UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

RIEŠENIE PROBLÉMOV CELOČÍSELNÉHO PROGRAMOVANIA VYUŽITÍM TECHNÍK LINEÁRNEHO PROGRAMOVANIA

Bratislava 2011 **Simona Chattová**

UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

RIEŠENIE PROBLÉMOV CELOČÍSELNÉHO PROGRAMOVANIA VYUŽITÍM TECHNÍK LINEÁRNEHO PROGRAMOVANIA

Bakalárska práca

Študijný program: **Ekonomická a finančná matematika**

Študijný odbor: **9.1.9 Aplikovaná matematika**

Školiace pracovisko: **Katedra aplikovanej matematiky a štatistiky**

Vedúci bakalárskej práce: **Mgr. Zuzana Macová**

Evidenčné číslo: **3ffa73a2-3412-45df-8742-3553c5317ca8**

Bratislava 2011 **Simona Chattová**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Simona Chattová
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský

Názov : Riešenie problémov celočíselného programovania využitím techník lineárneho programovania

Cieľ : Zhrnúť základné metódy použité pri riešení úloh celočíselného programovania so zameraním na metódy odvodené z lineárneho programovania (Gomoryho algoritmus), praktická prezentácia zvoleného algoritmu jeho naprogramovaním.

Literatúra : A. Schrijver, Theory of Linear and Integer Programming R. E. Gomory and W. J. Baumol, Integer Programming and Pricing.

Vedúci : Mgr. Zuzana Macová

Dátum zadania: 05.11.2010

Dátum schválenia: 08.11.2010

doc. RNDr. Margaréta Halická, CSc.
garant študijného programu

Chattová
študent

vedúci práce

Dátum potvrdenia finálnej verzie práce, súhlas s jej odovzdaním (vrátane spôsobu sprístupnenia)

31.05.2011
vedúci práce

Čestné prehlásenie

Čestne prehlasujem, že som túto bakalársku prácu vypracovala samostatne s využitím nadobudnutých teoretických poznatkov a s použitím uvedenej literatúry.

Bratislava, máj 2011 ....

Simona Chattová

Poďakovanie

Týmto sa chcem poďakovať Mgr. Zuzane Macovej za odborné vedenie, cenné rady a pripomienky pri vytváraní tejto bakalárskej práce.

Abstrakt

Chattová, Simona : **Riešenie problémov celočíselného programovania využitím techník lineárneho programovania**. [Bakalárska práca] - Univerzita Komenského v Bratislave. Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej matematiky a štatistiky. - Vedúci bakalárskej práce: Mgr. Zuzana Macová, - Bratislava: FMFI UK, 2011. /39 s./

Práca sa zaoberá analyzovaním základných metód na riešenie úloh celočíselného programovania, ktoré využívajú simplexový algoritmus lineárneho programovania. V prvej časti popisujeme Metódu vetvenia a hraníc a Gomoryho algoritmus. Tieto metódy sú založené na riešení postupnosti relaxovaných úloh k úlohe celočíselného programovania. Analyzujeme taktiež ekonomickú interpretáciu duálnych premenných v prípade úlohy LP a na konkrétnom príklade odôvodníme nedostatky tejto interpretácie pre úlohu IP. Druhá časť práce je zameraná na stručný popis najdôležitejších častí nami naprogramovaného Gomoryho algoritmu, ktorý sme použili na riešenie konkrétnych príkladov uvedených v práci. Okrem toho sa nakoniec okrajovo zaoberáme úlohami, ktorých riešenie sa dá získať využitím metód celočíselného programovania.

Kľúčové slová : celočíselné programovanie (IP) • IP úloha • Gomoryho algoritmus • metóda vetvenia a hraníc • sečná nadrovina • účelová funkcia • optimálne riešenie • relaxovaná úloha • duálna úloha • simplexová metóda

Abstract

Chattová, Simona : **Solving Integer Programming problems Using Linear Programming Techniques**. [Bachelor thesis] - Comenius University Bratislava. Faculty of mathematics, physics and informatics; Department of applied mathematics and statistics. - Supervisor: Mgr. Zuzana Macová, - Bratislava: FMFI UK, 2011. /39 pp./

This bachelor thesis deals with the most common integer programming algorithms that are based on the simplex method used for solving problems in linear programming. The first part describes the Branch and bound method and Gomory's cutting plane algorithm. These methods work by solving a sequence of linear programming relaxations of the integer programming problem. We also analyze the economic interpretation of dual variables in LP and explain why such an interpretation is not appropriate as far as an IP problem is concerned. The second part focuses on a brief description of the most important parts of the Gomory's cutting plane algorithm which we programmed in Matlab and that is used to solve the problems mentioned in this work. Finally we briefly deal with problems whose solution can be obtained using integer programming methods.

Keywords : integer programming (IP) • IP problem • Gomory's cutting plane algorithm • Branch and bound method • cutting plane • objective function • optimal solution • linear programming relaxation • dual problem • simplex method

Obsah

Úvod

Optimalizácia ako proces výberu z realizovateľných postupov, ktorý vedie k dosiahnutiu čo najlepšieho výsledku, patrí k najčastejšie sa vyskytujúcim problémom v praxi, či už ide o vytvorenie optimálneho výrobného programu, pridelenie úloh pre jednotlivé pracoviská, maximalizáciu zisku alebo o optimalizáciu procesov v telekomunikačných sieťach. Všetky tieto problémy sa dajú modelovať ako úlohy matematického programovania. Pojem programovanie sa v tomto prípade používa preto, lebo potrebujeme nájsť program resp. plán na optimalizovanie účelovej funkcie, ktorá je merateľným vyjadrením cieľa, ktorý chceme optimalizovať.

V tejto bakalárskej práci sa budeme zaoberať špeciálnym prípadom matematického programovania a to **celočíselným programovaním** (ďalej iba IP). Ide o **lineárne programovanie** s **celočíselnými rozhodovacími premennými**. Oboznámime sa s princípmi základných metód na riešenie úloh IP a pokúsime sa vysvetliť ich podstatu a význam v aplikáciách. Práca je koncipovaná tak, že sa predpokladá znalosť základných pojmov lineárneho programovania, predovšetkým simplexovej metódy.

V prvej kapitole vymedzíme predmet teórie celočíselného programovania a zdôvodníme, prečo nie je vhodné aproximovať celočíselné riešenie zaokrúhlením riešenia relaxovanej úlohy k danej úlohe IP. Požiadavka celočíselnosti komplikuje úlohu viac, než by sa na prvý pohľad mohlo zdať. Optimálne riešenie môže byť totiž vnútorným bodom množiny prípustných riešení.

Druhá kapitola obsahuje definované ciele práce. Je zameraná na popis a odvodenie dvoch základných metód na riešenie úloh celočíselného programovania, a to **Gomoryho algoritmu** a **Metódy vetvenia a hraníc**. Princíp týchto metód ilustrujeme na konkrétnych príkladoch. V tejto časti sa zaoberáme aj ekonomickou interpretáciou duálnych premenných v úlohe lineárneho programovanie a uvádzame jej nedostatky v prípade úlohy celočíselného programovania.

Kapitola **Programová realizácia Gomoryho algoritmu** zahŕňa stručnú charakteristiku najdôležitejších častí nami vytvoreného programu na riešenie úloh IP.

Posledná časť práce okrajovo popisuje úlohy, ktorých riešenie sa dá získať využitím metód celočíselného programovania a špeciálne algoritmy, ktoré sa na riešenie týchto úloh používajú. Je zameraná na *dopravnú úlohu* a *úlohu o batohu*.

1 Úvod do celočíselného programovania

1.1 Formulácia úlohy IP

Pod ***úlohou celočíselného programovania (integer programming problem)***, ďalej len **IP úlohou**, rozumieme úlohu optimalizovať t.j. maximalizovať alebo minimalizovať lineárnu účelovú funkciu pri lineárnych ohraničeniach, pričom na rozdiel od úlohy lineárneho programovania ešte predpokladáme, že všetky dané premenné sú celočíselné. Formulácia úlohy IP má nasledovný tvar:

$$\sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{c_j x_j} \to \mathbf{max}$$
$$\sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{a_{ij} x_j} \leq \mathbf{b_i},$$
$$\mathbf{x_j} \in \mathbb{Z} \quad \text{pre všetky} \quad j = 1, \ldots, n.$$

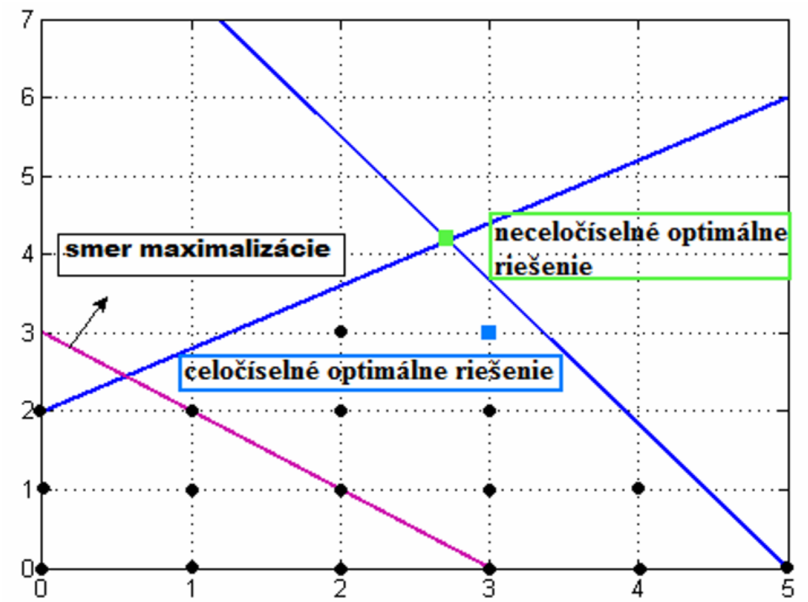
V tomto prípade ide o ***čisto celočíselnú*** úlohu, no existujú aj tzv. zmiešané úlohy celočíselného programovania, v ktorých iba časť premenných je obmedzená podmienkou celočíselnosti, no takýmito úlohami sa v tejto práci nebudeme zaoberať.

V niektorých optimalizačných úlohách majú premenné zmysel iba v prípade, keď nadobúdajú celočíselné hodnoty (napr. ak modelujeme počet vlakov, ktoré majú odísť zo stanice v danom časovom intervale, tak premenná reprezentujúca toto číslo musí byť celočíselná, v opačnom prípade je takýto model v praxi nepoužiteľný). Ide teda o **diskrétne optimalizačné problémy**.

1.2 Aproximácia riešenia úlohy IP

Jednou z metód, ako vyriešiť úlohu celočíselného programovania, je zanedbať podmienku celočíselnosti a riešiť úlohu lineárneho programovania (v takom prípade hovoríme, že riešime relaxovanú úlohu) a vypočítané optimálne riešenie ***zaokrúhliť*** na celočíselné. Avšak takto získané riešenie nemusí byť vhodnou aproximáciou. Takýto prístup totiž môže viesť k veľkým chybám v porovnaní s optimálnou hodnotou účelovej funkcie v celočíselnom optimálnom riešení, prípadne dáva neprípustné riešenie. Nie je ani úplne jasné, či máme dosiahnuté riešenie zaokrúhľovať nahor alebo nadol.

Problém je v tom, že ***celočíselné optimálne riešenie*** môže byť **vnútorným bodom** množiny prípustných riešení, ktorú tvoria **izolované celočíselné body** (tzv. ***lattice points***) – obr.1.



Obr. 1: **IP úloha**

Preto je výhodnejšie použiť exaktné metódy IP a tak vypočítať presné riešenie.

Napriek tomu sa v praxi zaokrúhľovanie využíva v úlohách, v ktorých vieme, že chyba, ktorej sa tým dopustíme, je prijateľná. To znamená, že v niektorých situáciách je úplne akceptovateľné aj suboptimálne riešenie. Ide predovšetkým o úlohy obsahujúce veľa premenných, kde problémom je ***časová*** prípadne ***pamäťová náročnosť***.

Pod pojmom časová náročnosť chápeme reálny čas, ktorý je potrebný na vyriešenie danej úlohy konkrétnym algoritmom. Čo sa týka pamäťovej náročnosti, ide o operačnú pamäť, ktorú zaberajú údaje vystupujúce v algoritme.

2 Algoritmy celočíselného programovania

Medzi najrozšírenejšie metódy riešenia úloh celočíselného programovania patrí **Gomoryho algoritmus (Gomory's cutting plane algorithm)** [1] a **Metóda vetvenia a hraníc (Branch and bound method)** [7].

Tieto metódy síce zaručujú nájdenie optimálneho riešenia **IP problému**, ale v ***nepolynomiálnom čase*** v závislosti od veľkosti úlohy (t.j. od počtu premenných n). To znamená, že počet aritmetických operácií potrebných na vyriešenie úlohy rozmeru n sa nedá zhora odhadnúť žiadnym polynómom v premennej n.

Aj samotný simplexový algoritmus, ktorý je sučasťou týchto metód, nie je polynomiálne zložitý a dokonca môže vyžadovať až exponenciálne veľa iterácií.

2.1 Metóda vetvenia a hraníc (Branch and bound method)

Princíp tejto iteračnej metódy spočíva v tom, že sa postupne ***menia ohraničenia daných premenných*** a vytvárajú sa nové úlohy LP. Jednotlivé úlohy riešime samostatne a dosiahnuté riešenia posudzujeme vzhľadom na podmienku ich celočíselnosti.

Vo všeobecnosti je možné zhrnúť ideovú schému algoritmu do nasledujúcich krokov:

- **vetvenie** (***branching***) - ide o rozklad množiny prípustných riešení na navzájom disjunktné podmnožiny; k danej relaxovanej úlohe, ktorej premenná $\mathbf{x_j}$ optimálneho riešenia má neceločíselnú hodnotu $\overline{\mathbf{b}}_\mathbf{j}$, skonštruujeme **2 čiastkové úlohy**, pričom k jednej pridáme obmedzenie

  $$\mathbf{x_j} \leq \lfloor \overline{\mathbf{b}}_\mathbf{j} \rfloor,$$

  k druhej obmedzenie

  $$\mathbf{x_j} \geq \lfloor \overline{\mathbf{b}}_\mathbf{j} \rfloor + \mathbf{1},$$

  avšak ostatné ohraničenia a účelová funkcia ostávajú v obidvoch úlohách nezmenené.

  Podmienka pre ukončenie vetvenia v danom uzle je splnená v týchto prípadoch:

  - relaxovaná LP úloha nemá prípustné riešenie;
  - optimálna hodnota kriteriálnej funkcie relaxovanej úlohy nie je lepšia ako hodnota účelovej funkcie v doteraz najlepšom nájdenom prípustnom riešení;
  - optimálne riešenie relaxovanej LP úlohy je celočíselné.

- **ohraničovanie** (***bounding***) - pre každú z podmnožín sa určí dolná (v prípade minimalizačnej úlohy) resp. horná (v prípade maximalizačnej úlohy) hranica hodnôt účelovej funkcie na danej podmnožine, cieľom je teda získať odhad, ktorý poskytuje informáciu o tom, nakoľko dobré (v zmysle hodnoty účelovej funkcie) môže byť najlepšie prípustné rišenie, preto aj v prípade tejto metódy budeme riešiť relaxovanú úlohu k danej IP úlohe.

Ukazuje sa, že táto metóda je veľmi efektívna pri riešení niektorých typov úloh IP, ide predovšetkým o praktické úlohy s menším počtom premenných, keďže počet riešených parciálnych úloh rastie exponenciálne.

V takýchto príkladoch je možné určiť dolnú a hornú hranicu pre jednotlivé premenné a teda optimálne riešenie leží niekde medzi týmito hranicami. Čím užší je inicializačný interval, tým menší priestor prehľadávania musí metóda vetvenia a hraníc spracovať.

Príklad

Letecká spoločnosť zabezpečuje spojenie z Londýna do dvoch cieľových miest-Madrid(A) a Bratislava(B). K dispozícii je jedno lietadlo a cieľom je rozhodnúť, koľko letov denne do každého z cieľov prevádzkovať, aby daná spoločnosť dosiahla maximálny zisk. Máme k dispozícií nasledujúce údaje :

- každý spiatočný let do A trvá 45 minút,
- každý spiatočný let do B trvá 100 minút,
- priemerný zisk z každého spiatočného letu do A je 100,
- priemerný zisk z každého spiatočného letu do B je 250,
- požiadavky cestujúcich na lety do B neprevyšujú 7 letov denne,
- požiadavky cestujúcich na lety do A neprevyšujú 13 letov denne,
- lietadlo môže lietať najviac 16 hodín denne,

Najprv zostavíme matematický model daného problému, v našom prípade teda sformulujeme nasledujúcu úlohu celočíselného programovania:

$$\begin{aligned} Z = 100x + 250y &\to \mathbf{max} \\ 9x + 20y &\leq 192, \\ 0 \leq x &\leq 13, \\ 0 \leq y &\leq 7, \\ x, y &\in \mathbb{Z}. \end{aligned}$$

Rozhodovacie premenné x, y predstavujú počet letov prevádzkovaných denne do miest A a B. Cieľ, ktorý chceme maximalizovať, je zisk danej spoločnosti. Ohraničenia v tomto prípade reprezentujú dodržanie určitých pravidiel a teda zavedenie limitov pre rozhodovacie premenné.

Ako sme už spomenuli, nie vždy výsledné optimálne riešenie vyhovuje charakteru riešeného problému, čo platí aj v prípade tejto úlohy, ktorej optimálne riešenie získané simplexovým algoritmom je $(\mathbf{x}, \mathbf{y})^{\mathbf{T}} = (\mathbf{5.78}, \mathbf{7})^{\mathbf{T}}$, pričom $\mathbf{Z} = \mathbf{2328}$.

Keďže premenná $\mathbf{x}$ nadobúda ***neceločíselnú hodnotu***, rozvetvíme danú úlohu podľa tejto premennej na 2 ďalšie úlohy:

<u>`Úloha 1`</u>

$$\begin{aligned} \mathbf{max} \quad & 100x + 250y \\ & 9x + 20y \leq 192, \\ & 0 \leq x \leq 5, \\ & 0 \leq y \leq 7. \end{aligned}$$

<u>`Úloha 2`</u>

$$\begin{aligned} \mathbf{max} \quad & 100x + 250y \\ & 9x + 20y \leq 192, \\ & x \geq 6, \\ & 0 \leq y \leq 7. \end{aligned}$$

Podľa schémy algoritmu riešenie jednej úlohy nahradíme riešením týchto 2 podúloh, ktoré taktiež vypočítame použitím simplexovej metódy.

Riešenie Úlohy 1 je $(\mathbf{x}, \mathbf{y})^{\mathbf{T}} = (\mathbf{5}, \mathbf{7})^{\mathbf{T}}$, s optimálnou hodnotou účelovej funkcie **2250**, no riešenie Úlohy 2 - $(\mathbf{x}, \mathbf{y})^{\mathbf{T}} = (\mathbf{6}, \mathbf{6.9})^{\mathbf{T}}$ je neceločíselné a $\mathbf{Z} = \mathbf{2325}$. Premenná $\mathbf{y}$ má ***neceločíselnú hodnotu***, a preto úlohu 2 rozvetvíme podľa tejto premennej na dve ďalšie úlohy.

Úloha 3

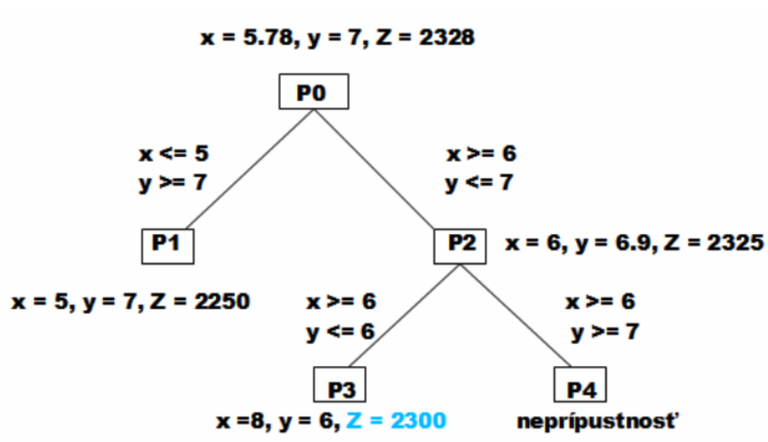
$$\begin{aligned} \textbf{max} \quad & 100x + 250y \\ & 9x + 20y \leq 192, \\ & x \geq 6, \\ & 0 \leq y \leq 6. \end{aligned}$$

Úloha 4

$$\begin{aligned} \textbf{max} \quad & 100x + 250y \\ & 9x + 20y \leq 192, \\ & x \geq 6, \\ & y \geq 7. \end{aligned}$$

Situácia po druhom vetvení je nasledovná - Úloha 4 nemá prípustné riešenie, Úloha 3 má celočíselné optimálne riešenie $(\mathbf{x}, \mathbf{y})^{\mathbf{T}} = (\mathbf{8}, \mathbf{6})^{\mathbf{T}}$ a hodnota účelovej funkcie v tomto bode je **2300**.

Takýmto spôsobom dostaneme rozhodovací strom s vrcholmi, ktoré zodpovedajú jednotlivým parciálnym úlohám:



Obr. 2: **Metóda vetvenia a hraníc**

2.2 Gomoryho algoritmus (Gomory's cutting plane algorithm)

Túto metódu označovanú aj ako ***metóda sečných nadrovín*** navrhol americký matematik ***R.E. Gomory*** [2]. Jej základnou myšlienkou je redukovať množinu prípustných riešení LP relaxovanej úlohy k danej úlohe IP zavedením tzv. **platných nerovností** (***valid inequalities***). Tieto nerovnosti predstavujú ďalšie ohraničenia relaxovanej úlohy, pričom každé celočíselné prípustné riešenie úlohy IP spĺňa dané nerovnosti, ktoré však neplatia pre aktuálne optimálne riešenie relaxovanej úlohy. Algoritmus teda spočíva v riešení postupnosti LP úloh vytvorených špeciálnym spôsobom.

Predpokladajme, že sme vyriešili úlohu IP bez ohľadu na podmienku celočíselnosti a optimálne riešenie LP úlohy je dané v nasledujúcej simplexovej tabuľke

	x_1	$\dots$	x_i	$\dots$	x_m	x_{m+1}	$\dots$	x_{m+n}	$\bar{b}$
z	0	$\dots$	0	$\dots$	0	$\bar{c}_1$	$\dots$	$\bar{c}_n$	$\bar{z}$
x_1	1	$\dots$	0	$\dots$	0	$\bar{a}_{11}$	$\dots$	$\bar{a}_{1n}$	$\bar{b}_1$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$
x_i	0	$\dots$	1	$\dots$	0	$\bar{a}_{i1}$	$\dots$	$\bar{a}_{in}$	$\bar{b}_i$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$
x_m	0	$\dots$	0	$\dots$	1	$\bar{a}_{m1}$	$\dots$	$\bar{a}_{mn}$	$\bar{b}_m$

kde x_i, $i = 1, 2, \dots, m$ sú bázické premenné a x_j, $j = m+1, \dots, m+n$ sú nebázické premenné. Ak získané optimálne riešenie je celočíselné, je to zároveň optimálne riešenie úlohy IP a výpočet končí. V opačnom prípade analyzujeme rovnicu i, pre ktorú platí, že bázická premenná x_i nadobúda neceločíselnú hodnotu. Túto rovnicu môžeme zapísať v tvare

$$x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{b}_i \tag{1}$$

pričom N je množina nebázických premenných.

Označme

$$\bar{b}_i = \lfloor \bar{b}_i \rfloor + f_i, \qquad \bar{a}_{ij} = \lfloor \bar{a}_{ij} \rfloor + f_{ij},$$

kde $0 < f_i < 1$ a $0 \leq f_{ij} < 1$.

Rovnica (1) sa dá prepísať na tvar

$$x_i + \sum_{j\in N} \lfloor \overline{a}_{ij} \rfloor x_j + \sum_{j\in N} f_{ij} x_j = \lfloor \overline{b}_i \rfloor + f_i,$$

čo je ekvivalentné nasledujúcej rovnici

$$x_i + \sum_{j\in N} \lfloor \overline{a}_{ij} \rfloor x_j - \lfloor \overline{b}_i \rfloor = f_i - \sum_{j\in N} f_{ij} x_j. \tag{2}$$

Chceme, aby premenná x_i nadobudla celočíselnú hodnotu, preto ľavá a následne aj pravá strana uvedenej rovnice musia byť celočíselné. Keďže $f_{ij} \geq 0$ a $x_j \geq 0$, tak

$$\left\lfloor f_i - \sum_{j\in N} f_{ij} x_j \right\rfloor \leq f_i < 1.$$

Aby bola splnená podmienka celočíselnosti pre hodnotu výrazu $f_i - \sum_{j\in N} f_{ij} x_j$, musí platiť

$$f_i - \sum_{j\in N} f_{ij} x_j \leq 0. \tag{3}$$

Nerovnica (3) predstavuje **sečnú nadrovinu** (***cutting plane***) – nové obmedzenie, ktoré sa označuje ako ***Gomoryho zlomkový rez*** (***Gomory's fractional cut***). Samozrejme môžeme použiť aj ľavú stranu rovnice (2) a vytvoriť tak tzv. ***Gomoryho celočíselný rez*** (***Gomory's integer cut***). Rez generovaný takýmto spôsobom má dve podstatné vlastnosti:

- ľubovoľné prípustné riešenie úlohy IP spĺňa danú nerovnosť,
- nerovnosť však neplatí pre aktuálne optimálne riešenie príslušnej relaxovanej úlohy.

Takto vytvorený rez teda redukuje oblasť spojitých riešení, čo znamená, že odrezáva časť množiny prípustných riešení, ktorá neobsahuje žiadny celočíselný bod. Tým pádom sa zmenší oblasť (množina prípustných riešení), ktorú budeme v nasledujúcom kroku prehľadávať. Daný postup opakujeme, kým nedostaneme celočíselné riešenie úlohy. Princíp Gomoryho algoritmu ilustrujeme na nasledujúcom príklade.

Príklad

Máme daný celočíselný optimalizačný problém:

$$\begin{aligned} \texttt{max} \quad & 7x_1 + 9x_2 \\ & -x_1 + 3x_2 \leq 6, \\ & 7x_1 + 1x_2 \leq 35, \\ & \quad x_1, x_2 \geq 0,\ x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

Najprv vyriešime relaxovanú úlohu k danej úlohe, t.j. neberieme do úvahy požiadavku celočíselnosti premenných a riešime klasickú úlohu LP simplexovým algoritmom. Takto dostaneme nasledujúcu konečnú simplexovú tabuľku:

	z	x_1	x_2	s_1	s_2	$\bar{b}$
z	1	0	0	28/11	15/11	63
x_2	0	0	1	7/22	1/22	7/2
x_1	0	1	0	-1/22	3/22	9/2

kde s_1, s_2 sú doplnkové premenné. Vo všeobecnosti na generovanie roviny rezu môžeme použiť ľubovoľnú rovnicu prislúchajúcu neceločíselnému riešeniu. Dá sa však ukázať, že ak sa rovina rezu generuje v rámci prvého možného riadku, tak Gomoryho algoritmus vyrieši úlohu celočíselného programovania za konečný počet iterácií [4].

Uvažujme teda prvú rovnicu danej konečnej simplexovej tabuľky, ktorá má nasledujúci tvar

$$x_2 + \frac{7}{22}s_1 + \frac{1}{22}s_2 = \frac{7}{2}$$

Koeficienty na obidvoch stranách rovnice rozložíme na ich celočíselné a zlomkové časti

$$x_2 + (0 + \frac{7}{22})s_1 + (0 + \frac{1}{22})s_2 = 3 + \frac{1}{2}$$

a rovnicu upravíme nasledovne

$$x_2 - 3 = \frac{1}{2} - \frac{7}{22}s_1 - \frac{1}{22}s_2$$

S využitím ľavej strany rovnice môžeme generovať Gomoryho celočíselný rez

$$x_2 - 3 \leq 0 \quad \Leftrightarrow \quad x_2 \leq 3$$

prípadne môžeme vytvoriť Gomoryho zlomkový rez zohľadňujúc pravú stranu rovnice

$$\frac{1}{2} - \frac{7}{22}s_1 - \frac{1}{22}s_2 \leq 0 \quad \Leftrightarrow \quad -\frac{7}{22}s_1 - \frac{1}{22}s_2 \leq -\frac{1}{2}$$

Môžeme si všimnúť, že všetky nenulové koeficienty generovanej reznej nadroviny sú menšie ako 1. Teraz pridáme nové ohraničenie $x_2 \leq 3$ k relaxovanej úlohe celočíselného problému a nové optimálne riešenie je vypočítané v nasledujúcej simplexovej tabuľke:

	z	x_1	x_2	s_1	s_2	s_3	$\bar{b}$
z	1	0	0	0	1	8	59
x_2	0	0	1	0	0	1	3
s_1	0	0	0	1	1/7	-22/7	11/7
x_1	0	1	0	0	1/7	-1/7	32/7

pričom s_3 je doplnková premenná prislúchajúca novému ohraničeniu $x_2 \leq 3$. Optimálne riešenie je však stále neceločíselné, preto budeme generovať ďalšiu sečnú nadrovinu s využitím posledného riadku aktuálnej simplexovej tabuľky

$$x_1 + (0 + \frac{1}{7})s_2 + (-1 + \frac{6}{7})s_3 = 4 + \frac{4}{7}.$$

Gomoryho celočíselný rez má teda tvar

$$x1 - s3 \leq 4,$$

a zlomkový rez vyzerá nasledovne

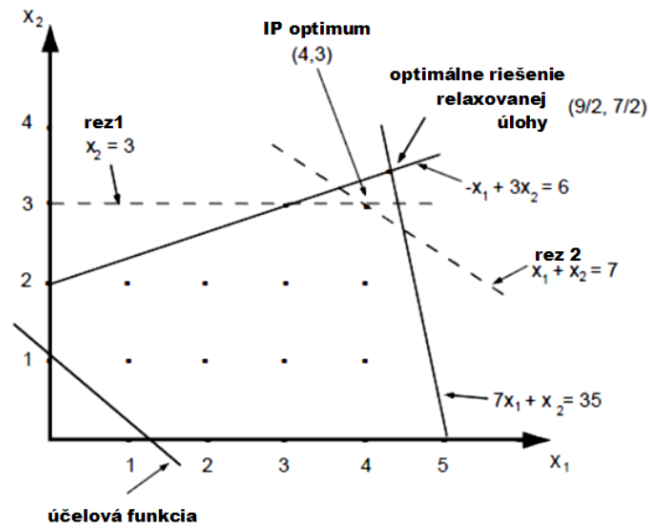
$$-\frac{1}{7}s_2 - \frac{6}{7}s_3 \leq -\frac{4}{7}.$$

Ak použijeme substitúciu $s_2 = 35 - 7x_1 - x_2$ a $s_3 = 3 - x_2$, tak Gomoryho rez môžeme zapísať v tvare

$$x_1 + x_2 \leq 7.$$

Pridaním ohraničenia $x_1 + x_2 \leq 7$ k relaxovanej úlohe daného IP problému a vyriešením novej úlohy simplexovým algoritmom dostaneme **celočíselné optimálne riešenie** $\mathbf{x}^* = (\mathbf{4}, \mathbf{3})^{\mathbf{T}}$ s optimálnou hodnotou účelovej funkcie $\mathbf{z}^* = \mathbf{55}$.

Grafické riešenie daného príkladu je znázornené na obrázku 3



Obr. 3: **Princíp rezných nadrovín**

Gomoryho algoritmus, popísaný v tejto časti práce, sa vyznačuje **pamäťovou** a **výpočtovou zložitosťou**, čo má za následok pomalú konvergenciu (veľký počet iterácií) algoritmu. Na druhej strane výhodou tejto metódy je, že ju môžeme v danej fáze zastaviť a aktuálnu hodnotu účelovej funkcie použiť ako lepšie horné ohraničenie v porovnaní s LP odhadom a pokračovať v riešení úlohy napríklad metódou vetvenia a hraníc.

Doteraz sme sa detailnejšie nezmienili o tom, ako postupovať pri riešení úlohy LP v prípade, že máme k dispozícii finálnu simplexovú tabuľku a k pôvodnej úlohe pridáme nové obmedzenie. Dostaneme tak novú simplexovú tabuľku obsahujúcu jeden riadok a jeden stĺpec navyše, ktorá je však *primárne neprípustná*.

Riešiť rozšírenú úlohu od začiatku je samozrejmé možné, ale rýchlejšie a teda jednoduchšie (najmä pri riešení úloh veľkých rozmerov) je použiť **duálnu simplexovú metódu**. Najväčší význam tejto metódy je práve v celočíselnom programovaní, kde viaceré postupy sú založené na reoptimalizácii [9].

2.3 Duálna medzera v úlohe celočíselného programovania

Už samotný názov tejto časti práce naznačuje, že sa budeme zaoberať teóriou duality v celočíselnom programovaní, pričom sa sústredíme predovšetkým na `ekonomickú interpretáciu` duálnej úlohy k úlohe IP.

Najprv objasníme základné pojmy týkajúce sa duálnej úlohy v lineárnom programovaní. V tejto práci budeme pod pojmom primárna úloha rozumieť vždy maximalizačnú úlohu a k nej prislúchajúca duálna úloha bude minimalizačná.

2.3.1 Duálna úloha k úlohe LP

V lineárnom programovaní ***teória duality*** poskytuje nielen nutné, ale aj postačujúce podmienky na overenie optimality získaného prípustného riešenia. ***Silná veta o dualite*** hovorí, že ak ***primárna úloha*** má optimálne riešenie, tak optimálne riešenie má aj ***duálna úloha*** a naopak, pričom ***optimálne hodnoty účelových funkcií*** sú ***rovnaké***. Presnú formuláciu tejto vety a jej dôkaz možno nájsť v [9].

Avšak vo všeobecnosti pre IP úlohy nemáme analogický aparát. A preto pojem duality v celočíselnom programovaní je nejednoznačný v prípade ekonomickej interpretácie [3]. Tento fakt ukážeme na konkrétnom príklade.

Pre niektoré úlohy LP, ktoré modelujú reálne problémy, sa aj duálne úlohy k nim prislúchajúce dajú interpretovať ako reálne problémy.
Ilustrujeme to na príklade **výrobného problému**.

Príklad - Optimálne riadenie výroby

Predpokladajme, že firma má k dispozícii zdroje / výrobné činitele / v množstvách $b_1, b_2, \ldots, b_m$, z ktorých môže vyrobiť n typov produktov s cenami $c_1, c_2, \ldots, c_n$ za jednotku produktu, pričom na výrobu jednotky produktu j treba a_{ij} jednotiek zdroja i. Cieľom je určiť také množstvá $x_1, x_2, \ldots, x_n$ jednotlivých produktov, aby zdroje, ktoré má firma k dispozícii, boli postačujúce a aby firma pri danej výrobnej aktivite *maximalizovala svoj celkový zisk*. Tento problém môžeme formulovať ako nasledujúcu úlohu lineárneho programovania:

$$
\begin{aligned}
\max \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
& a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1, \\
& a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2, \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m, \\
& x_1, x_2, \ldots, x_n \geq 0.
\end{aligned}
$$

Teraz vytvoríme duálnu úlohu k tejto úlohe:

$$
\begin{aligned}
\max \quad & b_1y_1 + b_2y_2 + \ldots + b_my_m \\
& a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \geq c_1, \\
& a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \geq c_2, \\
& \vdots \\
& a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \geq c_n, \\
& y_1, y_2, \ldots, y_m \geq 0.
\end{aligned}
$$

V prípade duálnej úlohy chceme určiť ***ocenenia*** $\mathbf{y_i}$ jednotlivých zdrojov tak, aby *celkové náklady* (v zmysle ocenenia výrobných činiteľov) pri množstvách zdrojov b_i a cenách produktov c_j boli minimálne. Koeficienty a_{ij} majú rovnakú interpretáciu ako v primárnej úlohe.

Firma F1 sa namiesto výroby produktu môže rozhodnúť predať disponibilné zdroje inej firme F2. Avšak firma F2 chce kúpiť dané zdroje za čo najnižšiu cenu. Čo sa týka firmy F1, tak stanoví takú cenu, aby za predaj príslušného počtu jednotiek daného zdroja získala aspoň toľko, koľko by získala za predaj produktu, ktorý by z toho zdroja vyrobila - a tento fakt vyjadrujú práve ***obmedzenia*** v danej ***duálnej úlohe***.

Príslušné **optimálne hodnoty duálnych premenných** sa označujú ako **duálne ceny** (***dual prices***) resp. **relatívne náklady** [9]. Duálne premenné totiž určujú, nakoľko je disponibilné množstvo výrobných prostriedkov limitujúce z hľadiska zmeny hodnoty funkcie, ktorú optimalizujeme.

V ekonomických analýzach je dôležitá práve informácia o tom, ako sa dá zlepšiť hodnota účelovej funkcie. Z toho dôvodu sa zaviedol pojem **tieňová cena** (***shadow cost***), ktorý udáva, o koľko sa zvýši hodnota účelovej funkcie, ak príslušné ohraničenie v primárnej úlohe bude o jednotku väčšie [11]. Inými slovami, za akú cenu sa oplatí predať jednotku príslušného zdroja. Tieto údaje sa v praxi využívajú na `riadenie výroby`[11].

Presné matematické zdôvodnenie predchádzajúcej interpretácie je nasledovné - nech x^*, y^* sú optimálne riešenia dvojice duálnych úloh, potom podľa ***silnej vety o dualite*** sú optimálne hodnoty účelových funkcií rovnaké, čiže:

$$c_1x_1^* + c_2x_2^* + \ldots + c_nx_n^* = b_1y_1^* + b_2y_2^* + \ldots + b_my_m^*.$$

Ak sa zvýši iba disponibilné množstvo b_i i-teho zdroja o jednotku, teda z b_i na $b_i + 1$, zmení sa optimálne riešenie x^* na nové optimálne riešenie $\bar{x}$ a optimálne hodnota účelovej funkcie primárnej úlohy bude $c^T\bar{x}$. Ak sa nezmení optimálne riešenie duálnej úlohy, teda $\bar{y} = y^*$, tak platí:

$$c^T\bar{x} = b_1y_1^* + b_2y_2^* + \ldots + (b_i + 1)y_i^* + \ldots + b_my_m^* = b^T\bar{y} + \bar{y}_i.$$

Premenná $\bar{y}_i$ preto udáva, o koľko sa zmení optimálna hodnota účelovej funkcie, ak je k dispozícii jednotka i-teho výrobného faktora naviac. Treba však zdôrazniť, že hodnotu primárnej účelovej funkcie nemôžeme zlepšovať len zvyšovaním množstva jedného výrobného prostriedku. Po prekročení určitej hranice sa zmení aj duálne optimálne riešenie a hodnota premennej $\bar{y}_i$ bude nulová. V takom prípade zvyšovanie hodnoty b_i nebude mať vplyv na zlepšovanie hodnoty účelovej funkcie primárnej úlohy.

Vráťme sa ešte k oceneniu jednotlivých výrobných faktorov, pričom použijeme tvrdenie **vety o komplementarite** [9], ktoré môžeme sformulovať v nasledovnom tvare:

$$\sum_{j=1}^{n} a_{ij}x_j^* < b_i \quad \Rightarrow \quad y_i^* = 0, \qquad \text{pre všetky} \quad i = 1, \ldots, m.$$

Táto implikácia znamená, že ak niektorý výrobný prostriedok nie je v optimálnom riešení úplne využitý, tak optimálne ocenenie tohto zdroja je nulové. To znamená, že nemá žiadnu cenu z hľadiska zlepšenia hodnoty účelovej funkcie.

Aplikovaním vety o komplementarite na duálnu úlohu dostaneme:

$$x_j^* > 0 \quad \Rightarrow \quad \sum_{i=1}^{m} a_{ij}y_i^* = c_j \qquad \text{pre všetky} \quad j = 1, \ldots, n.$$

Na základe toho môžeme povedať, že j-ty produkt sa má vyrábať a celkové relatívne náklady sa majú rovnať jeho cene. Z dôvodu uvedenej ekonomickej interpretácie vety o komplementarite, sa niekedy táto veta označuje ako ***veta o rovnováhe cien*** [11].

Vo všeobecnosti teda teória duality v lineárnom programovaní poskytuje 3 dôležité výsledky:

- **slabá veta o dualite**
  – podľa tejto vety **hodnota účelovej funkcie duálnej úlohy** v ľubovoľnom prípustnom riešení tejto úlohy predstavuje **hornú hranicu** pre optimálnu hodnotu účelovej funkcie primárnej úlohy
  – to znamená, že ak nájdeme duálne prípustné riešenie, tak môžeme odhadnúť nakoľko je primárne prípustné riešenie vzdialené od optimálneho riešenia
  – napriek tomu, že táto vlastnosť nie je až taká dôležitá pre LP úlohy, na riešenie ktorých sú dostupné robustné algoritmy, je však nevyhnutná pre celočíselné programovanie
  – ak teda predpokladáme, že primárna aj duálna úloha majú prípustné riešenie, tak pre primárnu úlohu nemôže nastať prípad neohraničenosti, lebo slabá veta o dualite zaručuje ohraničenosť v smere optimalizácie

- **analýza citlivosti (*sensitivity analysis*) riešenia na zmenu vstupných údajov**
  – nech $\bar{y}$ je optimálne riešenie duálneho problému, a teda $b^T\bar{y}$ je extrémnou hodnotou účelovej funkcie primárnej aj duálnej úlohy; ak sa pravá strana ohraničení zmení o hodnotu Δb, tak sa zmení iba účelová funkcia duálnej úlohy a $\bar{y}$ je stále prípustným riešením tejto úlohy, a teda hodnota $(b+\Delta b)^T\bar{y}$ je horným ohraničením optimálnej hodnoty zmeneného primárneho problému

- **komplementarita doplnkových premenných** (***complementary slackness***)
  – predchádzajúci výsledok implikuje, že ak pravá strana b_i daného ohraničenia predstavuje obmedzenie týkajúce sa množstva zdroja, ktoré máme k dispozícii na výrobu produktu, tak zmena tohto obmedzenia o Δb_i zvýši optimálny zisk maximálne o $b_i\bar{y}_i$
  – navyše podľa vety o komplementarite, zdroj, ktorého disponibilné množstvo nie je v opitmálnom riešení úplne využité (***surplus resource***), má **nulové ocenenie** (***zero marginal value***) [2].

2.3.2 Dualita v celočíselnom programovaní

Zavedieme teraz pojem **duálnej medzery** ***(duality gap)***, ktorá je definovaná ako ***odchýlka hodnôt účelových funkcií primárnej a duálnej úlohy*** [6]. ***Nulová duálna medzera*** v prípade úlohy LP je nielen postačujúcou, ale aj nutnou podmienkou optimality. Na konkrétnom príklade ukážeme, ako je to s duálnou medzerou v prípade IP úlohy.

Uvažujme nasledujúcu úlohu celočíselného programovania, ktorá predstavuje `výrobný problém` s `diskrétnymi hodnotami` zdrojov:

$$
\begin{aligned}
z = 3x_1 + 8x_2 &\rightarrow \mathbf{max} \\
2x_1 + 5x_2 &\leq 17, \\
x_1 + 4x_2 &\leq 9, \\
x_1, x_2 &\geq 0, \quad x_1, x_2 \in \mathbb{Z}.
\end{aligned} \tag{4}
$$

Celočíselné riešenie tejto úlohy získané Gomoryho algoritmom je $\mathbf{x}^* = (\mathbf{8}, \mathbf{0})^{\mathbf{T}}$ s hodnotou účelovej funkcie v tomto bode $\mathbf{z}^* = \mathbf{24}$. Optimálne riešenie relaxovanej úlohy k danej úlohe je $\bar{\mathbf{x}} = (\mathbf{\frac{23}{3}}, \mathbf{\frac{1}{3}})^{\mathbf{T}}$ s hodnotou účelovej funkcie $\bar{\mathbf{z}} = \mathbf{\frac{77}{3}}$.

Duálna úloha k danej relaxovanej úlohe má tvar:

$$
\begin{aligned}
w = 17y_1 + 9y_2 &\rightarrow \mathbf{min} \\
2y_1 + y_2 &\geq 3, \\
5y_1 + 4y_2 &\geq 8, \\
y_1, y_2 &\geq 0, \quad y_1, y_2 \in \mathbb{R}.
\end{aligned}
$$

Podľa ***silnej vety o dualite***, ktorá platí pre úlohu lineárneho programovania, je optimálna hodnota účelovej funkcie duálnej úlohy rovnaká ako v prípade primárnej úlohy, teda $\bar{\mathbf{w}} = \mathbf{\frac{77}{3}}$.

Duálne ceny získame z **finálnej simplexovej tabuľky** z riešenia primárnej úlohy – $\bar{\mathbf{y}} = (\mathbf{\frac{4}{3}}, \mathbf{\frac{1}{3}})^{\mathbf{T}}$ – ide o vektor cien, ktorého zložky sú v účelovom riadku tabuľky pod doplnkovými premennými, ktoré udávajú s akou rezervou je dané obmedzenie v tvare nerovnosti splnené oproti obmedzeniu s rovnosťou. Optimálna simplexová tabuľka z riešenia úlohy primárnou simplexovou metódou teda obsahuje aj ***optimálne riešenie duálnej úlohy***.

V tomto príklade **minimálne celkové ocenenie** všetkých výrobných činiteľov nemôže byť menšie ako $\mathbf{\frac{77}{3}}$ a túto hodnotu nie je možné znížiť na maximálnu celkovú hodnotu produktu **24**. Existuje tu **duálna medzera**.

Na tomto mieste vzniká otázka, prečo porovnávame **optimálne riešenie IP úlohy** (maximálnu hodnotu účelovej funkcie primárnej úlohy) s optimálnym riešením **duálnej LP úlohy** (minimálnu hodnotu účelovej funkcie duálnej úlohy). Dôvodom je fakt, že optimálne riešenie duálnej relaxovanej úlohy je vždy aspoň také dobré (v zmysle hodnoty účelovej funkcie) ako optimálne riešenie primárneho celočíselného problému.

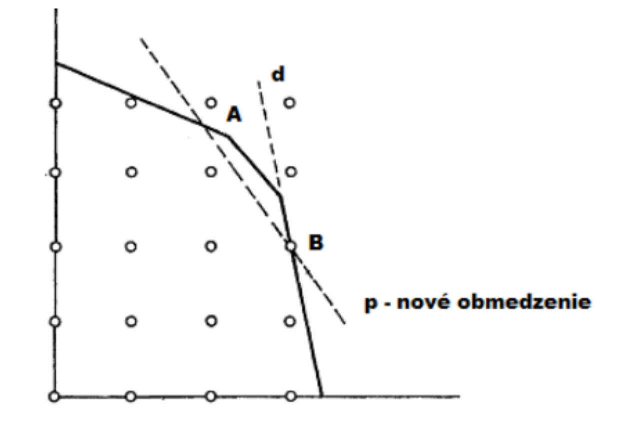
Ak ohraničujúca podmienka je v tvare nerovnosti, tak pri riešení úlohy simplexovým algoritmom je potrebné zaviesť pre takúto podmienku tzv. **doplnkovú premennú** (***slack variable***), ktorá predstavuje hodnotu, o ktorú treba doplniť ľavú stranu nerovnice, aby ohraničenie pre dané hodnoty premenných bolo splnené ako rovnica.

V lineárnom programovaní je ohraničenie s **nenulovou hodnotou doplnkovej premennej** v optimálnom riešení **redundantné**. To znamená, že toto ohraničenie môžeme vynechať a optimálne riešenie primárnej a teda aj duálnej úlohy sa nezmení.

V prípade úlohy celočíselného programovania môže byť dané ohraničenie v optimálnom riešení splnené ako ***ostrá nerovnosť*** (takéto ohraničenie sa označuje ako *non - binding constraint*), no jeho vynechanie môže mať za následok ***zmenu optimálneho riešenia úlohy***.

Prvé ohraničenie problému (4), $\mathbf{2x_1 + 5x_2 \leq 17}$, má v celočíselnom riešení optimálnom $\mathbf{x}^* = (\mathbf{8}, \mathbf{0})^{\mathbf{T}}$ **rezervu** (***slack***) **1** a vynechanie tohto ohraničenia zmení optimálne riešenie na $\mathbf{x}^* = (\mathbf{9}, \mathbf{0})^{\mathbf{T}}$. Tým pádom sa optimálna hodnota účelovej funkcie zmení z pôvodných **24** na **27**. Druhé ohraničenie, $\mathbf{x_1 + 4x_2 \leq 9}$, je tiež v optimálnom riešení splnené s rezervou 1, no vynechaním tohto ohraničenia sa hodnota účelovej funkcie v novom optimálnom riešení nezmení.

Pozrime sa na obr.4., ktorý slúži na ilustráciu ocenení zdrojov v úlohe IP. Predpokladajme, že bod A je **neceločíselným** optimálnym riešením rozšíreného problému. Bod B je **celočíselným** optimálnym riešením úlohy, ktoré získame pridaním nového obmedzenia **p** k rozšírenému problému. Potom daný zdroj nebude úplne využitý v bode **A** v rámci ohraničenia **d** a jeho ocenenie je nulové. Avšak bod T leží na priamke vytvárajúcej ohraničenie **d**, a preto ocenenie zdroja v tomto bode bude kladné.



Obr. 4: **Duálne ceny**

Tento príklad ukazuje, že **pojem komplementarity** definovaný pre úlohy lineárneho programovania sa nedá použiť pre úlohy celočíselného programovania. Ocenenia, ktoré sú nenulové pre neceločíselné riešenie, môžu klesnúť na nulu v prípade celočíselného optimálneho riešenia. Duálna informácia môže byť dokonca ovplyvnená ohraničeniami, ktoré pridávame v priebehu riešenia úlohy Gomoryho algoritmom [3].

Zaoberanie sa prístupmi, ktoré sa snažia zachytiť požadované ekonomické vlastnosti aj pre duálnu úlohu k úlohe ceočíselného programovania, hlavne tzv. ***Gomory - Baumol dual prices***, je nad rámec tejto práce. Informácie ohľadom tohto problému môže čitateľ nájsť v [2, 3].

3 Programová realizácia Gomoryho algoritmu

Jedným z cieľov tejto práce bolo naprogramovať Gomoryho algoritmus. Na realizáciu tohto cieľa sme použili programové prostredie **MatLab**, ktoré je ľahko aplikovateľné na prácu s vektormi a maticami. V tejto kapitole stručne popíšeme najdôležitejšie časti nami vytvoreného programu na riešenie úloh IP.

Najpodstatnejšou časťou programu, a teda aj Gomoryho algoritmu, je **pridávanie nových obmedzení** do **aktuálnej finálnej simplexovej tabuľky**. Tento problém umožňuje riešiť nasledovná funkcia, v ktorej vstupná premenná **type** je dátového typu reťazec a určuje, či ide o maximalizačnú alebo minimalizačnú úlohu, premenná **A** predstavuje maticu príslušnej sústavy obmedzení, **b** je pravá strana sústavy a **c** je vektor koeficientov účelovej funkcie.

```
function Cutting_Plane_Algorithm(type, A, b, c)
if strcmp(type,'min')
   t = -1;
   else
   t = 1;
end
```

Zadanú úlohu teraz riešime bez ohľadu na podmienku celočíselnosti (hovoríme, že riešime relaxovanú úlohu) simplexovým algoritmom. Ten je naprogramovaný v pomocnej funkcii **simplex(type,A,b,c)**, ktorej výstupnou premennou je množina indexov bázických premenných.

```
[m,n] = size(A);
[A,indices_basic_variables] = simplex(type,A,b,c);
[m,n]= size(A);
a = A(1:m-1,end);
f = fractional_part(a);
```

Nasledujúca pomocná dáva ako výstup zlomkové časti čísel v danom poli x.

```
function f = fractional_part(x)
f = zeros(1,length(x));
indexy = find(abs(x - round(x)) >= eps);
f(indexy) = x(indexy) - floor(x(indexy));
```

Algoritmus pokračuje, kým zlomkové časti čísel sú v počítačovej aritmetike rozoznateľné od nuly, pričom `eps` je najmenšie také kladné číslo, pre ktoré platí $(1 + eps) > 1$. V opačnom prípade získané optimálne riešenie je už celočíselné.

```
while norm(f,'inf') > eps
     [zlomkova_cast,i] = max(f);
     new_row = A(i,1:n-1);
```

V tejto časti algoritmu sa generuje **rovinu rezu** (**Gomoryho sečná nadrovina**) v rámci riadku, v ktorom je zlomková časť maximálna. Tým je totiž zabezpečená konvergencia Gomoryho algoritmu za konečný počet iterácií [2].

```
     new_row = [-fractional_part(new_row) 1 -zlomkova_cast];
     B = [A(1:m-1,1:n-1) zeros(m-1,1) A(1:m-1,end)];
     B = [B;new_row;[A(m,1:n-1) 0 A(end,end)]];
     A = B;
     [m,n] = size(A);
```

K pôvodnej úlohe teda pridáme nové obmedzenie, ktoré pomocou doplnkovej premennej zmeníme na rovnicu. Dostaneme novú simplexovú tabuľku s jedným riadkom a jedným stĺpcom navyše a riešime **rozšírenú úlohu** (***augmented program***) **duálnou simplexovou metódou**.

Nevýhodou tejto metódy je to, že prípustné riešenie získame až vtedy, keď optimálne, t.j. na konci výpočtov. To znamená, že ak výpočty ukončíme skôr, priebežný výsledok nie je možné použiť ako suboptimálne riešenie na rozdiel od primárnej simplexovej metódy [10].

V nasledujúcej časti algoritmu je na transformáciu duálnej simplexovej tabuľky aplikované **pravidlo maximálnych podielov** (***maximum ratio test***). Toto pravidlo zabezpečí, že nová simplexová tabuľka bude duálne prípustná. V tomto prípade vedúci riadok je riadok s najmenšou zápornou pravou stranou a vedúci stĺpec je podľa uvedeného pravidla určený tak, aby podiel prvku v účelovom riadku a príslušného záporného prvku vedúceho riadku bol maximálny.

Môžeme sa však stretnúť aj s *pravidlom minimálnych podielov* v prípade duálnej simplexovej transformácie - v takom prípade sa zmení iba to, že vedúci stĺpec je ten, v ktorom je podiel prvku v účelovom riadku a zodpovedajúceho záporného prvku vedúceho riadku s opačným znamienkom minimálny [5].

```
[bmin, row] = min(A(1:m-1,end));
while( bmin < 0 && abs(bmin) > eps )
      column = MRTD(A(m,1:n-1),A(row,1:n-1));
```

Ak nastane prípad, že do premennej `column`, ktorá predstavuje index vedúceho stĺpca, sa nepriradí žiadne číslo, tak množina prípustných riešení je prázdna.

```
if isempty(column)
   fprintf('\n Error.')
   return
end
```

V tejto fáze výpočtu pokračujeme transformáciou simplexovej tabuľky na novú simplexovú tabuľku. Prvok, ktorý sa nachádza vo vedúcom riadku a vedúcom stĺpci, sa nazýva **vedúci prvok**. Teraz vynásobíme vedúci prvok takou konštantou, aby sme dostali jednotku. Ostatné riadky upravíme tak, že od nich odčítame také násobky vedúceho riadku, aby sme dostali nulové koeficienty vo vedúcom stĺpci.

Simplexový algoritmus je v podstate Gauss–Jordanova eliminačná metóda riešenia sústav lineárnych rovníc so špeciálnou voľbou vedúceho prvku eliminácie [11].

```
   A(row,:)= A(row,:)./A(row,column);
   indices_basic_variables(row) = column;
   for i = 1:m
       if i ~= row
          A(i,:)= A(i,:)-A(i,col)*A(row,:);
       end
   end
   [bmin, row] = min(A(1:m-1,end));
   end
   a = A(1:m-1,end);
   f = fractional_part(a);
end
x = zeros(n-1,1);
x(indices_basic_variables) = A(1:m-1,end);
```

Jedným zo základných pravidiel na transformáciu simplexovej tabuľky je **pravidlo minimálnych podielov** (***minimum ratio test***), v ktorom zohráva úlohu minimálny podiel pravých strán a príslušných kladných koeficientov vedúceho stĺpca.

Toto pravidlo zabezpečí, že nová simplexová tabuľka je prípustná a **hodnota účelovej funkcie neklesne** [9]. Výstupná premenná *riadok_index* určuje index vedúceho riadku, premenná *minimum* vyjadruje minimálnu hodnotu podielu pravej strany a zodpovedajúceho kladného prvku vo vedúcom stĺpci.

Na rozdiel od duálnej simplexovej metódy sa v algoritme primárnej simplexovej metódy určí najskôr **vedúci stĺpec** - ako stĺpec s najmenším záporným koeficientom v účelovom riadku.

```
function [riadok_index, minimum] = MRT(a, b)
c = 1:length(a);
index = c(b > 0);
[minimum, riadok_index] = min(a(index)./b(index));
riadok_index = index(riadok_index);
```

Pri riešení LP úlohy ***simplexovým algoritmom*** sa môže stať, že sa opakuje rovnaká postupnosť niektorých simplexových tabuliek a výpočet sa zacyklí. V literatúre sa uvádza, že v praktických úlohach sa zacyklenie nevyskytuje často [9].

Aby sme sa takémuto problému vyhli a tým zabezpečili konečnosť simplexového algoritmu, v našom programe použijeme anticyklickú metódu a to konkrétne **Blandovo pravidlo** [6], označované aj ako **pravidlo najmenších indexov** (***double least-index rule***). Ide o spresnené základné pravidlo pre voľbu pivota (vedúceho prvku), ktoré spočíva v tom, že volíme vedúci stĺpec s najmenším indexom a následne z možných vedúcich riadkov zvolíme ten, ktorého bázická premenná má minimálny index (v tomto prípade teda determinujeme obe súradnice vedúceho prvku).

```
function [m, j] = Bland_rule(a)
ind = find(a < 0);
if ~isempty(ind)
   j = ind(1);
   m = d(j);
else
  m = [];
  j = [];
end
```

Algoritmus primárnej simplexovej metódy končí, ak sa v účelovom riadku nenachádza žiadny záporný koeficient, a teda by sme nemali ako zvoliť vedúci stĺpec [9]. V prípade duálnej simplexovej metódy sa algoritmus končí, ak vektor pravých strán obsahuje iba nezáporné zložky.

Ako sme už spomenuli, pomocná funkcia **simplex(type,A,b,c)** umožňuje riešiť úlohy LP **simplexovým algoritmom**. Tento algoritmus je založený na efektívnom prehľadávaní ***krajných bodov (vrcholov)*** množiny prípustných riešení. Predstavuje presun po `susedných vrcholoch` polyedrickej množiny určenej danými ohraničeniami.

Začneme v ľubovoľnom krajnom bode, ktorý zodpovedá bázickému riešeniu sústavy lineárnych rovníc úlohy LP. Potom prejdeme k takému vrcholu množiny prípustných riešení, ktorého hodnota účelovej funkcie je lepšia. Prechod od jedného krajného bodu k inému, znamená prechod od jedného bázického riešenia k inému bázickému riešeniu. Tento postup opakujeme dovtedy, kým už nie je možné nájsť krajný bod s lepšou hodnotou účelovej funkcie. Potom aktuálne bázické riešenie je optimálnym riešením danej úlohy [9].

Z dôvodu eliminácie chýb vzniknutých zo zaokrúhľovania, keďže počítač pracuje s tzv. *floating point aritmetikou*, je výhodné pracovať s tzv. **presnou aritmetikou**. Ide o spôsob reprezentovania čísel a vykonávania aritmetických operácií s nimi v tvare `zlomkov`. Takýto spôsob je však náročný z časového hľadiska a na druhej strane z hľadiska pamäťovej zložitosti algoritmu.

Konkrétne úlohy v tejto práci sme riešili pomocou naprogramovaného Gomoryho algoritmu. Na overenie správnosti riešenia sme použili softvér **XpressIve Student Version**.

4 Aplikácie celočíselného programovania

Medzi najznámejšie problémy, ktorých riešenie sa dá získať využitím metód celočíselného programovania, patria:

- úloha o batohu (knapsack problem),
- dopravná úloha,
- optimalizácia portfólia,
- vytváranie letových plánov (aircraft scheduling),
- efektívne rozmiestnenie zdrojov (allocation problem),
- distribučné problémy a toky v sieťach (transportation problems and network flow).

Cieľom tejto časti práce je stručne charakterizovať vymenované problémy a uviesť špeciálne metódy, ktoré sa na riešenie týchto úloh používajú. Zameriame sa predovšetkým na **úlohu o batohu** a **dopravnú úlohu**.

S **vytváraním letových plánov** sme sa čiastočne oboznámili v časti, kde sme práve na úlohe tohto typu ilustrovali princíp **metódy vetvenia a hraníc**. V praxi sú tieto úlohy oveľa komplexnejšie a ich riešenie je pomerne náročné. Vyžadujú splnenie ďalších podmienok týkajúcich sa letiskových obmedzení, posádky a lietadiel.

Čo sa týka **optimalizácie portfólia**, ide o úlohu kvadratického programovania s lineárnymi ohraničeniami, v ktorej rozhodovacie premenné môžu byť binárne a v takom prípade vyjadrujú, či dané aktívum je súčasťou portfólia alebo nie. Prípadne môže formulácia úlohy zahŕňať celočíselné obmedzenie počtu aktív v portfóliu. Najčastejšie používaným a najznámejším nástrojom na tvorbu portfólia je tzv. Markowitzova mean-variance analýza. V článku [14] sa uvádza alternatívny model pre malých investorov (*mean-variance integer programming portfolio selection*) založený na riešení postupnosti úloh kvadratického programovania s celočíselnými premennými. V prípade, že portfólio rebalancujeme a berieme do úvahy transakčné náklady, je výhodné použiť tzv. *quadratic mixed integer programming portfolio rebalancing model* [15].

Ku klasickým úlohám ekonomického rozhodovania patria tzv. **alokačné problémy** zaoberajúce sa efektívnym rozmiestnením výrobných zdrojov, napr. alokáciou výrobného procesu na konkrétne miesto s cieľom minimalizovať výrobné náklady.

Toky v sieťach umožňujú riešiť aplikácie, v ktorých je potrebné realizovať za sebou idúce operácie. Ide napríklad o modelovanie obslužných procesov v telekomunikačných sieťach alebo o hľadanie najkratšej trasy v cestnej sieti. Zaujímavou aplikáciou je práve tok dopravy, pričom chceme modelovať maximálny počet automobilov, ktoré sa pri danej infraštruktúre môžu dostať z danej oblasti za určité časové obdobie.

Špeciálnu kategóriu úloh lineárneho programovania tvoria **distribučné problémy**, ktoré modelujú úlohy týkajúce sa premiestňovania objektov z jedného miesta na druhé. Do tejto kategórie zaraďujeme aj **priraďovacie problémy** (*assignment problems*), ktoré sa v praxi zaoberajú napr. pridelením úloh pre jednotlivé pracoviská. Medzi distribučné problémy patrí taktiež dopravná úloha, ktorej sa venujeme v nasledujúcej časti práce.

4.1 Dopravná úloha (Transportation problem)

Dopravná úloha ako špeciálny prípad ***distribučných úloh*** patrí medzi problémy lineárneho programovania, s ktorými sa v praxi často stretávame. Nerieši sa však simplexovou metódou, lebo takýto prístup vyžaduje veľký počet simplexových iterácií. Vlastnosti dopravnej úlohy umožňujú jej riešenie inými metódami.

V tejto časti práce spomenieme **metódu severozápadného rohu** (***north-west corner rule***), ktorá za určitých predpokladov umožňuje získať ***celočíselné*** optimálne riešenie dopravnej úlohy.

Predpokladajme, že máme p výrobcov $s_1, \ldots, s_p$ toho istého produktu, ktorí ho vyrobili v množstvách $a_1, \ldots, a_p$ a q odberateľov $t_1, \ldots, t_q$, ktorých požiadavky sú $b_1, \ldots, b_q$. Náklady na prepravu jednotkového množstva od výrobcu s_i k dodávateľovi t_j označme c_{ij}. Cieľom je určiť také množstvo x_{ij} produktu prepraveného od dodávateľa s_i k odberateľovi t_j pre všetky uvažované i, j, aby **celkové dopravné náklady** boli **minimálne**. Pridaním dodatočného predpokladu tzv. **vybilancovanosti** úlohy, ktorý sa označuje ako **dodávateľsko - odberateľská rovnováha** a znamená, že spolu sa vyrobilo práve toľko ako je sumárna požiadavka, má dopravná úloha nasledujúci tvar:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{p}\sum_{j=1}^{q} c_{ij}x_{ij} \\
& \sum_{j=1}^{q} x_{ij} = a_i \quad \text{pre všetky} \quad i = 1, \ldots, p, \\
& \sum_{i=1}^{p} x_{ij} = b_j \quad \text{pre všetky} \quad j = 1, \ldots, q, \\
& x_{ij} \geq 0 \quad \text{pre všetky} \quad i = 1, \ldots, p, \quad j = 1, \ldots, q.
\end{aligned}
$$

Metóda severozápadného rohu sa dá odvodiť z primárnej simplexovej metódy. Výhodou tejto metódy je *prehľadnosť* a *jednoduchosť* výpočtov, keďže pracujeme s tabuľkou, v ktorej je prepísané celé zadanie úlohy, napríklad:

				a_i
				2
				5
				1
b_j	1	4	3	

2	5	3
1	3	1
4	6	3

Na nájdenie prípustného bázického riešenia použijeme nasledovný postup, ktorý je podrobnejšie popísaný v [6, 10]. Zvolíme ľubovoľnú dvojicu indexov $(\mathbf{i}, \mathbf{j})$ a určíme maximálne množstvo produktu, ktoré môžeme dopraviť od dodávateľa s_i k odberateľovi t_j. Je zrejmé, že $\mathbf{x_{ij}} = \min\{\mathbf{a_i}, \mathbf{b_j}\}$. Práve o toto množstvo znížime príslušné požiadavky, pričom aspoň jedna z nových požiadaviek už bude nulová. V ďalšom kroku vynecháme jeden riadok s nulovou požiadavkou. Ak máme viac ako dvoch účastníkov, tak vynecháme iba jedného a to takého, aby v tabuľke zostal aspoň jeden dodavateľ a aspoň jeden odberateľ. Ak je počet účastníkov v tabuľke rovný dvom, tak vynecháme obidvoch. Tento postup opakujeme, kým nenastane prípad, že nemáme žiadnych účastníkov.

Metóda severozápadného rohu je špeciálnym prípadom uvedeného postupu. Indexovú dvojicu (i, j) zvolíme tak, aby $\mathbf{i} + \mathbf{j}$ bolo minimálne – v tabuľke je to vždy *severozápadné políčko*.

Ak parametre úlohy predstavujúce požiadavky daného produktu a jeho dostupnosť sú celočíselné, tak metódou severozápadného rohu získame **celočíselné** optimálne riešenie dopravnej úlohy. Táto vlastnosť je sformulovaná v [6] - theorem 7.7 (*integrality property of basic variables*).

4.2 Úloha o batohu (Knapsack problem)

Predpokladajme, že máme batoh, ktorý unesie najviac 13 kilogramov a chceme doň vložiť knihy, ktorých hmotnosť a užitočnosť je uvedená v nasledujúcej tabuľke:

kniha	hmotnosť	užitočnosť
1	4	9
2	7	10
3	4	5
4	2	2

Potrebujeme zbaliť najdôležitejšie knihy a pritom neprekročiť hmotnostný limit. Tento problém môžeme formulovať v tvare nasledujúcej úlohy:

$$\begin{aligned} \mathbf{max} \quad & 9x_1 + 10x_2 + 5x_3 + 2x_4 \\ & 4x_1 + 7x_2 + 4x_3 + 2x_4 \leq 13, \\ & x_1, x_2, x_3, x_4 \in \{0, 1\}. \end{aligned}$$

Rozhodovacie premenné nadobúdajú hodnoty 1 resp. 0 podľa toho, či daný predmet je alebo nie je vložený do batohu. V úlohe teda máme tzv. ***binárne premenné***, ktoré sa používajú na modelovanie rozhodnutí typu áno - nie (*Yes/No decisions*). V rámci všeobecnejšieho delenia takýto špeciálny prípad úlohy IP označuje ako úloha **bivalentného (nula-jednotkového) programovania *binary programming***.

Daný problém ako typový model úloh sa taktiež označuje ako ***úloha o batohu (knapsack problem)*** a je charakteristický tým, že obsahuje iba jediné ohraničenie. Je zrejmé, že v praxi nejde o samotné zbalenie vecí do batohu, praktické úlohy sa týkajú predovšetkým prepravy tovaru maximalizujúc celkovú hodnotu nákladu.

Úloha o batohu ako optimalizačná úloha s premennými, ktorých hodnoty môžu mať tvar zlomku, je veľmi ľahko riešiteľná. Nech je vo všeobecnosti daná nasledovná relaxovaná úloha k úlohe o batohu:

$$\begin{aligned} \mathbf{max} \quad & \sum_{i=1}^{n} c_i x_i \\ & \sum_{i=1}^{n} a_i x_i \leq b, \\ & 0 \leq x_i \leq 1 \quad \text{pre} \quad i = 1, \ldots n. \end{aligned}$$

V tejto úlohe teda nevyžadujeme, aby daná položka bola resp. nebola vložená do batohu, ale pripúšťame aj možnosť zbalenia iba určitej časti tej položky.

Hoci je samozrejme možné riešiť túto úlohu LP simplexovým algoritmom, dá sa použiť aj oveľa jednoduchšia metóda riešenia. Tá spočíva v tom, že jednotlivé položky usporiadame v poradí (v rámci indexov) podľa neklesajúcej postupnosti čísel c_i/a_i . Bez ujmy na všeobecnosti teraz predpokladajme, že platí $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$. Nech i_0 je najväčší index taký, že $\sum_{i=1}^{i_0} a_i \leq b$. Zbalíme teda položky $1, \ldots, i_0$ (t.j. $x_i = 1$ pre $i = 1, \ldots i_0$) a zvyšný priestor $s = b - \sum_{i=1}^{i_0} a_i$ bude obsahovať s/a_{i_0+1} položky i_0+1 (t.j. $x_{i_0+1} = s/a_{i_0+1}$), ostatné položky do batohu nedáme. Ľahko sa dá nahliadnuť, že takýto postup dáva optimálne riešenie.

Čo sa však týka tzv. ***integer knapsack problems***, vo všeobecnosti ide o tzv. NP ťažké úlohy [1], na riešenie ktorých sa najčastejšie využíva prístup `dynamického programovania`. V takom prípade sa úloha o batohu chápe ako *viacstupňový rozhodovací proces*, v ktorom stav procesu na začiatku danej etapy je ostávajúca kapacita batohu a rozhodnutie v tej etape je určené hodnotou príslušnej binárnej premennej. V tejto práci sa však dynamickou optimalizáciou procesov na riešenie úloh celočíselného programovania nebudeme podrobnejšie zaoberať [1].

Vráťme sa k úlohe sformulovanej na začiatku tejto kapitoly. Jeden z prístupov k riešeniu tejto úlohy je preskúmať všetky možné kombinácie kníh. V tomto prípade by bolo potrebné vyskúšať 16 možností, avšak vo všeobecnosti je počet kombinácií 2^n a preto takáto metóda vyskúšania všetkých možností nie je vhodná. Preto túto úlohu budeme riešiť nasledovným spôsobom. Najprv vytvoríme relaxovanú LP úlohu:

$$\begin{aligned} \textbf{max} \quad & 9x_1 + 10x_2 + 5x_3 + 2x_4 \\ & 4x_1 + 7x_2 + 4x_3 + 2x_4 \leq 13, \\ & 0 \leq x_i \leq 1. \end{aligned}$$

Riešením je $(x_1, x_2, x_3, x_4)^T = (1, 1, \frac{1}{2}, 0)^T$. Pridaním obmedzenia (reznej nadroviny)

$$x_1 + x_2 + x_3 \leq 2,$$

k danej úlohe a jej vyriešením dostaneme celočíselné optimálne riešenie

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1.$$

Záver

Cieľom tejto bakalárskej práce bolo vysvetliť a analyzovať základné algoritmy na riešenie úloh celočíselného programovania. V prvej časti sme definovali úlohu celočíselného programovania a zdôvodnili sme, prečo zaokrúhlenie riešenia relaxovanej úlohy nemusí byť vhodnou aproximáciou. Ďalej sme analyzovali samotné algoritmy na riešenie úloh IP - **Metódu vetvenia a hraníc** a **Gomoryho algoritmus**. Tieto metódy využívajú simplexový algoritmus a zaručujú síce nájdenie optimálneho celočíselného riešenia, no v nepolynomiálnom čase.

Princíp metódy vetvenia a hraníc spočíva v tom, že sa postupne menia ohraničenia daných premenných a vytvárajú sa nové úlohy LP. Jednotlivé parciálne úlohy riešime samostatne a dosiahnuté riešenia posudzujeme vzhľadom na podmienku celočíselnosti. Gomoryho algoritmus využíva fakt, že pridaním nového obmedzenia sa optimálna hodnota účelovej funkcie nikdy nezvýši (v prípade maximalizačnej úlohy). Ukázali sme, že je možné generovať také nové ohraničenie, ktoré je platné pre každé celočíselné prípustné riešenie úlohy IP, avšak neplatí pre aktuálne optimálne riešenie relaxovanej úlohy. Postupným pridávaním takýchto ohraničení (platných nerovností) sa redukuje oblasť spojitých riešení - vynecháva sa časť množiny prípustných riešení, ktorá neobsahuje žiadny celočíselný bod. Daný postup opakujeme, kým nedostaneme celočíselné riešenie úlohy.

Teória duality v lineárnom programovaní poskytuje nielen nutné, ale aj postačujúce podmienky na overenie optimality získaného prípustného riešenia. Analogické tvrdenie v prípade úlohy celočíselného programovania vo všeobecnosti neplatí. Tento fakt sa dá odôvodniť pomocou existencie tzv. duálnej medzery v úlohe IP. Na konkrétnom príklade ukážeme, že ekonomická interpretácia duálnych premenných pre úlohu IP je nejednoznačná na rozdiel od takejto interpretácie pre úlohy LP.

Jedným z cieľov práce bolo naprogramovať Gomoryho algoritmus, ktorý sme použili pri riešení konkrétnych príkladov uvedených v práci. Zároveň nám to umožnilo hlbšie pochopenie tejto metódy určenej na riešenie úloh IP.

V štvrtej kapitole sme sa okrajovo zaoberali aplikáciami celočíselného programovania v rámci riešenia problémov typu úloha o batohu, optimalizáciou portfólia a distribučnými problémami.

Literatúra

[1] Schrijver, A. : **Theory of Linear and Integer Programming**, John Wiley & Sons 1999, Chichester, ISBN 0-471-98232-6.

[2] Floudas, C.A. ,Pardalos, P.M. : **Encyclopedia of Optimization**, 2nd ed., Springer 2009, Princeton University, pp. 1650-1667.

[3] Gomory, R.E. and Baumol, W.J. : **Integer Programming and Pricing**, Econometrica, Vol. 28, No. 3. (July, 1960), pp. 521-550.

[4] Gomory, R.E.: **Outline of an Algorithm for Integer Solutions to Linear Programs**, Bulletin of the American Mathematical Society, Vol. 64, September, 1958.

[5] Dantzig, G.B., Thapa, M.N. : **Linear programming, 1: Introduction**, Springer 1997, New York.

[6] Dantzig, G.B., Thapa, M.N. : **Linear programming, 2: Theory and extensions**, Springer 2003, New York.

[7] Williams, H.P. : **The Problem with Integer Programming**, Working Paper LSEOR 10-118, London School of Economics, 2009, ISSN 2041-4668 (Online).

[8] Bronson, R., Naadimuthu, G. : **Operations Research**, 2nd ed., Schaum's Outlines, McGraw-Hill 1997, New Jersey.

[9] Plesník, J., Dupačová, J., Vlach, M. : **Lineárne programovanie**, Alfa 1990, Bratislava.

[10] Plesník, J. : **Lineárne programovanie**, Seminár 2009/2010.

[11] Toma, V. :**Základy lineárneho programovania**, Knižničné a edičné centrum FMFI UK 2008, Bratislava.

[12] Máca, J., Leitner, B. : **Operačná analýza I**, FŠI ŽU, 2002, Žilina.

[13] Williams, H.P. : **Integer Programming and Pricing Revised**, IMA Journal of Mathematics Applied in Business and Industry, No. 7.(1997), pp. 203-213.

[14] Faaland, B. : **An Integer Programming Algorithm for Portfolio Selection**, Management Science, Vol. 20, No. 10.(June, 1974).

[15] Glen, J.J. : **Mean-variance portfolio rebalancing with transaction costs and funding changes**, Journal of the Operational Research Society, Vol. 62, No 4, April 2011, pp. 667-676.