>m2</sub>		a <sub>mn</sub>	b <sub>m</sub>
Cena	c <sub>1</sub> ,	c <sub>2</sub> ,	...	c <sub>n</sub>	

Tab. 1.2

$n$  – počet prostriedkov  $P_j$

$m$  – počet zložiek obsiahnutých v prostriedkoch  $P_j$

$P_j$  – jednotlivé prostriedky

$Z_i$  – jednotlivé zložky

$a_{ij}$  – množstvo jednotiek  $i$  – tej zložky obsiahnutej v jednotke  $j$  – teho prostriedku

$b_i$  – minimálne množstvo požadovanej  $i$  – tej zložky vo vyrobenej zmesi

$c_j$  – cena za jednotku  $j$  – teho prostriedku

$x_j$  – množstvo jednotiek  $j$  – teho prostriedku použitého k výrobe požadovanej zmesi

Potom riešiť úlohu o zmesiach znamená určiť:

$$\min z(\mathbf{x}) = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n \quad (1.15)$$

pri ohraničujúcich podmienkach

$$\begin{aligned} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n &\geq b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n &\geq b_2 \\ &\dots \end{aligned} \quad (1.16)$$

$$\begin{aligned} a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \dots + a_{mn} \cdot x_n &\geq b_m \\ x_j &\geq 0 \quad j = 1, 2, \dots, n. \end{aligned} \quad (1.17)$$

Tieto úlohy sa vyskytujú v praxi priemyslu chemického, potravinárskeho, farmaceutického, textilného, hutníckeho, ale aj v poľnohospodárstve (kde najznámejšie sú úlohy výživové a krmivové), v obchodnej sortimentácii a pod. Je to všade tam, kde sa produkujú zlúčeniny, zliatiny a zmesi. To znamená, že ak sú k dispozícii suroviny či prostriedky, z ktorých je treba podľa určitého receptu získať zmiešaný produkt tak, aby jeho cena bola minimálna.

### 1.5.1 Príklad – úlohy o zmesiach

V piatich zliatinách  $Z_1$ - $Z_5$  s rôznym obsahom látok  $L_1$ - $L_3$  (napr. olovo, cín, zinok) sa vyrába nová zliatina  $Z_0$ . Percentuálny obsah jednotlivých látok v zliatinách a ceny zliatin v Sk za 1 kg udáva tabuľka (1.3). V novej zliatine  $Z_0$  majú byť látky  $L_1$ ,  $L_2$  a  $L_3$  zastúpené v pomere 25:9:56, pričom táto zliatina neobsahuje žiadne iné látky. Treba stanoviť, aké množstvá zliatin  $Z_1 - Z_5$  je treba použiť na výrobu 450 kg zliatiny  $Z_0$  tak, aby sa dodržalo predpísané zloženie látok a aby sa pritom na použité zliatiny vynaložilo čo najmenej prostriedkov.

	$Z_1$	$Z_2$	$Z_3$	$Z_4$	$Z_5$
$L_1$	30	40	20	20	30
$L_2$	10	10	10	20	20
$L_3$	60	50	70	60	50
Ceny	25	28	26	27	23

Tab 1.3

Nakoľko  $25 + 9 + 56 = 90$  a  $450/90 = 5 \Rightarrow$  V 450 kg zliatiny  $Z_0$  bude 125 kg látky  $L_1$ , 45 kg látky  $L_2$  a 280 kg látky  $L_3$ .

Matematický model:  $\min z = 25x_1 + 28x_2 + 26x_3 + 27x_4 + 23x_5$

za podmienok:

$$0,3x_1 + 0,4x_2 + 0,2x_3 + 0,2x_4 + 0,3x_5 \geq 125$$

$$0,1x_1 + 0,1x_2 + 0,1x_3 + 0,2x_4 + 0,2x_5 \geq 45$$

$$0,6x_1 + 0,5x_2 + 0,7x_3 + 0,6x_4 + 0,5x_5 \geq 280$$

Príklad sa dá riešiť simplexovou metódou, ktorá je podrobne rozobratá v 2. kapitole. Zatiaľ však pre ukážku možno uviesť skript pre riešenie tejto úlohy.

```
type='min'
A=[0.3 0.4 0.2 0.2 0.3 ;0.1 0.1 0.1 0.2 0.2 ;0.6 0.5 0.7 0.6
0.5];
f=[25 28 26 27 23];
rel=['>>>']
b=[125;45;280];
simplex2p(type,f,A,rel,b)
```

### Zdrojový kód 1.7

Riešenie je nasledujúce:

```
Values of the legitimate variables:  
x(1)= 350.000000  
x(2)= 0.000000  
x(3)= 100.000000  
x(4)= 0.0000  
Objective value at the optimal point:  
Z = 11350.000000
```

Z hore uvedeného riešenia jednoznačne vyplýva, že hodnoty  $x_i$  minimalizujúce účelovú funkciu  $z$  sú takéto:  $x_1 = 350$ ,  $x_2 = 0$ ,  $x_3 = 100$ ,  $x_4 = 0$ , a hodnota účelovej funkcie je  $z = 11350$ . Na výrobu 450 kg zliatiny  $Z_0$  treba použiť 350 kg zliatiny  $Z_1$  a 100 kg zliatiny  $Z_4$ , pri dodržaní predpísaného zloženia látok. Prostriedky na použité zliatiny budú mať hodnotu 11 350 Sk.

## 2. SIMPLEXOVÁ METÓDA RIEŠENIA ÚLOH LP

Je to iteračná procedúra umožňujúca po konečnom počte krokov (iterácií) určiť optimálne riešenie úloh LP, alebo stanoviť, že optimálne riešenie neexistuje.

Východiskovým bodom tohto algoritmu je nájdenie východiskového základného riešenia úloh LP. Pokiaľ je takého riešenie k dispozícii, potom simplexová metóda v jednotlivých krokoch vypočíta vždy nové základné riešenie s lepšou alebo aspoň rovnakou – v prípade maximalizácie vyššou – hodnotou účelovej funkcie. Po konečnom počte krokov teda musí tento výpočtový postup viesť k nájdeniu základného riešenia s najlepšou hodnotou účelovej funkcie alebo k zisteniu, že takéto riešenie neexistuje. Pri jeho nájdení sa musí podľa základnej vety LP jednať o riešenie optimálne [7].

Simplexová metóda skúma postupne len tie prípustné riešenia, ktoré nie sú horšie ako predchádzajúce riešenia.

Postup výpočtu pomocou simplexovej metódy sa delí na dve fázy:

1. nájdenie východiskového základného riešenia
2. iteračný postup vedúci k optimalizácii účelovej funkcie.

### 2.1 Použitie simplexovej metódy v príkladoch

Pri riešení úloh sa veľmi dobre osvedčuje usporiadanie do tzv. simplexovej tabuľky, ktorá obsahuje koeficienty sústavy  $m+1$  lineárnych rovníc v kanonickom tvare o  $n+1$  neznámych.

Do hlavičky simplexovej tabuľky sa píše označenie všetkých premenných. Do prvého stĺpca sa píše označenie premenných, ktoré tvoria v príslušnom kroku základné riešenie, do posledného stĺpca sa píše pravé strany sústavy lineárnych rovníc. Posledný riadok zodpovedá rovnici pre účelovú funkciu.

Vstupujúcej premennej zodpovedá v tabuľke tzv. **kľúčový stĺpce**, riadok, obsahujúci vystupujúcu premennú, je **kľúčový riadok**. Prvok ležiaci v priesečníku kľúčového riadku a kľúčového stĺpca sa nazýva **kľúčovým prvkom**.

Použitie tejto metódy vyžaduje upraviť matematický model úlohy LP tak:

- a) aby pre všetky premenné platili podmienky nezápornosti,
- b) aby pravé strany ohraničujúcich podmienok mali nezáporné hodnoty ( $b_i > 0$  pre  $i = 1, 2, \dots, m$ ),
- c) aby sústava vlastných ohraničení bola sústavou rovníc (realizáciou a, b, c nadobudne model tzv. štandardný, normovaný tvar),
- d) aby sústava vlastných ohraničí obsahovala kladnú jednotkovú maticu (bázu)  $m$ -tého stupňa (realizácia c, d znamená úpravu sústavy na kanonický tvar) a
- e) aby úloha bola maximalizačná.

### 2.1.1 Simplexový algoritmus

Je to spôsob výpočtu, ktorý nám umožňuje konečným počtom krokov dosiahnuť optimálne riešenie danej úlohy. Každý z týchto krokov spočíva v nájdení nového lepšieho bázičného riešenia, ktorému

- a) pri maximalizácii zodpovedá väčšia hodnota účelovej funkcie ako pri predchádzajúcom kroku
- b) pri minimalizácii mu zodpovedá menšia hodnota účelovej funkcie ako pri predchádzajúcom kroku.

Proces sa opakuje elementárnou zmenou bázy (B) dovtedy, kým sa nedosiahne optimálne riešenie.

Preto so simplexovou metódou možno riešiť len matematické modely LP upravené do kanonického tvaru. Je to tvar, keď matica sústavy  $A$  danej úlohy LP (1.1 – 1.3) obsahuje jednotkovú submaticu  $I$ , ktorej hodnosť je rovná počtu rovníc sústavy  $m$ . Kanonický tvar automaticky poskytuje bázičné riešenie sústavy, a preto aj príslušnej úlohy lineárneho programovania. Pritom účelová funkcia  $z$  sa píše ako jedna z rovníc sústavy.

Simplexový algoritmus je zostavený pre úlohy LP v štandardnom tvare [8]:

$$\begin{aligned} A \cdot \mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \\ \mathbf{c} \cdot \mathbf{x} &\rightarrow \max_x \end{aligned} \quad (2.1)$$

Riešenie takto formulovanej úlohy LP sa nájde nasledujúcim postupom:

**A.** Nájdenie východiskového bázičného riešenia a zostavenie simplexovej tabuľky

1. zavedením prídavných (doplňkových) premenných  $u$  sa prevedie úloha do kanonického tvaru

$$\begin{aligned} A \cdot \mathbf{x} + \mathbf{u} &= \mathbf{b} \\ z - \mathbf{c} \cdot \mathbf{x} &= 0 \\ z &\rightarrow \max_{x,u} \end{aligned} \quad (2.2)$$

2. ako východiskové bázičné riešenie sa volí počiatok súradnicového systému, t.j.

$$x = 0, \quad u = b$$

3. Pre lepšiu orientáciu je výhodné zapísať úlohu v prehľadnej simplexovej tabuľke

$$\begin{aligned} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n + u_1 &= b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n + u_2 &= b_2 \\ \dots &\dots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \dots + a_{mn} \cdot x_n + u_m &= b_m \\ \dots &\dots \\ z - c_1 \cdot x_1 - c_2 \cdot x_2 - \dots - c_n \cdot x_n - 0u_1 - \dots - 0u_m &= 0 \end{aligned} \quad (2.3)$$

	x1	x2	...	xn	u1	u2	...	um	
<b>b</b>	$a_{1,1}$	$a_{1,2}$	...	$a_{1,n}$	1	0	...	0	$b_1$
<b>a</b>	$a_{2,1}$	$a_{2,2}$	...	$a_{2,n}$	0	1	...	0	$b_2$
<b>z</b>	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
<b>e</b>	$a_{m,1}$	$a_{m,2}$	...	$a_{m,n}$	0	0	...	1	$b_m$
<b>z</b>	$-c_1$	$-c_2$	...	$-c_n$	0	0	...	0	$d$

Obr. 2.1 Simplexová tabuľka

## B. Nájdenie bázičského riešenie s vyššou hodnotou cieľovej funkcie

Uvažuje sa simplexová tabuľka v  $k$ -tom kroku algoritmu

	$v'$	
$e^k$	$A^k$	$b^k$
$z$	$c^k$	$d^k$

a bázičské riešenie

$$\begin{aligned} v_i^k &= b_i^k && \Leftrightarrow && i \in e^k \\ v_i^k &= 0 && \Leftrightarrow && i \notin e^k \end{aligned}$$

Teraz je možné pokúsiť sa o nájdenie iného bázičského riešenia, ktoré zvýši hodnotu cieľovej funkcie.

### 1. Výber novej bázičkej premennej (kľúčového stĺpca)

Na základe hodnoty prvkov vektora  $c^k$  v poslednom riadku simplexovej tabuľky možno rozhodnúť o vývoji hodnoty cieľovej funkcie  $z$  v prípade zaradenia odpovedajúcej premennej medzi bázičké premenné. Ak je prvok  $c_i^k$

- o kladný – hodnota cieľovej funkcie sa zníži
- o nulový – hodnota cieľovej funkcie sa nezmení
- o záporný – hodnota cieľovej funkcie sa zvýši.

Z tejto skutočnosti vyplýva, že ako novú bázičnú premennú sa bude voliť len tá, ktorej zodpovedá záporný prvok vektora  $c^k$ . Tá zaručí v ďalšom kroku simplexovej metódy nárast hodnoty cieľovej funkcie. Vždy sa volí záporný prvok s najvyššou absolútnou hodnotou. Pokiaľ žiadny taký (záporný) prvok neexistuje, neexistuje mimo bázičká premenná, ktorá by zvýšila hodnotu cieľovej funkcie. Zostávajúce bázičské riešenie je hľadaným optimálnym riešením úlohy LP.

### 2. Výber bázičkej premennej, ktorá bude vyradená z bázy (kľúčový riadok)

Pre všetky nezáporné prvky matice  $A^k$  zodpovedajúce stĺpcu novej bázičkej premennej vybranej v predchádzajúcom bode sa spočíta:

$$\frac{b_i^k}{a_{i,j}^k} \tag{2.4}$$



kde  $b_i^k$  sú prvky vektora  $\mathbf{b}^k$ ,  $a_{i,j}^k$  sú prvky matice  $A^k$  a  $j$  je index premennej vybranej v predchádzajúcom kroku. Z bázy riešení sa potom vyradí taká premenná, pre ktorú je tento podiel najmenší. Pokiaľ vo vybranom stĺpci neexistuje kladný prvok, má riešená úloha LP riešenie v nekonečne.

### 3. Prepočet nového bázického riešenia a vyčíslenie hodnoty cieľovej funkcie.

Využitím Gaussovej eliminačnej metódy sa nahradí bázická premenná vybraná v bode **B2** premennou vybranou v bode **B1**.

### 4. Cyklus simplexovej metódy.

Pokiaľ nebol dosiahnutý maximálny počet krokov simplexového algoritmu, treba sa pokúsiť nájsť ďalší bod, ktorý by zvýšil hodnotu cieľovej funkcie a pokračuje v bode **B**. V opačnom prípade vznikol nekonečný cyklus simplexového algoritmu signalizujúci, že riešenie úlohy sa nachádza v nekonečne. Simplexová metóda je v tomto prípade ukončená.

### C. Riešenie úlohy LP

Simplexový algoritmus môže byť ukončený jedným z troch nasledujúcich výsledkov:

- Existuje práve jedno riešenie
- Existuje nekonečne veľa riešení
- Riešenie sa nachádza v nekonečne.

### *Kritérium optimálnosti*

- a) Ak sústave rovníc sú všetky koeficienty  $z_j$  pre  $j = 1, 2, \dots, n$  nezáporné, t.j.  $z_j \geq 0$ , nemožno už hodnotu účelovej funkcie prechodom na iné bázické riešenie zvýšiť, preto dané prípustné riešenie je optimálne a hodnota účelovej funkcie je maximálna.
- b) Ak v sústave rovníc sú všetky koeficienty  $z_j$  pre  $j = 1, 2, \dots, n$  nekladné, t.j.  $z_j \leq 0$ , nemožno už hodnotu účelovej funkcie prechodom na bázické prípustné riešenie znížiť, preto dané prípustné riešenie je optimálne a hodnota účelovej funkcie je minimálna.

Ako je známe, pri riešení úloh LP pomocou simplexovej metódy sa okrem riešenia primárnej úlohy nachádza aj riešenie duálnej úlohy. Zo simplexového algoritmu vyplýva, že v simplexovej tabuľke sa v stĺpcoch jednotkovej východiskovej bázy na konečnom kroku dosiahne *inverzia bázy*. Okrem toho prvky posledného riadka tabuľky pod vektormi východiskovej bázy v končenej tabuľke sú vlastne hodnoty duálnych premenných  $u_i$ , ak ocenenia východiskových bázičských premenných boli nulové.

### 2.1.2 Príklad 1

**Úloha maximalizačná, vlastné obmedzenie nerovnosti typu  $\leq$**

Modifikované zo [9].

Nájdienie maxima funkcie:  $z = 8x_1 + 4x_2$

za podmienok:

$$-2x_1 + 6x_2 \leq 18$$

$$4x_1 + 6x_2 \leq 36$$

$$4x_1 - 2x_2 \leq 20$$

$$x_i \geq 0 \quad (i = 1, 2)$$

Zavedením nezáporných prídavných (doplňkových) premenných  $x'_1, x'_2, x'_3$  (resp.  $u_1, u_2, u_3$ ) a pripojením rovnice pre účelovú funkciu sa dosiahne sústava štyroch lineárnych rovníc o šiestich neznámych v kanonickom tvare:

$$-2x_1 + 6x_2 + x'_1 = 18$$

$$4x_1 + 6x_2 + x'_2 = 36$$

$$4x_1 - 2x_2 + x'_3 = 20$$

$$z - 8x_1 - 4x_2 = 0$$

Rozšířená matice tejto sústavy sa zapíše do simplexovej tabuľky:

1. krok	z	x <sub>1</sub>	x <sub>2</sub>	x' <sub>1</sub>	x' <sub>2</sub>	x' <sub>3</sub>	b
x' <sub>1</sub>	0	-2	6	1	0	0	18
x' <sub>2</sub>	0	4	8	0	1	0	36
x' <sub>3</sub>	0	4	-2	0	0	1	20
z	1	-8	-4	0	0	0	0

Tab. 2.1

Východiskové základné riešenie:

$$x'_1 = 18, \quad x'_2 = 36, \quad x'_3 = 20, \quad x_1 = x_2 = 0, \quad z = 0.$$

Kritérium pre to, či je riešenie optimálne, dáva riadok **z**. Keďže obsahuje záporné koeficienty, nie je riešenie optimálne.

Za vstupujúcu premennú sa zvolí  $x_1$ , lebo tejto premennej zodpovedá v riadku **z** záporný koeficient s najväčšou absolútnou hodnotou (t.j. kľúčovým stĺpcom je stĺpec druhý).

Vystupujúcu premennú možno určiť tak, že sa podelia postupne čísla v poslednom stĺpci tabuľky číslami z kľúčového stĺpca, pokiaľ sú tieto čísla kladné. Získané podiely sú  $36/4$  a  $20/4$ . Menší podiel je v treťom riadku, preto vystupujúcou premennou bude  $x'_3$  (t.j. kľúčovým riadkom je tretí riadok). Červený koeficient je kľúčovým prvkom.

V ďalšom riešení sa postupuje takto:

- V prvom stĺpci sa nahradí označenie vystupujúcej premennej označením vstupujúcej premennej,
- prvky v riadku vstupujúcej premennej sa získajú tak, že sa podelia prvky kľúčového riadku kľúčovým prvkom,
- prvky v ostatných riadkoch sa získajú tak, že od každého riadku sa odčíta taký násobok riadku vstupujúcej premennej, aby v kľúčovom stĺpci boli nuly (okrem jednotky v riadku vstupujúcej premennej).

Jedná sa vlastne o použitie Gaussovej eliminačnej metódy zvláštnou voľbou kľúčového prvku, ktorá zabezpečuje nezápornosť riešenia.

Ďalší krok teda vyzerá takto:

2. krok	z	$x_1$	$x_2$	$x'_1$	$x'_2$	$x'_3$	b
$x'_1$	0	0	5	1	0	1/2	28
$x'_2$	0	0	8	0	1	-1	16
$x_1$	0	1	-1/2	0	0	1/4	5
z	1	0	-8	0	0	2	40

Tab. 2.2

Zlepšenie riešenia:

$$x_1 = 5, \quad x_2 = 0, \quad x'_1 = 28, \quad x'_2 = 16, \quad x'_3 = 0, \quad z = 40$$

Ešte stále je v riadku **z** záporný koeficient, teda riešenie nie je optimálne.

Ďalší krok:

3. krok	z	$x_1$	$x_2$	$x'_1$	$x'_2$	$x'_3$	b
$x'_1$	0	0	0	1	-5/8	9/8	18
$x_2$	0	0	1	0	1/8	-1/8	2
$x_1$	0	1	0	0	1/16	3/16	6
z	1	0	0	0	1	1	56

Tab. 2.3

**Optimálne riešenie:**

$$\begin{aligned} x_1 &= 6 & x_2 &= 2 \\ x'_1 &= 18 & x'_2 &= 0 & x'_3 &= 0 \\ z &= 56 \end{aligned}$$

Optimálne riešenie, ktoré sa získalo, je jediné, pretože v riadku **z** sú vo všetkých nezákladných premenných (t.j. u premenných  $x'_2, x'_3$ ) kladné koeficienty (zaradením týchto premenných do riešenia by hodnota účelovej funkcie klesla).

### 2.1.3 Príklad 2

#### Úloha minimalizačná, vlastné obmedzenie rovnice

Nájdenie minima funkcie:  $z = 30x_1 + 39x_2 + 33x_3 + 26x_4$

za podmienok:

$$0.5x_1 + 0.3x_2 + 0.4x_3 + 0.7x_4 = 186$$

$$0.5x_1 + 0.7x_2 + 0.6x_3 + 0.3x_4 = 114$$

$$x_i \geq 0 \quad (i = 1, 2, 3, 4)$$

Pretože sústava lineárnych rovníc nie je v kanonickom tvare, treba zaviesť pomocné premenné a tým sa získa rozšírená sústava:

$$0.5x_1 + 0.3x_2 + 0.4x_3 + 0.7x_4 + x''_1 = 186$$

$$0.5x_1 + 0.7x_2 + 0.6x_3 + 0.3x_4 + x''_2 = 114$$

$$\text{kde } x''_1 \geq 0, \quad x''_2 \geq 0$$

Aby sa pomocné premenné anulovali, treba riešiť v prvej fáze pomocnú úlohu (jej vyriešením sa získa východiskové základné riešenie vlastnej úlohy):

Nájdenie minima funkcie:

$$z'' = x''_1 + x''_2$$

Z funkcie  $z''$  je nutné vylúčiť pomocné premenné, pretože tieto premenné majú tvoriť východiskové základné riešenie:

$$x''_1 = 186 ; \quad x''_2 = 114 ; \quad x_i = 0 \text{ pre } i = 1, 2, 3, 4$$

Urobí sa to tak, že k rovnici  $z'' - x''_1 - x''_2 = 0$  sa pripočítajú obe rovnice rozšírenej sústavy. Získaná rovnica:

$$z'' + x_1 + x_2 + x_3 + x_4 = 300$$

Do simplexovej tabuľky sa zapíše táto sústava rovníc (okrem koeficientu u) premennej  $z$  (resp.  $z''$ ), pretože táto premenná bude základnou premennou v celom priebehu riešenia:

$$0.5x_1 + 0.3x_2 + 0.4x_3 + 0.7x_4 + x''_1 = 186$$

$$0.5x_1 + 0.7x_2 + 0.6x_3 + 0.3x_4 + x''_2 = 114$$

$$z - 30x_1 - 39x_2 - 33x_3 - 26x_4 = 0$$

$$z'' + x_1 + x_2 + x_3 + x_4 = 300$$

Bude sa minimalizovať funkcia  $z''$ , v okamihu keď  $z''$  dosiahne nulu, bude sa minimalizovať  $z$ . Pretože ide o problémy minimalizačné, možno riešenie zlepšiť, pokiaľ v riadku  $z''$  (resp.  $z$ ) je kladný koeficient.

Postup bude zrejmý z tabuliek (2.4-2.6):

1.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x''_1$	$x''_2$	b
$x''_1$	0.5	0.3	0.4	0.7	1	0	186
$x''_2$	0.5	0.7	0.6	0.3	0	1	114
$z$	-30	-39	-33	-26	0	0	0
$z''$	1	1	1	1	0	0	300

Tab. 2.4

2.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x''_1$	$x''_2$	b
$x''_1$	0	-0.4	-0.2	0.4	1	-1	72
$x_1$	1	1.4	1.2	0.6	0	2	228
$z$	0	3	3	-8	0	60	6840
$z''$	0	-0.4	1	0.4	0	-2	72

Tab. 2.5

3.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x''_1$	$x''_2$	b
$x_4$	0	-1	-0.5	1	2.5	-2.5	180
$x_1$	1	2	1.5	0	-1.5	3.5	120
$z$	0	-5	-1	0	20	40	8280
$z''$	0	0	0	0	-1	-1	0

Tab. 2.6

V prvom kroku sa zvolil prvý stĺpec za kľúčový (bez ohľadu na čo možný najväčší pokles hodnoty  $z''$ ). Po dvoch krokoch sa podarilo anulovať funkciu  $z''$ . Ďalej by sa dalo pokračovať v minimalizácii funkcie  $z$  (vynechal by sa riadok  $z''$  a stĺpce  $x''_1$  a  $x''_2$ ), ale v tomto prípade bola vyriešením pomocnej úlohy zároveň vyriešená pôvodná úloha, pretože v poslednom riadku  $z$  sú už všetky koeficienty záporné.

**Optimálne riešenie:**

$$x_1 = 120 \quad x_2 = x_3 = 0 \quad x_4 = 180$$

$$z = 8280$$

### 2.1.4 Príklad 3

**Úloha minimalizačná, vlastné obmedzenie nerovnosti typu  $\geq$**

Nájdenie minima funkcie:  $z = 15x_1 + 10x_2 + 12x_3 + 25x_4$

za podmienok:

$$2x_2 + 4x_3 + 5x_4 \geq 5000$$

$$2x_1 + 2x_2 + 4x_4 \geq 6000$$

$$10x_1 + 5x_2 + 4x_3 + 10x_4 \geq 18000$$

$$x_i \geq 0 \quad (i = 1, 2, 3, 4)$$

Po zavedení prídavných premenných sa získa obmedzenie v tvare rovníc:

$$\begin{aligned} 2x_2 + 4x_3 + 5x_4 - x'_1 &= 5000 \\ 2x_1 + 2x_2 + 4x_4 - x'_2 &= 6000 \\ 10x_1 + 5x_2 + 4x_3 + 10x_4 - x'_3 &= 18000 \\ x'_i &\geq 0 \quad (i = 1, 2, 3) \end{aligned}$$

Aby sa nemuseli zavádzať tri pomocné premenné, možno upraviť sústavu rovníc takto: Treba nájsť rovnicu s najväčším absolútnym členom a od nej sa odčítajú ostatné rovnice (od tretej rovnice sa odčíta prvá a druhá rovnica).

$$\begin{aligned} 10x_1 + 3x_2 + 5x_4 + x'_1 - x'_3 &= 13000 \\ 8x_1 + 3x_2 + 4x_3 + 6x_4 + x'_2 - x'_3 &= 12000 \\ 10x_1 + 5x_2 + 4x_3 + 10x_4 - x'_3 &= 18000 \end{aligned}$$

Teraz stačí zaviesť jedinú pomocnú premennú, a to v tretej rovnici. Obdobne ako v predchádzajúcom prípade, do simplexovej tabuľky sa zapíše táto sústava rovníc:

$$\begin{aligned} 10x_1 + 3x_2 + 5x_4 + x'_1 - x'_3 &= 13000 \\ 8x_1 + 3x_2 + 4x_3 + 6x_4 + x'_2 - x'_3 &= 12000 \\ 10x_1 + 5x_2 + 4x_3 + 10x_4 - x'_3 + x''_3 &= 18000 \\ z - 15x_1 - 10x_2 - 12x_3 - 25x_4 &= 0 \\ z'' + 10x_1 + 5x_2 + 4x_3 + 10x_4 - x'_3 &= 18000 \\ x''_3 &\geq 0 \end{aligned}$$

Postup riešenia bude opäť zrejmý z tabuliek (2.7-2.11):

1.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x'_1$	$x'_2$	$x'_3$	$x''_3$	b
$x'_1$	10	3	0	5	1	0	-1	0	13000
$x'_2$	8	3	4	6	0	1	-1	0	12000
$x''_3$	10	5	4	10	0	0	-1	1	18000
Z	-15	-10	-12	-25	0	0	0	0	0
$z''$	10	5	4	10	0	0	-1	0	18000

Tab. 2.7



2.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x'_1$	$x'_2$	$x'_3$	$x''_3$	b
$x'_1$	5	0.5	-2	0	1	0	-0.5	-0.5	4000
$x'_2$	2	0	1.6	0	0	1	-0.4	-0.6	1200
$x_4$	1	0.5	0.4	1	0	0	-0.1	0.1	1800
z	10	2.5	-2	0	0	0	-2.5	-	45000
$z''$	0	0	0	0	0	0	0	-1	0

Tab. 2.8

3.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x'_1$	$x'_2$	$x'_3$	$x''_3$	b
$x'_1$	0	0.5	-6	0	1	-2.5	-0.5	-	1000
$x_1$	1	0	0.8	0	0	0.5	-0.2	-	600
$x_4$	0	0.5	-0.4	1	0	-0.5	0.1	-	1200
z	0	2.5	-10	0	0	-5	-0.5	-	39000

Tab. 2.9

4.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x'_1$	$x'_2$	$x'_3$	$x''_3$	b
$x_2$	0	1	-12	0	2	-5	1	-	2000
$x_1$	1	0	0.8	0	0	0.5	-0.2	-	600
$x_4$	0	0	5.6	1	-1	2	-0.4	-	200
z	0	0	20	0	-5	-3	-3	-	34000

Tab. 2.10

5.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x'_1$	$x'_2$	$x'_3$	$x''_3$	b
$x_2$	0	1	2	2.5	-0.5	0	0.8	-	2500
$x_1$	1	0	-0.6	-0.25	0.25	0	0	-	550
$x'_2$	0	0	2.8	0.5	-0.5	1	-0.2	-	100
z	0	0	-1	-3.75	-1.25	0	-1.5	-	33250

Tab. 2.11

**Optimálne riešenie:**

$$\begin{aligned}x_1 &= 550 & x_2 &= 2500 & x_3 &= x_4 = 0 \\x'_1 &= 0 & x'_2 &= 100 & x'_3 &= 0 \\z &= 33250\end{aligned}$$

## 2.1.5 Príklad 4

**Úloha, ktorá nemá optimálne riešenie**

Nájdenie minima funkcie:  $z = -2x_3 - x_4 + x_5$

za podmienok:

$$\begin{aligned}x_1 + x_3 - 2x_4 &= 3 \\x_2 - x_3 + x_4 + 2x_5 &= 4 \\x_i &\geq 0 \quad (i = 1, 2, 3, 4, 5)\end{aligned}$$

Simplexová tabuľka:

1.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	b
$x_1$	1	0	1	-2	0	3
$x_2$	0	1	-1	1	2	4
$z$	0	0	2	1	-1	0

Tab. 2.12

2.krok	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	b
$x_3$	1	0	1	-2	0	3
$x_2$	1	1	0	-1	1	7
$z$	-2	0	0	5	-1	-6

Tab. 2.13

Riešenie nie je optimálne, pretože v riadku  $z$  je kladný koeficient (minimalizačná úloha). Kľúčový stĺpec možno zvoliť ( $x_4$ ), ale nemožno zvoliť kľúčový riadok, pretože oba koeficienty sú záporné.

**Úloha teda nemá optimálne riešenie.**

## 2.1.6 Príklad 5

### Interaktívne riešenie úlohy LP pomocou JavaScriptu pre Gaussovú eliminačnú metódu

Pre edukačné účely sú na internete dostupné rôzne interaktívne moduly na riešenie úloh LP.

V tejto diplomovej práci bolo použité interaktívne riešenie z programu uvedeného na www stránke [10].

Tento interaktívny modul má nasledujúce podmienky pre úlohu LP:

- maximalizácia účelovej funkcie
- najviac tri obmedzenie (rovnice)
- najviac šesť premenných (nezáporných)

Interaktívna simplexová tabuľka je zobrazená na obr.(2.2).

Simplexová tabuľka 2

Promenne zakladni	x1	x2	x3	x4	x5	x6	Prava strana	Volba radku
x <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
x <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
x <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
Z (účelova funkce)	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
Volba sloupce	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ITEROVAT	VYMAZAT

Obr. 2.2 Interaktívna simplexová tabuľka

Je zadaný nasledujúci príklad a ohraničeniami:

$$\max z(x_1, x_2) = 80x_1 + 60x_2 + 100x_3 + 300x_4$$

$$x_1 + 4x_2 + 2x_3 + x_4 \leq 4$$

$$2x_1 + 3x_3 + x_4 \leq 6$$

$$8x_3 + 4x_4 \leq 12$$

$$x_i \geq 0, i = 1, 4$$

### Postup riešenia:

1. zadať príslušné koeficienty (t.j. vyplniť simplexovú tabuľku) vrátane indexov základných premenných,
2. pomocou kruhových tlačidiel zvoliť kľúčový prvok (podľa známych pravidiel simplexového algoritmu),
3. zadať prevedenie jedného iteračného kroku,
4. rozhodnúť, či zlepšené riešenie je optimálne - ak áno, tak skončiť, pokiaľ nie, tak opakovať body 2 až 4.

Za vstupujúcu premennú sa zvolí  $x_4$ , lebo tejto premennej zodpovedá v riadku **z** záporný koeficient s najväčšou absolútnou hodnotou (t.j. kľúčovým stĺpcom je stĺpec druhý).

Vystupujúca premenná je  $x_7$  (t.j. kľúčovým riadkom je prvý riadok). Pomocou kruhových tlačidiel sa zvolí kľúčový prvok.

Po vyplnení a zvolení kľúčového prvku vyzerá simplexová tabuľka takto:

Promenne zakladni	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	Prava strana	Volba radku
$x_5$	1	4	2	1	1	0	4	<input type="radio"/>
$x_6$	2	0	3	1	0	1	6	<input type="radio"/>
$x_7$	0	0	8	4	0	0	12	<input type="radio"/>
<b>z</b> (cieľová funkcia)	-80	-60	-100	-300	0	0		<input type="radio"/>
Volba sloupce	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ITEROVAT	VYMAZAT

Obr. 2.3 Vyplnená simplexová tabuľka

Po kliknutí na tlačidlo iterovať, sa vykoná prvá iterácia a získa sa zlepšené riešenie. Treba rozhodnúť či je optimálne. Ak nie je optimálne, treba zopakovať výber kľúčového prvku a opäť iterovať. Za vstupujúcu premennú sa zvolí  $x_1$ , lebo tejto premennej zodpovedá v riadku **z** záporný koeficient s najväčšou absolútnou hodnotou (t.j. kľúčovým stĺpcom je stĺpec druhý). Vystupujúca premenná je  $x_5$  (t.j. kľúčovým riadkom je prvý riadok). Pomocou kruhových tlačidiel sa opäť zvolí kľúčový prvok.

Promenne zakladni	x1	x2	x3	x4	x5	x6	Prava strana	Volba radku
x5	1	4	0	0	1	0	1	<input type="radio"/>
x6	2	0	1	0	0	1	3	<input type="radio"/>
x4	0	0	2	1	0	0	3	<input type="radio"/>
Z (ucelova funkce)	-80	-60	500	0	0	0	900	<input type="radio"/>
Volba sloupce	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ITEROVAT	VYMAZAT

Obr. 2.4 Simplexová tabuľka po zadání klíčového řádku

Po ďalšej iterácii je zrejmé z tabuľky na obr. (2.5), že sa jedná o optimálne riešenie, pretože v riadku **z** sú vo všetkých nezákladných premenných kladné koeficienty.

Promenne zakladni	x1	x2	x3	x4	x5	x6	Prava strana	Volba radku
x1	1	4	0	0	1	0	1	<input type="radio"/>
x6	0	-8	1	0	-2	1	1	<input type="radio"/>
x4	0	0	2	1	0	0	3	<input type="radio"/>
Z (ucelova funkce)	0	260	500	0	80	0	980	<input type="radio"/>
Volba sloupce	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ITEROVAT	VYMAZAT

Obr. 2.5 Simplexová tabuľka po druhej iterácii

**Optimálne riešenie:**

$$x_1 = 1 \quad x_2 = 0 \quad x_3 = 0 \quad x_4 = 3$$

$$x'_1 = 0 \quad x'_2 = x_6 = 1 \quad x'_3 = x_7 = 0$$

$$z = 980$$

Z výsledkov vyplýva, že aj použitím tejto metódy, aj prostredníctvom príkazov a funkcií programu MATLAB<sup>®</sup> je výsledok rovnaký, a teda sa jedná o optimálne riešenie.

### 2.1.6 Príklad 6

**Dvojfázová metóda – funkcia *simplex2p* použitá prostredníctvom toolboxu programu MATLAB®**

Dvojfázová metóda je jednou zo základných výpočtových pomôcok používaných v lineárnom programovaní.

V priebehu vykonávania funkcie *simplex2p* v prostredí MATLAB®, používateľ má možnosť sledovania postupu výpočtu, a to kliknutím na možnosť "áno" v zobrazenom okne.

Zadávanie *rel* v zdrojovom kóde funguje na nasledovnom princípe: napríklad, ak *rel* = '<<<', potom systém ohraničení pozostáva z troch nerovností typu <=.

Zdrojový kód danej funkcie je uvedený v prílohe [11].

#### Príklad 1

Motocyklový závod vyrába tri typy vozidiel A, B, C. Výroba jedného typu môže byť po malých úpravách vystriedaná výrobou iného typu. Pri stanovení výrobného programu sa musí vziať do úvahy denná kapacita šiestich prevádzok, ktoré sú obmedzené podľa tabuľky (2.14). Ostatné zdroje – suroviny, pracovné sily a pod. – sú k dispozícii v dostatočnom množstve. Je treba určiť výrobný plán závodu tak, aby závod mal maximálny zisk.

	Denná kapacita pri výrobe		
	Vozidlo A	Vozidlo B	Vozidlo C
Zlievareň	100	125	75
Úpravovňa odliatkov	150	125	100
Lisovňa	125	100	100
Montáž A	75	-	-
Montáž B	-	80	-
Montáž C	-	-	80
Ceny	45000	40000	60000

Tab. 2.14

Matematický model:  $\max z = 45000x_1 + 40000x_2 + 60000x_3$

s ohraňčeniami

$$\frac{100}{100}x_1 + \frac{100}{125}x_2 + \frac{100}{75}x_3 \leq 100$$

$$\frac{100}{150}x_1 + \frac{100}{125}x_2 + \frac{100}{100}x_3 \leq 100$$

$$\frac{100}{125}x_1 + \frac{100}{100}x_2 + \frac{100}{100}x_3 \leq 100$$

$$x_1 \leq 75$$

$$x_2 \leq 80$$

$$x_3 \leq 80$$

Skript pre spustenie výpočtu simplexovej metódy je uvedený zdrojovým kódom (2.1).

```
type='max'
f=[45000 40000 60000 0 0 0];
A=[100/100 100/125 100/75;100/150 100/125 100/100 ;100/125
100/100 100/100; 1 0 0;0 1 0; 0 0 1];
b=[100;100;100;75;80;80];
rel='<<<<<<';
simplex2p(type,f,A,rel,b)
```

Zdrojový kód 2.1

Výsledok výpočtu simplexovej metódy prostredníctvom optimalizačného toolboxu programu MATLAB<sup>®</sup> je nasledujúci:

```
Values of the legitimate variables:
x(1)= 0.000000
x(2)= 62.500000
x(3)= 37.500000
Objective value at the optimal point:
z = 4750000.000000
```

Z výsledkov vyplýva, že ak chce podnik dosiahnuť maximálny zisk, mal by vyrábať vozidlá typu B vozidlá typu C. Matematicky:  $x_1 = 0$ ,  $x_2 = 62,5$ ,  $x_3 = 37,5$ , a hodnota účelovej funkcie je  $z = 4\,750\,000$ .

## Príklad 2

Podnik vyrába tri výrobky  $V_1 - V_3$  pomocou štyroch technologických postupov  $T_1 - T_4$ . Údaje o produktivite jednotlivých technológií (počet jednotiek každého výrobku vyprodukovaných pri jednotkovej intenzite príslušnej technológie, náklady spojené s jednotkovou intenzitou využitia každej technológie a minimálne požadované množstvá jednotlivých výrobkov, ktoré podnik musí vyrobiť v skúmanom období) sú dané v tabuľke (2.17). Je treba určiť výrobný program s minimálnymi úhrnnými nákladmi.

	<b>T<sub>1</sub></b>	<b>T<sub>2</sub></b>	<b>T<sub>3</sub></b>	<b>T<sub>4</sub></b>	<b>Požadované množstvo</b>
<b>V<sub>1</sub></b>	4	3	3	2	190
<b>V<sub>2</sub></b>	4	3	3	3	180
<b>V<sub>3</sub></b>	3	2	4	4	130
<b>Náklady</b>	70	50	90	70	

Tabuľka 2.15

Matematický model:  $\min z = 70x_1 + 50x_2 + 90x_3 + 70x_4$

s ohraničeniami

$$4x_1 + 3x_2 + 3x_3 + 2x_4 \geq 190$$

$$4x_1 + 3x_2 + 3x_3 + 3x_4 \geq 180$$

$$3x_1 + 2x_2 + 4x_3 + 4x_4 \geq 130$$

Zápis pre minimalizačnú funkciu vyzerá nasledovne:

```
type='min'
A=[4 3 3 2;4 3 3 3;3 2 4 4];
f=[70 50 90 70];
rel=['>>>']
b=[190; 180; 130];
simplex2p(type,f,A,rel,b)
```



Riešenie:

```

Values of the legitimate variables:
x(1)= 10.000000
x(2)= 50.000000
x(3)= 0.000000
x(4)= 0.000000
Objective value at the optimal point:
Z = 3200.000000

```

Z výsledkov vyplýva, že ak chce podnik dosiahnuť maximálny zisk, mal by vyrábať iba výrobky  $V_1$  a výrobky  $V_2$ . Matematicky:  $x_1 = 10$ ,  $x_2 = 50$ ,  $x_3, x_4 = 0$ , a hodnota účelovej funkcie je  $z = 3200$ .

### 2.1.7 Príklad 7

#### Duálny simplexový algoritmus – funkcia *dsimplex*

Ďalšia metóda dôležitá pri lineárnom programovaní je metóda duálneho simplexového algoritmu. Pre obsiahlosť tejto témy sa predložená diplomová práca nezaobrá týmto algoritmom, ale je tu predložený spôsob výpočtu prostredníctvom optimalizačného toolboxu programu MATLAB® - tento algoritmus sa nazýva *dsimplex*.

Zdrojový kód danej funkcie je uvedený v prílohe [11].

#### Príklad:

Matematický model:  $\min z = 3x_1 + 4x_2 + x_3 + 2x_4$

s ohraničeniami

$$3x_1 + 4x_2 + x_3 + 2x_4 \geq 5$$

$$x_1 + 2x_2 - x_3 + x_4 \geq 4$$

Skript pre minimalizačnú funkciu vyzerá nasledovne:

```
type='min';  
A=[1 1 1 1;1 2 -1 1];  
f=[3 4 1 2];  
b=[5;4];  
subs=dsimplex(type, f, A, b)
```

Zdrojový kód 2.3

Riešenie je nasledovné:

```
Hodnoty vyslednych premennych:  
x(1)= 0.000000  
x(2)= 0.000000  
x(3)= 0.500000  
x(4)= 4.500000  
Realna hodnota optimalneho riesenia:  
Z = 9.500000  
Indexy bazických premenných vo finalnej tabulke :  
3  
4
```

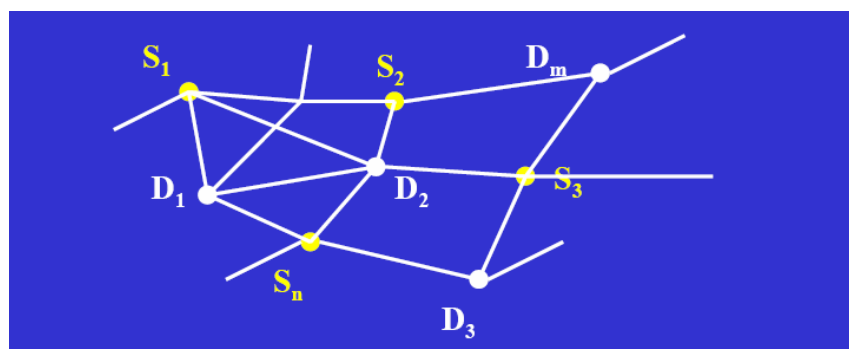
### 3. DOPRAVNÉ ÚLOHY

#### 3.1 Dopravná úloha a jej aplikácie

Ide o najstaršiu aj najlepšie prepracovanú úlohu lineárneho programovania, nazývanú tiež úlohou Hitchcockovou, ktorý sa ňou ako prvý podrobne zaoberal.

Cieľ dopravných úloh – minimalizácia nákladov na prepravu materiálov medzi výrobcami, resp. medzi výrobcami a miestom odbytu.

##### 3.1.1 Formulácia dopravnej úlohy



Obr. 3.1 Grafické znázornenie siete odberateľov a dodávateľov

$D_1, D_2, \dots, D_m$  - dodávateľia

$S_1, S_2, \dots, S_n$  - spotrebitelia

$a_1, a_2, \dots, a_m$  - dodávky od dodávateľov  
spotrebitel'ov

$b_1, b_2, \dots, b_n$  - požiadavky

Majme  $m$  výrobcov (dodávateľov)  $D_1, D_2, \dots, D_m$  deliteľného homogénneho tovaru, produkovaného v množstvách  $a_1, a_2, \dots, a_m$  a na strane druhej  $n$  spotrebitel'ov (odberateľov)  $S_1, S_2, \dots, S_n$ , požadujúcich tento tovar v množstvách  $b_1, b_2, \dots, b_n$ . Bude sa pritom predpokladať, že sú prípustné väzby každého z výrobcov s každým spotrebitel'om [10].

Označí sa hľadané množstvo prepravovaného tovaru od  $i$ -teho výrobcu  $v_i$  k  $j$ -temu spotrebiteľovi  $S_j$  symbolom  $x_{ij}$ . Sú to premenné, ktoré musia spĺňať podmienky nezápornosti:

$$x_{ij} \geq 0 \quad \text{pre} \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m \quad (3.1)$$

Pre zjednodušenie úlohy sa bude predpokladať, že prepravné náklady medzi výrobcom (dodávateľom)  $D_i$  a spotrebiteľom  $S_j$  rastú lineárne s prepravovaným množstvom  $x_{ij}$ . Hodnotu týchto nákladov pri jednotkovom množstve tovaru  $x_{ij}$  označíme  $c_{ij}$ .

### 3.1.2 Vyvážená dopravná úloha

Je najjednoduchšou variantov dopravnej úlohy a slovné ju možno charakterizovať takto podľa [3]:

Za minimálny celkový náklad je potrebné prepraviť určitý počet jednotiek tovaru z rôznych výrobných centier do rôznych spotrebných centier. Každé spotrebné centrum potrebuje určité množstvo jednotiek tovaru a každé výrobné centrum má k dispozícii obmedzené množstvo jednotiek tovaru.

Predpoklad: koľko spotrebiteľia požadujú, toľko dodávatelia dodajú

$$a_1 + a_2 + \dots + a_m = b_1 + b_2 + \dots + b_n \quad (3.2)$$

$$\sum_{i=1}^m a_i = \sum_{k=1}^n b_k \rightarrow \text{podmienka vyváženosti} \quad (3.3)$$

Sadzby za dopravu:

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix} = [c_{ik}]_1^{m,n} \quad (3.4)$$

$c_{ik}$  – jednotkové prepravné náklady na tovar prepravený od i-teho dodávateľa  $D_i$  ku k-temu spotrebiteľovi  $S_k$ .

Úloha: Stanovenie matematického modelu optimalizácie celkových prepravných nákladov, t.j. minimalizácia celkových prepravných nákladov.

Riešenie: Určiť prepravný program, t.j. množstvá  $x_{ik}$ , ktoré dodá i-ty dodávateľ  $D_i$  k-temu spotrebiteľovi  $S_k$ , tak aby sa minimalizovali celkové prepravné náklady.

$$x = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} = [x_{ik}]_1^{m,n} \quad (3.5)$$

$x_{ik}$  – množstvo ktoré dodávateľ  $D_i$  dodá spotrebiteľovi  $S_k$

## 3.2 Konštrukcia modelu

Podmienky:

Dodávateľské:

$$D_1 : x_{11} + x_{12} + \dots + x_{1n} = a_1$$

$$D_2 : x_{21} + x_{22} + \dots + x_{2n} = a_2$$

.....

$$D_m : x_{m1} + x_{m2} + \dots + x_{mn} = a_m$$

Odberateľské:

$$S_1 : x_{11} + x_{21} + \dots + x_{m1} = b_1$$

$$S_2 : x_{12} + x_{22} + \dots + x_{m2} = b_2 \quad (3.6)$$

.....

$$S_n : x_{1n} + x_{2n} + \dots + x_{mn} = b_n$$

Nezápornosti:  $x_{ik} \geq 0$

$$\text{Vyváženosť: } \sum_{i=1}^m a_i = \sum_{k=1}^n b_k \quad (3.7)$$

Prepravné náklady:

$$\begin{aligned} \min Z = & c_{11} \cdot x_{11} + c_{12} \cdot x_{12} + \dots + c_{1n} \cdot x_{1n} + \\ & + c_{21} \cdot x_{21} + c_{22} \cdot x_{22} + \dots + c_{2n} \cdot x_{2n} + \\ & + \dots + \\ & + c_{m1} \cdot x_{m1} + c_{m2} \cdot x_{m2} + \dots + c_{mn} \cdot x_{mn} \end{aligned} \quad (3.8)$$

$$\min Z = \sum_{i=1}^m \sum_{k=1}^n c_{ik} \cdot x_{ik} \quad (3.9)$$

### 3.2.1 Zápis modelu

	$S_1$	$S_2$	....	$S_n$	$a_i$
$D_1$	$c_{11}$ $x_{11}$	$c_{12}$ $x_{12}$	...	$c_{1n}$ $x_{1n}$	$a_1$
$D_2$	$c_{21}$ $x_{21}$	$c_{22}$ $x_{22}$	...	$c_{2n}$ $x_{2n}$	$a_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$D_m$	$c_{m1}$ $x_{m1}$	$c_{m2}$ $x_{m2}$	...	$c_{mn}$ $x_{mn}$	$a_m$
$b_k$	$b_1$	$b_2$	...	$b_n$	

Obr. 3.2 Zápis modelu dopravnej úlohy

Sústavu  $(m+n-1)$  rovníc s  $(m \cdot n)$  premennými možno riešiť simplexovou metódou ako úlohu lineárneho programovania. Avšak už pri malých hodnotách  $m$  a  $n$  sa možno stretnúť s veľkou sústavou ohraničení, ktorá je pre ručný výpočet nevhodná.

G. B. Dantzig upravil simplexovú metódu vzhľadom na špeciálnu štruktúru matice koeficientov modelu dopravného typu tak, že je možné určiť riešenie úlohy bez zostavenia simplexovej tabuľky.

Všetky zvláštnosti dopravných úloh umožňujú riešiť tieto úlohy mnohými jednoduchými, matematicky nenáročnými metódami, ktoré sú ľahko zvládnuteľné i bez pomoci výpočtovej techniky. Tieto metódy sa nazývajú *približné (iteračné, praktické alebo suboptimálne)*.

Dopravný problém je výhodné riešiť pomocou modifikovanej dvojrozmernej tabuľky. Pomocou tejto tabuľky je možné nájsť prípustné a potom aj optimálne riešenie dopravnej úlohy.

### 3.3 Vlastnosti dopravnej úlohy

Vlastnosti dopravnej úlohy podľa [2]:

1. Dopravná úloha má vždy prípustné riešenie, t.j. konvexná množina prípustných riešení je neprázdna.
2. Dopravná úloha má vždy bázičné riešenie.
3. Ak dodávky  $a_i$  a požiadavky  $b_k$  sú celé čísla, potom optimálne riešenie je celočíselné.
4. Dopravná úloha má vždy optimálne riešenie.
5. Kantorovič: Nevyhnutnou a postačujúcou podmienkou k tomu, aby prípustné riešenie  $x = [x_{ik}]_{I_1}^{m,n}$  bolo optimálnym riešením je, aby existovali také čísla  $u_i$  a  $v_j$ , pre ktoré platí:
  - $u_i + v_j = d_{ik} \leq c_{ik}$
  - pre tie  $i, k$ , pre ktoré je  $x_{ik} > 0$ , platí:  $u_i + v_j = c_{ik}$ , kde:  
 $c_{ik} \rightarrow$  dopravné sadzby  
 $d_{ik} \rightarrow$  duálne sadzby  
 $u_i, v_j \rightarrow$  duálne premenné.

#### 3.3.1 Riešenie vybilancovanej dopravnej úlohy

Postup riešenia:

1. metódou SZ rohu (indexovou, Vogelovou aproximačnou...) sa zostaví prvé prípustné riešenie
2. vypočítajú sa duálne premenné  $u_i, v_j$ . (V prvom kroku sa ľubovoľná duálna premenná volí = 0, ostatné sa dopočítajú). Pre bázičné (obsadené) premenné  $x_{ik} > 0$  platí:  
 $u_i + v_j = c_{ik}$
3. pre neobsadené polia sa vypočítajú duálne dopravné sadzby  $d_{ik} = u_i + v_j$
4. kontrola optima:  $d_{ik} \leq c_{ik}$
5. Ak:  $d_{ik} \geq c_{ik}$ , treba vyhľadať:  $\max [d_{ik} - c_{ik}] \rightarrow$  zaradí sa dopravu na pozíciu  $i.k$  +
6. vytvorí sa uzavretý cyklus, tak aby sa neporušili bilančné podmienky
7. zmena  $\delta = \min \{ x_{ik}^- \}$ ,  $\delta$  je najmenší prvok

## 8. nové bážické riešenie.

Na popis hľadanie riešenie dopravného problému možno využiť nasledujúce fakty [12]:

Dá sa dokázať, že prípustné riešenie dopravného problému je také, ktoré obsahuje maximálne  $m+n-1$  nenulových premenných  $x_{ij}$ . Pri hľadaní riešenia sa vychádza zo spomenutej dvojrozmernej tabuľky a do príslušných políčok tejto tabuľky sa zapisujú hodnoty premenných  $x_{ij}$ . Pokiaľ sú tieto hodnoty nulové, tak ich do políčok nezapíšujeme. Políčko, v ktorom je hodnota premennej kladná, sa nazýva *obsadené políčko*. Tieto obsadené políčka sa pospájajú vodorovnými a zvislými čiarami, aby sa zistilo, či tieto obsadené políčka vytvárajú *uzavretý cyklus*. Pri riešení dopravného problému sa vychádza z nejakého prípustného riešenia a testuje sa, či je toto riešenie optimálne. Ak obsahuje maximálne  $m+n-1$  obsadených políčok a tieto políčka nevytvárajú uzavretý cyklus (a spĺňa podmienku optimality), tak sa našlo hľadané riešenie dopravnej úlohy, v opačnom prípade sa prejde na ďalšie prípustné riešenie a opakuje sa test optimality. Ak počet obsadených políčok je menší ako  $m+n-1$ , tak riešenie je *degenerované*.

Test optimality: Ak všetky diferencie sú nulové alebo nezáporné, tak dané prípustné riešenie je optimom.

Najčastejšie používané dopravné úlohy:

- metóda severozápadného rohu
- indexova metóda
- Vogelova aproximačná metóda
- modifikovaná metóda
- Habrova metóda úzkych profilov.



### 3.4 Metóda severozápadného rohu

Polia sa začínajú obsadzovať v ľavom hornom rohu dopravnej tabuľky a postupuje sa smerom doprava, pričom každé pole sa obsadzuje najvyššou možnou prepravou s ohľadom na kapacity  $a_i$ , požiadavky  $b_k$  a už obsadené polia. Na každom kroku sa obsadí jeden riadok (stĺpec), výnimočne dva rady (ak napr.  $a_1 = b_1$  a dosadí sa  $x_{11} = a_1$ , obsadí sa súčasne prvý riadok a prvý stĺpec). Na poslednom kroku sa obsadzuje zostávajúci riadok alebo stĺpec súčasne. To znamená, že riešenie, ktoré sa opísaným spôsobom získa, bude mať maximálne  $(m+n-1)$  obsadených polí. Je to práve toľko, koľko je v sústave (3.6-3.8) lineárne nezávislých ohraničení. Táto metóda neberie do úvahy výšku dopravných sadzieb (nákladov alebo dopravných vzdialeností) [13].

**Poznámka.** Východiskové riešenie možno v zásade získať tak, že pri obsadzovaní sa začne ktorýmkoľvek poľom a pokračuje sa ľubovoľne. Je však nutné zachovať pravidlo, že do každého poľa sa dosadzuje maximálne možná hodnota, t.j. pri každom obsadení poľa sa musí dbať na to, aby bol obsadený celý riadok (stĺpec).

Metóda severozápadného rohu umožňuje jednoducho a rýchlo získať východiskové riešenie. Riešenie dosiahnuté touto metódou je prijateľné, ale nie konečné.

### 3.4.1 Príklad – Metóda severozápadného rohu

Nájdite taký plán prepravy tovaru od troch dodávateľov štyrom spotrebiteľom, aby náklady na prepravu boli minimálne pri úplnom vyčerpaní kapacít dodávateľov a plnom uspokojení požiadaviek spotrebiteľov.

Kapacity dodávateľov, požiadavky spotrebiteľov a prepravné sadzby sú uvedené v tabuľke:

Dodávatelia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6	8	8	5	300
D <sub>2</sub>	5	11	8	7	400
D <sub>3</sub>	8	9	7	13	500
Požiadavky	350	280	320	250	1200

Tab. 3.1

#### Riešenie:

Začína sa v ľavom hornom (severozápadnom) rohu nasledujúcim postupom:  $x_{ij} = \min(\text{kapacita}, \text{požiadavka})$ . Snahou je dosiahnuť vyčerpanie kapacity prvého dodávateľa, alebo požiadavky prvého spotrebiteľa.

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6 300	8	8	5	300
D <sub>2</sub>	5	11	8	7	400
D <sub>3</sub>	8	9	7	13	500
Požiadavky	350	280	320	250	1200

Tab. 3.2

Ak nie je uspokojená požiadavka prvého spotrebiteľa, tak treba pokračovať v jej uspokojovaní z kapacít ďalších dodávateľov.

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6 300	8	8	5	300
D <sub>2</sub>	5 50	11	8	7	400
D <sub>3</sub>	8	9	7	13	500
Požiadavky	350	280	320	250	1200

Tab. 3.3

Po uspokojení požiadavky prvého spotrebiteľa sa pokračuje v uspokojovaní požiadaviek ďalších spotrebiteľov z kapacít ďalších dodávateľov.

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6 300	8	8	5	300
D <sub>2</sub>	5 50	11 280	8 70	7	400
D <sub>3</sub>	8	9	7	13	500
Požiadavky	350	280	320	250	1200

Tab. 3.4

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6 300	8	8	5	300
D <sub>2</sub>	5 50	11 280	8 70	7	400
D <sub>3</sub>	8	9	250	250	500
Požiadavky	350	280	320	250	1200

Tab. 3.5

$$z_1 = 6 * 300 + 5 * 50 + 11 * 280 + 8 * 70 + 7 * 250 + 13 * 250 = 10690$$

Riešením metódou severozápadného rohu sa dosiahla vyššie uvedená hodnota.

### 3.5 Indexová metóda

**Použitie:** Na získanie základného, východiskového riešenia zohľadnením sadzieb a ich veľkosti.

Základným postupom je obsadiť políčka maximálne možným množstvom tak, aby sa riadkové a stĺpcové súčty rovnali okrajovým hodnotám.

Pri rovnakých sadzbách uprednostniť políčko, ktoré je možné obsadiť väčším množstvom.

#### 3.5.1 Príklad – Indexova metóda

Zadanie je to isté ako pri metóde severozápadného rohu.

Kapacity dodávateľov, požiadavky spotrebiteľov a prepravné sadzby sú rovnaké ako pri príklade na metódu severozápadného rohu a sú uvedené v tabuľke (3.1).

#### Riešenie:

- pôvodnú tabuľku treba previesť do tabuľky:

Políčko	Kapacita	Požiadavka	Kapacita/Požiadavka	$c_{ij}$
$D_1S_1$	300	350	0,85714	6
$D_1S_2$	300	280	1,07143	8
$D_1S_3$	300	320	0,9375	8
$D_1S_4$	300	250	1,2	5
$D_2S_1$	400	350	1,14286	5
$D_2S_2$	400	280	1,42857	11
$D_2S_3$	400	320	1,25	8
$D_2S_4$	400	250	1,6	7
$D_3S_1$	500	350	1,42857	8
$D_3S_2$	500	280	1,78571	9
$D_3S_3$	500	320	1,5625	7
$D_3S_4$	500	250	2	13

Tab. 3.6

- údaje v tabuľke zoradíme podľa veľkostí sadzieb vzostupne:

Políčko	Kapacita	Požiadavka	Kapacita/Požiadavka	$c_{ij}$	Poradie
D <sub>2</sub> S <sub>1</sub>	400	350	1,14286	5	1
D <sub>1</sub> S <sub>4</sub>	300	250	1,2	5	2
D <sub>1</sub> S <sub>1</sub>	300	350	0,85714	6	3
D <sub>3</sub> S <sub>3</sub>	500	320	1,5625	7	4
D <sub>2</sub> S <sub>4</sub>	400	250	1,6	7	5
D <sub>3</sub> S <sub>1</sub>	500	350	1,42857	8	6
D <sub>2</sub> S <sub>3</sub>	400	320	1,25	8	7
D <sub>1</sub> S <sub>3</sub>	300	320	0,9375	8	8
D <sub>1</sub> S <sub>2</sub>	300	280	1,0714	8	9
D <sub>3</sub> S <sub>2</sub>	500	280	1,78571	9	10
D <sub>2</sub> S <sub>2</sub>	400	280	1,42857	11	11
D <sub>3</sub> S <sub>4</sub>	500	250	2	13	12

Tab. 3.7

- na základe zoradenej tabuľky sa budú podľa poradia obsadzovať políčka maximálnym možným množstvom, pričom ani riadkové a ani stĺpcové súčty nesmú prekročiť ani hranicu kapacít ani hranicu požiadaviek

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6 300	8 50	8	5 250	300
D <sub>2</sub>	5 350	11 50	8	7	400
D <sub>3</sub>	8	9 180	7 320	13	500
Požiadavky	350	280	320	250	1200

Tab. 3.8

Všetko sa pre kontrolu zapíše do vzostupne zoradenej tabuľky:

Políčko	Kapacita	Požiadavka	Kapacita/Požiadavka	$c_{ij}$	Poradie	$x_{ij}$
$D_2S_1$	400	350	1,14286	5	1	350
$D_1S_4$	300	250	1,2	5	2	250
$D_1S_1$	300	350	0,85714	6	3	0
$D_3S_3$	500	320	1,5625	7	4	320
$D_2S_4$	400	250	1,6	7	5	0
$D_3S_1$	500	350	1,42857	8	6	0
$D_2S_3$	400	320	1,25	8	7	0
$D_1S_3$	300	320	0,9375	8	8	0
$D_1S_2$	300	280	1,0714	8	9	50
$D_3S_2$	500	280	1,78571	9	10	180
$D_2S_2$	400	280	1,42857	11	11	50
$D_3S_4$	500	250	2	13	12	0
						<b>1200</b>

Tab. 3.9

$$z_2 = 5 \cdot 350 + 5 \cdot 250 + 7 \cdot 320 + 8 \cdot 50 + 9 \cdot 180 + 11 \cdot 50 = 7810$$

Riešenie indexovou metódou poskytl hore uvedenú hodnotu účelovej funkcie.

### 3.6 Vogelova aproximačná metóda

Pre každý riadok a stĺpec treba vyrátať tzv. diferenciu. Diferencia je rozdiel medzi najmenšími cenovými koeficientmi v danom riadku resp. stĺpci. Nájde sa riadok (stĺpec) s maximálnou diferenciou a tam sa obsadí políčko s najnižšou sadzbou. V prípade, že majú riadky (stĺpce) rovnaké diferencie, obsadzuje sa najnižšia sadzba. Po obsadení poľa treba znížiť kapacitu i požiadavky a znova sa prepočítajú diferencie a zopakuje sa postup.

## **Postup:**

### **1. Výpočet rozdielov „ $R_i$ “ a „ $R_j$ “**

- a) k tabuľke sa pridá stĺpec pre „ $R_i$ “ a riadok pre „ $R_j$ “
- b) v každom riadku a stĺpci sa vypočíta rozdiel  $R$  medzi najnižšou a najbližšou vyššou hodnotou sadzby prepravných nákladov. Ak je viac ako jedna najnižšia hodnota sadzby (buď v riadku alebo v stĺpci), tak  $R = 0$ .

### **2. Vyhľadanie najväčšieho rozdielu „ $R$ “ v stĺpcoch alebo riadkoch**

Vo vypočítaných rozdieloch „ $R$ “ sa nájde najväčšiu hodnotu, a v príslušnom riadku alebo stĺpci sa obsadí políčko s najnižšou hodnotou sadzby prepravných nákladov najväčším možným objemom (menšia hodnota z porovnania požiadavky s kapacitou).

Ak existuje niekoľko najväčších rozdielov „ $R$ “ treba uprednostniť políčko s najnižšou hodnotou sadzby prepravných nákladov. Ak by existovali aj rovnaké hodnoty sadzieb prepravných nákladov zvolí sa spomedzi políčok s rovnakými hodnotami sadzieb prepravných nákladov to políčko, do ktorého možno umiestniť najväčší objem (z porovnania požiadavky a kapacity)

V riadku alebo stĺpci kde sa objem v políčku rovná kapacite alebo požiadavke zapísať do zostávajúcich políčok „0“.

### **3. Opakovanie krokov 1. a 2.**

- sadzba v obsadenom/ých políčku/ach sa pri výpočte rozdielov neberie do úvahy
- pri obsadzovaní políčok sa nepočíta s políčkami s „nulovým“ objemom
- opakovanie sa realizuje do naplnenia všetkých požiadaviek pri súčasnom vyčerpaní všetkých kapacít.

#### **3.6.1 Príklad – Vogelova aproximačná metóda**

Zadanie je to isté ako pri metóde severozápadného rohu.

Kapacity dodávateľov, požiadavky spotrebiteľov a prepravné sadzby sú uvedené v tabuľke (3.1).



### Riešenie:

- tabuľku treba doplniť o stĺpec „ $R_i$ “ a riadok „ $R_j$ “
- vypočítajú sa rozdiely „ $R$ “ vo všetkých riadkoch a stĺpcoch
- po výpočte rozdielom sa zistilo, že sú dve najväčšie hodnoty „2“ a zároveň sú aj dve políčka ( $D_1S_4$  a  $D_2S_1$ ) s najnižšími sadzbami prepravných nákladov „5“, takže si treba zvoliť z týchto dvoch políčok to, do ktorého možno umiestniť najväčší objem (porovnanie požiadavky a kapacity), teda políčko  $D_2S_1$

Dodávateľia	Spotrebitelia				Kapacity	$R_i$
	$S_1$	$S_2$	$S_3$	$S_4$		
$D_1$	6	8	8	5	300	1
$D_2$	5 350	11	8	7	400	2
$D_3$	8	9	7	13	500	1
Požiadavky	350	280	320	250	1200	
$R_j$	1	1	1	2		

Tab. 3.10

- bola naplnená požiadavka spotrebiteľa  $S_1$  preto možno do políčok  $D_1S_1$  a  $D_3S_1$  zapísať „0“

Dodávateľia	Spotrebitelia				Kapacity	R <sub>i</sub>
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>		
D <sub>1</sub>	6 0	8	8	5	300	1
D <sub>2</sub>	5 350	11	8	7	400	2
D <sub>3</sub>	8 0	9	7	13	500	1
Požiadavky	350	280	320	250	1200	
R <sub>j</sub>	1	1	1	2		

Tab. 3.11

- vypočítajú sa rozdiely „R“ vo všetkých riadkoch a zostávajúcich troch stĺpcoch, pričom pri výpočtoch rozdielov v riadkoch sa neberú do úvahy sadzby prepravných nákladov z políček prvého stĺpca
- najväčší rozdiel je „3“ a políčko s najnižšou sadzbou „5“ je políčko D<sub>1</sub>S<sub>4</sub>

Dodávateľia	Spotrebitelia				Kapacity	R <sub>i</sub>
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>		
D <sub>1</sub>	6 0	8	8	5 250	300	3
D <sub>2</sub>	5 350	11	8	7	400	1
D <sub>3</sub>	8 0	9	7	13	500	2
Požiadavky	350	280	320	250	1200	
R <sub>j</sub>	-	1	1	2		

Tab. 3.12

- bola naplnená požiadavka spotrebiteľa S<sub>4</sub> preto možno do políček D<sub>2</sub>S<sub>4</sub> a D<sub>3</sub>S<sub>4</sub> zapísať „0“

- vypočítajú sa rozdiely „R“ vo všetkých riadkoch a zostávajúcich dvoch stĺpcoch, pričom pri výpočtoch rozdielov v riadkoch sa neberú do úvahy sadzby prepravných nákladov z políček prvého a štvrtého stĺpca

Dodávateľia	Spotrebitelia				Kapacity	R <sub>i</sub>
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>		
D <sub>1</sub>	6 0	8	8	5 250	300	0
D <sub>2</sub>	5 350	11	8	7 0	400	3
D <sub>3</sub>	8 0	9	7	13 0	500	2
Požiadavky	350	280	320	250	1200	
R <sub>j</sub>	-	1	1	-		

Tab. 3.13

- najväčší rozdiel je „3“ a políčko s najnižšou sadzbou „8“ je políčko D<sub>2</sub>S<sub>3</sub> ale do tohto políčka možno umiestniť už len „50“ jednotiek (rozdiel medzi kapacitou „400“ a objemom políčka D<sub>2</sub>S<sub>1</sub> „350“)

Dodávateľia	Spotrebitelia				Kapacity	R <sub>i</sub>
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>		
D <sub>1</sub>	6 0	8	8	5 250	300	0
D <sub>2</sub>	5 350	11	8 50	7 0	400	3
D <sub>3</sub>	8 0	9	7	13 0	500	2
Požiadavky	350	280	320	250	1200	
R <sub>j</sub>	-	1	1	-		

Tab. 3.14

- bola vyčerpaná kapacita dodávateľa D<sub>2</sub> preto možno do políčka D<sub>2</sub>S<sub>2</sub> zapísať „0“
- vypočítajú sa rozdiely „R“ v zostávajúcich dvoch riadkoch a zostávajúcich dvoch stĺpcoch, pričom pri výpočtoch rozdielov v riadkoch sa neberú do úvahy sadzby prepravných nákladov z políček prvého a štvrtého stĺpca a pri výpočtoch rozdielov v stĺpcoch sa neberú do úvahy sadzby prepravných nákladov z políček prostredného riadka

Dodávateľia	Spotrebitelia				Kapacity	R <sub>i</sub>
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>		
D <sub>1</sub>	6 0	8	8	5 250	300	0
D <sub>2</sub>	5 350	11 0	8 50	7 0	400	-
D <sub>3</sub>	8 0	9	7	13 0	500	2
Požiadavky	350	280	320	250	1200	
R <sub>j</sub>	-	1	1	-		

Tab. 3.15

- najväčší rozdiel je „2“ a políčko s najnižšou sadzbou „7“ je políčko  $D_3S_3$  ale do tohto políčka možno umiestniť už len „270“ jednotiek (rozdiel medzi požiadavkou „320“ a objemom políčka  $D_2S_3$  „50“)

Dodávateľia	Spotrebitelia				Kapacity	$R_i$
	$S_1$	$S_2$	$S_3$	$S_4$		
$D_1$	6 0	8	8	5 250	300	0
$D_2$	5 350	11 0	8 50	7 0	400	-
$D_3$	8 0	9	7 270	13 0	500	2
Požiadavky	350	280	320	250	1200	
$R_j$	-	1	1	-		

Tab. 3.16

- bola naplnená požiadavka spotrebiteľa  $S_3$  preto možno do políčka  $D_1S_3$  zapísať „0“
- do políčka  $D_1S_3$  zostáva zapísať rozdiel medzi kapacitou dodávateľa  $D_1$  a políčkom  $D_1S_4$
- do políčka  $D_3S_2$  zostáva zapísať rozdiel medzi požiadavkou spotrebiteľa  $S_2$  a políčkom  $D_1S_2$

- získa sa konečné riešenie

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6 0	8 50	8 0	5 250	300
D <sub>2</sub>	5 350	11 0	8 50	7 0	400
D <sub>3</sub>	8 0	9 230	7 270	13 0	500
Požiadavky	350	280	320	250	1200

Tab. 3.17

- treba rozhodnúť, či získané riešenie je konečné (optimálne) alebo východiskové pre „modifikovanú metódu“

Dodávateľia	Spotrebitelia				Kapacity
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	
D <sub>1</sub>	6	8	8	5	300
D <sub>2</sub>	5	11	8	7	400
D <sub>3</sub>	8	9	7	13	500
Požiadavky	350	280	320	250	1200

Tab. 3.18

- tabuľka sa doplní o stĺpec „ $u_i$ “ a riadok „ $v_j$ “

Dodávateľia	Spotrebitelia				Kapacity	$u_i$
	$S_1$	$S_2$	$S_3$	$S_4$		
$D_1$	6 250	8 50	8	5	300	
$D_2$	5 350	11	8 50	7	400	
$D_3$	8	9 230	7 270	13	500	
Požiadavky	350	280	320	250	1200	
$v_j$						

Tab. 3.19

- v riadkoch 1, 2 a 3 ako aj v stĺpcoch 2 a 3 je najväčší počet nenulových hodnôt a preto „0“ môže byť ktorékoľvek z riadkových čísel ako aj jedno zo stĺpcových čísel  $v_2$  a  $v_3$

Dodávateľia	Spotrebitelia				Kapacity	$u_i$
	$S_1$	$S_2$	$S_3$	$S_4$		
$D_1$	6 250	8 50	8	5	300	
$D_2$	5 350	11	8 50	7	400	
$D_3$	8	9 230	7 270	13	500	
Požiadavky	350	280	320	250	1200	
$v_j$		0				

Tab. 3.20

- najprv sa vypočíta „ $u_1$ “ a „ $u_3$ “:

$$u_1 + v_2 = 8 \quad u_1 = 8 - 0 = 8$$

$$u_3 + v_2 = 9 \quad u_3 = 9 - 0 = 9$$

Dodávateľia	Spotrebitelia				Kapacity	$u_i$
	$S_1$	$S_2$	$S_3$	$S_4$		
$D_1$	6 350	8 50	8 50	5 250	300	8
$D_2$	5 350	11	8 50	7	400	
$D_3$	8	9 230	7 270	13	500	9
Požiadavky	350	280	320	250	1200	
$v_j$		0				

Tab. 3.21

- vypočítajú sa ostatné „ $u_i$ “ a „ $v_j$ “:

$$u_3 + v_3 = 7 \quad v_3 = 7 - 9 = -2$$

$$u_2 + v_3 = 8 \quad u_2 = 8 - (-2) = 10$$

$$u_2 + v_1 = 5 \quad v_1 = 5 - 10 = -5$$

$$u_1 + v_4 = 5 \quad v_4 = 5 - 8 = -3$$



Dodávateľia	Spotrebitelia				Kapacity	$u_i$
	$S_1$	$S_2$	$S_3$	$S_4$		
$D_1$	6 350	8 50	8 50	5 250	300	8
$D_2$	5 350	11	8 50	7	400	10
$D_3$	8	9 230	7 270	13	500	9
Požiadavky	350	280	320	250	1200	
$v_j$	-5	0	-2	-3		

Tab. 3.22

- vypočítajú sa rozdiely „ $(u_i + v_j) - c_{ik}$ “ pre všetky neobsadené políčka:

$$(u_1 + v_1) - c_{11} = (8 + (-5)) - 6 = 3 - 6 = -3$$

$$(u_1 + v_3) - c_{13} = (8 + (-2)) - 8 = 6 - 8 = -2$$

$$(u_2 + v_2) - c_{22} = (10 + 0) - 11 = 10 - 11 = -1$$

$$(u_2 + v_4) - c_{24} = (10 + (-3)) - 7 = 7 - 7 = 0$$

$$(u_3 + v_1) - c_{31} = (9 + (-5)) - 8 = 4 - 8 = -4$$

$$(u_3 + v_4) - c_{34} = (9 + (-3)) - 13 = 6 - 13 = -7$$

Z výpočtov rozdielov  $(u_i + v_j) - c_{ik}$  vyplýva, že ide o konečné (optimálne) riešenie a náklady na prepravu sú pri tomto spôsobe uspokojenia požiadaviek spotrebiteľov danými kapacitami dodávateľov najnižšie:

$$z_3 = 5 \cdot 350 + 8 \cdot 50 + 9 \cdot 230 + 8 \cdot 50 + 7 \cdot 270 + 5 \cdot 250 = 7760$$

### 3.7 Modifikovaná metóda (metóda riadkových a stĺpcových čísel)

**Postup:**

**1. Výpočet riadkových a stĺpcových čísel  $u_i$  a  $v_j$**

1. k tabuľke sa pridá stĺpec pre riadkové čísla  $u_i$  a riadok pre stĺpcové čísla  $v_j$
2. v riadku alebo stĺpci s najväčším počtom nenulových premenných sa dosadí za  $u_i$  resp.  $v_j$  nula
3. ostatné čísla  $u_i$  a  $v_j$  sa volia tak, aby sa súčet riadkového a stĺpcového čísla rovnal prepravným nákladom obsadeného políčka, ležiaceho v priesečníku riadka a stĺpca, v ktorých sa tieto čísla nachádzajú. Vždy sa postupuje od riadku alebo stĺpca s najväčším počtom políčok s nenulovou premennou k riadku alebo stĺpcu s najmenším počtom políčok s nenulovou premennou.

**2. Výpočet rozdielov  $(u_i + v_j) - c_{ik}$  pre všetky neobsadené políčka**

- Hodnota sa zapisuje do ľavého dolného rohu každého neobsadeného políčka

**3. Zistenie kladných rozdielov  $(u_i + v_j) - c_{ik}$**

- Políčko s najväčším kladným rozdielom (začiatočné políčko „kruhu“) označiť do ľavého horného rohu znamienkom +. Najbližšie obsadené políčka (vždy v stĺpci alebo v riadku) sa označia znamienkami – (ak by sa pri začiatočnom políčku nachádzali dve obsadené políčka či už v riadku alebo stĺpci, počká sa s označením podľa potreby – uzavretie „kruhu“) a ďalšie striedavo aby sa uzavrel „okruh“ (ak sú v riadku alebo stĺpci viac ako 2 obsadené políčka, označí sa znamienkom len prvé a posledné políčko).

**4. Zmena premenných podľa znamienka**

- Najmenšiu hodnotu v políčku so znamienkom – treba odpočítať vo všetkých políčkach so znamienkom +. Políčka mimo uzavretého okruhu treba nechať bez zmeny.

**5. Kroky 1b až 4. sa opakujú až po získanie optimálneho riešenia dopravnej úlohy, t.j. do „vyčistenia priestoru“ od kladných hodnôt rozdielu**

$$(u_i + v_j) - c_{ik}$$

**6. Určenie hodnoty účelovej funkcie  $z$**

### 3.8 Habrova metóda úzkych profilov

Opiera sa hlavne o ekonomické poznatky optimalizačných systémov a o logický rozbor. Pretože úvahy odvodené z úzkoprofilových situácií sú bežných javom pri ekonomických rozhodovaniach, je táto metóda blízka mysleniu pracovníkov praxe. Dovoľuje v priebehu celého výpočtového postupu bezprostrednú ekonomickú interpretáciu a tým aj kontrolu riešenia, nakoľko sa dá previesť odhad maximálnej možnej odchýlky od optimálneho riešenia. Využíva sa k tomu duálny princíp lineárneho programovania. Výhodou však je, že sa aj riešenie duálnej úlohy prevádza len približnou metódou, čím je kontrola výsledku podstatne uľahčená.

Postup riešenia úloh LP metódou úzkych profilov je veľmi jednoduchý:

1. zvolí sa poradie premenných podľa výšky koeficientov účelovej funkcie (od najmenšieho po najväčší koeficient),
2. keď sa úloha LP zapíše do tabuľky, tak postupne sa každej premennej  $x_{ij}$  (podľa zvoleného poradia) priradí najlepšia možná hodnota  $a_i$  alebo  $b_k$ , a to je  $\min \{ a_i ; b_k \}$
3. ak sa niektorej premennej  $x_{ij}$  priradila hodnota  $a_i$ , potom už ostatné (ešte nevybavené) premenné v  $i$ -tom riadku budú nulové
4. ak sa niektorej premennej  $x_{ij}$  priradila hodnota  $b_j$ , potom už ostatné (ešte nevybavené) premenné v  $(m+j)$ -tom riadku tabuľky budú nulové
5. po každom jednom takomto kroku sa napíše vedľa tabuľky nový stĺpec a riadok platných hodnôt:

$$a_i - x_{ij} \text{ a } b_j - x_{ij} \text{ pre } i = 1, 2, \dots, m \text{ a } j = 1, 2, \dots, n. \quad (3.10)$$

Porovnaním riešení jednotlivých metód jednoznačne vyplýva, že Vogelova aproximačná metóda je optimálnym riešením danej úlohy.

## 4. CELOČÍSELNÉ LINEÁRNE PROGRAMOVANIE

V praxi sa prichádza často do styku s úlohami, že premenné  $x_1, x_2, \dots, x_n$  môžu byť len celočíselné. Znamená to ďalšie ohraničujúce podmienky, a ak v niektorom kroku simplexovej metódy sa získajú neceločíselné hodnoty premenných  $x_1, x_2, \dots, x_n$ , tak také riešenie sa považuje za neprípustné. Zaokrúhlenie takéhoto riešenia úlohy LP k najbližším celým číslam nemusí byť ešte to pravé optimálne celočíselné riešenie.

Úlohou celočíselného lineárneho programovania (CLS) v štandardnom tvare sa rozumie úloha:

$$\begin{aligned} \mathbf{c}^T \mathbf{x} &\rightarrow \min \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}, \text{ celé,} \end{aligned} \tag{4.1}$$

kde matica  $A$  a vektory  $c, b$  sú celočíselné. Ak sa vynechá zo zadania úlohy (4.1) podmienka celočíselnosti premenných  $x$ , získa sa úloha LP, ktorá je relaxáciou úlohy (4.1) [14].

**Veta:** Ak optimálne riešenie relaxácie úlohy (4.1) je celočíselné, tak je to zároveň aj optimálne riešenie úlohy (4.1).

Predpokladá sa, že sa vyriešila relaxácia úlohy (4.1) simplexovou metódou a získané optimálne bázičné prípustné riešenie  $x_B$  primárnej úlohy nie je celočíselné. V ďalšom, pre zjednodušenie vyjadrovania, sa budú prvky optimálnej simplexovej tabuľky označovať gréckymi písmenami  $\xi$  a písmenká  $x$  budú rezervované pre premenné úlohy. Symbol  $\lfloor \xi \rfloor$  označuje celú časť čísla  $\xi$ , t.j. najväčšie číslo, ktoré nie je väčšie než  $\xi$  a  $\{\xi\} = \xi - \lfloor \xi \rfloor$  je zlomková časť čísla  $\xi$ .

Neceločíselnosť optimálneho riešenia znamená, že niektorá jeho bázičná zložka  $x_{B(i)}$  nie je celočíselná, čo sa v tabuľke prejaví tak, že  $\xi_{i0}$  nie je celé. Príslušná rovnica tabuľky má tvar:

$$x_{B(i)} + \sum_{j \in B} \xi_{ij} x_j = \xi_{i0} \quad (4.2)$$

**Veta:** Nech  $\xi_{i0}$  v optimálnej simplexovej tabuľke nie je celočíselné. Ak sa k nej pridá rovnica

$$-\sum_{j \in B} \{\xi_{ij}\} x_j + s = -\xi_{i0}, s \geq 0 \quad (4.3)$$

tak žiadne celočíselné prípustné riešenie sa nevylúči a nová tabuľka je primárne neprípustná, duálne prípustná a bázická.

Rovnica (4.3) sa nazýva Gomoryho rez.

Úlohy celočíselného programovania sa v praxi vyskytujú často, preto bolo vyvinuté množstvo metód pre riešenie týchto problémov. Bohužiaľ sú ale tieto úlohy zložito riešiteľné v porovnaní s úlohami neceločíselnými, kde rôzne formy simplexového algoritmu spolu s výpočtovou technikou umožňujú riešiť úlohy veľkých rozmerov (tisíce premenných a stovky obmedzení). Pri riešení niektorých celočíselných úloh počet iterácií a spotrebovaný čas na počítači sú značne vysoké i u úloh malých rozmerov, ktoré by boli bez ťažkostí riešené bez podmienok celočíselnosti. Väčšie úlohy celočíselné sú pre veľkú výpočtovú náročnosť niekedy i neriešiteľné prakticky.

#### 4.1 Príklad - Gomoryho algoritmus rezných nadrovín

Niektoré problémy, ktoré sa vyskytujú v ekonomike, môžu byť formulované ako problémy lineárneho programovania s premennými obmedzenými na celočíselné hodnoty.

Medzi početné algoritmy navrhnuté pre riešenie takéhoto problému patrí aj Gomoryho algoritmus rezných nadrovín. Jeho implementácia sa v programe MATLAB<sup>®</sup> vykonáva pomocou funkcie *cpa*, pričom zdrojový kód danej funkcie je uvedený v prílohe [11].

Je zadaný príklad:

$$\max z = 5x_1 + 8x_2$$

$$6x_1 + 7x_2 \leq 10$$

$$8x_1 + 4x_2 \leq 16$$

$$x_{1-2} \geq 0, \text{ celé}$$

```
type='max'  
A=[6 7;8 4];  
c=[5 8];  
b=[10;16];  
cpa(type, c, A, b)
```

Zdrojový kód 4.1

Na základe zdrojového kódu 4.1 získa program MATLAB<sup>®</sup> toto riešenie Gomoryho algoritmu rezu nadrovín:

```
Hodnoty vyslednych premennych:  
x(1)= 0  
x(2)= 1  
Realna hodnota optimalneho riesenia:  
Z = 8.000000
```

## 5. ZÁVER

Cieľom tejto diplomovej práce bolo pokryť klasické partie lineárneho programovania, ktoré sú súčasťou optimalizácie. Je určená študentom ako výučbová pomôcka pri riešení úloh LP, príp. na oboznámenie sa s danou problematikou. Tiež je podkladom pre e-learningový výučbový systém predmetu Optimalizácia.

Keďže e-learning sa stáva bežnou súčasťou procesu výučby, je možné použiť informácie v predloženej diplomovej práci na elektronické vzdelávanie.

Keďže je optimalizačný toolbox programu MATLAB<sup>®</sup> využitý pri výučbe predmetu Optimalizácia, snahou bolo uviesť riešenie mnohých príkladov v predloženej diplomovej práci práve v tomto prostredí. Základné partie, ako sú napr. teória duality a primárna simplexová metóda, sú rozobrané dosť podrobne, s viacerými príkladmi, ktoré majú ukázať spôsob riešenia takýchto úloh lineárneho programovania.

Pri riešení úloh LP je použitý toolbox programu MATLAB<sup>®</sup>, pričom je využitý nielen na matematické výpočty, ale aj na grafické riešenie problému LP. Sú zvolené rôzne typy príkladov, a rôzne spôsoby možných výpočtov. Časť je venovaná aj úlohám o zmesiach, ktoré sa vyskytujú v praxi priemyslu chemického, potravinárskeho, farmaceutického atď.

Značná časť diplomovej práce je venovaná simplexovej metóde, ktorá je základnou metódou pre riešenie úloh LP. Príklady, ktoré sú riešené touto metódou, boli vybraté za účelom toho, aby umožňovali riešenie vlastnej úlohy LP postupom uvedeným v niektorom z príkladov. Snahou je obsiahnuť všetky možné varianty príkladov, ktoré by sa mohli pri riešení úloh LP vyskytnúť.

Tiež je uvedená web stránka [11], kde danú úlohu LP možno riešiť interaktívne pomocou krokov, ktoré sa používajú aj pri ručnom riešení. Ďalšie uvedené príklady sú riešené pomocou optimalizačného toolboxu programu MATLAB<sup>®</sup>, pričom je použitá funkcia *simplex2p*, a pri duálnom simplexovom algoritme je to funkcia *dsimplex*. Teória k duálnemu algoritmu nie je uvedená kvôli obmedzeniu z hľadiska rozsahu tejto témy, ale pre prípad potreby je uvedený jeden vzorový príklad.

Pri dopravnom probléme sú na jeden príklad aplikované rôzne metódy na riešenie dopravného problému. Po aplikácii týchto metód na daný príklad sa zistilo, že Vogelova aproximačná metóda je optimálnym riešením, pretože náklady na prepravu sú

pri tomto spôsobe uspokojenia požiadaviek spotrebiteľov danými kapacitami dodávateľov najnižšie.

V tejto diplomovej práci je spomenutá len základná teória k celočíselnému programovaniu jednak pre obmedzenie z hľadiska rozsahu predloženej diplomovej práce a tiež pre zložitosť riešení týchto úloh oproti úlohám neceločíselným. Pre bližšie oboznámenie sa s danou problematikou je vyriešený príklad prostredníctvom programu MATLAB<sup>®</sup>. Pre ďalšie oboznámenie sa s problematikou celočíselného programovania je potrebné odporučiť ďalšie štúdium tejto rozsiahlej témy.



## 6. LITERATÚRA

- [1] LAGOVÁ, M.; PELIKÁN J.: Metody lineárního programování. Praha: Vysoká škola technická, 1988.
- [2] FRIEBELOVÁ, J. 2007. Lineární programování. [online]. [Cit. 2007-02-15]. Dostupné na: Jihočeská univerzita v Českých Budějovicích, Zemědělská fakulta, Katedra aplikované matematiky a informatiky.  
URL: <http://www2.zf.jcu.cz/~janaklic/emm.html>
- [3] POBIŠOVÁ, M: Vybrané úlohy z lineárneho programovania. Zvolen: MAT-CENTRUM, 1995.
- [4] HUDZOVIČ, P.: Optimalizácia. Bratislava: Vydavateľstvo STU, 2000.
- [5] 2007. Linear Programming. [online]. [Cit. 2007-02-18]. Dostupné na: Luisiana State University, Minerals processing, Recherche institute.  
URL: <http://www.mpri.lsu.edu/textbook/Chapter4-d.htm#problems>
- [6] BORRE, J. 2005. [online]. [Cit. 2007-02-20]. Dostupné na: Aalborg University, Department of communication technology .  
URL: <http://kom.aau.dk/~borre/matlab/strang/>
- [7] LIŠKA, M. 2007. Simplexová metoda. [online]. [Cit. 2007-02-20]. Dostupné na: Ostravská univerzita, Přírodovědecká fakulta, Katedra informatiky a počítačů.  
URL: <http://www1.osu.cz/studium/mopv2/simplex/>
- [8] PEŠEK, P. 2003. Základy lineárního programování. [online]. [Cit. 2007-03-01]. Dostupné na: Západočeská univerzita, Katedra kybernetiky.  
URL: <http://control.zcu.cz/~pesek/OA2000/user/data/OA5.pdf>
- [9] KLINGEROVÁ, P. 1997. Lineární programování. [online]. [Cit. 2007-03-03]. Dostupné na: Technická univerzita v Liberci, Fakulta mechatroniky, Katedra softvérového inženýrství.  
URL: <http://www.fm.vslib.cz/cgi-bin/toASCII/~ksi/cz/mater/oa/linprog/>
- [10] KLINGEROVÁ, P. 1997. Interaktivní řešení úlohy lineárního programování. [online]. [Cit. 2007-03-03]. Dostupné na: Technická univerzita v Liberci, Fakulta mechatroniky, Katedra softvérového inženýrství.  
URL: <http://www.fm.vslib.cz/cgi-bin/toASCII/~ksi/cz/mater/oa/linprog/gauss.htm>
- [11] MAYER, J. 2007. [online]. [Cit. 2007-03-05]. Dostupné na: Západočeská univerzita, Fakulta elektrotechnická, Katedra teoretické elektrotechniky.  
URL: <http://home.zcu.cz/~jmayer/>

- [12] 2006. Dopravná úloha a jej aplikácie. [online]. [Cit. 2007-03-10]. Dostupné na: Slovenská technická univerzita, Stavebná fakulta, Katedra ekonomiky a riadenia stavebníctva. URL: <http://web.svf.stuba.sk/kat/ERS/doc/>
- [13] PLESNÍK, J., DUPAČOVÁ, J., VLACH, M.: Lineárne programovanie. Bratislava: Alfa, 1990.
- [14] POLAČKO, V. 2007. Metóda severozápadného rohu. [online]. [Cit. 2007-03-15]. Dostupné na: Ekonomická univerzita, Fakulta podnikového manažmentu, Katedra manažmentu výroby a logistiky. URL: [http://fpm.euba.sk/KMVaL/logistika/log\\_dopr\\_ul.htm](http://fpm.euba.sk/KMVaL/logistika/log_dopr_ul.htm)
- [13] CECHLÁROVÁ, K., SEMANIŠIN, G.: Lineárna optimalizácia. Košice: Prírodovedecká fakulta UPJŠ, 1999.

## 7. PRÍLOHY

CD obsahujúce pdf a doc súbor predloženej diplomovej práce.

### Príloha 1 – funkcia *linprog*

```
%LINPROG This code uses the revised simplex method to solve the
linear
% programming problem: Minimize the cost c'x subject to
% equations Ax=b and nonnegativity x >=0:
%
% f = Min { c'x; Ax = b, x >= 0 },
%
% You must define an m by n matrix A , a column vector b with
% m components, and a column vector c with n components.
% You may define k to change the first k equations of Ax=b to
% inequalities: thus [A(1:k,:)]x <= b(1:k).
%
% The output vector x gives the minimum cost, which is the
output f.
%
% [x,f,itn] = linprog(A,b,c,k,maxit,tol)
%
% At most "maxit" iterations (default 10*length(b)) are applied
% and the actual number of iterations is returned in "itn".
%
% If the optimal solution is unbounded or the constraints are
% inconsistent then a diagnostic is displayed.
% Bland's rule is used to resolve degeneracies. In exact
% arithmetic cycling is not possible. But in real life!!!
% If x has more than 20 components it is returned in the
% sparse format. Note that if A has many zeros it is worth
% passing it to linprog in sparse format.
%
% Although written for teaching purposes this routine has
% successfully solved some problems with size(A) = [50,100000]!
%
% Please report any difficulties to: idc@math.canterbury.ac.nz
%
% New version (c) I.D.Coope, 1988, 1993
function [x,f,it,B] = linprog(A,b,c,k,maxit,tol)
[m,n]=size(A); b=b(:); c=c(:); it=0;
if (length(c)~=n | length(b)~=m),error('wrong dimensions'); end
if (nargin<6), tol=1e-10; end
if (nargin<5), maxit=10*m; end
if (nargin<4), k=0; elseif isempty(k), k=0; end
D=sign(sign(b)+.5); if k, D(1:k)=ones(k,1); end
D = diag(D); % initial (inverse) basis matrix
A = [A D]; % incorporate slack/artificial
variables
B = n+1:n+m; % initial basis
N = 1:n; % non-basis
[bmin,j]=min(b(1:k));
if bmin<0,
phase=1; xb=ones(m,1); s=[zeros(n+k,1);ones(m-k+1,1)]; %
supercost
N=[N,B(j)]; J=B; J(j)=[]; B(j)=n+m+1;
a=b-sum(A(:,J)')'; A=[A a];
D(:,j)= -a/a(j); D(j,j)=1/a(j);
```

```

elseif k==m,
    phase=2; xb=b; s=[c;zeros(m,1)]; % cost function
else
    % k==[] or bmin>=0
    phase=1; xb=abs(b); s=[zeros(n+k,1);ones(m-k,1)]; % supercost
end
while phase<3,
    df=-1; t=inf;
    yb= D'*s(B); % multipliers for Ax=b
    while (it < maxit)
        if isempty(N), break, end % no freedom for minimization
        r = s(N) - [A(:,N)]'*yb; % reduced costs
        [rmin,q] = min(r); % determine new basic variable
        if rmin>=-tol*(norm(s(N),inf)+1), break, end % optimal!
        it=it+1;
        if df>=0 % apply Bland's rule to avoid
cycling
            if maxit==inf,
                disp(['LINPROG(',int2str(it),'): warning! degenerate
vertex']);
            end
            J=find(r<0); Nq=min(N(J)); q=find(N==Nq);
            end
            d = D*A(:,N(q));
            I=find(d>tol); % I=find(d>0);
            if isempty(I), disp('Solution is unbounded'); it=-it; break; end
            xbd=xb(I)./d(I); [r,p]=min(xbd); p=I(p);
            if df>=0, % apply Bland's rule to avoid
cycling
                J=find(xbd==r); Bp=min(B(I(J))); p=find(B==Bp);
            end
            xb= xb - r*d; xb(p)=r; % update x
            df=r*rmin; % change in f
            v = D(p,:)/d(p); % row vector
            yb= yb + v'*( s(N(q)) - d'*s(B) );
            d(p)=d(p)-1;
            D = D - d*v; % update inverse basis matrix
            t=B(p); B(p)=N(q);
            if t>n+k, N(q)=[]; else N(q)=t; end
        end % end of phase
        xb=xb+D*(b-A(:,B)*xb); % iterative refinement
        I=find(xb<0); % must be due to rounding error
        if I, xb(I)=xb(I)-xb(I); end % so correct
        if phase==2 | it<0, break; end; % B, xb,n,m,res=A(:,B)*xb-b
        if xb'*s(B)>tol,it=-it; disp('no feasible solution'); break; end
        phase=phase+1; % re-initialise for Phase 2
        s=1e6*norm(c,'inf')*s; s(1:n)=c;% tol=tol*norm(s,inf);
    end
    x=sparse(n,1); x(B)=xb; x=x(1:n); if n<21, x=full(x); end
    f=c'*x;
    if it>=maxit, disp('too many iterations'); it=-it; end

```

## Príloha 2 – funkcia lp

```

function [x,lambda,how]=lp(f,A,B,vlb,vub,x,neqcstr,verbosity)
%LP      Linear programming.
%  LP has been replaced with LINPROG.  LP currently works but
%  will be removed in the future.  Use LINPROG instead.
%
%  X=LP(f,A,b) solves the linear programming problem:
%

```

```

%           min f'x      subject to:   Ax <= b
%           x
%
%   X=LP(f,A,b,VLB,VUB) defines a set of lower and upper
%   bounds on the design variables, X, so that the solution is always
in
%   the range VLB <= X <= VUB.
%
%   X=LP(f,A,b,VLB,VUB,X0) sets the initial starting point to X0.
%
%   X=LP(f,A,b,VLB,VUB,X0,N) indicates that the first N constraints
defined
%   by A and b are equality constraints.
%
%   X=LP(f,A,b,VLB,VUB,X0,N,DISPLAY) controls the level of warning
%   messages displayed.  Warning messages can be turned off with
%   DISPLAY = -1.
%
%   [x,LAMBDA]=LP(f,A,b) returns the set of Lagrangian multipliers,
%   LAMBDA, at the solution.
%
%   [X,LAMBDA,HOW] = LP(f,A,b) also returns a string how that
indicates
%   error conditions at the final iteration.
%
%   LP produces warning messages when the solution is either unbounded
%   or infeasible.

%   Copyright 1990-2002 The MathWorks, Inc.
%   $Revision: 1.18 $   $Date: 2002/03/12 20:36:12 $
%   Andy Grace 7-9-90. Mary Ann Branch 9-30-96.

%   This function is old, the current version should be used.
WarningStr=sprintf(...
['The function LP is obsolete and has been replaced by\n',...
'LINPROG.  LP will be removed in a future release of the\n',...
'Optimization Toolbox.  Update your code to call LINPROG to\n',...
'suppress this warning message.  See "Converting your code to\n',...
'Version 2 Syntax" in the Optimization Toolbox User's Guide\n',...
'for more information.\n']);

warning(WarningStr)

if nargin<8, verbosity = 0;
    if nargin<7, neqcstr = 0;
        if nargin<6, x = [];
            if nargin<5, vub = [];
                if nargin<4, vlb = [];
                    end, end, end, end, end
[ncstr,nvars]=size(A);
nvars = max([length(f),nvars]); % In case A is empty

if isempty(verbosity), verbosity = 0; end
if isempty(neqcstr), neqcstr = 0; end
if isempty(x), x=zeros(nvars,1); end

if isempty(A), A=zeros(0,nvars); end
if isempty(B), B=zeros(0,1); end

% Set to column vectors
f = f(:);

```

```

B = B(:);
x = x(:);

caller = 'lp';
[x,lambda,how]=qpsubold([],f,A,B,vlb,vub,x,neqcstr,verbosity,caller,nc
str,nvars);

```

### Príloha 3 – funkcia *simplex2p*

```

function simplex2p(type, c, A, rel, b)

% The Two-Phase Method for solving the LP problem

%               min(or max) z = c*x
%       Subject to   Ax rel b
%                   x >= 0

% The input parameter type holds information about type of the LP
% problem to be solved. For the minimization problem type = 'min'
% and for the maximization problem type = 'max'.
% The input parameter rel is a string holding the relation signs.
% For instance, if rel = '<=>', then the constraint system consists
% of one inequality <=, one equation, and one inequality >=.

if (type == 'min')
    mm = 0;
else
    mm = 1;
    c = -c;
end
b = b(:);
c = c(:)';
[m, n] = size(A);
n1 = n;
les = 0;
neq = 0;
red = 0;
if length(c) < n
    c = [c zeros(1,n-length(c))];
end
for i=1:m
    if(rel(i) == '<')
        A = [A vr(m,i)];
        les = les + 1;
    elseif(rel(i) == '>')
        A = [A -vr(m,i)];
    else
        neq = neq + 1;
    end
end
ncol = length(A);
if les == m
    c = [c zeros(1,ncol-length(c))];
    A = [A;c];
    A = [A [b;0]];
    [subs, A, z, p1] = loop(A, n1+1:ncol, mm, 1, 1);
    disp('                               End of Phase 1')
    disp(' *****')
else
    A = [A eye(m) b];

```

```

if m > 1
    w = -sum(A(1:m,1:ncol));
else
    w = -A(1,1:ncol);
end
c = [c zeros(1,length(A)-length(c))];
A = [A;c];
A = [A;[w zeros(1,m) -sum(b)]];
subs = ncol+1:ncol+m;
av = subs;
[subs, A, z, pl] = loop(A, subs, mm, 2, 1);
if pl == 'y'
    disp('
                                End of Phase 1')
    disp('
                                *****')
    end
    nc = ncol + m + 1;
    x = zeros(nc-1,1);
    x(subs) = A(1:m,nc);
    xa = x(av);
    com = intersect(subs,av);
    if (any(xa) ~= 0)
        disp(sprintf('\n\n
                                Empty feasible region\n'))
        return
    else
        if ~isempty(com)
            red = 1;
        end
    end
    end
    A = A(1:m+1,1:nc);
    A = [A(1:m+1,1:ncol) A(1:m+1,nc)];
    [subs, A, z, pl] = loop(A, subs, mm, 1, 2);
    if pl == 'y'
        disp('
                                End of Phase 2')
        disp('
                                *****')
    end
end
if (z == inf | z == -inf)
    return
end
[m, n] = size(A);
x = zeros(n,1);
x(subs) = A(1:m-1,n);
x = x(1:n1);
if mm == 0
    z = -A(m,n);
else
    z = A(m,n);
end
disp(sprintf('\n\n
                                Problem has a finite optimal
                                solution\n'))
disp(sprintf('\n Values of the legitimate variables:\n'))
for i=1:n1
    disp(sprintf(' x(%d)= %f ',i,x(i)))
end
disp(sprintf('\n Objective value at the optimal point:\n'))
disp(sprintf(' z= %f',z))
t = find(A(m,1:n-1) == 0);
if length(t) > m-1
    str = 'Problem has infinitely many solutions';
    msgbox(str,'Warning Window','warn')
end
end

```

```

if red == 1
    disp(sprintf('\n Constraint system is redundant\n\n'))
end

function [subs, A, z, pl]= loop(A, subs, mm, k, ph)

% Main loop of the simplex primal algorithm.
% Bland's rule to prevent cycling is used.

tbn = 0;
str1 = 'Would you like to monitor the progress of Phase 1?';
str2 = 'Would you like to monitor the progress of Phase 2?';
if ph == 1
    str = str1;
else
    str = str2;
end
question_ans = questdlg(str,'Make a choice Window','Yes','No','No');
if strcmp(question_ans,'Yes')
    pl = 'y';
end
if pl == 'y' & ph == 1
    disp(sprintf('\n\n                                Initial tableau'))
    A
    disp(sprintf(' Press any key to continue ...\n\n'))
    pause
end
if pl == 'y' & ph == 2
    tbn = 1;
    disp(sprintf('\n\n                                Tableau %g',tbn))
    A
    disp(sprintf(' Press any key to continue ...\n\n'))
    pause
end
[m, n] = size(A);
[mi, col] = Br(A(m,1:n-1));
while ~isempty(mi) & mi < 0 & abs(mi) > eps
    t = A(1:m-k,col);
    if all(t <= 0)
        if mm == 0
            z = -inf;
        else
            z = inf;
        end
        disp(sprintf('\n                                Unbounded optimal solution with z=
%s\n',z))
        return
    end
    [row, small] = MRT(A(1:m-k,n),A(1:m-k,col));
    if ~isempty(row)
        if abs(small) <= 100*eps & k == 1
            [s,col] = Br(A(m,1:n-1));
        end
        if pl == 'y'
            disp(sprintf('                                pivot row-> %g    pivot column->
%g',...
                                row,col))
        end
        A(row,:)= A(row,+)/A(row,col);
        subs(row) = col;
        for i = 1:m

```



```

        if i ~= row
            A(i,:) = A(i,:) - A(i,col)*A(row,:);
        end
    end
    [mi, col] = Br(A(m,1:n-1));
end
tbn = tbn + 1;
if p1 == 'y'
    disp(sprintf('\n\n                        Tableau %g',tbn))
    A
    disp(sprintf(' Press any key to continue ...\n\n'))
    pause
end
end
z = A(m,n);

```

#### Príloha 4 – funkcia *dsimplex*

```

function varargout = dsimplex(type, c, A, b)

% The Dual Simplex Algorithm for solving the LP problem

%               min (max) z = c*x
%       Subject to  Ax >= b
%                   x >= 0
%

if type == 'min'
    mm = 0;
else
    mm = 1;
    c = -c;
end
str = 'Would you like to monitor the progress of computations?';
A = -A;
b = -b(:);
c = c(:)';
[m, n] = size(A);
A = [A eye(m) b];
A = [A; c zeros(1,m+1)];
question_ans = questdlg(str, 'Make a choice Window', 'Yes', 'No', 'No');
if strcmp(question_ans, 'Yes')
    p1 = 'y';
else
    p1 = 'n';
end
if p1 == 'y'
    disp(sprintf('\n\n                        Initial tableau'))
    A
    disp(sprintf(' Press any key to continue ...\n\n'))
    pause
end
subs = n+1:n+m;
[bmin, row] = Br(b);
tbn = 0;
while ~isempty(bmin) & bmin < 0 & abs(bmin) > eps
    if A(row,1:m+n) >= 0
        disp(sprintf('\n\n                        Empty feasible region\n\n'))
        varargout(1) = {subs(:)};
    end
end

```

```

        varargout(2)={A};
        varargout(3) = {zeros(n,1)};
        varargout(4) = {0};
        return
    end
    col = MRTD(A(m+1,1:m+n),A(row,1:m+n));
    if p1 == 'y'
        disp(sprintf('                pivot row-> %g    pivot column-> %g',...
            row,col))
    end
    subs(row) = col;
    A(row,:)= A(row,:)/A(row,col);
    for i = 1:m+1
        if i ~= row
            A(i,:)= A(i,:)-A(i,col)*A(row,:);
        end
    end
    end
    tbn = tbn + 1;
    if p1 == 'y'
        disp(sprintf('\n\n                Tableau %g',tbn))
        A
        disp(sprintf(' Press any key to continue ...\n\n'))
        pause
    end
    [bmin, row] = Br(A(1:m,m+n+1));
end
x = zeros(m+n,1);
x(subs) = A(1:m,m+n+1);
x = x(1:n);
if mm == 0
    z = -A(m+1,m+n+1);
else
    z = A(m+1,m+n+1);
end
disp(sprintf('\n\n                Problem has a finite optimal
solution\n\n'))
disp(sprintf('\n Values of the legitimate variables:\n'))
for i=1:n
    disp(sprintf(' x(%d)= %f ',i,x(i)))
end
disp(sprintf('\n Objective value at the optimal point:\n'))
disp(sprintf(' z= %f',z))
disp(sprintf('\n Indices of basic variables in the final tableau:'))
varargout(1)={subs(:)};
varargout(2)={A};
varargout(3) = {x};
varargout(4) = {z};

```

## Príloha 5 – funkcia cpa

```

function cpa(type, c, A, b)

% Gomory's cutting plane algorithm for solving
% the integer programming problem

%
%               min(max) z = c*x
%       Subject to: Ax <= b
%
%                   x >= 0 and integral

```

```

str = 'Would you like to monitor the progress of computations?';
question_ans = questdlg(str,'Make a choice Window','Yes','No','No');
if strcmp(question_ans,'Yes')
    p1 = 'y';
else
    p1 = 'n';
end
if type == 'min'
    tp = -1;
else
    tp = 1;
end
[m,n] = size(A);
nlv = n;
b = b(:);
c = c(:)';
if p1 == 'y'
    [A,subs] = simplex(type,c,A,b,p1);
else
    [A,subs] = simplex(type,c,A,b);
end
[m,n] = size(A);
d = A(1:m-1,end);
pc = fractp(d);
tbn = 0;
if p1 == 'y'
    disp(sprintf('
_____'))
    disp(sprintf('\n                Tableaux of the Dual Simplex
Method'))
    disp(sprintf('
_____'))
end
while norm(pc,'inf') > eps
    [el,i] = max(pc);
    nr = A(i,1:n-1);
    nr = [-fractp(nr) 1 -el];
    B = [A(1:m-1,1:n-1) zeros(m-1,1) A(1:m-1,end)];
    B = [B;nr;[A(m,1:n-1) 0 A(end,end)]];
    A = B;
    [m,n] = size(A);
    [bmin, row] = Br(A(1:m-1,end));
    while ~isempty(bmin) & bmin < 0 & abs(bmin) > eps
        col = MRTD(A(m,1:n-1),A(row,1:n-1));
        if p1 == 'y'
            disp(sprintf('\n                pivot row-> %g    pivot column->
%g',...
                row,col))
            tbn = tbn + 1;
            disp(sprintf('\n                Tableau %g',tbn))
            A
            disp(sprintf(' Press any key to continue ... \n'))
            pause
        end
        if isempty(col)
            disp(sprintf('\n Algorithm fails to find an optimal
solution. '))
            return
        end
        subs(row) = col;
    end
end

```

```

        A(row,:) = A(row,:)./A(row,col);
        for i = 1:m
            if i ~= row
                A(i,:) = A(i,:)-A(i,col)*A(row,:);
            end
        end
        [bmin, row] = Br(A(1:m-1,end));
    end
    d = A(1:m-1,end);
    pc = fractp(d);
end
if p1 == 'y'
    disp(sprintf('\n                        Final tableau'))
    A
    disp(sprintf(' Press any key to continue ...\n'))
    pause
end
x = zeros(n-1,1);
x(subs) = A(1:m-1,end);
x = x(1:nlv);
disp(sprintf('\n                        Problem has a finite optimal
solution\n\n'))
disp(sprintf('\n Values of the legitimate variables:\n'))
for i=1:nlv
    disp(sprintf(' x(%d)= %g ',i,x(i)))
end
disp(sprintf('\n Objective value at the optimal point:\n'))
disp(sprintf(' z= %f',tp*A(m,n)))

function y = fractp(x)

% Fractional part y of x, where x is a matrix.

y = zeros(1,length(x));
ind = find(abs(x - round(x)) >= 100*eps);
y(ind) = x(ind) - floor(x(ind));

```