

MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY

Diplomová práce

BRNO 2020

JANA ZUZAŇÁKOVÁ

**MASARYKOVA
UNIVERZITA**
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY

Lineární programování v praxi

Diplomová práce

Jana Zuzáňáková

Vedoucí práce: doc. Mgr. Petr Zemánek, Ph.D.

Brno 2020

Bibliografický záznam

- Autor:** Bc. Jana Zuzaňáková
Přírodovědecká fakulta, Masarykova univerzita
Ústav matematiky a statistiky
- Název práce:** Lineární programování v praxi
- Studijní program:** Matematika
- Studijní obor:** Aplikovaná matematika pro víceoborové studium, Ekonomie
- Vedoucí práce:** doc. Mgr. Petr Zemánek, Ph.D.
- Akademický rok:** 2019/2020
- Počet stran:** xiv + 150
- Klíčová slova:** Lineární programování; Optimalizace; Simplexová metoda; Simplexová tabulka; Pivotovací pravidla; Dualita; Řešené příklady; Aplikace; Problém jídelníčku; Dopravní problém

Bibliographic Entry

Author: Bc. Jana Zuzaňáková
Faculty of Science, Masaryk University
Department of Mathematics and Statistics

Title of Thesis: Linear Programming in Practice

Degree Programme: Mathematics

Field of Study: Applied Mathematics for Multi-Branched Study, Economics

Supervisor: doc. Mgr. Petr Zemánek, Ph.D.

Academic Year: 2019/2020

Number of Pages: xiv + 150

Keywords: Linear Programming; Optimization; Simplex method; Simplex table; Pivoting rules; Duality; Solved problems; Applications; The diet problem; The transportation problem

Abstrakt

V této diplomové práci se věnujeme lineárnímu programování a jeho využití v praxi. Práce je rozčleněna do šesti kapitol. Úvodní kapitoly pokrývají stručnou historii oboru, motivační příklady a teoretické základy. Třetí kapitola je věnována simplexové metodě, a to jak jejímu odvození, tak různým situacím, které mohou při řešení úloh lineárního programování nastat. Čtvrtá kapitola se zabývá teorií duality a duální simplexovou metodou. Závěrečné kapitoly představují čtenáři možnosti využití lineárního programování v praxi, a to včetně speciálního typu úlohy, kterým je dopravní problém. Všechny kapitoly obsahují řešené příklady ilustrující vysvětlovanou teorii.

Abstract

In this thesis we study linear programming and its practical applications. The thesis consists of six chapters. The introductory chapters cover the brief history of the discipline, motivation and theory behind linear programs. The third chapter deals with the simplex method and different situations that can occur while solving linear programs. The fourth chapter focuses on the duality theory and the dual simplex method. The concluding chapters bring examples of utilization of linear programming in practice, including a special structured problem, the transportation problem. Every chapter includes solved problems that illustrate the explained theory.

ZADÁNÍ
DIPLOMOVÉ PRÁCE

Akademický rok: 2019/2020

Ústav:	Ústav matematiky a statistiky
Studentka:	Bc. Jana Zuzaňáková
Program:	Matematika
Obor:	Aplikovaná matematika pro víceoborové studium Ekonomie

Ředitel Ústavu matematiky a statistiky PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje diplomovou práci s názvem:

Název práce:	Lineární programování v praxi
Název práce anglicky:	Linear Programming in Practice
Jazyk závěrečné práce:	čeština

Oficiální zadání:

Cílem práce je vytvoření uceleného textu s praktickými aplikacemi lineárního programování. Text bude doplněn také potřebným matematickým aparátem v přiměřeném rozsahu.

Literatura:

DOŠLÝ, Ondřej. *Základy konvexní analýzy a optimalizace v R^n* . 1. vyd. Brno: Masarykova univerzita, 2005. 194 s. ISBN 80-210-3905-1.

Vedoucí práce:	doc. Mgr. Petr Zemánek, Ph.D.
Datum zadání práce:	19. 9. 2018
V Brně dne:	2. 4. 2020

Souhlasím se zadáním (podpis, datum):

.....
Bc. Jana Zuzaňáková
studentka

.....
doc. Mgr. Petr Zemánek, Ph.D.
vedoucí práce

.....
prof. RNDr. Jan Slovák, DrSc.
ředitel Ústavu matematiky a sta-
tistiky

Poděkování

Na tomto místě bych chtěla poděkovat doc. Mgr. Petru Zemánkovi, Ph.D. za odborné vedení, cenné rady, podnětné připomínky, ochotu, trpělivost a v neposlední řadě za čas, který mi při psaní textu věnoval.

Prohlášení

Prohlašuji, že jsem svoji diplomovou práci vypracovala samostatně pod vedením vedoucího práce s využitím informačních zdrojů, které jsou v práci citovány.

Brno 20. května 2020

.....
Jana Zuzanáková

Obsah

Úvod	1
Kapitola 1. Úvod do lineárního programování	3
1.1 Historie	3
1.2 Motivační příklady	5
1.3 Základní pojmy lineárního programování	10
1.4 Převody mezi různými tvary úlohy LP	12
1.5 Grafická metoda řešení	15
Kapitola 2. Teoretické základy lineárního programování	23
Kapitola 3. Simplexová metoda	31
3.1 Odvození simplexové metody	32
3.2 Simplexová tabulka	34
3.2.1 Metody pro výběr indexu vstupujícího do báze	43
3.2.2 Zacyklení simplexové metody	47
3.3 Výchozí bazický přípustný bod	53
3.3.1 Metoda velkého M	53
3.3.2 Metoda umělé báze	55
3.4 Alternativní metody řešení úloh lineárního programování	61
Kapitola 4. Dualita v lineárním programování	63
4.1 Duální simplexová metoda	68
4.1.1 Výchozí duálně přípustný bazický bod	72
Kapitola 5. Příklady	81
5.1 Problém jídelníčku	81
5.2 Minimalizace nákladů na zásobování	85
5.3 Pracovní rozvrhy	91
5.4 Řezný problém	95
5.5 Investice – tvorba portfolia	100
5.6 Plánování zásob a výroby	106
Kapitola 6. Dopravní problém	113
6.1 Výchozí bazický přípustný bod	117

6.1.1 Metoda severozápadního rohu	117
6.1.2 Metoda minimální ceny	119
6.1.3 Vogelova aproximační metoda (VAM)	121
6.2 Modifikovaná distribuční metoda	124
Příloha	143
Seznam použité literatury	147

Úvod

Tato diplomová práce se věnuje lineárnímu programování, jež je jednou z významných disciplín aplikované matematiky. Vznik lineárního programování, nazývaného též lineární optimalizace, je úzce spojen s ekonomickým vývojem ve 20. století. Řešené problémy jsou tak obvykle motivovány ekonomickými požadavky, jakými jsou například maximalizace zisku, minimalizace nákladů atp. Práce má za cíl představit čtenáři jak konkrétní příklady využití lineárního programování, tak potřebný matematický aparát.

Text práce je rozčleněn do šesti kapitol, ve kterých se čtenář seznámí se základními pojmy a nejnámějšími typy úloh lineárního programování. Ve všech kapitolách jsou teoretické poznatky doplněny řešenými ilustrujícími příklady. První kapitola poskytuje čtenáři úvod do studia lineárního programování. Kromě stručné historické poznámky obsahuje též čtyři motivační příklady a definici obecné úlohy lineárního programování. V této kapitole se čtenář také dozví o různých tvarech úlohy a způsobech převodu úloh na jiný tvar. Poslední část kapitoly tvoří grafická metoda řešení, pomocí níž je vyřešeno šest příkladů. Druhá kapitola je věnována teoretickým základům lineárního programování. Čtenář se zde seznámí s řadou tvrzení, např. se základní větou lineárního programování, a pojmů, kterými jsou např. nosič bodu, bazický přípustný bod či degenerovanost bodu.

Ve třetí kapitole se seznámíme se simplexovou metodou, a to jak s jejím odvozením, tak s konkrétními příklady využití algoritmu. V této kapitole též na příkladech ilustrujeme několik metod pro výběr indexu vstupujícího do báze. Čtenář se zde také seznámí s pojmem zacyklení algoritmu a s pravidly, která zacyklení zabrání. Tyto anticyklické metody opět ilustrujeme řešenými příklady. V poslední části kapitoly představíme dvě metody pro nalezení výchozího bazického přípustného bodu a stručně okomentujeme alternativní způsoby řešení úloh lineárního programování. Čtvrtá kapitola je věnována duální úloze, a to jak teoretickému pojetí, tak její interpretaci. V kapitole je též uveden popis duální simplexové metody a řešené příklady. Kromě samotného algoritmu se i zde seznámíme se dvěma metodami pro hledání výchozího bazického přípustného bodu.

Pátá kapitola je věnována praktickým příkladům lineárního programování, a to např. problému jídelníčku, problému vytváření rozvrhů, řeznému problému a dalším. Kromě obecného popisu jednotlivých typů úloh je zde vyřešeno šest konkrétních příkladů. V šesté kapitole se čtenář seznámí s jedním ze speciálních problémů lineárního programování, a to s dopravním problémem. V kapitole jsou uvedeny tři metody pro nalezení výchozího bazického přípustného bodu a jedna z metod řešení dopravních problémů. Všechny popsání metody jsou ilustrovány na příkladech. Kapitulu završuje praktický příklad dopravní úlohy.

V textu je předpokládána základní znalost lineární algebry. Cílem práce je ukázat praktické příklady lineárního programování a metody, které můžeme při jejich řešení využít. V textu je z tohoto důvodu kladen důraz spíše na popis metod a ilustrující příklady,

než na podrobné důkazy uvedených tvrzení, na které se odkazujeme do uvedených zdrojů. Práce tak může sloužit všem, kteří se chtějí seznámit s lineárním programováním a jeho úlohami i pouze praktickou cestou.

Práce je vysázena v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, k vytvoření grafů doplňujících text byl využit program GeoGebra, příklady praktických úloh byly vypočteny s využitím softwaru Microsoft Excel. K práci je přiložen excelový soubor s daty a řešeními příkladů z páté kapitoly. Stručný návod aktivace doplňku Řešitel v Excelu najde čtenář spolu s ukázkovým příkladem využití Řešitele v příloze.

Kapitola 1

Úvod do lineárního programování

Matematické programování je disciplína, která se zabývá hledáním vázaných extrémů. Lineární programování je jednou z částí této disciplíny. Společně s lineárním programováním do tohoto oboru spadá např. i celočíselné, kvadratické či dynamické programování. Lineární programování, společně s např. teorií her a teorií grafů, je také jednou z nejrozšířenějších oblastí operačního výzkumu. Operačním výzkumem se označují principy a metody matematického modelování rozhodovacích situací. Samotný název „programování“ může být na první pohled poněkud zavádějící. Slovo programování zde totiž primárně neodkazuje na něco informatického (týkajícího se programu ve smyslu softwaru), jeho význam je zde spíše plánování či optimalizace. Program, neboli vojenský plán či rozvrh, totiž stál u zrodu této disciplíny, která se původně označovala jako programování v lineární struktuře. Samotné slovo program tak tento význam neslo mnohem dříve, než vůbec programy, jak je běžně dnes chápeme, vznikly.

Název lineární programování se poté ustálil díky T. C. Koopmansovi, který jej Georgi B. Dantzigovi navrhl v roce 1948. Původní problémy byly motivovány hlavně potřebami armády či ekonomickými požadavky zemí, jejich cílem bylo najít optimální řešení nějakého praktického problému. Rozvoj lineárního programování (někdy též lineární optimalizace) jako matematické disciplíny je ovšem úzce spojen i s vývojem výpočetní techniky. Úlohu lineárního programování si prozatím představme jako problém nalezení extrému lineární funkce za podmínek daných lineárními rovnostmi a nerovnostmi.

1.1 Historie

Ještě před tím, než se začneme zabývat teoretickými základy lineárního programování, typickými úlohami nebo způsoby řešení, si uvedeme stručný přehled historického vývoje této matematické disciplíny a některých jejích významných problémů. V této části vycházíme z [6], [8], [32], [47] a dalších zdrojů uvedených přímo v textu.

Lineární programování je poměrně mladá oblast matematiky, její hlavní rozvoj začal až v období 2. světové války. Již v 18. století však můžeme najít první předchůdce, a to např. ekonomickou tabulku z roku 1758 francouzského ekonoma Françoise Quesnayeho (viz [6]). V této tabulce se Quesnay rozhodl ukázat vzájemné vztahy statkářů, sedláků a řemeslníků, a to právě s využitím soustavy lineárních rovnic a nerovnostních omezení zaručujících nezápornost proměnných (tzv. omezení na znaménka). Více informací

o Quesnayeho ekonomické tabulce lze nalézt např. v [31].

Jedním ze způsobů, které lze využít při hledání vázaných extrémů, je metoda Lagrangeových multiplikátorů. Tato metoda se poprvé objevila v knize *Mécanique analytique* publikované v roce 1788. Při hledání extrémů lineárních funkcí, které jsou omezené lineárními rovnostmi bez omezení na znaménko, převede Lagrangeova metoda úlohu na soustavu lineárních rovnic, kterou lze vyřešit např. Gaussovou eliminací (na konci 18. století již byla v řadě učebnic, ovšem pod jiným názvem, více viz [18]).

Dalším významným příspěvkem je Fourierův článek z 20. let 19. století, ve kterém autor uvádí algoritmus pro řešení soustav lineárních nerovnic. Podobný přístup může být využit k odvození teorie duality v lineárním programování, více čtenář najde např. v [12]. Jedním ze základů lineárního programování je také Farkasova teorie soustav lineárních nerovnic z přelomu 19. a 20. století. Této oblasti se kromě Farkase dále věnovali např. Hermann Minkowski a Constantin Carathéodory.

Za zmínku stojí také uvedení návrhu řešení dopravního problému, který byl poprvé formulován již na konci 18. století francouzským matematikem Gaspardem Mongem. Toto řešení navrhl v roce 1930 A. N. Tolstoj, obecný postup pro řešení dopravního problému pak v roce 1941 formuloval americký matematik Hitchcock. Více o historii dopravního problému najde čtenář např. v [37]. Ve 30. letech byl také řešen přiřazovací problém, jenž spolu s dopravním problémem řadíme mezi speciální úlohy lineárního programování. Základní zadání přiřazovacího problému je spárovat dvě stejně velké skupiny co nejefektivněji. Jako příklad si můžeme představit množinu strojů a množinu zaměstnanců, o nichž víme, za jak dlouho se s jednotlivými stroji naučí pracovat. Úkolem je pak najít takové přiřazení, které minimalizuje celkový čas potřebný k zaučení pracovníků. Řešení přiřazovacího problému je založeno na kombinatorických úvahách, výsledný algoritmus je nyní kvůli národnosti autorů (König a Egerváry) znám pod názvem Maďarská metoda.

Další významný příspěvek nazvaný *Mathematical Methods in the Organization and Planning of Production* (1939) pochází od ruského matematika L. V. Kantoroviče. Tato kniha, která byla kvůli ideologickým důvodům vydána až o dvě desetiletí později, je považována za první práci o lineárním programování. Kantorovič poté spolu s ekonomem Koopmansem získal v roce 1975 Nobelovu cenu za ekonomii za přínosy k rozvoji lineárního programování a jeho ekonomických aplikací. S přínosem Kantoroviče se lze blíže seznámit např. v [45] a [46].

Jedním z významných problémů lineárního programování je zcela jistě problém jídelníčku. Tento problém byl formulován při 2. světové válce Jerryem Cornfieldem (více o problému je možné najít např. v [11] a [7]). Hlavní motivací bylo najít co nejlevnější, ale zároveň nutričně dostatečný jídelníček pro vojáky. Přibližné řešení tohoto problému nabídnul ekonom George Stigler, jenž se řadí mezi laureáty Nobelovy ceny za ekonomii. Původní (Stiglerův) model jídelníčku sestával ze 77 surovin a 9 pozorovaných výživových hodnot. V roce 1947 byla na tomto problému vyzkoušena Dantzigova simplexová metoda, optimální řešení s její pomocí bylo nalezeno za 120 človeko-dnů. Stiglerův problém jídelníčku je možné najít např. v [52]. Zajímavostí jistě je, že Stiglerovo přibližné řešení se od optimálního řešení nalezeného simplexovou metodou lišilo pouze o 0,24 dolaru za roční výdaje na stravování. Více informací lze najít např. v [17].

Rozvoj lineárního programování byl tedy poháněn hlavně nutností řešení ekonomických a válečných problémů v době 2. světové války i po ní. Nešlo tak jen o oblast zájmu matema-

tiků, ale i ekonomů, a to např. již zmiňovaného Koopmanse, W. Leontiefa, J. von Neumanna, jehož přínos je v teorii duality a teorii her, či O. Morgensterna. George B. Dantzig, jenž je považován za zakladatele lineárního programování a jehož algoritmus zvaný simplexová metoda (1947) je dodnes používán k řešení problémů lineárního programování, sám uvedl, že formulace algoritmu byla vyústěním několikaletých diskuzí s těmito ekonomy, viz [32]. Simplexová metoda (či simplexový algoritmus), která je považována za jeden z nejvýznamnějších algoritmů vyvinutých ve 20. století, byla původně navržena pro potřeby amerického letectva. V roce 1951, kdy se konalo 1. symposium o lineárním programování ve Washingtonu, byl již zájem o tuto disciplínu velmi rozšířený, a to jak v akademických kruzích, tak ve státních úřadech. Další oblastí zájmu, kterou se zabývali např. A. Charnes a W. W. Cooper, bylo využití lineárního programování ve výrobě.

S rozvojem výpočetní techniky v 50. a 60. letech minulého století výrazně vzrostly výpočetní možnosti, v potaz se začala brát také efektivnost a složitost algoritmů používaných k řešení úloh lineárního programování. Vzhledem k náročnosti simplexové metody se začaly objevovat další metody, které jsou sice výpočetně složitější, ale efektivnější. Mezi tyto metody patří např. metoda elipsoidů zveřejněná v roce 1979 či metoda vnitřních bodů z roku 1984. Postup těchto metod a jejich porovnání se simplexovou metodou stručně okomentujeme v Části 3.4.

1.2 Motivační příklady

Před tím, než si zadefinujeme obecnou úlohu lineárního programování a další důležité pojmy, si ukážeme čtyři příklady, které vedou na úlohu lineárního programování a se kterými se, více či méně, můžeme setkat v běžném životě. Řešení následujícího příkladu získané pomocí doplňku Řešitel v Excelu najde čtenář v příloze.

Příklad 1.1. Úloha o výrobě.

Představme si, že máme firmu, která vyrábí 2 druhy výrobků – diáře a stolní kalendáře. Pro zjednodušení předpokládejme, že k výrobě obou výrobků jsou potřeba pouze 3 zdroje: papír, kroužková vazba a lidská práce. Zisk z jednoho diáře je 100 Kč, zisk z jednoho kalendáře pak 180 Kč. K výrobě jednoho diáře potřebujeme 18 ks papíru, jednu kroužkovou vazbu a 30 minut lidské práce. K výrobě stolního kalendáře využijeme 28 ks papíru, 2 kroužkové vazby a 20 minut práce. Všechny zdroje máme omezené množstvím, které je dáno např. tím, na kolik hodin zaměstnáváme pracovníky, nebo kolik kroužkových vazeb máme v zásobách firmy. Maximální počet papírů, který můžeme za den využít, je 1000, maximální počet kroužkových vazeb je 80 a maximální čas výroby je 900 minut denně (naše firma zaměstnává pouze 2 zaměstnance).

Jaký je optimální plán denní výroby této firmy, která chce maximalizovat zisk z prodeje svých výrobků?

Řešení. Ze slovního zadání si nejprve sestavíme matematickou úlohu. Označme si d počet diářů a k počet kalendářů, které firma vyrobí za den. Papír, vazbu a práci označíme postupně p , v , ℓ . Firma vlastně hledá maximum funkce

$$f(d, k) = 100d + 180k$$

za daných podmínek, kterými jsou omezení na zdroje. První omezení se týká využití papíru, kterého potřebujeme 18 na výrobu diáře a 28 na výrobu kalendáře. Naše celková spotřeba

papíru je tedy dána $18d + 28k$. Tato spotřeba je ale omezená naší zásobou papíru, která je 1000. Z těchto informací již můžeme sestavit první omezení, jemuž naše firma čelí, a to

$$18d + 28k \leq 1000.$$

S ostatními zdroji postupujeme obdobně. Pro kroužkové vazby tímto způsobem získáme nerovnostní omezení

$$d + 2k \leq 80,$$

z omezeného času práce pak dostaneme

$$30d + 20k \leq 900.$$

Poslední omezení, která k úloze přidáme, jsou omezení

$$d \geq 0 \quad \text{a} \quad k \geq 0,$$

neboť firma chce vyrábět nezáporná množství diářů a kalendářů. Výroba záporného množství možná na první pohled nedává smysl, ekonomicky ji však můžeme chápat jako nakupování výsledných produktů, tj. zde diářů a kalendářů.

△

Příklad 1.2. Směšovací úloha.

Nyní si představme, že jsme poradci prodejců benzínu, kteří chtějí namíchat paliva s různými oktanovými čísly (např. Natural 91, 95 a 98). K dispozici mají 5 různých druhů surových paliv, která se liší oktanovými čísly, množstvím, které má prodejce k dispozici, a cenou (v dolarech za barel). Údaje pro konkrétní typy (označené čísly 1-5) jsou zobrazeny v Tabulce 1.1.

palivo	oktanové číslo	cena	dostupné množství
1	75	10	2200
2	80	12,5	4000
3	85	13	5000
4	90	27,5	5000
5	99	27,5	3000

Tabulka 1.1: Vlastnosti paliv

Oktanové číslo paliva, které chce prodejce namíchat, získáme jako vážený průměr oktanových čísel použitých surových paliv. Za prodej barelu Naturalu 91, 95 a 98, vydělá prodejce postupně 36,5, 39 a 42 dolarů. Pokud mu zbude nějaké množství surových paliv, které nakoupil, může ho prodat za tržní cenu. Cílem prodejce je maximalizovat zisk. Jakou strategii poradíme prodejci, který disponuje daným množstvím jednotlivých surových paliv, aby dosáhl svého cíle?

Řešení. Stejně jako v předchozím příkladu začneme s formálním zápisem úlohy. Označme množství Naturalu 91, 95 a 98 jako y_1 , y_2 a y_3 . Dále označme z_1, \dots, z_5 množství

surových paliv, které jsme nakoupili, ale nespotřebovali. Zisk z jejich prodeje tedy také vstoupí do ziskové funkce, kterou můžeme zapsat jako

$$36,5y_1 + 39y_2 + 42y_3 + 10z_1 + 12,5z_2 + 13z_3 + 27,5z_4 + 27,5z_5.$$

Tuto funkci chce prodejce maximalizovat. Čelí však několika omezením. Označme si x_{ij} množství surového paliva i využitého k přípravě prodejného paliva j , přičemž $i \in \{1, \dots, 5\}$, $j \in \{1, 2, 3\}$. Proměnná x_{23} tak reprezentuje množství paliva 2 využitého k přípravě Naturalu 98. Z tabulky známe maximální dostupná množství jednotlivých paliv, s jejich využitím získáme podmínky

$$x_{11} + x_{12} + x_{13} + z_1 = 2200,$$

$$x_{21} + x_{22} + x_{23} + z_2 = 4000,$$

$$x_{31} + x_{32} + x_{33} + z_3 = 5000,$$

$$x_{41} + x_{42} + x_{43} + z_4 = 5000,$$

$$x_{51} + x_{52} + x_{53} + z_5 = 3000.$$

Tato omezení jsou ve tvaru rovností, neboť surové palivo buď použijeme na výrobu směsi, nebo prodáme za tržní cenu, využijeme tedy vše, co máme. Pro množství jednotlivých paliv, která vyrobíme, platí rovnosti

$$y_1 = x_{11} + x_{21} + x_{31} + x_{41} + x_{51},$$

$$y_2 = x_{12} + x_{22} + x_{32} + x_{42} + x_{52},$$

$$y_3 = x_{13} + x_{23} + x_{33} + x_{43} + x_{53}.$$

Posledním faktorem, který prodejce zajímá, je oktanové číslo směsi, kterou vyrobí. Pro y_1 máme podmínku

$$\frac{75x_{11} + 80x_{21} + 85x_{31} + 90x_{41} + 99x_{51}}{x_{11} + x_{21} + x_{31} + x_{41} + x_{51}} = \frac{75x_{11} + 80x_{21} + 85x_{31} + 90x_{41} + 99x_{51}}{y_1} \geq 91,$$

kterou můžeme také zapsat jako

$$75x_{11} + 80x_{21} + 85x_{31} + 90x_{41} + 99x_{51} \geq 91y_1.$$

Pro ostatní paliva obdobně získáme další podmínky, kterým při hledání maximálního zisku čelíme. Pro y_2 a y_3 to jsou podmínky

$$75x_{12} + 80x_{22} + 85x_{32} + 90x_{42} + 99x_{52} \geq 95y_2,$$

$$75x_{13} + 80x_{23} + 85x_{33} + 90x_{43} + 99x_{53} \geq 98y_3.$$

Poslední podmínkou je podmínka nezápornosti všech proměnných, které v úloze uvažujeme, tj.

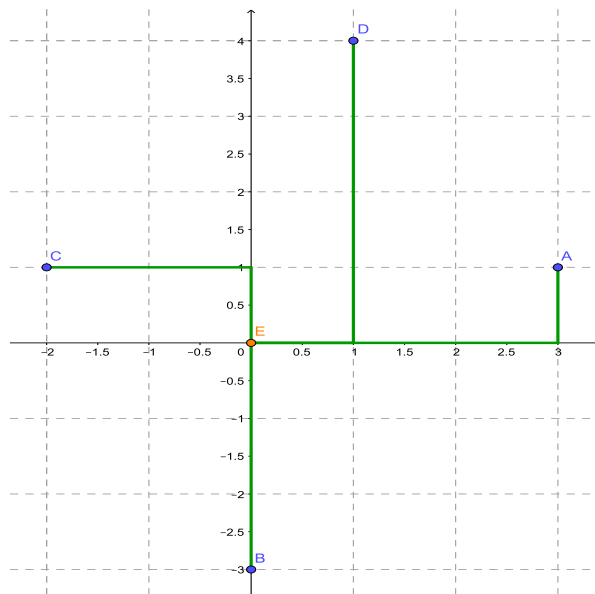
$$x_{ij}, y_j, z_i \geq 0 \quad \text{pro} \quad \forall i \in \{1, \dots, 5\}, \forall j \in \{1, 2, 3\}.$$

Prodejce v této úloze zajímá primárně maximalizace zisku, nejsou zde tedy žádné podmínky na maximální rozpočet, který chce využít. Vzhledem k omezené dostupnosti všech druhů surových paliv však není úloha neomezená. Ekonomické úlohy jsou ve většině případů formulované v maximalizačním tvaru, s minimalizací se v ekonomii setkáme např. při hledání minima nákladové funkce.

△

Příklad 1.3. Úloha s absolutní hodnotou.

V tomto příkladu uvažujeme továrnu, ve které jsou umístěny čtyři stroje. Jejich souřadnice jsou $[3, 1]$, $[0, -3]$, $[-2, 1]$ a $[1, 4]$. Vedení továrny zakoupilo nový stroj, který chce umístit tak, aby byl součet vzdáleností nového stroje od všech ostatních strojů minimální. Uvažujeme součtovou (Manhattanskou) vzdálenost, ve které je vzdálenost bodu $[x_1, x_2]$ od bodu $[1, 3]$ rovna $|x_1 - 1| + |x_2 - 3|$. Pro bod $E = [0, 0]$ jsou vzdálenosti od jednotlivých strojů znázorněny na Obrázku 1.1. Naformulujte problém, kterému vedení čelí, ve formě lineárních rovnic a nerovnic.



Obrázek 1.1: Nákres Příkladu 1.3

Řešení. Označme x_1, x_2 souřadnice nového stroje. Ze zadání víme, že chceme minimalizovat součet vzdáleností tohoto stroje od ostatních strojů. Funkce, kterou chceme minimalizovat, je tedy

$$f(x_1, x_2) = |x_1 - 3| + |x_2 - 1| + |x_1| + |x_2 + 3| + |x_1 + 2| + |x_2 - 1| + |x_1 - 1| + |x_2 - 4|.$$

Nyní je potřeba převést tuto funkci na funkci lineární. Označme si pro přehlednost y_1, y_3, y_5 a y_7 horizontální vzdálenosti nového stroje od původních strojů. Dále označme y_2, y_4, y_6 a y_8 příslušné vertikální vzdálenosti. Funkci, kterou chceme minimalizovat, tedy můžeme nyní zapsat jako

$$f = \sum_{i=1}^8 y_i.$$

Všechna y_i vyjadřují vzdálenosti, a jsou tedy nezáporné. V úloze musíme doplnit podmínky $y_i \geq 0, i \in \{1, \dots, 8\}$. Nyní už chybí jen propojení původních proměnných x_i s novými proměnnými y_i , které si nyní odvodíme pro proměnnou y_1 . Pro absolutní hodnotu platí $|x| = \max\{x, -x\}$. Pro novou proměnnou y_1 platí, že

$$y_1 = \begin{cases} x_1 - 3, & \text{pro } x_1 \geq 3, \\ 3 - x_1, & \text{pro } x_1 < 3. \end{cases}$$

Pokud tedy k úloze přidáme podmínky $y_1 \geq x_1 - 3$ a $y_1 \geq 3 - x_1$, zajistíme, že

$$y_1 = \max \{x_1 - 3, 3 - x_1\} = |x_1 - 3|.$$

Pro y_2 tedy musíme přidat podmínky $y_2 \geq x_2 - 1$ a $y_2 \geq 1 - x_2$, pro ostatní proměnné postupujeme obdobně. Úloha, kterou při umisťování nového stroje řeší vedoucí továrny, je

$$\begin{aligned} \min \quad & \sum_{i=1}^8 y_i \\ \text{za podmínek} \quad & y_1 \geq x_1 - 3, y_1 \geq 3 - x_1, y_2 \geq x_2 - 1, y_2 \geq 1 - x_2, \\ & y_3 \geq x_1, y_3 \geq -x_1, y_4 \geq x_2 + 3, y_4 \geq -3 - x_2, \\ & y_5 \geq x_1 + 2, y_5 \geq -2 - x_1, y_6 \geq x_2 - 1, y_6 \geq 1 - x_2, \\ & y_7 \geq x_1 - 1, y_7 \geq 1 - x_1, y_8 \geq x_2 - 4, y_8 \geq 4 - x_2, \\ & y_i \geq 0. \end{aligned}$$

△

Příklad 1.4. Přiřazovací úloha – úloha s binárními proměnnými.

Představme si čtyřčlennou rodinu, ve které jsou obě děti minimálně patnáctileté. Rodinu čeká jarní úklid celého domu, který si chce pořádně naplánovat. Na úklidu se budou podílet všichni. Činnosti si rodina rozdělila do čtyř kategorií (můžeme si je představit např. jako umytí oken, umytí podlah, úklid komory a přichystání jarní výzdoby). Každý z rodiny je jinak efektivní při vykonávání jednotlivých činností – efektivita se zde bude měřit časem, jinak předpokládáme, že všichni jsou schopni úkol splnit stejně dobře. Úkolem je přiřadit každému členovi rodiny právě jednu úklidovou činnost tak, aby rodina minimalizovala čas, který uklízením stráví. Čas, který jednotlivým členům zaberou jednotlivé úkony, je zapsán v Tabulce 1.2.

člen rodiny \ činnost	činnost			
	1	2	3	4
1	110	55	90	65
2	50	75	100	80
3	50	110	120	95
4	70	150	95	130

Tabulka 1.2: Úklidový čas (v minutách) pro členy domácnosti

Řešení. Rodina, která chce minimalizovat celkový čas úklidu, vlastně stojí před problémem, jenž nyní popíšeme matematicky. Označme c_{ij} čas, který i -tému členu domácnosti trvá provést j -tou činnost. Proměnná c_{23} např. vyjadřuje, kolik času trvá 2. členu rodiny úklidová činnost číslo 3, v tabulce vidíme, že $c_{23} = 100$. Dále si označme x_{ij} proměnnou, která vyjadřuje, zda i -tý člověk vykonává j -tou činnost či ne. Všechny proměnné x_{ij} , $i, j \in \{1, 2, 3, 4\}$ vyjadřují přiřazení člověka a činnosti a nabývají pouze hodnot $\{0, 1\}$.

Z tohoto důvodu se podobné úlohy nazývají též úlohami s binárními proměnnými. Funkce, kterou chce rodina minimalizovat, je ve tvaru

$$110x_{11} + 50x_{21} + 50x_{31} + 70x_{41} + 55x_{12} + 75x_{22} + 110x_{32} + 150x_{42} + \\ + 90x_{13} + 100x_{23} + 120x_{33} + 95x_{43} + 65x_{14} + 80x_{24} + 95x_{34} + 130x_{44}.$$

Podle zadání musí každý člen vykonávat právě jednu práci. Tuto skutečnost zajistíme využitím podmínek

$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &= 1, \\ x_{21} + x_{22} + x_{23} + x_{24} &= 1, \\ x_{31} + x_{32} + x_{33} + x_{34} &= 1, \\ x_{41} + x_{42} + x_{43} + x_{44} &= 1, \end{aligned}$$

kteří říkají, že součet proměnných, které udávají přiřazení člověka k určité činnosti, je roven jedné.

Podobná omezení naformulujeme pro jednotlivé činnosti, každá činnost musí být vykonána právě jednou, což zajistíme s využitím podmínek

$$\begin{aligned} x_{11} + x_{21} + x_{31} + x_{41} &= 1, \\ x_{12} + x_{22} + x_{32} + x_{42} &= 1, \\ x_{13} + x_{23} + x_{33} + x_{43} &= 1, \\ x_{14} + x_{24} + x_{34} + x_{44} &= 1. \end{aligned}$$

Při formulaci problému jsme využili podmínky bivalence, tj. $x_{ij} \in \{0, 1\}$. Úloha tedy není, na rozdíl od předchozích úloh, tvořena pouze lineární funkcí omezenou afinními funkcemi. Vzhledem k vlastnostem úlohy je však možné ji vyřešit metodami pro úlohy lineárního programování. Když při formulaci úlohy nahradíme podmínky bivalence podmínkami $0 \leq x_{ij} \leq 1$ pro všechny proměnné, můžeme úlohu vyřešit jako problém lineárního programování. Hodnoty proměnných v řešení, které tímto způsobem získáme, jsou však binární, což byl náš požadavek. To vyplývá ze struktury dané úlohy, která je speciálním typem tzv. dopravní úlohy, se kterou se seznámíme v Kapitole 6.

Úloha s binárními proměnnými je speciálním případem úlohy celočíselného programování, ve které, jak lze vyčíst z názvu, požadujeme celočíselnost některých nebo všech proměnných. Metody řešení úloh celočíselného programování jsou často náročnější než metody řešení úloh lineárního programování, pro získání přibližného řešení je tak i v případě celočíselné úlohy možné využít např. simplexovou metodu.

△

1.3 Základní pojmy lineárního programování

V předchozí části jsme si ukázali několik motivačních úloh, které jsme ze slovního zadání převedli do matematického popisu. Všechny úlohy, které jsme viděli, měly velmi podobnou strukturu. V této části nejprve definujeme pojem obecné úlohy lineárního programování, poté se podíváme na jednotlivé tvary úloh v závislosti na přítomnosti nerovnostních a rovnostních omezení.

Definice 1.1. Obecná úloha lineárního programování.

Obecnou úlohu lineárního programování definujeme jako

$$\left. \begin{aligned} \min f(x) \quad x \in X \subseteq \mathbb{R}^n, \\ f(x) = \langle c, x \rangle = c_1x_1 + \dots + c_nx_n, \\ x \in X := \{x \in \mathbb{R}^n \mid \langle a_i, x \rangle \leq b_i; i = 1, \dots, k; \\ \langle a_i, x \rangle = b_i; i = k + 1, \dots, m; \\ x_1, \dots, x_s \geq 0\}, \\ k \in \{0, 1, \dots\}, m \in \{k, \dots\}, s \in \{0, \dots, n\}, \\ c \in \mathbb{R}^n, a_i \in \mathbb{R}^n, b_i \in \mathbb{R}. \end{aligned} \right\} \quad (1.1)$$

Funkci $f(x)$ nazýváme *účelovou funkcí*, v literatuře se můžeme setkat i s dalšími názvy, např. *cílová, ztrátová, nákladová, cenová* či *užitková funkce*. Množina X se nazývá *přípustná množina*, body $x \in X$ pak *přípustné body*. Pro přípustný bod $x^* \in X$, který je řešením dané úlohy (tj. platí pro něj $\langle c, x^* \rangle \leq \langle c, x \rangle$ pro každé $x \in X$), lze v literatuře najít označení *optimum* či *optimální řešení*. V této práci však pod pojmem řešení úlohy lineárního programování rozumíme optimální řešení. Prvky matice A , tj. a_i , se v ekonomické literatuře nazývají *strukturální koeficienty*, prvky b_i pak *výrobní kapacity*. Povšimněme si, že v Definici 1.1 není v účelové funkci uvedena konstanta. Tato konstanta by totiž neměla vliv na řešení úlohy, znamenala by pouze posun, neboť $\min_{x \in X} (f + c) = \min_{x \in X} (f) + c$. Konstanta tedy ovlivní pouze hodnotu účelové funkce v řešení, ne řešení samotné.

V ekonomii se typicky setkáme s problémem najít maximum zadané účelové funkce, tyto úlohy můžeme do obecného tvaru převést s využitím skutečnosti, že

$$\max f(x) = -\min [-f(x)].$$

Stejně tak můžeme převést podmínky s opačnou nerovností do tvaru nerovností v obecném tvaru, a to vynásobením číslem -1 , např.

$$\langle a_i, x \rangle \geq b_i \text{ převedeme na } -\langle a_i, x \rangle \leq -b_i.$$

Vzhledem k tomu, že obecná úloha (1.1) je ve tvaru neostrých nerovností a rovností, může dojít k několika významným situacím. Pokud máme úlohu s větším počtem rovnostních omezení takových, že žádné není nadbytečné (tj. nepřináší nám žádnou novou informaci), než počtem proměnných, přípustná množina je jistě prázdná. Nadbytečné omezení si můžeme představit např. jako omezení, které je již zahrnuté v jiném omezení, nebo omezení, které vznikne lineární kombinací jiných. Počet nerovnostních omezení není ve všeobecnosti nijak závislý na počtu proměnných (n), i v tomto případě se však může stát, že přípustná množina bude prázdná. Nadbytečná omezení, jak uvidíme dále, mohou vést k degenerovanosti úlohy. Teoreticky by tedy bylo vhodné si je z úlohy odstranit, prakticky tak však nepostupujeme, neboť se nejedná o jednoduchý úkol a samotná degenerovanost úlohy nemusí vést k žádným problémům, což blíže vysvětlíme v následující kapitole.

Pro úlohu lineárního programování (příp. pro množinu X) mohou v závislosti na počtu rovnostních a nerovnostních omezení a omezení na znaménko nastat zejména čtyři situace:

1. pouze nerovnostní omezení, bez omezení na znaménka (tj. $k = m, s = 0$)
V tomto případě nazýváme úlohu (1.1) úlohou v *základním tvaru*.
2. pouze rovnostní omezení, všechny proměnné omezené na znaménka (tj. $k = 0, s = n$)
V tomto případě nazýváme úlohu (1.1) úlohou v *kanonickém tvaru*.
3. pouze nerovnostní omezení, všechny proměnné omezené na znaménka (tj. $k = m, s = n$)
V tomto případě nazýváme úlohu (1.1) úlohou ve *standardním tvaru*.
4. pouze rovnostní omezení, bez omezení na znaménka (tj. $k = 0, s = 0$)
Soustavu v tomto případě můžeme zapsat jako $Ax = b$ právě tehdy, když $x \in X$.
V případě, že jsou všechna omezení lineárně nezávislá, se tedy jedná o hledání volného extrému v \mathbb{R}^{n-m} , jež není předmětem této práce.

Poznámka 1.2. Terminologie při označování různých typů úloh není zcela ustálená, v literatuře se lze setkat s nazváním úlohy v kanonickém tvaru (podle vlastností popsaných výše) úlohou ve standardním tvaru. Námi definovanou úlohu v kanonickém tvaru lze též najít pod názvem úloha v *rovníkovém tvaru* či úloha ve *tvaru rovností*, standardní tvar pak pod označením *tvar nerovností*, viz např. [47].

V této práci se budeme např. při odvozování a využívání simplexové metody věnovat úlohám v kanonickém tvaru. V Kapitole 4 si pomocí primární a duální úlohy ukážeme, jaký je vztah kanonického a základního tvaru. Abychom se v dalších částech práce mohli zaměřit výhradně na úlohy v kanonickém tvaru, ukážeme si nyní, jak mezi sebou úlohy v různých tvarech převádět. Naším běžným cílem bude získat úlohy v kanonickém tvaru, které budou obsahovat co nejmenší počet proměnných, což je užitečné zejména při ručních výpočtech.

1.4 Převody mezi různými tvary úlohy LP

1. Převod podmínek ve tvaru nerovností do tvaru rovností provedeme přidáním tzv. *doplňkové (přídavné, skluzové)* proměnné. Nerovnost $g(x) \leq 0$ tedy převedeme na $g(x) + s = 0$, kde $s \in \mathbb{R}, s \geq 0$. Přidáním doplňkové proměnné se nezmění tvar účelové funkce $f(x)$ (resp. účelová funkce $f(x)$ je nyní funkcí více proměnných, ale přídavná proměnná se v ní vyskytuje s nulovým koeficientem).
2. Podmínky ve tvaru rovností převedeme na soustavu 2 nerovností, tj. rovnost $g(x) = 0$ převedeme na $g(x) \leq 0$ a $-g(x) \leq 0$.
3. K získání úlohy v kanonickém tvaru potřebujeme omezit všechny proměnné na znaménko. Pro proměnné, které nejsou takto omezené, tedy omezení přidáme, a to následujícím způsobem:

$$x \in \mathbb{R} : \quad x = y - z, \quad \text{kde } y, z \geq 0.$$

Touto úpravou již změníme tvar účelové funkce.

4. V případě, že je proměnná omezena způsobem $a \leq x \leq b$ (resp. $a \leq x$), převedeme nerovnost $a \leq x$ na rovnost $x - y = a$, kde $y \geq 0$, a podmínku $x \leq b$ obdobně převedeme na $x + z = b$, kde $z \geq 0$. Tímto způsobem získáme nové rovnostní omezení $y + a = b - z$, ve kterém vystupují dvě doplňkové proměnné omezené na znaménko.

Těmito způsoby jsme schopni mezi sebou převádět úlohy lineárního programování v jednotlivých tvarech. Tyto transformace neovlivňují optimální řešení daných úloh. Přesněji řečeno, vyřešením transformované úlohy získáme řešení, které se liší od řešení původní úlohy, ale ze kterého již řešení původní úlohy umíme získat.

Poznámka 1.3. Při omezování proměnných na znaménko (bod 3 výše) jsme si k jedné proměnné, která nebyla původně omezená na znaménko, vytvořili dvě nové proměnné omezené na znaménko. Úloha se tedy zvětšila o jednu proměnnou. V praxi nezávisí na tom, kolik omezení na znaménko potřebujeme přidat. Vždy si vystačíme s přidáním pouze jedné nové proměnné, neboť všechna omezení můžeme dodat pomocí této jedné proměnné. Potřebujeme-li např. omezit na znaménko proměnné x_1, x_2, x_3 , můžeme využít rovnosti

$$x_1 = a - z,$$

$$x_2 = b - z,$$

$$x_3 = c - z,$$

kde $a, b, c, z \geq 0$. Nyní si vysvětlíme, proč tyto rovnosti můžeme využít. Uvažme např. zápis $x_i = y_i - z_i$ a definujme $z := \max \{z_i\} \geq 0$. Proměnnou x_i si můžeme rozepsat jako

$$x_i = (y_i - z_i + z) - z = a_i - z,$$

výraz v závorce jsme si označili a_i . Protože $y_i \geq 0$ a $z - z_i \geq 0$, určitě platí $a_i \geq 0$. Pro proměnnou x_i můžeme psát $x_i = a_i - z$, kde $a_i, z \geq 0$. Konkrétní hodnoty a_i a z nemají vliv na tvar účelové funkce, neboť v ní vystupuje pouze jejich rozdíl. S využitím přidání pouze jedné proměnné tedy můžeme omezit na znaménko libovolný počet proměnných.

Poznatky uvedené výše si nyní ukážeme na konkrétním příkladu.

Příklad 1.5. Převedení úlohy na kanonický tvar.

Převedte následující úlohu do kanonického tvaru.

$$\begin{aligned} \min \quad & 5x_1 + 6x_2 + 3x_3 - x_4 \\ \text{za podmínek} \quad & x_1 - 3x_3 = 2 \\ & 3x_1 + x_2 \leq 6 \\ & x_2 - 5x_3 + 2x_4 \geq 2 \\ & -1 \leq x_4 \leq 12 \\ & x_1 \geq 0, x_3 \geq 0 \end{aligned}$$

Řešení. Nejprve se zaměříme na proměnnou x_2 , která není omezená na znaménko. Omezíme ji tedy rozepsáním

$$x_2 = a - z, \text{ kde } a \geq 0, z \geq 0,$$

čímž získáme úlohu ve tvaru

$$\begin{aligned} \min \quad & 5x_1 + 6a - 6z + 3x_3 - x_4 \\ \text{za podmíněk} \quad & x_1 - 3x_3 = 2 \\ & 3x_1 + a - z \leq 6 \\ & a - z - 5x_3 + 2x_4 \geq 2 \\ & -1 \leq x_4 \leq 12 \\ & x_1 \geq 0, x_3 \geq 0, a \geq 0, z \geq 0. \end{aligned}$$

Nyní si nerovnosti pro x_4 přepíšeme do tvaru rovností, tj.

$$-1 + b = x_4, \quad x_4 + c = 12, \quad b \geq 0, c \geq 0.$$

Proměnnou x_4 tedy můžeme vyjádřit jako

$$x_4 = -1 + b = 12 - c.$$

V účelové funkci i v omezeních za x_4 dosadíme např. $b - 1$, v úloze se tedy změní původní nerovnostní omezení na proměnné omezené na znaménko a rovnostní omezení (z předchozí rovnosti), které zapíšeme např. ve tvaru

$$b + c = 13.$$

Úloha je tedy nyní ve tvaru

$$\begin{aligned} \min \quad & 5x_1 + 6a - 6z + 3x_3 - b + 1 \\ \text{za podmíněk} \quad & x_1 - 3x_3 = 2 \\ & 3x_1 + a - z \leq 6 \\ & a - z - 5x_3 + 2b \geq 4 \\ & b + c = 13 \\ & x_1 \geq 0, x_3 \geq 0, a \geq 0, b \geq 0, c \geq 0, z \geq 0. \end{aligned}$$

V této úloze již máme všechny proměnné omezené na znaménko. Nerovnostní podmínky převedeme do tvaru rovností přidáním doplňkový proměnných (jež v této práci vždy budeme značit písmenem s), např. takto:

$$\begin{aligned} 3x_1 + a - z + s_1 &= 6, \quad s_1 \geq 0, \\ a - z - 5x_3 + 2b - s_2 &= 4, \quad s_2 \geq 0. \end{aligned}$$

Lze si povšimnout, že přidáním doplňkových proměnných se formálně nezměnila účelová funkce. Ve skutečnosti jde nyní o funkci osmi proměnných oproti původním čtyřem. Výslednou úlohu v kanonickém tvaru tedy zapíšeme jako

$$\begin{aligned} \min \quad & 5x_1 + 6a - 6z + 3x_3 - b + 1 \\ \text{za podmíněk} \quad & x_1 - 3x_3 = 2 \\ & 3x_1 + a - z + s_1 = 6 \\ & a - z - 5x_3 + 2b - s_2 = 4 \\ & b + c = 13 \\ & x_1 \geq 0, x_3 \geq 0, a \geq 0, b \geq 0, c \geq 0, z \geq 0, s_1 \geq 0, s_2 \geq 0. \end{aligned}$$

△

1.5 Grafická metoda řešení

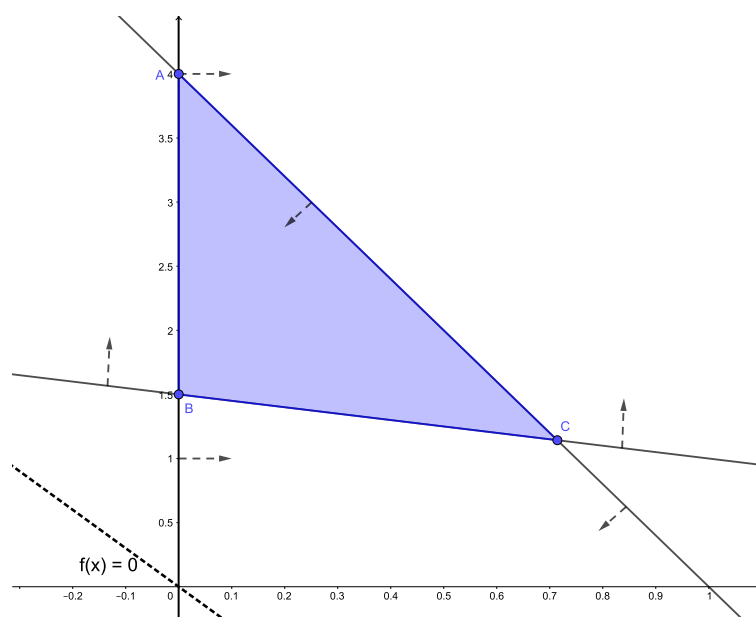
Nejjednodušším způsobem řešení úlohy lineárního programování v případě úloh s funkcemi dvou proměnných je grafická metoda řešení. Tu si nyní ukážeme na několika příkladech. V dalších částech práce již nebudeme mít možnost si úlohy a jejich výsledky znázornit, grafickou metodu tedy použijeme k ilustraci různých situací, které mohou při řešení úlohy lineárního programování v \mathbb{R}^2 nastat.

Příklad 1.6. Grafická metoda řešení 1.

Grafickou metodou najděte řešení úlohy

$$\begin{aligned} \min \quad & 3x_1 + x_2 \\ \text{za podmínek} \quad & 4x_1 + x_2 \leq 4 \\ & x_1 + 2x_2 \geq 3 \\ & x_1 \geq 0. \end{aligned}$$

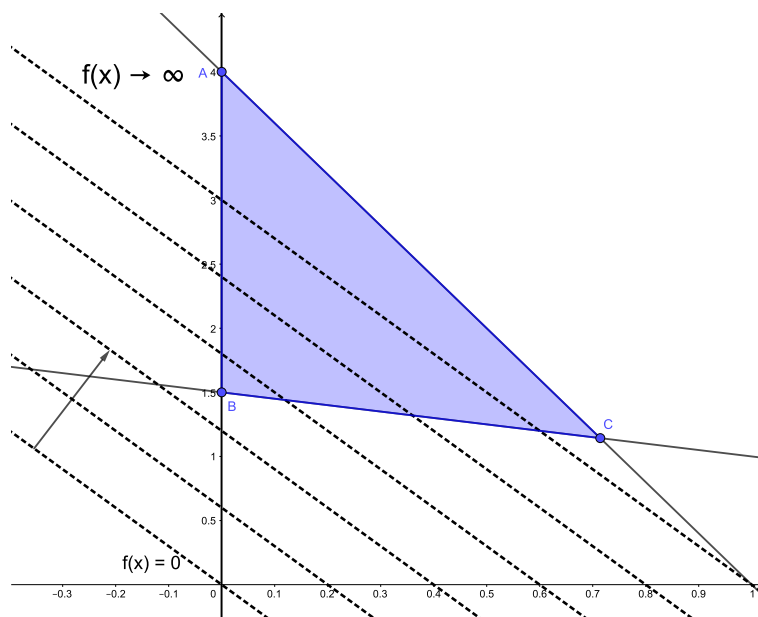
Řešení. Jako první si vyznačíme přípustnou množinu, která vznikne splněním všech tří podmínek. Dále si nakreslíme vrstevnici účelové funkce, která prochází počátkem souřadnicového systému. Přípustná množina (trojúhelník ABC) i daná vrstevnice (vyznačená přerušovaně) jsou zobrazeny na Obrázku 1.2.



Obrázek 1.2: Přípustná množina X a vrstevnice účelové funkce $f(x) = 0$

Na obrázku je šipkami vyznačeno, jakou část roviny vymezují omezení ze zadání. V úloze hledáme vrstevnici účelové funkce s nejnižší možnou hodnotou tak, aby byla splněna všechna omezení. Při řešení si tedy musíme rozmyslet, jakým směrem se hodnota účelové funkce zmenšuje. Pro ilustraci zde uvádíme následující obrázek, na kterém je

několik vrstevnic účelové funkce. Směrem doprava nahoru hodnota účelové funkce roste, hledáme tedy první bod, který splní všechna omezení a kterého se dotkne vrstevnice s nejmenší hodnotou. Z Obrázku 1.3 je možné vidět, že hledaným minimalizujícím bodem je bod označený písmenem B, který leží v průsečíku omezení $x_1 + 2x_2 = 3$ a $x_1 = 0$. Hodnota účelové funkce v bodě B o souřadnicích $[0, 1,5]$ je rovna 1,5.



Obrázek 1.3: Přípustná množina X a vrstevnice účelové funkce

△

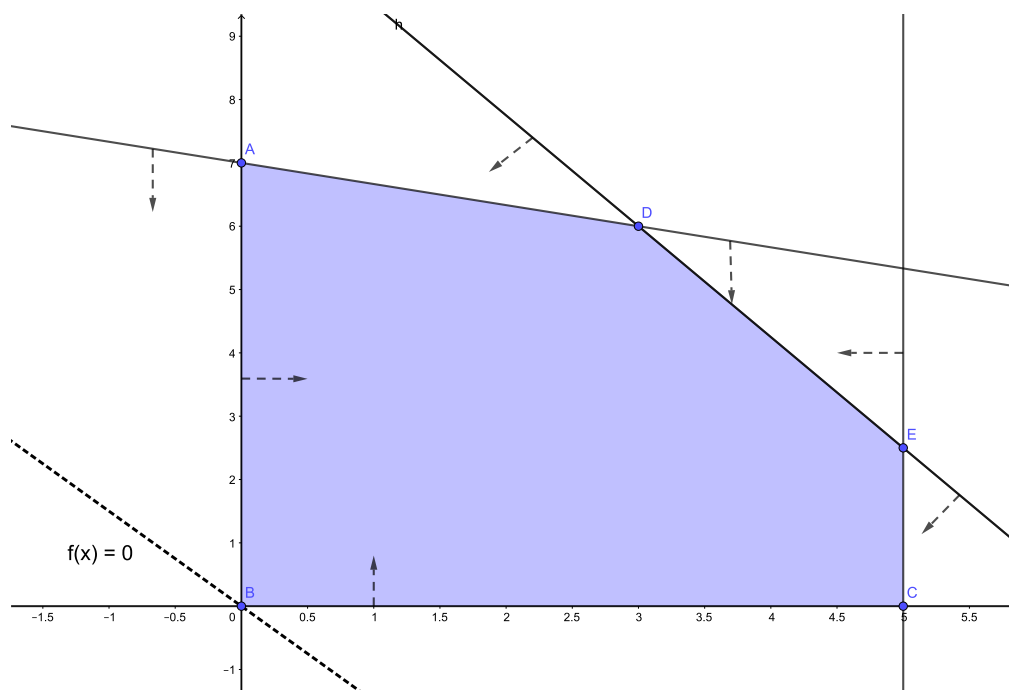
Ve všech následujících příkladech na grafickou metodu bude šipka u vrstevnic znamenat směr, kterým se zlepšuje účelová funkce (tj. jde k mínus nekonečnu pro minimalizaci a k nekonečnu pro maximalizaci).

Příklad 1.7. Grafická metoda řešení 2.

Grafickou metodou najděte řešení úlohy

$$\begin{aligned} \max \quad & 3x_1 + 2x_2 \\ \text{za podmíněk} \quad & x_1 + 3x_2 \leq 21 \\ & \frac{7}{2}x_1 + 2x_2 \leq \frac{45}{2} \\ & x_1 \leq 5 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Řešení. Stejně jako v předchozím příkladu si nejprve znázorníme přípustnou množinu a nulovou vrstevnici účelové funkce, výsledek můžeme vidět na Obrázku 1.4. Přípustná množina je v tomto příkladu pětiúhelník. Na rozdíl od předchozího příkladu nyní hledáme


 Obrázek 1.4: Přípustná množina X a vrstevnice účelové funkce $f(x) = 0$

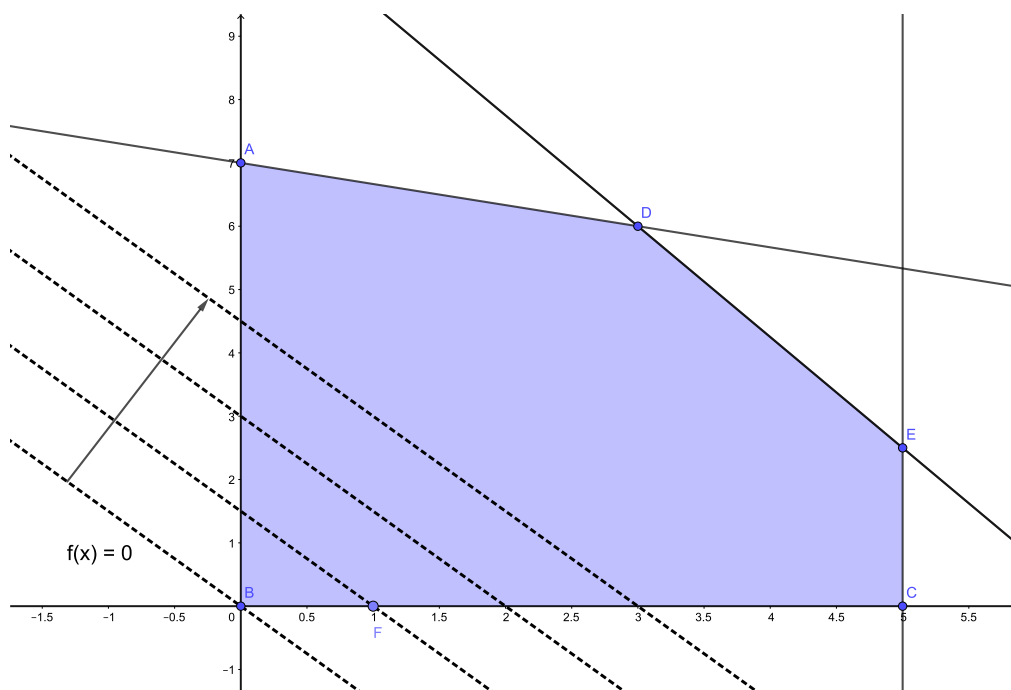
maximum účelové funkce, graficky tedy bod, kterého se dotkne vrstevnice s nejvyšší možnou hodnotou.

Stejně jako v minulém příkladu si do obrázku načrtneme další vrstevnice účelové funkce, např. jako na Obrázku 1.5. Hodnoty vrstevnice se směrem doprava nahoru zvětšují, např. bodem $[1,0]$ (na Obrázku 1.5 označeným písmenem F) prochází vrstevnice o hodnotě 3. Na rozdíl od minulého příkladu však není na první pohled zřejmé, jaký bod je hledaným optimem. Jednou z možností, jak nyní postupovat, je určit hodnotu účelové funkce v kandidátech na optimum. Body, které připadají v úvahu jsou nyní D a E. Jejich souřadnice získáme využitím nerovností, na jejichž průsečících body leží. Např. bod označený D vznikl jako průsečík nerovností (1) a (2) ze zadání, platí pro něj tedy:

$$x_1 + x_2 = 21 \quad \text{a} \quad \frac{7}{2}x_1 + 2x_2 = \frac{45}{2}.$$

Vyřešením této soustavy získáme souřadnice bodu $D = [3,6]$. Pro bod E pak obdobně dostaneme $E = [5,5/2]$. Porovnáním funkčních hodnot v těchto bodech pak najdeme optimální řešení, kterým je v tomto případě bod D, neboť $f(D) = 21 > f(E) = 20$.

Další způsob, kterým lze určit, který z bodů D a E je hledaným maximum, využívá sklon vrstevnice funkce a přímky omezení, na kterém oba tyto body leží. Sklon účelové funkce je roven $-3/2$, sklon druhého omezení je $-7/4$. Druhé omezení, na kterém leží body D a E, má v absolutní hodnotě větší sklon, než účelová funkce. Omezení je tedy strmější. Pokud budeme přidávat další vrstevnice účelové funkce, poslední bod, kterého se dotkeme pro největší hodnotu vrstevnice, a tedy hledané maximum, bude bod D. Bod E protne vrstevnice s nižší hodnotou. Pokud by byl sklon účelové funkce i omezení stejný, řešením by mohla být i celá úsečka DE.

Obrázek 1.5: Přípustná množina X a vrstevnice účelové funkce

△

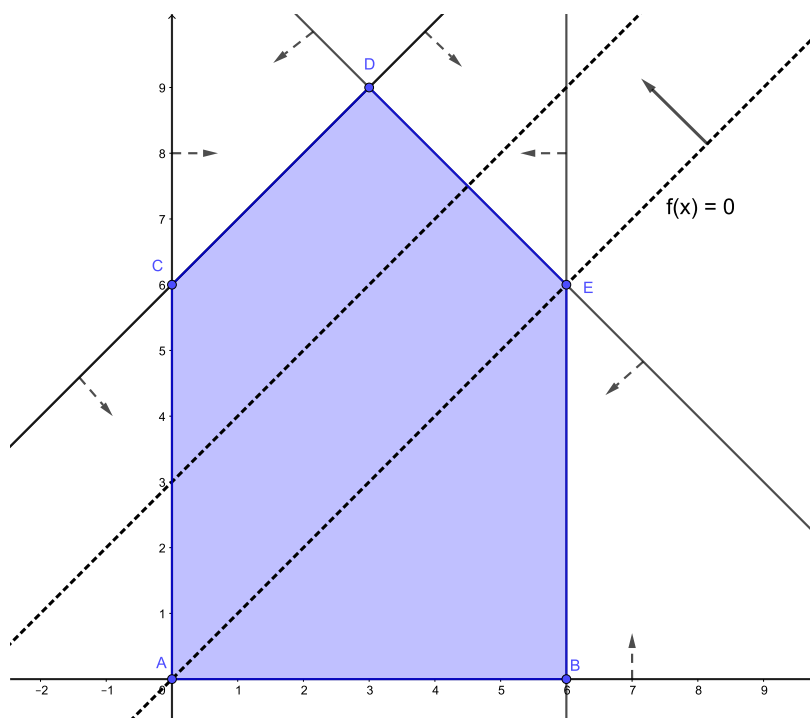
Příklad 1.8. Grafická metoda řešení 3.

Grafickou metodou najděte řešení úlohy

$$\begin{array}{ll}
 \min & x_1 - x_2 \\
 \text{za podmíněk} & x_1 + x_2 \leq 12 \\
 & -x_1 + x_2 \leq 6 \\
 & x_1 \leq 6 \\
 & x_1, x_2 \geq 0.
 \end{array}$$

Řešení. Opět začneme s načrtnutím přípustné množiny. Ta je spolu s vrstevnicemi na úrovni $f(x) = 0$ a $f(x) = -3$ znázorněna na Obrázku 1.6. Nejprve si rozmyslíme, jakým směrem se bude hodnota účelové funkce zmenšovat. Vzhledem k tomu, že hledáme minimum $x_1 - x_2$, klesá hodnota účelové funkce směrem doleva nahoru, což je na Obrázku 1.6 znázorněno plnou šipkou. Kandidáti na řešení jsou tedy body C a D.

Již z tohoto náčrtku si můžeme všimnout, že vrstevnice účelové funkce je rovnoběžná s omezením $-x_1 + x_2 \leq 6$, na kterém body C a D leží. Pokud nedůvěřujeme náčrtku, můžeme si rovnoběžnost vektorů ověřit výpočtem. Vektory $(1, -1)$ a $(-1, 1)$ jsou jistě rovnoběžné. Vzhledem k úvahám, jež jsme zde provedli, již můžeme uvést řešení, kterým je celá úsečka CD. Hodnota minima je -6 , k čemuž snadno dojdeme s využitím příslušného rovnoběžného omezení.


 Obrázek 1.6: Přípustná množina X a vrstevnice účelové funkce

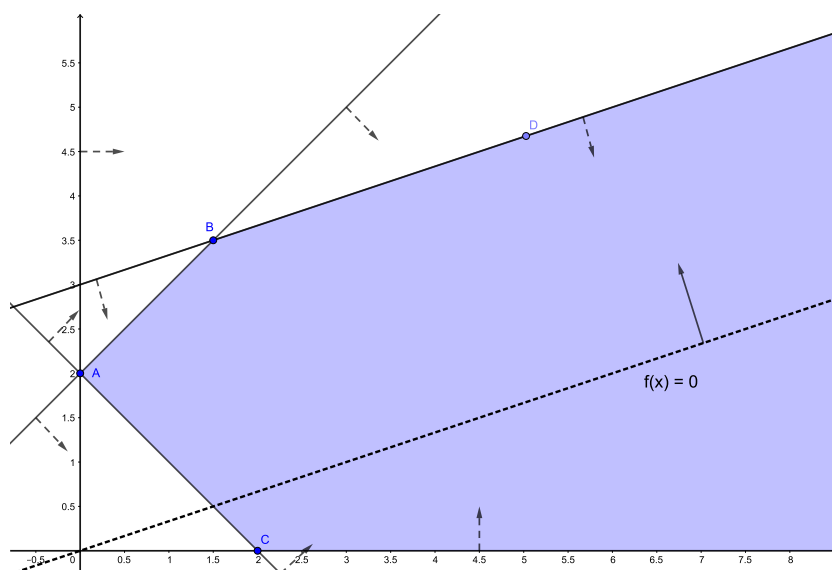
△

Příklad 1.9. Grafická metoda řešení 4.

Grafickou metodou najděte řešení úlohy

$$\begin{aligned}
 \min \quad & x_1 - 3x_2 \\
 \text{za podmínek} \quad & x_1 - x_2 \geq 2 \\
 & x_1 + x_2 \geq 2 \\
 & -\frac{x_1}{3} + x_2 \leq 3 \\
 & x_1, x_2 \geq 0.
 \end{aligned}$$

Řešení. Stejně jako v předchozích příkladech začneme s načrtnutím přípustné množiny. Ta je spolu s vrstevnicí na úrovni $f(x) = 0$ znázorněna na Obrázku 1.7. Dále si rozmyslíme, jakým směrem se bude hodnota účelové funkce zmenšovat. Účelová funkce nyní klesá směrem doleva nahoru, na obrázku je tento fakt znázorněn plnou šipkou. Stejně jako v Příkladu 1.8 je jedno z omezení rovnoběžné s vrstevnicí účelové funkce. Jde o omezení $-\frac{x_1}{3} + x_2 \leq 3$. Vzhledem k tomu, že hledáme minimum a hodnota účelové funkce klesá vyznačeným směrem, má úloha nekonečně mnoho řešení určených polopřímkou BD .

Obrázek 1.7: Přípustná množina X a vrstevnice účelové funkce $f(x) = 0$

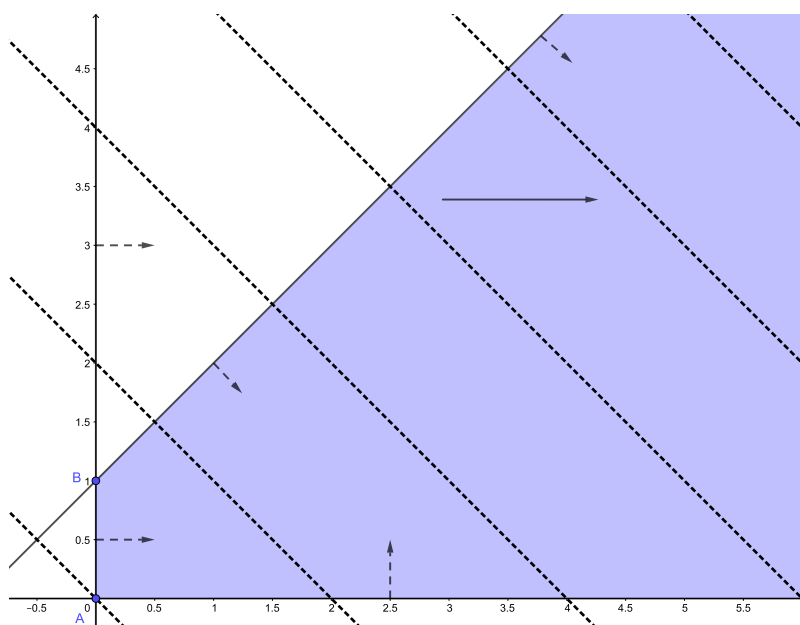
△

Příklad 1.10. Grafická metoda řešení 5.

Grafickou metodou najděte řešení úlohy

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{za podmíněk} \quad & x_1 - x_2 \geq -1 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Řešení. Náčrtkem přípustné množiny a vrstevnic účelové funkce získáme Obrázek 1.8.

Obrázek 1.8: Přípustná množina X a vrstevnice účelové funkce

Šipka znázorňuje směr, kterým se účelová funkce zlepšuje. Ihned vidíme, že účelová funkce je na přípustné množině neohraničená, tj. nemá na množině X minimum. Tato úloha tedy nemá řešení. Situaci, ve které je účelová funkce na přípustné množině neohraničená, označujeme též jako neohraničenou úlohu.

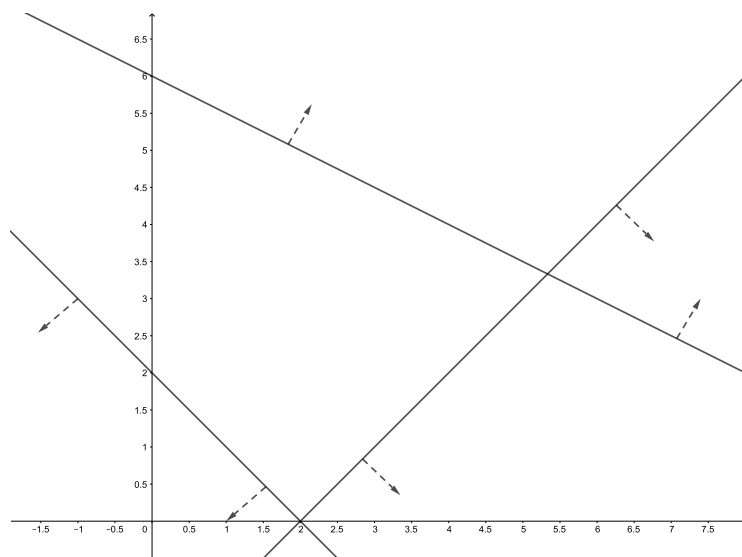
△

Příklad 1.11. Grafická metoda řešení 6.

Grafickou metodou najděte řešení úlohy

$$\begin{array}{ll} \min & 2x_1 - x_2 \\ \text{za podmínek} & \frac{x_1}{2} + x_2 \geq 6 \\ & x_1 - x_2 \geq 2 \\ & x_1 + x_2 \leq 2. \end{array}$$

Řešení. Pokud si v tomto případě načrtne omezení ze zadání, zjistíme, že jejich průnik je prázdný. To je možné vidět na Obrázku 1.9. Situaci, ve které je přípustná množina prázdná, nazveme též nepřipustnou úlohou. Úloha v tomto případě nemá řešení.



Obrázek 1.9: Prázdná přípustná množina X

△

Ve všech předchozích úlohách na grafickou metodu, ve kterých existovalo řešení, vyšla hledaná optima ve vrcholech mnohoúhelníků vymezených danými podmínkami, případně na celé úsečce mezi dvěma vrcholy či na polopřímce. Je tomu tak ale vždy? Než na tuto otázku odpovíme, definujeme několik základních pojmů.

Kapitola 2

Teoretické základy lineárního programování

Lineární programování je, jak již víme, disciplína, která řeší úlohy na vázané extrémní funkce více proměnných, ve kterých je účelová funkce i omezení ve tvaru lineárních rovností a nerovností. V této kapitole se seznámíme se základními teoretickými pojmy a vyslovíme důležitá tvrzení o řešitelnosti úloh lineárního programování. Vystavění teorie v případě lineárního programování lze v různých zdrojích najít různě (např. v [22] a [32]). My zde využijeme postup, který odpovídá např. [26] a [47]. Začneme s některými důležitými vlastnostmi přípustné množiny.

Definice 2.1. Konvexní množina.

Nechť $X \subset \mathbb{R}^n$. Množina X se nazývá *konvexní*, jestliže pro všechna $x_1, x_2 \in X$ a pro každé $\lambda \in [0, 1]$ platí

$$\lambda x_1 + (1 - \lambda)x_2 \in X.$$

Definice 2.2. Konvexní kombinace.

Nechť $x_1, \dots, x_m \in \mathbb{R}^n$. Lineární kombinaci $\lambda_1 x_1 + \dots + \lambda_m x_m$ nazýváme *konvexní*, jestliže $\lambda_1, \dots, \lambda_m \geq 0$ a $\sum_{i=1}^m \lambda_i = 1$.

Definice 2.3. Krajní (extrémní) bod.

Nechť $X \subset \mathbb{R}^n$. Bod $x \in X$ se nazývá *krajním (extrémním) bodem* množiny X , jestliže jej nelze vyjádřit ve tvaru netriviální konvexní kombinace jiných bodů množiny X .

Definice 2.4. Polyedr a polytop.

Poloprostor je část prostoru, který vznikne rozdělením prostoru jednou rovinou. *Polyedr* (v \mathbb{R}^n) je konečný průnik poloprostorů. *Polytopem* nazýváme ohraničený polyedr.

Poznámka 2.5. Množina X zdefinovaná v (1.1) je tedy nutně polyedr, neboť je omezená konečným počtem lineárních rovností a nerovností. Polyedr je z definice konvexní, neboť průnik konvexních množin je konvexní množinou.

Definice 2.6. Konvexní obal.

Nechť $X \subseteq \mathbb{R}^n$. Průnik všech konvexních množin obsahujících množinu X se nazývá *konvexní obal* množiny X a značí se $\text{conv}X$.

Nyní již můžeme vyslovit větu, jež v literatuře najdeme pod označením *základní (fundamentální) věta konvexního programování*.

Věta 2.7. Základní věta konvexního programování.

Máme-li konvexní funkci $f : X \rightarrow \mathbb{R}$ na polytopu $X := \text{conv}\{x_1, \dots, x_m\} \subseteq \mathbb{R}^n$, pak je maxima funkce f na X dosaženo v některém z bodů x_1, \dots, x_m , případně ve více z nich (tj. na celé hraně).

Důkaz. Důkaz je založen na Jensenově nerovnosti. Lze ho najít např. v [4]. □

Obecněji můžeme předchozí větu formulovat následujícím způsobem.

Věta 2.8. *Je-li X konvexní a kompaktní množina, pak maximum nastává v extrémním bodě.*

Důkaz. Důkaz lze najít např. v [26]. □

Důsledkem výše uvedených vět je základní věta lineárního programování. Její formulace se v různých zdrojích liší (např. [22], [47]), její hlavní smysl je ovšem vždy zachován.

Důsledek 2.9. Základní věta lineárního programování.

Máme-li afinní funkci $f : X \rightarrow \mathbb{R}$ na polytopu $X := \text{conv}\{x_1, \dots, x_m\} \subseteq \mathbb{R}^n$, pak globální minimum nastává v některém z bodů x_1, \dots, x_m (tj. ve vrcholu polytopu).

Důkaz. Afinní funkce je konvexní i konkávní zároveň. Při hledání minima funkce na polytopu můžeme využít faktu, že $\min f = -\max -f$. Funkce $-f$ je také afinní. Platí tedy předpoklady Věty 2.7 a funkce extrému nabývá v jednom z vrcholů (případně na celé hraně) polytopu. □

Nyní vyslovíme jednu z nejdůležitějších vět – větu o řešitelnosti úlohy lineárního programování. Z této věty totiž plynou situace, které mohou při řešení úlohy lineárního programování nastat. Tyto situace si popíšeme a ilustrujeme po zformulování věty.

Věta 2.10. O řešitelnosti úlohy lineárního programování.

Je-li v úloze lineárního programování v základním tvaru, tj.

$$\langle c, x \rangle \rightarrow \min, x \in X = \{x \in \mathbb{R}^n, Ax \geq b\},$$

kde $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, množina X neprázdná a účelová funkce $f(x) := \langle c, x \rangle$ zdola ohraničená na X , pak má úloha alespoň jedno řešení.

Důkaz. Důkaz je založen na Farkasově–Minkowského větě. □

Z teorie uvedené v této kapitole vyplývá, že při řešení úlohy lineárního programování může nastat několik situací. Tyto situace si nyní pro \mathbb{R}^2 popíšeme.

1. Úloha nemá žádné řešení. Tato situace může nastat ze dvou důvodů.

- (a) Přípustná množina X je prázdná (úloha je nepřipustná). S touto situací jsme se setkali v Příkladu 1.11. Ilustrace je zobrazena na Obrázku 1.9.

- (b) Účelová funkce je na přípustné množině neohraničená (v literatuře též označované jako neohraničená úloha). Tato situace byla popsána v Příkladu 1.10, její znázornění lze nalézt na Obrázku 1.8.
2. Úloha má jedno řešení. Pro účelovou funkci $f : X \rightarrow \mathbb{R}$, kde X je polytop, je toto řešení podle základní věty lineárního programování v jednom z vrcholů přípustné množiny. Tuto situaci ilustrují Příklady 1.6 a 1.7.
3. Úloha má nekonečně (nespočetně) mnoho řešení, což může nastat ze tří důvodů.
- (a) Řešením je úsečka (tj. 2 vrcholy a strana mezi nimi). S touto situací jsme se setkali při řešení Příkladu 1.8.
- (b) Řešením je polopřímka (tj. 1 vrchol a strana). Tato situace může nastat pouze v případě neohraničené přípustné množiny, což jsme viděli na Příkladu 1.9.
- (c) Řešením je přímka. Tuto situaci si můžeme představit např. tak, že přípustná množina je poloprostor, na jehož hranici nabývá účelová funkce hledaného extrému.

Situace, které mohou nastat při řešení úlohy lineárního programování v \mathbb{R}^n , neumíme takto jednoduše popsat. Výsledným objektem může být např. rovina, polorovina nebo stěna polyedru. Jak již bylo uvedeno výše, v následujících částech práce se budeme zabývat úlohou v kanonickém tvaru, kterou si pro přehlednost nyní definujeme.

Definice 2.11. Úloha lineárního programování v kanonickém tvaru.

Úlohu lineárního programování v kanonickém tvaru zapíšeme jako

$$\left. \begin{aligned} \min f(x) \quad & x \in X \subseteq \mathbb{R}^n, \\ f(x) = \langle c, x \rangle &= c_1x_1 + \dots + c_nx_n, \\ x \in X := \{x \in \mathbb{R}^n \mid &\langle a_i, x \rangle = b_i, i = 1, \dots, m; x_1, \dots, x_n \geq 0\}, \\ a_i \in \mathbb{R}, b_i \in \mathbb{R}, c &\in \mathbb{R}^n. \end{aligned} \right\} \quad (2.1)$$

Definice 2.12. Nosič bodu a bazický přípustný bod.

Pro bod $x = [x_1, \dots, x_n] \in \mathbb{R}^n$ definujeme jeho *nosič* jako množinu

$$J(x) := \{j \in \{1, \dots, n\}; x_j > 0\}.$$

Je-li navíc bod $x \in X$, nazveme jej *bazickým přípustným bodem* úlohy (2.1), jestliže sloupce matice A (tj. a_j) jsou lineárně nezávislé pro každé $j \in J(x)$.

Věta 2.13. *Nechť $X = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ je neprázdna množina a $x \in X$. Pak jsou následující tvrzení ekvivalentní.*

1. Bod x je krajní (extrémní) bod množiny X .
2. Bod x je bazický přípustný bod.

Důkaz. Důkaz lze najít např. v [22]. □

Ekvivalenci v předchozí větě lze rozšířit o tvrzení, že bazický přípustný bod je pro úlohu (2.1) totožný s vrcholem přípustné množiny. Pod pojmem vrchol množiny rozumíme bod $\hat{x} \in X$, pro který existuje nenulový vektor $a \in \mathbb{R}^n$ takový, že $a^T \hat{x} < a^T y$ pro každé $y \in X \setminus \{\hat{x}\}$. Simplexový algoritmus, který bude odvozen v následující části práce, postupuje při hledání optimálního řešení přes bazické přípustné body, neboli přes vrcholy, které v případě úlohy (2.1) snadno získáme. Existují však bazické přípustné body vždy? Tuto otázku zodpoví následující věta.

Věta 2.14. *Množina bazických přípustných bodů množiny $X \neq \emptyset$ je neprázdná. Navíc pro libovolné $x \in X$ existuje bazický přípustný bod $\hat{x} \in X$ tak, že platí $J(\hat{x}) \subseteq J(x)$.*

Důkaz. Důkaz lze najít např. v [19]. □

K podobnému závěru o neprázdnosti množiny bazických přípustných bodů bychom mohli dojít i s využitím vlastností úlohy (2.1). V této úloze má díky omezením na znaménko a rovnostním omezením neprázdná přípustná množina vždy vrchol, a tudíž i množina bazických přípustných bodů je neprázdná. Jedná se o vlastnost přípustné množiny, která nezáleží na tvaru účelové funkce.

Nyní již víme, že pro libovolnou množinu $X \neq \emptyset$ existuje alespoň jeden bazický přípustný bod. Kolik bazických přípustných bodů (vrcholů) ale může polyedr mít? Odpověď nabízí následující věta.

Věta 2.15. *Každý polyedr má konečný počet bazických přípustných bodů.*

Důkaz. Důkaz lze najít např. v [32]. □

Počet vrcholů, neboli bazických přípustných bodů, je tedy konečný. Jak ale bazické přípustné body najít? Možný způsob řešení tohoto problému nabízí lemma, ve kterém najdeme další způsob charakterizace bazických přípustných bodů.

Lemma 2.16. *Bod $x \in X$ je bazickým přípustným bodem právě tehdy, když existuje m -prvková množina $B \subseteq \{1, \dots, n\}$ taková, že*

1. $m \times m$ matice A_B , jež vznikne z matice A ponecháním sloupců, jejichž index je v B , je regulární (tj. a_j jsou lineárně nezávislé pro $j \in B$).
2. $x_j = 0$ pro $j \notin B$.

Důsledkem tvrzení uvedených výše je nejen fakt, že množina bazických přípustných bodů je konečně-prvková, ale i její nejvyšší možný počet prvků, který je $\binom{n}{m}$. Tato množina je totiž jednoznačně určena množinami B , neboť pro každou množinu $B \subseteq \{1, \dots, n\}$ existuje nejvýše jeden bazický přípustný bod $x \in X$ takový, že a_j jsou lineárně nezávislé pro $j \in B$.

Je vhodné zde zdůraznit, že jednomu bazickému přípustnému bodu může odpovídat i více různých množin B . Tato skutečnost bude ilustrována na Příkladu 2.2.

Definice 2.17. Báze, degenerovaný bod.

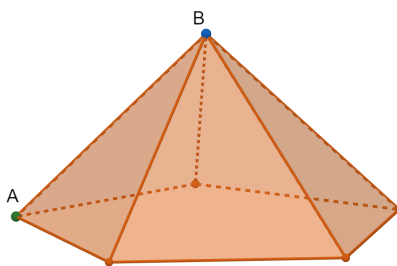
Množina B z Lemmatu 2.16 se nazývá *báze* bazického přípustného bodu $x \in X$. O prvcích s indexy $j \in B$ říkáme, že jsou v *bázi*, o prvcích s indexy $j \notin B$ říkáme, že jsou *mimo*

bázi. Bazický přípustný bod $x \in X$ je *nedegenerovaný*, jestliže $x_j > 0$ pro každé $j \in B$, v opačném případě říkáme, že bod je *degenerovaný*. Úloha lineárního programování je *nedegenerovaná*, jestliže každý bazický přípustný bod je nedegenerovaný. V opačném případě jde o *degenerovanou* úlohu.

Degenerovanost může být způsobena např. nadbytečným omezením. Pro degenerovanou úlohu je typické, že nám různé báze dají stejný bazický přípustný bod. Pro nedegenerovaný bod platí rovnost $|J(x)| = |B|$, tj. množiny mají stejný počet prvků, dokonce $J(x) = B$. Degenerovanou úlohu můžeme poznat také tak, že lze vektor b vyjádřit pomocí méně než m sloupců matice A . Degenerovaný bod v \mathbb{R}^n vznikne jako průnik více než n nadrovin. Pro \mathbb{R}^3 si tento fakt ilustrujeme v Příkladu 2.1. S degenerovaným bodem se můžeme setkat i při řešení praktických úloh, což uvidíme např. v Příkladu 5.4.

Příklad 2.1. Degenerovanost bodů v \mathbb{R}^3 .

Na Obrázku 2.1 je zobrazen pravidelný pětiboký jehlan. Bod označený písmenem A vznikl průnikem 3 nadrovin (2 stěn a podstavy), a je tedy nedegenerovaný. Bod označený písmenem B však vznikl průnikem 5 nadrovin (všech stěn jehlanu), jedná se tak o degenerovaný bod. Degenerovanost bodu B zde však není způsobena existencí nadbytečného omezení.



Obrázek 2.1: Pětiboký jehlan

△

Vzhledem k výše uvedeným vztahům můžeme pro vektor b psát $b = A_B x_B$, kde A_B, x_B vzniknou z A (případně z x) vynecháním nebazických pozic. Z tohoto vztahu pak pro bod x_B platí $x_B = A_B^{-1} b$, a tedy v případě, že známe bázi, můžeme bazický bod získat s využitím tohoto vztahu. Na nebazické pozice poté dosadíme nuly. Takto získaný bod ovšem není nutně přípustný.

Pro lepší představu si nyní pojmy bazický přípustný bod, nosič, báze a degenerovanost ukážeme na konkrétním příkladu.

Příklad 2.2. Bazické přípustné body pro různé báze v degenerované úloze.

Množina X je určena maticí A a vektorem b , kde

$$A = \begin{pmatrix} 1 & 2 & 1 & 2 & 1 \\ 3 & 0 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 8 \\ 7 \end{pmatrix}.$$

Najděte bazické body pro báze $B_1 = \{1, 2, 3\}$, $B_2 = \{2, 3, 4\}$, $B_3 = \{1, 3, 5\}$ a $B_4 = \{2, 3, 5\}$. Určete nosič těchto bodů a rozhodněte o jejich přípustnosti. Dále zjistěte, zda je zadaná úloha degenerovaná či ne.

Řešení: Nejprve ověříme, zda je B_1 báze. To dokážeme lineární nezávislostí sloupců matice, která vznikne ponecháním sloupců s indexy $\{1, 2, 3\}$. Determinant této matice je 4, jde tedy o regulární matici a B_1 je báze. Obdobně postupujeme pro B_2 , B_3 a B_4 . Ve všech případech dojdeme k výsledku, že se jedná o báze. Nyní můžeme přejít k hledání bazických přípustných bodů, neboli k řešení soustavy $A_B x_B = b$. K jejímu řešení můžeme využít více postupů, my zde zmíníme 2, a to postup přes řešení lineární soustavy a přes inverzi matice A_B . První způsob, který si ukážeme, využívá Gaussovu eliminaci. Pro bázi $B_1 = \{1, 2, 3\}$ hledáme řešení soustavy

$$\begin{aligned} a + 2b + c &= 5, \\ 3a + 2c &= 8, \\ 2a + 2b + c &= 7. \end{aligned}$$

Čtenář se snadno přesvědčí, že výsledkem je $a = 2$, $b = 1$, $c = 1$. Na nebazických pozicích doplníme nuly. Hledaný bod je tedy

$$x^{[1]} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Tento bod je přípustný, neboť je řešením uvedené soustavy. Zároveň je bazický, protože má na bazických pozicích pouze kladná čísla. Jeho nosičem je tedy $J(x) = \{1, 2, 3\} = B_1$. Bod $x^{[1]}$ je nedegenerovaný.

Druhým způsobem, kterým můžeme hledat bazický přípustný bod, je využití vztahu $x_B = A_B^{-1}b$. Pro bázi B_1 tímto způsobem získáme

$$x_{B_1} = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 5 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\ \frac{3}{2} & \frac{1}{2} & -\frac{3}{2} \end{pmatrix} \begin{pmatrix} 5 \\ 8 \\ 7 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

Tento způsob je ovšem pro větší matice náročnější.

Pro bázi $B_2 = \{2, 3, 4\}$ postupujeme obdobně, pro jednoduchost využijeme opět postup přes řešení soustavy rovnic. V tomto případě tak řešíme soustavu

$$\begin{aligned} 2a + b + 2c &= 5, \\ 2b + c &= 8, \\ 2a + b + c &= 7. \end{aligned}$$

Jejím řešením je $a = 2$, $b = 5$, $c = -2$. Bod, který jsme tímto způsobem získali, je tedy

$$x^{[2]} = \begin{pmatrix} 0 \\ 2 \\ 5 \\ -2 \\ 0 \end{pmatrix}.$$

Tento bod je sice bazický, není ovšem přípustný, neboť má na bazické pozici záporné číslo. Pro bázi $B_3 = \{1, 3, 5\}$ obdobně dostaneme bod

$$x^{[3]} = \begin{pmatrix} 0 \\ 0 \\ 3 \\ 0 \\ 2 \end{pmatrix}.$$

Vidíme, že na bazické pozici je 0, bod $x^{[3]}$ je sice bazickým přípustným bodem, pro jeho nosič však platí $J(x) = \{3, 5\} \neq B_3$. Vektor b jsme schopni získat s využitím pouze dvou sloupců matice A , úloha je tedy degenerovaná. Stejný bazický bod navíc dostaneme i pro bázi $B_4 = \{2, 3, 5\}$. Podrobnější ověření správnosti těchto výsledků ponecháme na čtenáři. △

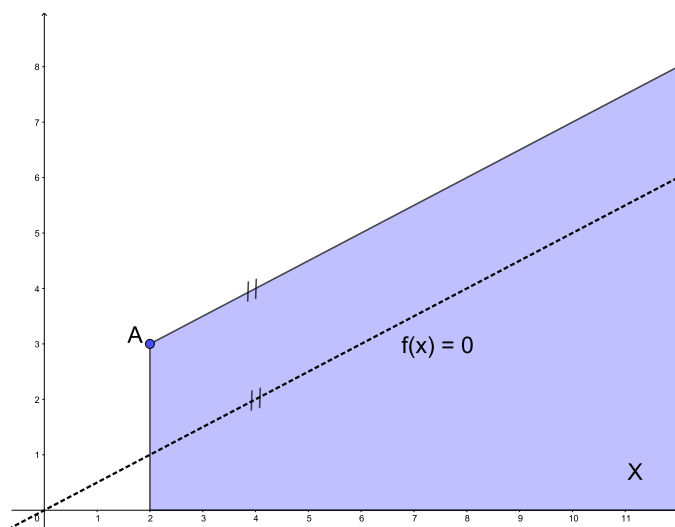
Již víme, jaké možnosti mohou nastat pro řešení úlohy lineárního programování. Jak ale toto řešení získáme? Z teorie uvedené výše víme, že pro úlohu (2.1) je přípustná množina polyedr a podle základní věty lineárního programování nastává minimum v některém z vrcholů přípustné množiny. Platí také, že bazický přípustný bod je ekvivalentní vrcholu množiny. Víme tedy, že jestliže má úloha optimální řešení, pak existuje bazický přípustný bod, který je řešením. Důkaz tohoto tvrzení je možné najít např. v [22].

Předchozí odstavec neříká nic o situaci, ve které úloha (2.1) nemá řešení, ani o tom, že každé řešení je bazickým přípustným bodem. To je pravda pouze v případě, že úloha (2.1) má jediné řešení. Tento fakt je ilustrován na Příkladu 2.3. Teorie popsaná výše je odvozená pro úlohu v kanonickém tvaru. V případě, že uvažujeme úlohu ve tvaru nerovností, se může stát, že je řešením celá přímka, neexistuje tedy žádný vrchol, který by byl řešením.

Příklad 2.3. Řešení, která nejsou bazickými přípustnými body.

Uvažujme úlohu lineárního programování, která je znázorněna na Obrázku 2.2, a předpokládejme, že hledáme minimum a hodnota účelové funkce klesá směrem nahoru. Hrana přípustné množiny je rovnoběžná s vrstevnicí účelové funkce, řešením tedy bude celá polopřímka (hrana). Existuje tak nekonečně mnoho řešení dané úlohy, pouze vrchol přípustné množiny označený písmenem A je i bazickým přípustným bodem. △

Při řešení úlohy (2.1) tedy hledáme bazický přípustný bod, který je řešením. Metoda, která postupně prochází přes jednotlivé vrcholy polyedru (a tedy přes bazické přípustné body), se nazývá *simplexová metoda*. Tuto algoritmickou metodu si podrobně představíme v následující části textu.



Obrázek 2.2: Přípustná množina a vrstevnice $f(x) = 0$

Kapitola 3

Simplexová metoda

Simplexová metoda (či simplexový algoritmus) byla zveřejněna americkým matematikem Georgem Bernardem Dantzigem v roce 1947. Metoda nese název simplexová podle simplexu, což je n -rozměrné zobecnění trojúhelníku, po kterém se při aplikování tohoto algoritmu na úlohu lineárního programování pohybujeme. Simplex je speciálním případem polyedru. Podstatou algoritmu je přechod mezi jednotlivými bazickými přípustnými body (tj. vrcholy) až do chvíle, kdy najdeme optimální řešení, případně zjistíme, že úloha je neohraňovaná. Geometricky si podstatu algoritmu můžeme představit tak, že vybírá hranu, podél které se hodnota účelové funkce zlepšuje (tj. zmenšuje). Po této hraně se pak z jednoho vrcholu dostaneme do dalšího, ve kterém účelová funkce nabývá menší, případně stejné hodnoty. Jedná se o metodu, která využívá principu Gaussovy eliminace s využitím speciálních pravidel, jak vybírat prvek, podle kterého eliminaci provádíme. V této kapitole si odvodíme simplexový algoritmus a ukážeme si, jak ho použít při řešení úloh lineárního programování.

Odvození a popis přesného postupu algoritmu simplexové metody se v různých zdrojích liší. To je dáno tím, zda v úloze hledáme maximum či minimum zadané funkce, nebo rozdílností tvarů zadaných úloh. V této práci je simplexová metoda odvozena pro úlohu lineárního programování v kanonickém tvaru (tj. pouze rovnostní omezení a všechny proměnné omezené na znaménko). Dále předpokládáme, že matice A má plnou hodnost, tj. levé strany omezení jsou lineárně nezávislé, což můžeme zajistit vynecháním přebytečných a odporujících si omezení.

Pro zahájení simplexové metody potřebujeme libovolný bazický přípustný bod. Tento bod nalezneme velmi snadno pouze pro speciální tvar úlohy (2.1), předpokládejme ale nyní, že výchozí bazický přípustný bod známe. To také znamená, že přípustná množina je neprázdná (a tedy úloha je přípustná). Způsoby nalezení výchozího bazického přípustného bodu rozebereme podrobněji v Části 3.3.

Pokud máme výchozí bazický přípustný bod $x^{[0]}$, můžeme zahájit algoritmus. Metoda pak generuje posloupnost bazických přípustných bodů (ozn. $x^{[1]}, \dots, x^{[p]}$) tak, že body ve dvou po sobě následujících krocích algoritmu mají až na jednu složku stejnou bázi. Těmto bodům říkáme *sousední*. V p -tém kroku pak zjistíme, zda

1. Nalezený bod $x^{[p]}$ je řešením úlohy.
2. Úloha nemá řešení.

3. Existuje bazický přípustný bod $x^{[p+1]}$ takový, že $c^T x^{[p+1]} < c^T x^{[p]}$ (tj. hodnota účelové funkce v bodě $x^{[p+1]}$ je menší než v $x^{[p]}$). Současně tento bod konstrukčně získáme, viz Věta 3.3.

3.1 Odvození simplexové metody

Minimalizační úlohu lineárního programování v kanonickém tvaru můžeme také zapsat maticově, tj.

$$\begin{aligned} \min c^T x \\ Ax = b \\ x \geq 0, \end{aligned}$$

kde $x = (x_1, \dots, x_n)^T$, $c = (c_1, \dots, c_n)^T$, $b = (b_1, \dots, b_m)^T$ a $A = (a_1, \dots, a_n)$. Označme $x := x^{[p]}$, $J := J(x)$ a $B := B(x)$, přičemž daná báze nemusí být jediná. Dále definujme hodnotu účelové funkce v daném bazickém přípustném bodu jako

$$z_0 := \sum_{j \in B} c_j x_j.$$

S využitím vztahu $A_{B \times B} x_B = b$, se kterým jsme se setkali v předchozí kapitole, můžeme psát

$$\sum_{j \in B} a_j x_j = \sum_{j \in J} a_j x_j = b.$$

Jelikož sloupce matice a_j jsou lineárně nezávislé pro $j \in B$, libovolný vektor a_1, \dots, a_n může být vyjádřen pouze pomocí vektorů a_j pro $j \in B$ s koeficienty, které označíme λ_{jk} . Libovolný sloupec matice A tedy můžeme zapsat ve tvaru

$$a_k = \sum_{j \in B} \lambda_{jk} a_j, \quad k = 1, \dots, n. \quad (3.1)$$

Z tohoto vyjádření si můžeme všimnout, že jistě platí

$$\begin{aligned} \lambda_{kk} &= 1 & \text{pro } k \in B, \\ \lambda_{jk} &= 0 & \text{pro } j \in B \setminus \{k\}. \end{aligned}$$

Definujme nyní

$$z_k := \sum_{j \in B} \lambda_{jk} c_j, \quad k = 1, \dots, n. \quad (3.2)$$

Dále si přeznačíme pořadí vektorů a_1, \dots, a_n tak, že báze B bodu x je $B = \{1, \dots, m\}$. Koeficienty λ_{jk} uspořádáme do matice, kterou označíme L , tj.

$$L := \begin{pmatrix} \lambda_{11} & \dots & \lambda_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1} & \dots & \lambda_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

Matici koeficientů omezení A můžeme s využitím vztahu (3.1) zapsat ve tvaru

$$A = (a_1, \dots, a_n) = A_B L, \quad \text{a tedy} \quad L = A_B^{-1} A,$$

neboť matice A_B je regulární. Odtud plyne, že k -tý sloupec matice L můžeme získat jako

$$\begin{pmatrix} \lambda_{1k} \\ \vdots \\ \lambda_{mk} \end{pmatrix} = A_B^{-1} \begin{pmatrix} a_{1k} \\ \vdots \\ a_{mk} \end{pmatrix}.$$

Pro z_k pak můžeme tohoto vztahu využít a získat tak vyjádření

$$z_k = c_1 \lambda_{1k} + \dots + c_m \lambda_{mk} = (c_1, \dots, c_m) \begin{pmatrix} \lambda_{1k} \\ \vdots \\ \lambda_{mk} \end{pmatrix} = c_B^T A_B^{-1} a_k. \quad (3.3)$$

Uvažujme nyní libovolný bod $y \in X$, tj. platí $Ay = b$ a $y \geq 0$, a označme N množinu nebazických indexů, tj. $N := \{1, \dots, n\} \setminus B = \{m+1, \dots, n\}$. Pro tento bod můžeme rozepsat rovnostní omezení na bazickou a nebazickou část a získat tak rovnost

$$A_B y_B + A_N y_N = b.$$

Odtud y_B vyjádříme postupně jako

$$\begin{aligned} A_B y_B &= b - A_N y_N \\ y_B &= A_B^{-1} b - A_B^{-1} A_N y_N = A_B^{-1} b - \sum_{j=m+1}^n A_B^{-1} a_j y_j, \end{aligned} \quad (3.4)$$

kde $k = 1, \dots, n$. Pokud stejné rozepsání na bazickou a nebazickou část provedeme pro účelovou funkci $c^T y$, dostaneme $c^T y = c_B^T y_B + c_N^T y_N$. Dosazením (3.4) za y_B pak získáme

$$c^T y = c_B^T A_B^{-1} b - c_B^T \sum_{j=m+1}^n A_B^{-1} a_j y_j + c_N^T y_N.$$

S využitím znalosti (3.3) a vztahu $A_B x_B = b$ můžeme psát

$$c^T y = c_B^T x_B - \sum_{j=m+1}^n z_j y_j + \sum_{j=m+1}^n c_j y_j = c_B^T x_B - \sum_{j=m+1}^n (z_j - c_j) y_j.$$

Pro libovolné $y \in X$ tedy platí

$$\langle c, y \rangle = \langle c, x \rangle - \sum_{j=m+1}^n (z_j - c_j) y_j.$$

Z této rovnosti vidíme, že bod x je řešením (hledaným minimem) pouze tehdy, když $\sum_{j=m+1}^n (z_j - c_j) y_j \leq 0$. Vzhledem k libovolnosti bodu $y \in X$, pro který jistě platí $y \geq 0$, bude tato nerovnost splněna zejména když $\sum_{j=m+1}^n (z_j - c_j) \leq 0$.

Z odvození simplexové metody vyplývá několik důsledků, které v následujícím odstavci zformulujeme do vět. Věty svým pořadím odpovídají situacím, které mohou v p -tém kroku algoritmu nastat, uvedeným výše. Princip důkazů lze najít např. v [35] a [47], důkaz následující věty uvádí též [9].

Věta 3.1. *Nechť je úloha (2.1) nedegenerovaná. Bod x je řešením úlohy (2.1) právě tehdy, když pro každé $k \notin B$ platí $z_k - c_k \leq 0$.*

Věta 3.2. *Jestliže existuje $s \notin B$ takové, že $z_s - c_s > 0$, a pro každé $j \in B$ je $\lambda_{js} \leq 0$, pak je úloha (2.1) neohraničená.*

Věta 3.3. *Nechť je úloha (2.1) nedegenerovaná. Existují-li nějaké indexy $s \notin B$ (pro nedegenerovanou úlohu platí rovnost $J = B$) a $j \in B$ takové, že $z_s - c_s > 0$ a $\lambda_{js} > 0$, pak existuje nový bazický přípustný bod $\hat{x} \in X$ takový, že $c^T \hat{x} < c^T x$.*

Souřadnice nového bazického přípustného bodu \hat{x} najdeme s využitím

$$\hat{x} = \begin{cases} x_j - \alpha \lambda_{js}, & j \in B, \\ \alpha, & j = s, \\ 0, & j \notin B \setminus \{s\}, \end{cases} \quad \text{kde } \alpha := \min \left\{ \frac{x_j}{\lambda_{js}} \mid j \in B, \lambda_{js} > 0 \right\} \geq 0.$$

Jak jsme odvodili v předchozí části, počet bazických přípustných bodů je vždy konečný. Algoritmus je tedy za předpokladu nedegenerovanosti úlohy vždy konečný. Z Věty 3.1 vyplývají následující dva důsledky o jednoznačnosti řešení.

Důsledek 3.4. *Jestliže pro každé $k \notin B$ platí $z_k - c_k < 0$, pak je bod x jediným řešením úlohy (2.1).*

Platnost tohoto důsledku nezávisí na možné degenerovanosti úlohy (2.1). Pro následující důsledek však nedegenerovanost opět požadujeme. V případě degenerované úlohy by se mohlo stát, že bod, který považujeme za jiné řešení, je stejný degenerovaný bod vyjádřený pomocí jiných bází. Daný bod by tedy byl sousední sám se sebou.

Důsledek 3.5. *Nechť je úloha (2.1) nedegenerovaná. Jestliže je x řešením úlohy (2.1) a pro nějaké $k \notin B$ platí $z_k - c_k = 0$, pak existuje další řešení. Jestliže pro nějaké $j \in B$ je $\lambda_{jk} > 0$, pak existuje nový bazický přípustný bod. Naopak, jestliže pro každé $j \in B$ je $\lambda_{jk} \leq 0$, pak neexistuje žádný další bazický přípustný bod, který by byl řešením.*

3.2 Simplexová tabulka

Již víme, jak podle hodnoty rozdílu $z_k - c_k$ pro $k \notin B$ a hodnot příslušných λ_{jk} určit, zda se jedná o optimální řešení či ne, případně zjistit, zda je úloha neohraničená. Pro přehlednou práci při výpočtu řešení úlohy (2.1) pomocí simplexové metody využijeme tzv. *simplexovou tabulku*. Do této tabulky zapíšeme všechny potřebné informace pro provedení jednotlivých kroků metody tak, že jsme schopni snadno přecházet další bazický přípustný bod, hodnotu účelové funkce, atd. Průběh simplexového algoritmu, jak jsme již zmínili, připomíná Gaussovu eliminaci s využitím speciálních pravidel, která jsou popsána níže. Podobnost s Gaussovou eliminací bude při využití simplexové tabulky zřejmá. Simplexová tabulka je ve tvaru, ve kterém ji v této práci budeme využívat, znázorněna v Tabulce 3.1.

c_B	B	x_1	x_2	\dots	x_n	x_B	$\frac{x_B}{\lambda}$
bazické složky vektoru c	bazické proměnné	matice L				bazické složky x	podíly $\frac{x_{b_i}}{\lambda_{b_i,s}}$
	c	vektor c					
	z_k	z_1	z_2	\dots	z_n		
	$\Delta = z_k - c$	$z_1 - c_1$	$z_2 - c_2$	\dots	$z_n - c_n$	$c^T x$	

Tabulka 3.1: Simplexová tabulka

Tabulku si nyní popíšeme podrobněji. V tabulce pod označením c_B rozumíme koeficienty jednotlivých bazických proměnných v účelové funkci. Písmeno B symbolicky označuje proměnné, jež jsou v bázi (ve sloupci budou příslušná x_i). Matice L je, jak již víme, definovaná jako matice koeficientů λ_{jk} . Výše jsme odvodili, že matici L můžeme zapsat jako součin matic $A_B^{-1}A$. V případě, že je matice A_B rovna jednotkové matici (ozn. I), platí rovnost $L = A$. V tomto případě je navíc $x_B = b$, což vyplývá ze vztahů uvedených výše.

Podíly, které jsou uvedeny v posledním sloupci tabulky, si rozebereme podrobněji v další části textu. Při výpočtu je však vždy počítáme v závěru daného kroku algoritmu, až po určení tzv. *klíčového sloupce*. Vektor c je vektor koeficientů účelové funkce, hodnoty v řádku označeném z_k získáme pomocí vztahu (3.2). Výpočet řádku z_k bude podrobněji popsán v následujícím příkladu.

V dalších krocích simplexového algoritmu není potřeba využívat takto podrobnou tabulku, vystačíme si s tzv. *redukovanou simplexovou tabulkou*, ve které chybí sloupec označený c_B a řádky označené c a z_k . Pro ilustraci si na Příkladu 3.1 ukážeme, jak sestavit výchozí simplexovou tabulku.

Příklad 3.1. Sestavení výchozí simplexové tabulky.

Pro následující úlohu sestavte výchozí simplexovou tabulku.

$$\begin{aligned} \min & && -3x_1 + x_2 - x_3 + x_4 \\ \text{za podmíněk} & && x_1 + 2x_2 + x_3 = 10 \\ & && 3x_1 + x_2 + x_4 = 15 \\ & && x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Řešení. Zadaná úloha je již v kanonickém tvaru. V souladu se značením maticového zápisu úlohy pro tento příklad máme

$$c = (-3, 1, -1, 1)^T \quad A = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 1 \end{pmatrix} \quad b = (10, 15)^T.$$

Úloha je nedegenerovaná, neboť k získání vektoru b potřebujeme alespoň 2 sloupce matice A . Matice A navíc obsahuje jednotkovou matici (sloupce 3 a 4), za výchozí bázi tedy zvolíme $B = \{3, 4\}$. Jak již bylo uvedeno výše, matice L je v tomto případě rovna matici A , neboť obecně platí vztah $L = A_B^{-1}A$. Nyní můžeme sestavit výchozí simplexovou tabulku

zobrazenou v Tabulce 3.2. Řádek z_k získáme skalárním součinem sloupce c_B a příslušného sloupce x_i . Např. pro první prvek v řádku z_k tak dostaneme $c_B^T a_1 = -1 \cdot 1 + 1 \cdot 3 = 2$, pro druhý prvek máme $c_B^T a_2 = -1 \cdot 2 + 1 \cdot 1 = -1$.

V tabulce záměrně nevyplňujeme poslední sloupec podílů, k jejich výpočtu se potřebujeme blíže seznámit se simplexovým algoritmem. Z tabulky vidíme, že hodnota účelové funkce ve vybraném výchozím bazickém přípustném bodě je rovna 5 (údaj v posledním řádku tabulky a sloupci x_B).

c_B	B	x_1	x_2	x_3	x_4	x_B	$\frac{x_B}{\lambda}$
-1	x_3	1	2	1	0	10	
1	x_4	3	1	0	1	15	
	c	-3	1	-1	1		
	z_k	2	-1	-1	1		
	$\Delta = z_k - c$	5	-2	0	0	5	

Tabulka 3.2: Výchozí simplexová tabulka pro Příklad 3.1

△

Na základě výše uvedené teorie můžeme s využitím tabulky simplexovou metodu popsat následujícím algoritmem. Kroky 6 a 7 tohoto algoritmu vychází z důkazu Věty 3.3, který zde neuvádíme. Čtenář ho však najde např. v [47].

1. Úlohu převedeme do kanonického tvaru tak, aby pro složky vektoru b navíc platilo $b \geq 0$.
2. Najdeme výchozí bazický přípustný bod a jeho bázi. V případě, že matice A obsahuje všechny sloupce jednotkové matice, vezmeme za bázi indexy určené těmito sloupci. V opačném případě využijeme tzv. *metodu velkého M* nebo *dvoufázovou simplexovou metodu*, jejichž postupy jsou popsány v Části 3.3.
3. Pro přehlednější výpočet si pro zvolený výchozí bazický přípustný bod sestavíme simplexovou tabulku.
4. Vypočítáme rozdíly $z_j - c_j$ pro každé $j \notin B$.
 - (a) Jestliže pro všechny rozdíly platí $z_j - c_j \leq 0$, našli jsme podle Věty 3.1 řešení. Algoritmus tak končí.
 - (b) Naopak, jestliže pro nějaké $s \notin B$ platí $z_s - c_s > 0$, vybereme tento index s . Sloupec, který v tomto kroku vybereme, nazýváme *klíčový sloupec*. V případě, že je indexů splňujících tuto podmínku více, vybereme jeden z těchto indexů podle námi vybraného pravidla. S různými metodami pro výběr indexu sloupce vstupujícího do báze se blíže seznámíme v Části 3.2.1.
5. Podíváme se na hodnoty příslušných λ_{js} .

- (a) Je-li $\lambda_{js} \leq 0$ pro každé $j \in B$, pak podle Věty 3.2 víme, že se jedná o neohraňovanou úlohu a algoritmus končí.
- (b) Naopak, jestliže je pro nějaké $j \notin B$ splněno $\lambda_{js} > 0$, pak existuje lepší bazický přípustný bod.
6. Proměnná příslušná vybranému klíčovému sloupci, tj. x_s , vstupuje v tomto kroku do báze. Nyní musíme určit, která proměnná bázi opustí. Tou je proměnná, pro kterou platí

$$\frac{x_r}{\lambda_{rs}} = \min_{j \in B, \lambda_{js} > 0} \frac{x_s}{\lambda_{js}}.$$

Řádku, který tímto způsobem vybereme, říkáme *klíčový řádek*. Z báze tedy vystupuje proměnná x_r . V případě rovnosti podílů pro více indexů můžeme využít např. Blandovo či lexikografické pravidlo, se kterými se seznámíme v Části 3.2.2.

7. Prvek, který je v klíčovém řádku a klíčovém sloupci, nazýváme *pivot* či *klíčový prvek*. Pro nalezení bazického přípustného bodu s nižší hodnotou účelové funkce provedeme s pomocí pivota Gaussovu eliminaci tak, aby se pivot rovnal jedné (tj. klíčový řádek vynásobíme číslem $1/\lambda_{rs}$). Dále pouze pomocí pivota vynulujeme všechny ostatní pozice v klíčovém sloupci, a to včetně Δ -řádku. V této eliminaci využíváme vždy pouze vhodně vynásobený klíčový řádek, který přičítáme k ostatním řádkům. Tímto krokem získáme další simplexovou tabulku příslušnou novému bazickému přípustnému bodu. V algoritmu se tedy nemusíme vracet ke 3. kroku, ani počítat rozdíly $z_j - c_j$, neboť ty jsou pro nový bazický přípustný bod uvedeny v Δ -řádku nové tabulky. Ve sloupci označeném x_B s bazickými proměnnými místo x_r (proměnné, která opouští bázi) napíšeme x_j (proměnná, která vstupuje do báze).
8. Vráťme se ke kroku 4(a) a algoritmus opakujeme, dokud nedojdeme k jednomu ze závěrů popsaných výše. V případě, že najdeme řešení úlohy, odečteme hodnoty jednotlivých bazických proměnných ve sloupci označeném x_B , na nebazické pozice doplníme nuly. Minimální hodnota účelové funkce je pak uvedena v Δ -řádku ve stejném sloupci.

Nyní již víme, jak postupovat při sestavení simplexové tabulky i při jednotlivých krocích algoritmu simplexové metody. Pokračujme tedy v Příkladu 3.1, pro který provedeme první krok simplexového algoritmu.

Příklad 3.2. První krok simplexového algoritmu.

S využitím simplexové tabulky proveďte první krok simplexové metody pro úlohu zadanou v Příkladu 3.1.

Řešení. Pro přehlednost si zde uvedeme Tabulku 3.2, ve které budeme pokračovat. V tabulce vidíme, že jediné kladné číslo v Δ -řádku je v prvním sloupci. Tento sloupec tedy vybereme a proměnná x_1 vstoupí do báze, což v tabulce označíme šípkou. Nyní již můžeme vypočítat podíly, které zapíšeme do posledního sloupce tabulky. Nejmenší podíl se nachází ve druhém řádku, klíčovým řádkem je tedy řádek pro proměnnou x_4 , která vystoupí z báze, což opět naznačíme šípkou. Pivotem je číslo 3 v průsečíku klíčového řádku a sloupce, v simplexové tabulce si pro přehlednost toto číslo zakroužkujeme.

c_B	B	x_1	x_2	x_3	x_4	x_B	$\frac{x_B}{\lambda}$
-1	x_3	1	2	1	0	10	$10/1 = 10$
1	x_4	3	1	0	1	15	$15/3 = 5 \rightarrow$
	c	-3	1	-1	1		
	z_k	2	-1	-1	1		
	Δ	$5 \uparrow$	-2	0	0	5	
	x_3	0	$5/3$	1	$-1/3$	5	
	x_1	1	$1/3$	0	$1/3$	5	
	Δ	0	$-11/3$	0	$-5/3$	-20	

Tabulka 3.3: První krok simplexové metody pro Příklad 3.2

Nyní provedeme Gaussovu eliminaci pomocí pivota. Druhý řádek tedy vynásobíme $-1/3$ a přičteme ho k prvnímu řádku. S využitím pivota eliminujeme též číslo 5 v Δ -řádku, a to přičtením Δ -řádku vynásobeného číslem $-1/5$ k prvnímu řádku. Výsledek tohoto postupu je v druhé části Tabulky 3.3. Jak jsme uvedli výše, není mimo výchozí simplexovou tabulku nutné uvádět i řádky označené c a z_k a sloupec označený c_B . V Δ -řádku jsou nyní pro nebazické prvky pouze záporná čísla, našli jsme tedy řešení dané úlohy. Minimum nastane v bodě $[5, 0, 5, 0]$, ve kterém je hodnota účelové funkce rovna -20 (viz sloupec x_B).

△

Pro lepší pochopení a procvičení simplexového algoritmu si nyní uvedeme několik řešených příkladů. Na těchto příkladech si ukážeme různé situace, které mohou při řešení úloh lineárního programování nastat.

Příklad 3.3. Simplexová metoda – postup řešení.

Pomocí simplexové tabulky najděte řešení následující minimalizační úlohy.

$$\begin{aligned} \min \quad & -6x_1 + x_2 \\ \text{za podmíněk} \quad & x_1 + 2x_2 \leq 8 \\ & 4x_1 - 2x_2 \leq 16 \\ & x_1 + x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Řešení. Prvním krokem při řešení této úlohy bude její převod do kanonického tvaru. Přidáním doplňkových proměnných označených s_1, s_2, s_3 omezených na znaménko získáme podmínky ve tvaru rovností. Matice A je tedy ve tvaru

$$A = \begin{pmatrix} 1 & 2 & 1 & 0 & 0 \\ 4 & -2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Matice A obsahuje všechny sloupce jednotkové matice, snadno tak sestavíme výchozí simplexovou tabulku. Výchozí báze je, na rozdíl od Příkladu 3.1, tvořená výhradně doplň-

kovými proměnnými, hodnoty ve sloupci c_B i řádku z jsou tedy nyní pouze nuly. Výchozí tabulka, společně s dalšími kroky algoritmu, je znázorněna v Tabulce 3.4.

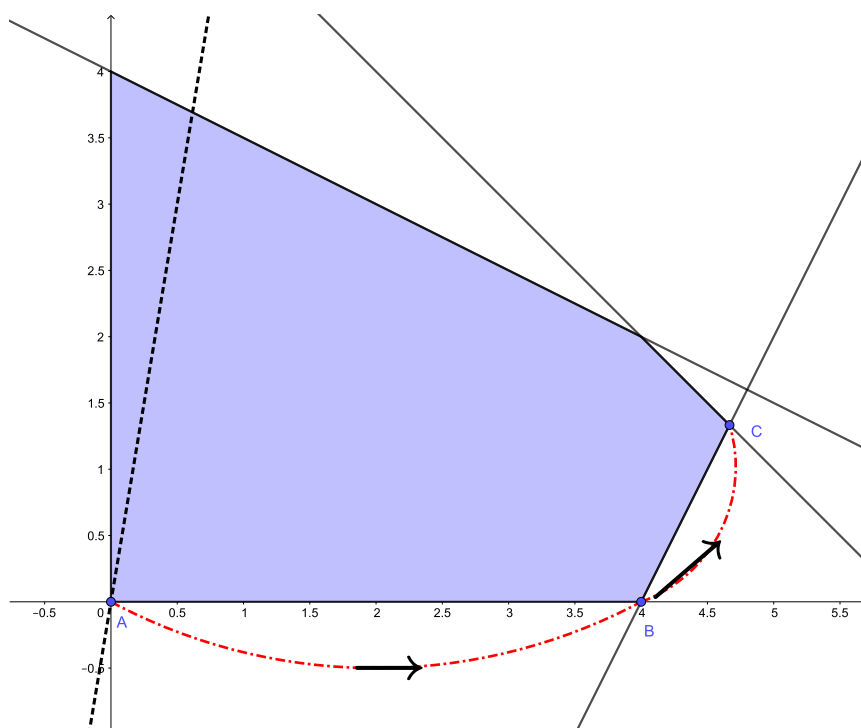
c_B	B	x_1	x_2	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
0	s_1	1	2	1	0	0	8	8/1 = 8
0	s_2	4	-2	0	1	0	16	16/4 = 4 →
0	s_3	1	1	0	0	1	6	6/1 = 6
	c	-6	1	0	0	0		
	z_k	0	0	0	0	0		
	Δ	6 ↑	-1	0	0	0	0	
	s_1	0	5/2	1	-1/4	0	4	8/5
	x_1	1	-1/2	0	1/4	0	4	-
	s_3	0	3/2	0	-1/4	1	2	4/3 →
	Δ	0	2 ↑	0	-3/2	0	-24	
	s_1	0	0	1	1/6	-5/3	2/3	
	x_1	1	0	0	1/6	1/3	14/3	
	x_2	0	1	0	-1/6	2/3	4/3	
	Δ	0	0	0	-7/6	-4/3	-80/3	

Tabulka 3.4: Simplexová metoda pro Příklad 3.3

V prvním kroku simplexového algoritmu je v Δ -řádku jediné kladné číslo, a to v prvním sloupci. Tento sloupec je tedy klíčovým sloupcem. Z vypočítaných podílů je nejmenší ve druhém řádku, v bázi tedy místo s_2 bude v další tabulce x_1 . Pivotem je tedy číslo 4. Druhý řádek dále vynásobíme číslem 1/4, abychom na místě pivota dostali číslo 1. Pomocí tohoto řádku pak provedeme Gaussovu eliminaci, čímž získáme druhou simplexovou tabulku.

Ve druhém kroku postupujeme obdobně. V Δ -řádku je opět pouze jedno kladné číslo, do báze tedy vstoupí proměnná x_2 , a to na místo proměnné s_3 . Pivotem je tedy číslo 2/3. K poslední simplexové tabulce se dostaneme vynásobením třetího řádku číslem 3/2 a následnou Gaussovou eliminací s využitím třetího řádku. Tímto způsobem získáme výsledek, neboť v Δ -řádku jsou pro všechny nebazické proměnné záporná čísla. Řešením úlohy v kanonickém tvaru je bod $x^* = [14/3, 4/3, 2/3, 0, 0]$, v němž je hodnota účelové funkce $-80/3$. Pro původní úlohu je tedy řešením bod $x^* = [14/3, 4/3]$.

Postup simplexové metody si nyní ukážeme také na náčrtku. Na Obrázku 3.1 je zobrazen postup přes bazické přípustné body v původních proměnných v jednotlivých krocích metody. Bod A označuje výchozí bazický přípustný bod, kterým je v původních proměnných počátek. Souřadnice tohoto bodu jsou ve sloupci x_B ve výchozí simplexové tabulce. Souřadnice nebazických proměnných jsou rovny nule. Bod označený písmenem B představuje bod, který získáme po prvním kroku simplexové metody. Tento bod je $[4, 0, 4, 0, 2]$, v původních proměnných jde tedy o bod $B = [4, 0]$. Řešení, označené v obrázku bodem C , opět najdeme ve sloupci x_B .



Obrázek 3.1: Postup simplexové metody pro Příklad 3.3

△

Příklad 3.4. Simplexová metoda – neohraničená úloha.

Pomocí simplexové tabulky najděte řešení následující minimalizační úlohy.

$$\begin{aligned}
 \min \quad & x_1 - 2x_2 + x_3 + 3x_4 \\
 \text{za podmíněk} \quad & 2x_1 - x_2 + x_3 - x_4 \leq 10 \\
 & -5x_1 + 2x_2 - 2x_3 + x_4 \leq 20 \\
 & 3x_1 - 4x_2 + 4x_3 - 2x_4 \leq 30 \\
 & x_1, x_2, x_3, x_4 \geq 0
 \end{aligned}$$

Řešení. Při řešení postupujeme stejně jako v předchozím příkladu, začneme tedy přidáním doplňkových proměnných s_1, s_2 a s_3 omezených na znaménko. Tímto krokem v matici A vytvoříme jednotkovou matici a snadno určíme výchozí bazický přípustný bod. Postup řešení si opět zapíšeme do simplexové tabulky.

První krok provedeme nám již známým způsobem. Klíčovým sloupcem je druhý sloupec, klíčovým řádkem je druhý řádek. Pomocí pivota, kterým je zakroužkované číslo 2, provedeme Gaussovou eliminaci, čímž dostaneme druhou část simplexové tabulky zobrazené v Tabulce 3.5. V tomto kroku máme dvě kladná čísla v Δ -řádku. Zkusíme tedy vybrat např. první sloupec. V tomto sloupci ovšem nejsou žádná kladná čísla, nemůžeme tedy vybrat klíčový řádek. Situace je stejná i pro třetí sloupec. Z popisu simplexového algoritmu víme, že v tomto případě řešení neexistuje, neboť se jedná o neohraničenou úlohu.

c_B	B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
0	s_1	2	-1	1	-1	1	0	0	10	—
0	s_2	-5	2	-2	1	0	1	0	20	$20/2 = 10 \rightarrow$
0	s_3	3	-4	4	-2	0	0	1	30	—
	c	1	-2	1	3	0	0	0		
	z_k	0	0	0	0	0	0	0		
	Δ	-1	2↑	-1	-3	0	0	0	0	
	s_1	-1/2	0	0	-1/2	1	1/2	0	20	
	x_2	-5/2	1	-1	1/2	0	1/2	0	10	
	s_3	-7	0	0	0	0	2	1	70	
	Δ	4	0	1	-4	0	-1	0	-20	

Tabulka 3.5: Simplexová metoda pro Příklad 3.4

△

Příklad 3.5. Simplexová metoda – úsečka řešení.

Pomocí simplexové tabulky najděte řešení následující minimalizační úlohy.

$$\begin{aligned}
 \min \quad & x_1 - x_2 \\
 \text{za podmínek} \quad & x_1 + x_2 \leq 6 \\
 & 3x_1 + x_2 \leq 15 \\
 & -x_1 + x_2 \leq 3 \\
 & x_1, x_2 \geq 0
 \end{aligned}$$

Řešení. Úlohu s využitím nezáporných doplňkových proměnných s_1, s_2, s_3 převedeme na kanonický tvar. Poté již sestavíme výchozí simplexovou tabulku, která je společně s dalšími kroky algoritmu zobrazena v Tabulce 3.6.

V příkladu postupujeme standardně. V prvním kroku do báze na místo s_3 vstoupí proměnná x_2 . Z Tabulky 3.6 vidíme, že bod, který jsme tímto krokem získali, je řešením. Ve druhém kroku ale máme v prvním sloupci v Δ -řádku hodnotu 0 pro nebazickou proměnnou x_1 , což znamená, že x_1 může vstoupit do báze, jejím vstupem se však nezmění hodnota účelové funkce. Z báze v tomto kroku vystoupí proměnná s_1 . Pokračujeme-li dalším krokem algoritmu, zjistíme, že v Δ -řádku je pro proměnnou s_1 hodnota 0. Tato proměnná tedy může vstoupit do báze. V příslušném sloupci jsou dvě kladná čísla, není zde tedy problém s neohrazeností úlohy. Pokud bychom ale vybrali klíčový sloupec s_1 , proměnná s_1 by v tabulce nahradila proměnnou x_1 , čímž bychom se vrátili k předchozí tabulce. Simplexový algoritmus by tímto způsobem mohl pokračovat dál a dál, střídat v bázi proměnné x_1 a s_1 , a to beze změny hodnoty účelové funkce. Mezi řešeními této úlohy jsou tedy dva bazické přípustné body $[0, 3, 3, 12, 0]$ a $[3/2, 9/2, 0, 6, 0]$. Řešením této úlohy je tak celá úsečka mezi danými bazickými přípustnými body, pro původní proměnné jde o úsečku spojující body $[0, 3]$ a $[3/2, 9/2]$.

c_B	B	x_1	x_2	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
0	s_1	1	1	1	0	0	6	6
0	s_2	3	1	0	1	0	15	15
0	s_3	-1	①	0	0	1	3	3→
	c	1	-1	0	0	0		
	z_k	0	0	0	0	0		
	Δ	-1	1↑	0	0	0	0	
	s_1	②	0	1	0	-1	3	3/2 →
	s_2	4	0	0	1	-1	12	3
	x_2	-1	1	0	0	1	3	-
	Δ	0↑	0	0	0	-1	-3	
	x_1	1	0	1/2	0	-1/2	3/2	3
	s_2	0	0	-2	1	1	6	-
	x_2	0	1	1/2	0	1/2	9/2	9
	Δ	0	0	0	0	-1	-3	

Tabulka 3.6: Simplexová metoda pro Příklad 3.5

△

Příklad 3.6. Simplexová metoda – jeden bazický přípustný bod.

Pomocí simplexové tabulky najděte řešení následující minimalizační úlohy.

$$\begin{aligned}
 \min \quad & 4x_1 - 12x_2 \\
 \text{za podmínek} \quad & -x_1 + 2x_2 + s_1 = 16 \\
 & -x_1 + 3x_2 + s_2 = 32 \\
 & -x_1 - x_2 + s_3 = 1 \\
 & x_1, x_2, s_1, s_2, s_3 \geq 0
 \end{aligned}$$

Řešení. Zadaná úloha je již v kanonickém tvaru a matice A obsahuje jednotkovou matici. Můžeme tedy sestavit simplexovou tabulku, kterou vidíme v Tabulce 3.7.

První i druhý krok metody provedeme obdobně jako v předchozích příkladech, na konci druhého kroku již máme řešení úlohy. Ve třetím kroku metody je v Δ -řádku pro třetí sloupec hodnota 0. Proměnná s_1 by tedy mohla vstoupit do báze, nedošlo by však ke změně hodnoty účelové funkce. V příslušném sloupci jsou pouze záporná čísla, nemůžeme tak vybrat klíčový řádek. Tento příklad se liší od Příkladu 3.4, neboť v tomto příkladu nemáme v Δ -řádku žádné kladné číslo, nalezený bazický přípustný bod $[16, 16, 0, 0, 33]$ je tedy řešením. Vezmeme-li úlohu pouze pro proměnné x_1 a x_2 , je zadaná úloha ve tvaru nerovností. Omezení, které je rovnoběžné s vrstevnicí účelové funkce je omezení číslo 2, jež je pro funkci dvou proměnných ve tvaru $-x_1 + 3x_2 \leq 32$. Řešením je tedy polopřímka $-x_1 + 3x_2$ začínající bodem $[16, 16]$.

c_B	B	x_1	x_2	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
0	s_1	-1	2	1	0	0	16	$8 \rightarrow$
0	s_2	-1	3	0	1	0	32	$32/3$
0	s_3	-1	-1	0	0	1	1	-
	c	4	-12	0	0	0		
	z_k	0	0	0	0	0		
	Δ	-4	$12 \uparrow$	0	0	0	0	
	x_2	-1/2	1	1/2	0	0	8	-
	s_2	1/2	0	-3/2	1	0	8	$16 \rightarrow$
	s_3	-3/2	0	1/2	0	1	9	-
	Δ	$2 \uparrow$	0	-6	0	0	-96	
	x_2	0	1	-1	1	0	16	
	x_1	1	0	-3	2	0	16	
	s_3	0	0	-4	3	1	33	
	Δ	0	0	0	-4	0	-128	

Tabulka 3.7: Simplexová metoda pro Příklad 3.6

△

V předchozích příkladech (s výjimkou Příkladu 3.4) jsme pro výběr klíčového sloupce měli pouze jednu možnost. To se ale nemusí stát vždy. Podívejme se tedy na různé metody pro výběr indexu, který vstoupí do báze. Tyto metody si poté ukážeme na Příkladu 3.7.

3.2.1 Metody pro výběr indexu vstupujícího do báze

V této části uvažujeme, že existuje více indexů $j \notin B$, pro které platí $z_j - c_j > 0$. Jaký index máme v této situaci vybrat? Náš požadavek by mohl být, abychom vždy vybrali takový index, pro který proběhne algoritmus co nejrychleji (neboli v co nejmenším počtu iterací). Žádná z metod, které si v tomto výčtu uvedeme, však tento požadavek globálně nesplňuje. Při výběru klíčového sloupce se využívají pravidla založená na hodnotě účelové funkce. V této práci zmíníme pět pravidel, se kterými se lze v literatuře setkat nejčastěji. Další pravidla čtenář najde např. v [9] a [13].

Pravidla pro výběr klíčového sloupce (potažmo řádku) jsou v literatuře souhrnně označována jako *pivotovací pravidla* (viz např. [28], [36] a [43]).

1. Náhodné pravidlo a pravidlo nejmenšího indexu

Prvním pravidlem, které nás snadno napadne, je vybírat klíčový sloupec náhodně. V případě takového výběru dojdeme k optimálnímu řešení, G. B. Dantzig však ukázal, že náhodný výběr vede ve všeobecnosti k většímu počtu iterací, než využití jiných pravidel (více viz [10]). Výběr sloupce, který má nejmenší index, je podobný

náhodnému výběru. Zlepšení efektivity těchto dvou pravidel nabízí všechna další uvedená pravidla.

2. Dantzigovo pravidlo maximálního relativního ocenění

Toto pravidlo je kvůli své jednoduchosti jedním z nejvíce využívaných pravidel. Princip spočívá v tom, že vybíráme největší hodnotu rozdílu (v simplexové tabulce největší hodnotu v Δ -řádku). V případě rovnosti vybereme náhodně. Dantzigovo pravidlo tak dává největší možné zlepšení hodnoty účelové funkce na jednotku vstupující proměnné. Nevýhodou tohoto pravidla je, že zlepšení hodnoty účelové funkce závisí i na velikosti vstupující proměnné. Pokles tedy může být poměrně malý i při výběru největší hodnoty rozdílu.

3. Pravidlo maximálního zmenšení účelové funkce

Jedná se o další Dantzigovo pravidlo (1952), které vybírá klíčový sloupec tak, abychom v daném kroku zajistili co největší pokles hodnoty účelové funkce. Nejprve si pro všechny indexy splňující $z_j - c_j > 0$ spočítáme podíly $\frac{x_{Bi}}{\lambda_{ij}}$. Z nich poté pro každé x_i vybereme ten nejmenší, který vynásobíme s hodnotou v Δ -řádku. Do báze pak vstoupí proměnná, pro kterou je tento součin maximální. Tímto pravidlem tedy vybíráme proměnnou s takovým indexem k , že platí

$$(z_k - c_k) \frac{x_k}{\lambda_{kj}} = \max \left\{ (z_j - c_j) \frac{x_{Bi}}{\lambda_{ij}}, i \in B, j \notin B, z_j - c_j > 0, \lambda_{ij} > 0 \right\}.$$

4. Pravidlo nejstrmější hrany

Posledním pravidlem, které zde uvedeme, je pravidlo nejstrmější hrany (Dickson a Frederick (1960), Kuhn and Quandt (1963)). Tímto pravidlem vybíráme vstupující index tak, aby došlo k maximálnímu možnému poklesu na jednotku vzdálenosti podél hrany poklesu. Jestliže jsme v p -tém kroku algoritmu, vybíráme další bazický přípustný bod $x^{[p+1]}$ tak, abychom maximalizovali výraz

$$\frac{|c^T(x^{[p+1]} - x^{[p]})|}{\|x^{[p+1]} - x^{[p]}\|}. \quad (3.5)$$

Znění tohoto pravidla se v různých zdrojích liší. V [28] vybíráme klíčový sloupec tak, abychom maximalizovali výraz

$$\frac{c^T(x^{[p+1]} - x^{[p]})}{\|c\| \|x^{[p+1]} - x^{[p]}\|},$$

což je kosinus úhlu sevřeného vektory c a $(x^{[p+1]} - x^{[p]})$. Těmito dvěma verzemi však vybereme stejný klíčový sloupec, neboť dělení $\|c\|$ kritérium neovlivní.

Výše uvedená pravidla jsou nejčastěji využívanými pravidly. Ač se může zdát, že pravidlo maximálního zmenšení účelové funkce a pravidlo nejstrmější hrany urychlují výpočet, nemusí tomu tak být. Obě pravidla totiž vyžadují výpočet hodnoty všech proměnných, které

mohou vstoupit do báze. Je možné, že jimi dosáhneme menšího počtu iterací, jednotlivé iterace jsou však s využitím těchto pravidel náročnější. Pokles počtu iterací většinou nevyhradí jejich větší náročnost. Tato pravidla také ovlivňují proměnnou, která z báze vystoupí. Při ručních výpočtech v této práci bude využito Dantzigovo pravidlo. Výše uvedená pravidla si ukážeme na následujícím příkladu.

Příklad 3.7. Pravidla pro výběr klíčového sloupce.

Pro zadanou minimalizační úlohu sestavte výchozí simplexovou tabulku. Dále rozhodněte, jaký klíčový sloupec bychom vybrali pomocí všech pravidel uvedených výše.

$$\begin{aligned}
 \min \quad & -x_1 - 2x_2 - 3x_3 \\
 \text{za podmíněk} \quad & -x_1 + x_2 + x_3 + x_4 = 2 \\
 & x_1 + x_2 + x_3 + x_5 = 4 \\
 & x_3 + x_6 = 1 \\
 & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
 \end{aligned}$$

Řešení. Začneme se sestavením výchozí simplexové tabulky. Úloha je již v kanonickém tvaru, snadno tedy určíme výchozí bazický přípustný bod. Výchozí tabulka je zobrazena v Tabulce 3.8.

c_B	B	x_1	x_2	x_3	x_4	x_5	x_6	x_B	$\frac{x_B}{\lambda}$
0	x_4	-1	1	1	1	0	0	2	
0	x_5	1	1	1	0	1	0	4	
0	x_6	0	0	1	0	0	1	1	
	c	-1	-2	-3	0	0	0		
	z_k	0	0	0	0	0	0		
	Δ	1	2	3	0	0	0	0	

Tabulka 3.8: Výchozí tabulka pro ilustraci pravidel výběru klíčového sloupce

V simplexové tabulce vidíme, že v Δ -řádku jsou tři kladná čísla. Máme tedy tři kandidáty na výběr klíčového sloupce, mezi kterými s využitím různých pravidel rozhodneme.

1. Podle náhodného pravidla bychom vybrali zcela náhodně, např. druhý sloupec. Podle pravidla nejmenšího indexu bychom se podívali na sloupec s nejmenším indexem, ve kterém se v Δ -řádku vyskytuje kladné číslo. Zde bychom tedy vybrali první sloupec.
2. Podle Dantzigova pravidla vybereme největší kladné číslo, zde třetí sloupec.
3. Pro pravidlo maximálního zmenšení účelové funkce si musíme napočítat podíly x_B/λ pro všechny kandidáty na klíčový sloupec. Výsledky uvádíme v Tabulce 3.9. V každém sloupci tabulky je zakroužkován nejmenší podíl. V Tabulce 3.9 tedy vidíme, který řádek by pro jednotlivé sloupce opustil bázi. Např. pro výběr klíčového sloupce x_1 by bázi opustila proměnná ve druhém řádku, tj. x_5 . Tyto vybrané podíly

B	x_B	x_B/λ_{i1}	x_B/λ_{i2}	x_B/λ_{i3}
x_4	2	—	②	2
x_5	4	④	4	4
x_6	1	—	—	①
Δ	0	1	2	3
$\Delta \cdot x_B/\lambda$		$1 \cdot 4 = 4$	$2 \cdot 2 = 4$	$3 \cdot 1 = 3$

Tabulka 3.9: Využití pravidla maximálního zmenšení účelové funkce

nyní vynásobíme příslušnými čísly v Δ -řádku, což je naznačené v posledním řádku tabulky. Z tohoto řádku podle pravidla maximálního zmenšení účelové funkce vybereme největší číslo. Čísla, která jsme tímto způsobem získali, nám říkají, o kolik se zmenší účelová funkce vybráním daného sloupce a řádku. Pro první a druhý sloupec se účelová funkce zmenší o 4 (tj. na -4), v tomto případě tedy vybereme jeden z nich. Pro třetí sloupec by se hodnota zmenšila pouze na -3 .

4. Využití pravidla nejstrmější hrany vyžaduje nejvíce pomocných výpočtů. Musíme si totiž vypočítat souřadnice všech možných bodů, do kterých se můžeme v dalším kroku metody dostat. Nepotřebujeme nutně celé tabulky, postačí nám vždy výpočet sloupce se souřadnicemi x_B . Z tohoto důvodu jsou v Tabulce 3.10 uvedeny pouze báze a sloupec x_B pro všechny možné další bazické přípustné body. První část tabulky odpovídá situaci, kdy do báze vstoupí proměnná x_1 a vystoupí x_5 . Druhá část odpovídá situaci, ve které proměnná x_2 vystřídá v bázi proměnnou x_4 . Poslední část tabulky zobrazuje poslední možnost, do které se můžeme dostat, a to nahrazení proměnné x_6 v původní bázi proměnnou x_3 .

B	x_B	B	x_B	B	x_B
x_4	6	x_2	2	x_4	1
x_1	4	x_5	2	x_5	3
x_6	1	x_6	1	x_3	1
Δ	-4	Δ	-4	Δ	-3

Tabulka 3.10: Využití pravidla nejstrmější hrany

Označme si výchozí bazický přípustný bod $x = [0, 0, 0, 2, 4, 1]$. V dalším kroku simplexové metody se tedy můžeme dostat do bodu $x^{[1]} = [4, 0, 0, 6, 0, 1]$, bodu $x^{[2]} = [0, 2, 0, 0, 2, 1]$ či do bodu $x^{[3]} = [0, 0, 1, 1, 3, 0]$. Pro každý z těchto bodů nyní vypočteme rozdíl $x_i - x$, $i = \{1, 2, 3\}$, tyto rozdíly jsou postupně $y^{[1]} = [4, 0, 0, 4, -4, 0]$, $y^{[2]} = [0, 2, 0, -2, 2, 0]$, $y^{[3]} = [0, 0, 1, -1, -1, -1]$. Nyní podle vzorce (3.5) vypočítá-

táme hodnoty daného výrazu pro jednotlivé body. Získáme tak

$$\begin{aligned}x^{[1]} &: \frac{(-1, -2, -3, 0, 0, 0)^T (4, 0, 0, 4, -4, 0)}{\|(4, 0, 0, 4, -4, 0)\|} = -\frac{4}{4\sqrt{3}} = -\frac{1}{\sqrt{3}}, \\x^{[2]} &: \frac{(-1, -2, -3, 0, 0, 0)^T (0, 2, 0, -2, 2, 0)}{\|(0, 2, 0, -2, 2, 0)\|} = -\frac{4}{2\sqrt{3}} = -\frac{2}{\sqrt{3}}, \\x^{[3]} &: \frac{(-1, -2, -3, 0, 0, 0)^T (0, 0, 1, -1, -1, -1)}{\|(0, 0, 1, -1, -1, -1)\|} = -\frac{3}{\sqrt{4}} = -\frac{3}{2}.\end{aligned}$$

Výraz v nezkráceném čitateli nám říká, jaká bude hodnota účelové funkce v novém bazickém přípustném bodu (tj. je stejná jako příslušné hodnoty v Δ -řádku v Tabulce 3.10). Výraz ve jmenovateli pak vyjadřuje vzdálenost bazických přípustných bodů, mezi kterými v tomto kroku přecházíme. Chceme, aby se hodnota účelové funkce na jednotkový krok zmenšila co nejvíce, podle pravidla nejstrmější hrany tedy vybereme situaci číslo 3, ve které je dalším bazickým přípustným bodem bod $x^{[3]}$. Pokles na jednotkový krok je totiž v tomto případě největší možný, neboť

$$|-3/2| > |-2/\sqrt{3}| > |-1/\sqrt{3}|.$$

△

3.2.2 Zacyklení simplexové metody

Simplexová metoda je pro nedegenerované úlohy konečná. Pro degenerované úlohy však může dojít k zacyklení, tj. k tomu, že se po několika krocích simplexového algoritmu dostaneme ke stejné tabulce, kterou jsme již při výpočtu získali předtím. Zacyklení je způsobeno výběrem pravidel, která při řešení využíváme. Výběr těchto pravidel ale musíme učinit před řešením úloh, simplexovou metodu bez jejich určení totiž nemůžeme „plně“ algoritmizovat. Ukázkou zacyklení simplexové metody si nyní ukážeme na Příkladu 3.8. V této části textu také uvedeme metody, pomocí kterých můžeme zacyklení zabránit. Více na toto téma je možné najít např. v [36].

Příklad 3.8. Zacyklení simplexové metody.

Příklad, který si uvedeme jako ilustraci zacyklení simplexové metody, vytvořil Václav Chvátal v roce 1983. Simplexovou metodou s využitím Dantzigova pravidla vyřešte tuto minimalizační úlohu. V případě rovnosti podílů x_B/λ vyberte vždy ten, který se nachází v nejvýše položeném řádku.

$$\begin{aligned}\min & \quad -10x_1 + 57x_2 + 9x_3 + 24x_4 \\ \text{za podmíněk} & \quad \frac{1}{2}x_1 - \frac{11}{2}x_2 - \frac{5}{2}x_3 + 9x_4 \leq 0 \\ & \quad \frac{1}{2}x_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3 + x_4 \leq 0 \\ & \quad x_1 \leq 1 \\ & \quad x_1, x_2, x_3, x_4 \geq 0\end{aligned}$$

Řešení. Úlohu nejprve převedeme na kanonický tvar přidáním tří nezáporných doplňkových proměnných označených s_1, s_2, s_3 . Matice A nyní obsahuje všechny sloupce jednotkové matice, můžeme tedy snadno sestavit výchozí simplexovou tabulku pro bázi tvořenou doplňkovými proměnnými. Tato tabulka je zobrazena v Tabulce 3.11. Při řešení této úlohy postupujeme obdobně jako v předchozích úlohách. V prvním kroku do báze vstoupí proměnná x_1 a díky pravidlu výběru nejmenšího indexu klíčového řádku vystoupí proměnná s_1 . Pokračování tohoto postupu zde nebudeme podrobněji komentovat, prvních šest kroků simplexového algoritmu je uvedeno v Tabulce 3.11.

Při postupu simplexové metody se při využití námi vybraných pravidel vždy můžeme dostat k dalšímu bodu, ve kterém je hodnota účelové funkce stejná. Pokud ale provedeme uvedených šest kroků algoritmu, dostaneme se k výchozí simplexové tabulce, v úloze tedy došlo k cyklu. Cyklus má v tomto příkladě délku šest a s námi zvolenými pravidly se z něj při výpočtu nedostaneme. Algoritmus by tak nikdy neskončil, stále by obíhal mezi jedním bodem vyjádřeným pomocí různých bází.

△

Nyní si můžeme položit otázku, jak se z cyklu dostat a zjistit, zda pro úlohu existuje řešení nebo zda je neohraničená. Pravidla (kromě náhodného), která jsme si představili v předchozí části textu, nám nezaručí, že se simplexová metoda nezacyklí. Proto si nyní představíme pravidla, která zacyklení zabrání. Pro zabránění cyklu bylo vyvinuto několik anticyklických metod. V této práci stručně popíšeme myšlenky tří z nich – *Blandova pravidla* (někdy též označovaného jako pravidlo nejmenších indexů), *lexikografického pravidla* a *perturbační metody*. V [41] lze kromě dalších anticyklických pravidel najít také obecné vlastnosti, které zacyklení zabrání. Důkazy nezacyklení Blandova a lexikografického pravidla lze najít např. v [43]. Důkaz věty, která říká, že při využití perturbační metody nejsou v úloze degenerované body, uvádí např. [38].

1. Blandovo pravidlo (pravidlo nejmenších indexů)

Blandovo pravidlo, nazvané podle svého tvůrce, bylo zveřejněno v roce 1977. Podle Blandova pravidla vybíráme klíčový sloupec s nejmenším indexem a klíčový řádek (v případě rovnosti více podílů) taktéž. V případě klíčového řádku nejde o jeho pořadí v tabulce, ale o index bazické proměnné, které odpovídá. Postupujeme tedy, jak název napovídá, po nejmenších indexech proměnných, přičemž pořadí proměnných je dáno záhlavím simplexové tabulky. Toto pravidlo nám sice zajistí, že nedojde k cyklu, z hlediska rychlosti výpočtu je však méně efektivní, než jiná pravidla pro výběr klíčového sloupce a řádku, a proto se v praxi používá jen zřídka. Blandovo pravidlo je ilustrováno na Příkladu 3.9.

2. Lexikografické pravidlo

Lexikografické pravidlo publikovali v roce 1955 Dantzig, Orden a Wolfe. Lexikografické pravidlo, na rozdíl od Blandova pravidla, neudává pravidlo pro výběr klíčového sloupce, doporučuje ovšem využít Dantzigovo pravidlo (tj. vybírat největší kladnou hodnotu v Δ -řádku). Při výběru klíčového řádku (v případě shody podílů) pak podle tohoto pravidla vynásobíme celé řádky příslušnými prvky $1/\lambda$ z klíčového sloupce, čímž na pozici potenciálních pivotů dostaneme jedničku. Z řádků, které tímto způsobem získáme, pak vybereme ten lexikograficky nejmenší. Ze dvou vektorů je

c_B	B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
0	s_1	$\textcircled{1/2}$	$-11/2$	$-5/2$	9	1	0	0	0	$0 \rightarrow$
0	s_2	$1/2$	$-3/2$	$-1/2$	1	0	1	0	0	0
0	s_3	1	0	0	0	0	0	1	1	1
	c	-10	57	9	24	0	0	0	0	
	z_k	0	0	0	0	0	0	0	0	
	Δ	$10 \uparrow$	-57	-9	-24	0	0	0	0	
	x_1	1	-11	-5	18	2	0	0	0	-
	s_2	0	$\textcircled{4}$	2	-8	-1	1	0	0	$0 \rightarrow$
	s_3	0	11	5	-18	-2	0	1	1	1/11
	Δ	0	$53 \uparrow$	41	-204	-20	0	0	0	
	x_1	1	0	$\textcircled{1/2}$	-4	$-3/4$	$11/4$	0	0	$0 \rightarrow$
	x_2	0	1	$1/2$	-2	$-1/4$	$1/4$	0	0	0
	s_3	0	0	$-1/2$	4	$3/4$	$-11/4$	1	1	-
	Δ	0	0	$29/2 \uparrow$	-98	$-27/4$	$-53/4$	0	0	
	x_3	2	0	1	-8	$-3/2$	$11/2$	0	0	-
	x_2	-1	1	0	$\textcircled{2}$	$1/2$	$-5/2$	0	0	$0 \rightarrow$
	s_3	1	0	0	0	0	0	1	1	-
	Δ	-29	0	0	$18 \uparrow$	15	-93	0	0	
	x_3	-2	4	1	0	$\textcircled{1/2}$	$-9/2$	0	0	$0 \rightarrow$
	x_4	$-1/2$	$1/2$	0	1	$1/4$	$-5/4$	0	0	0
	s_3	1	0	0	0	0	0	1	1	-
	Δ	-20	-9	0	0	$21/2 \uparrow$	$-141/2$	0	0	
	s_1	-4	8	2	0	1	-9	0	0	-
	x_4	$1/2$	$-3/2$	$-1/2$	1	0	$\textcircled{1}$	0	0	$0 \rightarrow$
	s_3	1	0	0	0	0	0	1	1	-
	Δ	22	-93	-21	0	0	$24 \uparrow$	0	0	
	s_1	$\textcircled{1/2}$	$-11/2$	$-5/2$	9	1	0	0	0	$0 \rightarrow$
	s_2	$1/2$	$-3/2$	$-1/2$	1	0	1	0	0	0
	s_3	1	0	0	0	0	0	1	1	1
	Δ	$10 \uparrow$	-57	-9	-24	0	0	0	0	

Tabulka 3.11: Chvátalův příklad zacyklení

lexikograficky menší ten, který má na první pozici, ve které se vektory liší, menší číslo, tj. např. vektor $u = (0, 3, 1)$ je lexikograficky menší než vektor $v = (0, 5, 0)$, neboť první složka vektorů je stejná (rovna 0) a ve druhé složce má vektor u menší číslo. Výpočet pomocí lexikografického pravidla je ukázán v Příkladu 3.10.

3. Perturbační metoda

Perturbační metoda byla představena v roce 1952 matematikem Abrahamem Charneem. Myšlenkou perturbační metody je malá změna pravé strany tak, aby vznikla nedegenerovaná úloha. Metoda postupuje tak, že ke každému z prvků b_i přičteme $1 > \varepsilon_i > 0$, které je menší než všechna kladná čísla, která se vyskytují v úloze a platí $\varepsilon_i > \varepsilon_{i+1}$. Jednou z možností (viz [38]) je zvolit např. $\varepsilon_1 = \varepsilon > 0$, $\varepsilon_2 = \varepsilon^2, \dots, \varepsilon_m = \varepsilon^m$. Místo vektoru x_B tedy počítáme s vektorem $x_B + \varepsilon$. Touto perturbací získáme tzv. ε -úlohu, která je nedegenerovaná, a tedy v ní nedojde k zacyklení. Nedegenerovanost ε -úlohy je způsobena tím, že díky perturbacím se v žádném bodě v \mathbb{R}^n neprotne více než n nadrovin. Po vyřešení ε -úlohy (s využitím simplexové metody) získáme řešení původní úlohy odstraněním perturbací. Tuto metodu si ukážeme na Příkladu 3.11.

Degenerovanost úlohy nezajistí cyklus vždy, jedná se spíše o výjimečný případ. Příklady zacyklení není lehké sestavit. Pro představu, v roce 1963 Judin a Goľštejn dokázali, že minimální délka cyklu je 6. Další zajímavostí je, že každý příklad, ve kterém existuje řešení a k cyklu dojde mimo toto řešení, musí mít alespoň 6 proměnných a 3 omezení, což v roce 1969 dokázali Marshall a Suurbale. Více o zacyklení čtenář najde např. v [20] a [32].

Při řešení úloh vycházejících z praxe se tak se zacyklením prakticky nesetkáme. To může být způsobeno i vlivem výpočetních zaokrouhlovacích chyb, díky kterým se do cyklu nedostaneme. K dalšímu druhu cyklů, tzv. výpočetním cyklům, může naopak dojít i pro nedegenerované úlohy, a to kvůli zaokrouhlování při využití výpočetních softwarů.

Příklad 3.9. Blandovo pravidlo.

Pro úlohu z Příkladu 3.8 využijte Blandovo pravidlo.

Řešení. Pokud se znovu podíváme na Tabulku 3.11, všimneme si, že až do pátého kroku výběrem podle Dantzigova pravidla s rozšířením pro výběr klíčového řádku a výběrem podle Blandova pravidla vybereme stejné klíčové sloupce i řádky. Prvních pět simplexových tabulek by tedy vypadalo úplně stejně jako v Příkladu 3.8, není nutné je zde znovu uvádět. V šestém kroku však podle Blandova pravidla vstoupí do báze x_1 (ne s_2) a postup dalšího výpočtu se již liší. Postup pomocí Blandova pravidla je uveden v Tabulce 3.12.

S využitím Blandova pravidla najdeme řešení úlohy, neboť v Δ -řádku jsou pro nebazické proměnné pouze záporná čísla. K nalezení řešení jsme potřebovali osm kroků simplexové metody. Řešením úlohy v kanonickém tvaru je bod $[1, 0, 1, 0, 2, 0, 0]$, minimální hodnota účelové funkce na přípustné množině je rovna -1 . Pro původní úlohu je řešením bod $x^* = [1, 0, 1, 0]$.

△

Příklad 3.10. Lexikografické pravidlo.

Pro úlohu z Příkladu 3.8 využijte lexikografické pravidlo.

Řešení. Stejně jako v Příkladu 3.9 se podíváme, zda bychom s využitím lexikografického pravidla postupovali podobně jako s využitím Dantzigova pravidla. V Tabulce 3.11 vidíme,

c_B	B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
	s_1	-4	8	2	0	1	-9	0	0	-
	x_4	1/2	-3/2	-1/2	1	0	1	0	0	0 \rightarrow
	s_3	1	0	0	0	0	0	1	1	1
	Δ	22 \uparrow	-93	-21	0	0	24	0	0	
	s_1	0	-4	-2	8	1	-1	0	0	-
	x_1	1	-3	-1	2	0	2	0	0	-
	s_3	0	3	1	-2	0	-2	1	1	1 \rightarrow
	Δ	0	-27	1 \uparrow	-44	0	-20	0	0	
	s_1	0	2	0	4	1	-5	2	2	
	x_1	1	0	0	0	0	0	1	1	
	x_3	0	3	1	-2	0	-2	1	1	
	Δ	0	-30	0	-42	0	-18	-1	-1	

Tabulka 3.12: Blandovo pravidlo pro Chvátalův příklad zacyklení

že již v prvním kroku algoritmu máme na výběr dva podíly. Podle lexikografického pravidla tedy vynásobíme řádek hodnotou $1/\lambda$, čímž získáme řádky, které pro přehlednost uvedeme v Tabulce 3.13.

x_1	x_2	x_3	x_4	s_1	s_2	s_3
1	-11	-5	18	2	0	0
1	-3	-1	2	0	2	0

Tabulka 3.13: Výpočet klíčového řádku pomocí lexikografického pravidla

První řádek v této tabulce je lexikograficky menší, neboť čísla na první pozici jsou stejná a na druhé pozici je v prvním řádku menší číslo. V tomto případě tedy vybereme lexikografickým pravidlem stejný řádek jako při výpočtu Příkladu 3.8. Jak výchozí tabulka, tak následující tabulka se tedy nebudou lišit, neboť ve druhém kroku simplexové metody nedojde ke shodě podílů. Ve třetím kroku již k rozdílnosti postupů dojde, výpočet pomocí lexikografického pravidla uvádíme v Tabulce 3.14.

V prvním kroku uvedeném v Tabulce 3.14 (ve 3. kroku simplexové metody pro Chvátalův příklad) se pro odchod z báze rozhodujeme mezi proměnnými x_1 a x_2 . Podle Blandova pravidla bychom vybrali x_1 , neboť má menší index. Podle lexikografického pravidla vynásobíme oba řádky číslem 2 ($1/\lambda$), čímž získáme vektory zobrazené v Tabulce 3.15.

Na první pohled vidíme, že druhý vektor je lexikograficky menší, z báze tedy odchází x_2 . V dalším kroku již máme pouze jednu možnost výběru pivota. Tou se dostaneme k řešení. Toto řešení je samozřejmě stejné, jako v Příkladu 3.9, v simplexové tabulce jsou pouze prohozené řádky. Pro výpočet Chvátalova příkladu lexikografickým pravidlem jsme potřebovali pět kroků simplexové metody.

c_B	B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
	x_1	1	0	1/2	-4	-3/4	11/4	0	0	0
	x_2	0	1	1/2	-2	-1/4	1/4	0	0	0 \rightarrow
	s_3	0	0	-1/2	4	3/4	-11/4	1	1	-
	Δ	0	0	29/2 \uparrow	-98	-27/4	-53/4	0	0	
	x_1	1	-1	0	-2	-1/2	5/2	0	0	-
	x_3	0	2	1	-4	-1/2	1/2	0	0	-
	s_3	0	1	0	2	1/2	-5/2	1	1	2 \rightarrow
	Δ	0	-29	0	-40	1/2 \uparrow	-41/2	0	0	
	x_1	1	0	0	0	0	0	1	1	
	x_3	0	3	1	-2	0	-2	1	1	
	s_1	0	2	0	4	1	-5	2	2	
	Δ	0	-30	0	-42	0	-18	-1	-1	

Tabulka 3.14: Lexikografické pravidlo pro Chvátalův příklad zacyklení

x_1	x_2	x_3	x_4	s_1	s_2	s_3
2	0	1	-8	-3/2	11/2	0
0	2	1	-4	-1/2	1/2	0

Tabulka 3.15: Výpočet klíčového řádku pomocí lexikografického pravidla

△

Příklad 3.11. Perturbační metoda.

Pro úlohu z Příkladu 3.8 využijte perturbační metodu.

Řešení. Pro výpočet perturbační metodou zvolíme $e_1 = e$, $e_2 = e^2$, $e_3 = e^3$, kde pro e platí $1/2 \gg e > 0$ (symbolem \gg rozumíme „mnohem menší“). Vektor x_B je tedy roven

$$b + \varepsilon = (e, e^2, 1 + e^3).$$

Nyní obdobně jako v Příkladu 3.8 převedeme Chvátalovu úlohu na kanonický tvar a sestavíme výchozí simplexovou tabulku. Ta se bude od Tabulky 3.11 lišit pouze sloupcem x_B . Pro přehlednost zde neuvádíme sloupec c_B . Výchozí simplexová tabulka pro ε -úlohu je spolu s výpočtem uvedena v Tabulce 3.16.

V prvním kroku do báze vstoupí proměnná x_1 . Bázi v tomto kroku opustí proměnná s_2 , neboť platí $1 + e^3 > 2e > 2e^2$. Ve druhém kroku máme při výběru pivota pouze jednu možnost, do báze musí vstoupit x_3 a odejít z báze musí s_3 . Tímto postupem jsme se v pouhých 3 krocích dostali k řešení ε -úlohy. K řešení původní úlohy se dostaneme vynecháním perturbací. „Vynulováním“ všech e v poslední simplexové tabulce bychom se dostali ke stejné tabulce, kterou jsme viděli při využití Blandova a lexikografického pravidla. Tímto postupem jsme Chvátalovu úlohu vyřešili ve třech krocích.

B	x_1	x_2	x_3	x_4	s_1	s_2	s_3	x_B	$\frac{x_B}{\lambda}$
s_1	1/2	-11/2	-5/2	9	1	0	0	e	$2e$
s_2	1/2	-3/2	-1/2	1	0	1	0	e^2	$2e^2 \rightarrow$
s_3	1	0	0	0	0	0	1	$1+e^3$	$1+e^3$
c	-10	57	9	24	0	0	0	0	
z_k	0	0	0	0	0	0	0	0	
Δ	10 \uparrow	-57	-9	-24	0	0	0	0	
s_1	0	-4	-2	8	1	-1	0	$e - e^2$	-
x_1	1	-3	-1	2	0	2	0	$2e^2$	-
s_3	0	3	1	-2	0	-2	1	$1 - e^2(2+e)$	$1 - e^2(2+e) \rightarrow$
Δ	0	-27	1 \uparrow	-44	0	-20	0	$-20e^2$	
s_1	0	2	0	4	1	-5	2	$2(1+e^3) + e - 5e^2$	
x_1	1	0	0	0	0	0	1	$1+e^3$	
x_3	0	3	1	-2	0	-2	1	$1 - 2e^2 + e^3$	
Δ	0	-30	0	-42	0	-18	-1	$-1 - 18e^2 - e^3$	

Tabulka 3.16: Perturbační metoda pro Chvátalův příklad zacyklení

△

Více o perturbační metodě najde čtenář např. v [23] a [38].

3.3 Výchozí bazický přípustný bod

Ve všech předchozích úlohách jsme snadno našli výchozí bazický přípustný bod, neboť matice A vždy obsahovala všechny sloupce jednotkové matice. To se ale nemusí stát vždy. Jak tedy výchozí bazický přípustný bod najít? V Kapitole 2 jsme viděli, že k nalezení bazického přípustného bodu lze využít vztah $x_B = A_B^{-1}b$. Tato rovnost nám ovšem nezajistí přípustnost daného bodu. Hledání výchozího bazického přípustného bodu by nás tak mohlo stát velké množství výpočtů i času. V této části si proto ukážeme, jak najít výchozí bazický přípustný bod efektivněji.

V literatuře (např. v [1] a [15]) se nejčastěji setkáme se dvěma metodami pro výpočet výchozího bazického přípustného bodu – metodou velkého M (penalizační metodou) a metodou umělé báze (dvoufázovou metodou). V této práci si obě metody představíme a ukážeme na jednoduchých příkladech. Obě tyto metody využívají k nalezení výchozího bazického přípustného bodu řešení upravené úlohy.

3.3.1 Metoda velkého M

Začneme metodou velkého M, která místo původní úlohy (1.1) řeší úlohu ve tvaru

$$\left. \begin{array}{l} \min \quad c^T x + m^T w \\ \text{za podmíněk} \quad Ax + Iw = b \\ \quad \quad \quad x \geq 0, w \geq 0, \end{array} \right\} \quad (3.6)$$

kde I je jednotková matice a $m = (M, M, \dots, M)^T$ je vektor konstant $M > 0$, jenž je větší než všechna čísla v úloze. Vzhledem k tomu, že dopředu nevíme, jaká čísla se v průběhu výpočtu objeví, počítáme s hodnotou M jako s parametrem. Úlohu rozšíříme tak, aby obsahovala jednotkovou matici. Pokud je v úloze nějaký sloupec jednotkové matice (nebo ho násobením konstantou snadno získáme), můžeme tento sloupec využít a přidat tak méně proměnných w . Cílem této metody je zavést umělé proměnné tak, abychom je při řešení dostali z báze (tj. aby ve výsledném bodě platilo $w = 0$). Proto je potřebné přičíst umělou proměnnou s velkým koeficientem M , který zajistí, že v případě řešitelnosti původní úlohy, umělá proměnná jistě bázi opustí.

Pokud řešení rozšířené úlohy (3.6) závisí na parametru M (tj. umělá proměnná zůstane v bázi s nenulovou hodnotou), původní úloha je nepřijatelná. V opačném případě vyřešením rozšířené úlohy získáme výchozí bazický přípustný bod původní úlohy. Jestliže při výpočtu umělou proměnnou z báze vyloučíme, nemusíme již počítat jí příslušný sloupec, neboť optimální řešení jistě nenajdeme tak, že ji do báze opět vrátíme. Po vyloučení umělých proměnných z báze postupujeme nám dobře známým způsobem. Nyní již řešíme původní úlohu s příslušným výchozím bazickým přípustným bodem. Tuto metodu si nyní ukážeme na příkladu.

Příklad 3.12. Metoda velkého M .

S využitím metody velkého M najděte výchozí bazický přípustný bod pro zadanou minimalizační úlohu.

$$\left. \begin{array}{l} \min \quad x_1 + 4x_2 + 2x_3 \\ \text{za podmíněk} \quad x_1 - x_2 + x_3 = 0 \\ \quad \quad \quad -x_2 + 4x_3 = 1 \\ \quad \quad \quad x_1, x_2, x_3 \geq 0 \end{array} \right\}$$

Řešení. Na první pohled si můžeme všimnout, že proměnná x_1 je obsažena pouze v prvním omezení, navíc s koeficientem 1. Tento sloupec můžeme využít při tvorbě jednotkové matice v matici A . Ke druhé rovnosti pak přidáme umělou proměnnou (ozn. w), čímž získáme i druhý sloupec jednotkové matice. Rozšířená úloha je tedy ve tvaru

$$\left. \begin{array}{l} \min \quad x_1 + 4x_2 + 2x_3 + Mw \\ \text{za podmíněk} \quad x_1 - x_2 + x_3 = 0 \\ \quad \quad \quad -x_2 + 4x_3 + w = 1 \\ \quad \quad \quad x_1, x_2, x_3, w \geq 0. \end{array} \right\}$$

Tuto úlohu umíme vyřešit simplexovou metodou, jejíž postup je uveden v Tabulce 3.17. Po sestavení tabulky je v Δ -řádku pouze jedno kladné číslo, a to ve třetím sloupci. Stejně jako v předchozích příkladech pak vybereme klíčový řádek a provedeme Gaussovu eliminaci. Nová báze je tvořena proměnnými x_3 a w . K eliminaci čísla $-1 + 4M$ přičteme k Δ -řádku

klíčový řádek vynásobený číslem $1 - 4M$. V dalším kroku se tak dostaneme k tabulce, ve které je opět v Δ -řádku jedno kladné číslo. Do báze tedy vstoupí x_2 a vystoupí umělá proměnná w . Eliminací s využitím pivota (číslo 3) pak získáme poslední tabulku.

V této tabulce již nemusíme počítat sloupec pro w , v poslední tabulce je proto nevyplněný. V Δ -řádku jsou všechny hodnoty pro nebazické proměnné záporné, simplexový algoritmus tedy končí. V bázi nemáme žádnou umělou proměnnou, nalezený bod je tak výchozím bazickým přípustným bodem původní úlohy. S touto simplexovou tabulkou můžeme nyní pokračovat. Z posledního kroku algoritmu vidíme, že již nelze vybrat klíčový sloupec. Nalezený výchozí bazický přípustný bod je tak zároveň řešením původní úlohy.

c_B	B	x_1	x_2	x_3	w	x_B	$\frac{x_B}{\lambda}$
1	x_1	1	-1	①	0	0	$0 \rightarrow$
M	w	0	-1	4	1	1	$1/4$
	c	1	4	2	M		
	z_k	1	$-1 - M$	$1 + 4M$	M		
	Δ	0	$-5 - M$	$-1 + 4M \uparrow$	0	M	
	x_3	1	-1	1	0	0	-
	w	-4	③	0	1	1	$1/3 \rightarrow$
	Δ	$1 - 4M$	$-6 + 3M \uparrow$	0	0	M	
	x_3	$-1/3$	0	1		$1/3$	
	x_2	$-4/3$	1	0		$1/3$	
	Δ	-7	0	0		2	

Tabulka 3.17: Simplexová metoda pro Příklad 3.12

△

3.3.2 Metoda umělé báze

Druhou metodou, kterou můžeme využít pro nalezení výchozího bazického přípustného bodu, je metoda umělé báze. Princip této metody opět spočívá v tom, že původní problém rozšíříme na problém, pro něj snadno najdeme výchozí bazický přípustný bod. Řešením rozšířené úlohy pak zjistíme, zda je původní úloha nepřipustná či zda existuje výchozí bazický přípustný bod, který zároveň najdeme. Postupujeme následovně.

Nejprve úlohu (1.1) převedeme do kanonického tvaru. Poté tuto úlohu rozšíříme o umělé proměnné $w_1, \dots, w_m \geq 0$ tak, že platí $Ax + Iw = b$. Stejně jako pro metodu velkého M nemusíme nutně přidat právě m proměnných. Opět stačí přidat umělé proměnné tak, aby matice I obsahovala sloupce jednotkové matice, které nejsou v matici A .

Simplexovým algoritmem pak vyřešíme úlohu

$$\begin{array}{ll} \min & w_1 + \dots + w_m \\ \text{za podmínek} & Ax + Iw = b \\ & x, w \geq 0. \end{array}$$

Tuto část postupu dvoufázové metody označujeme jako první fázi. Výchozí bazický přípustný bod v první fázi najdeme snadno – je jím bod $x = 0$ a $w = b$. Vzhledem k tomu, že $w_1, \dots, w_m \geq 0$, nemůže být tato úloha neohrazená. V závislosti na nalezeném řešení $x^{[0]}$, w^* mohou na konci první fáze nastat tři situace.

1. Platí $\sum_{i=1}^m w_i^* = 0$ a všechny umělé proměnné jsou mimo bázi. V tomto případě je tedy $w^* = 0$ a nalezený $x^{[0]}$ je výchozí bazický přípustný bod původní úlohy.
2. Platí $\sum_{i=1}^m w_i^* = 0$, ale v bázi zůstala alespoň jedna umělá proměnná (s nulovou hodnotou). V tomto případě jsme získali výchozí bazický přípustný bod původní úlohy, ve výchozí bázi je však stále umělá proměnná w . K této situaci dojde v případě, kdy původní úloha obsahuje nadbytečná omezení nebo je degenerovaná. Nyní je možné postupovat různě. Můžeme se snažit pokračovat se simplexovou tabulkou z první fáze tak dlouho, dokud nebudou všechny umělé proměnné mimo bázi. Tímto způsobem získáme ve druhé fázi úlohu s menším počtem proměnných. V případě nadbytečných omezení ale nemusí být vyvedení všech umělých proměnných z báze možné, což uvidíme v Příkladu 3.15.

Můžeme také pokračovat do druhé fáze s umělou proměnnou v bázi. Jakmile ji při řešení druhé fáze z báze vyloučíme, můžeme jí příslušný sloupec vyškrtnout, neboť optimální řešení, stejně jako v metodě velkého M, určitě nenajdeme jejím zavedením zpět do báze.

3. Jestliže je $\sum_{i=1}^m w_i^* > 0$, původní úloha není přípustná.

Pokud nastala první nebo druhá situace, můžeme pokračovat s druhou fází metody. V té ze simplexové tabulky odstraníme všechny sloupce umělých proměnných, které nejsou v bázi. Dále nahradíme hodnotu pomocné účelové funkce $w_1 + \dots + w_m$ hodnotou účelové funkce původní úlohy (tj. přepočítáme Δ -řádek). Tímto způsobem získáme výchozí simplexovou tabulku pro původní úlohu, kterou vyřešíme nám již dobře známým způsobem. Nastane-li druhá situace, musíme si při řešení druhé fáze dát pozor, aby hodnota umělých proměnných, které zůstaly v bázi, nebyla kladná. To by totiž způsobilo nepřípustnost daného bazického přípustného bodu.

Dvoufázovou metodu si nyní ukážeme na třech příkladech. Nejprve ji ilustrujeme na stejném příkladu jako metodu velkého M.

Příklad 3.13. Dvoufázová metoda.

S využitím dvoufázové metody vyřešte minimalizační úlohu zadanou v Příkladu 3.12.

Řešení. Stejně jako v Příkladu 3.12 začneme převedením na kanonický tvar a přidáním umělé proměnné. V tomto případě však začneme první fází, ve které řešíme úlohu

$$\begin{aligned} \min \quad & w \\ \text{za podmínek} \quad & x_1 - x_2 + x_3 = 0 \\ & -x_2 + 4x_3 + w = 1 \\ & x_1, x_2, x_3, w \geq 0. \end{aligned}$$

Sestavíme tedy výchozí simplexovou tabulku, která je společně s dvěma kroky algoritmu zobrazená v Tabulce 3.18.

c_B	B	x_1	x_2	x_3	w	x_B	$\frac{x_B}{\lambda}$
0	x_1	1	-1	①	0	0	$0 \rightarrow$
1	w	0	-1	4	1	1	$1/4$
	c	0	0	0	1		
	z_k	0	-1	4	1		
	Δ	0	-1	$4 \uparrow$	0	1	
<hr/>							
	x_3	1	-1	1	0	0	-
	w	-4	③	0	1	1	$1/3 \rightarrow$
	Δ	-4	3	0	0	1	
<hr/>							
	x_3	-1/3	0	1		1/3	
	x_2	-4/3	1	0		1/3	
	Δ	0	0	0		0	

Tabulka 3.18: První fáze dvoufázové metody pro Příklad 3.13

Čtenář si jistě povšimne, že v tomto příkladu postupuje algoritmus podobně jako v Příkladu 3.12. Jediný rozdíl je v účelové funkci, v Tabulce 3.18 je tedy jiný Δ -řádek. Ve druhém kroku první fáze vystupuje z báze umělá proměnná w , jí příslušný sloupec tak už nemusíme počítat. Jsme tedy v situaci, kdy $w^* = 0$. V bázi navíc není žádná umělá proměnná. Nalezli jsme tak bazický přípustný bod původní úlohy $x^{[0]} = [0, 1/3, 1/3]$. Nyní můžeme přejít ke druhé fázi metody, která je zobrazena v Tabulce 3.19.

c_B	B	x_1	x_2	x_3	x_B	$\frac{x_B}{\lambda}$
2	x_3	-1/3	0	1	1/3	
4	x_2	-4/3	1	0	1/3	
	c	1	4	2		
	z	-6	4	2		
	Δ	-7	0	0	2	

Tabulka 3.19: Druhá fáze dvoufázové metody pro Příklad 3.13

Ve druhé fázi řešíme původní úlohu zadanou v Příkladu 3.12. Tuto tabulku jsme získali přepočítáním řádků c , z a Δ pro účelovou funkci původní úlohy. Vidíme, že v Δ -řádku již pro nebazické proměnné není žádné nezáporné číslo. Výchozí bazický přípustný bod je tak zároveň řešením úlohy. Čtenář si jistě všimne, že jsme se dostali ke stejné tabulce, jako v případě využití metody velkého M, tj. k Tabulce 3.17.

△

Z příkladu vypočítaného jak metodou velkého M, tak dvoufázovou metodou, je zřejmé, že obě metody jsou založeny na stejném principu. Vždy totiž přidáme umělé proměnné s cílem je vyvést z báze a získat tak výchozí bazický přípustný bod původní úlohy. Rozdíl je v tom, že v metodě velkého M pracujeme s původní úlohou, ve které rozšíříme účelovou funkci. Po vyloučení umělých proměnných z báze proto nemusíme přepočítávat Δ -řádek. Na druhou stranu ve dvoufázové metodě si problém rozdělíme na dvě fáze, přičemž v první fázi řešíme úlohu s novou účelovou funkcí, ve druhé pak s účelovou funkcí původní úlohy. Dalším rozdílem je, že při využití metody velkého M pracujeme s parametrem M, jehož hodnotu neznáme. V případě využití dvoufázové metody pracujeme pouze s čísly. Dvoufázovou metodu si nyní procvičíme na následujících dvou příkladech.

Příklad 3.14. Dvoufázová metoda – nepřípustná úloha.

S využitím dvoufázové metody vyřešte následující minimalizační úlohu.

$$\begin{aligned} \min & && -3x_1 - 5x_2 \\ \text{za podmíněk} & && x_1 - 2x_2 \leq 6 \\ & && 2x_1 + x_2/2 \leq 7 \\ & && x_1 + x_2 \geq 16 \\ & && x_1, x_2 \geq 0 \end{aligned}$$

Řešení. Stejně jako obvykle začneme převedením na kanonický tvar a přidáním umělé proměnné. Tím dostaneme následující úlohu, kterou vyřešíme v první fázi výpočtu.

$$\begin{aligned} \min & && w \\ \text{za podmíněk} & && x_1 - 2x_2 + s_1 = 6 \\ & && 2x_1 + x_2/2 + s_2 = 7 \\ & && x_1 + x_2 - s_3 + w = 16 \\ & && x_1, x_2, s_1, s_2, s_3, w \geq 0 \end{aligned}$$

Pro tuto úlohu nyní sestavíme výchozí simplexovou tabulku, která je společně s postupem algoritmu zobrazená v Tabulce 3.20. Při výpočtu postupujeme standardně. V prvním kroku máme na výběr dvě proměnné, které by mohly vstoupit do báze, vybereme tedy náhodně. Kdybychom v prvním kroku vybrali místo x_1 proměnnou x_2 , úlohu bychom vyřešili o jeden krok rychleji.

c_B	B	x_1	x_2	s_1	s_2	s_3	w	x_B	$\frac{x_B}{\lambda}$
0	s_1	1	-2	1	0	0	0	6	6
0	s_2	2	1/2	0	1	0	0	7	7/2 →
1	w	1	1	0	0	-1	1	16	16
	c	0	0	0	0	0	1		
	z_k	1	1	0	0	-1	1		
	Δ	1 ↑	1	0	0	-1	0	16	
	s_1	0	-9/4	1	-1/2	0	0	5/2	-
	x_1	1	1/4	0	1/2	0	0	7/2	14 →
	w	0	3/4	0	-1/2	-1	1	25/2	50/3
	Δ	0	3/4 ↑	0	-1/2	-1	0	25/2	
	s_1	9	0	1	4	0	0	34	
	x_2	4	1	0	2	0	0	14	
	w	-3	0	0	-2	-1	1	2	
	Δ	-3	0	0	-2	-1	0	2	

Tabulka 3.20: První fáze dvoufázové metody pro Příklad 3.14

Pro nebazické proměnné jsou v Δ -řádku pouze záporná čísla, tj. našli jsme řešení. Jsme ale v situaci, kdy $w^* = 2$, tj. umělá proměnná je v bázi s nenulovou hodnotou a není způsob, jak ji z báze dostat. Příklad tak můžeme uzavřít s tím, že původní úloha je nepřipustná. \triangle

Příklad 3.15. Dvoufázová metoda – nadbytečné omezení.

Pomocí dvoufázové metody vyřešte následující minimalizační úlohu.

$$\begin{aligned}
 \min \quad & -x_1 - 2x_2 - x_3 \\
 \text{za podmínek} \quad & 3x_1 + x_2 - x_3 = 15 \\
 & 8x_1 + 4x_2 - x_3 = 50 \\
 & 2x_1 + 2x_2 + x_3 = 20 \\
 & x_1, x_2, x_3 \geq 0
 \end{aligned}$$

Řešení. Zadaná úloha je již v kanonickém tvaru, matice A však neobsahuje žádný sloupec jednotkové matice. Úlohu tedy rozšíříme o 3 umělé proměnné w_1, w_2, w_3 , čímž pro první fázi získáme úlohu ve tvaru

$$\begin{aligned}
 \min \quad & w_1 + w_2 + w_3 \\
 \text{za podmínek} \quad & 3x_1 + x_2 - x_3 + w_1 = 15 \\
 & 8x_1 + 4x_2 - x_3 + w_2 = 50 \\
 & 2x_1 + 2x_2 + x_3 + w_3 = 20 \\
 & x_1, x_2, x_3, w_1, w_2, w_3 \geq 0.
 \end{aligned}$$

Pro tuto úlohu si sestavíme výchozí simplexovou tabulku. Tato tabulka je společně s řešením první fáze zobrazena v Tabulce 3.21.

c_B	B	x_1	x_2	x_3	w_1	w_2	w_3	x_B	$\frac{x_B}{\lambda}$
1	w_1	3	1	-1	1	0	0	15	$5 \rightarrow$
1	w_2	8	4	-1	0	1	0	50	$25/4$
1	w_3	2	2	1	0	0	1	20	10
	c	0	0	0	1	1	1		
	z_k	13	7	-1	1	1	1		
	Δ	$13 \uparrow$	7	-1	0	0	0	85	
	x_1	1	$1/3$	$-1/3$		0	0	5	-
	w_2	0	$4/3$	$5/3$		1	0	10	$6 \rightarrow$
	w_3	0	$4/3$	$5/3$		0	1	10	6
	Δ	0	$8/3$	$10/3 \uparrow$		0	0	20	
	x_1	1	$3/5$	0			0	7	
	x_3	0	$4/5$	1			0	6	
	w_3	0	0	0			1	0	
	Δ	0	0	0			0	0	

Tabulka 3.21: První fáze dvoufázové metody pro Příklad 3.15

Při postupu první fáze po vyloučení umělé proměnné nepočítáme jí příslušný sloupec. V Tabulce 3.21 vidíme, že již po dvou krocích je umělá proměnná w_3 v bázi pouze s nulovou hodnotou. V tomto momentě bychom mohli do báze zkusit zavést proměnnou x_2 . Ta by však jistě nenahradila umělou proměnnou, kterou nyní z báze nemůžeme vyloučit, neboť jsou v příslušném řádku pro ostatní proměnné pouze nuly. Pokračujeme tedy ke druhé fázi, ve které si nejprve sestavíme výchozí tabulku bez umělých proměnných w_1 a w_2 (neboť nejsou v bázi). Přepočteme také Δ -řádek, aby odpovídal účelové funkci původní úlohy.

Výchozí tabulka pro druhou fázi je spolu s prvním krokem algoritmu zobrazená v Tabulce 3.22. V této tabulce vidíme, že v řádku pro umělou proměnnou w_3 jsou všechny prvky ve sloupcích pro původní proměnné nulové. To znamená, že se v dalších krocích tento řádek nezmění. Tento řádek vlastně k původní úloze přidává omezení $w_3 = 0$. V tomto případě můžeme řádek pro w_3 z tabulky odstranit, neboť představuje nadbytečné omezení. Čtenář si jistě všimne, že třetí omezení původního problému vzniklo lineární kombinací prvních dvou omezení (jako $o_3 = o_2 - 2o_1$). Ve druhé fázi tak pokračujeme pouze s tabulkou se dvěma řádky.

V prvním kroku do báze na místo x_3 vstoupí proměnná x_2 . V dalším kroku již nemůžeme vybrat klíčový sloupec, našli jsme tedy řešení, kterým je (v původních proměnných) bod $x^* = [5/2, 15/2, 0]$. Kdybychom ve druhé fázi měli v tabulce umělé proměnné, museli bychom hlídat, aby jejich hodnota nebyla kladná. To by totiž rozbilo přípustnost aktuálního bazického přípustného bodu.

c_B	B	x_1	x_2	x_3	w_3	x_B	$\frac{x_B}{\lambda}$
-1	x_1	1	3/5	0	0	7	35/3
-1	x_3	0	4/5	1	0	6	15/2 →
-0	w_3	0	0	0	1	0	---
	c	-1	-2	-1	0		
	z_k	-1	-7/5	-1	0		
	Δ	0	3/5 ↑	0	0	-13	
	x_1	1	0	-3/4		5/2	
	x_2	0	1	5/4		15/2	
	Δ	0	0	-3/4		-35/2	

Tabulka 3.22: Druhá fáze dvoufázové metody pro Příklad 3.15

△

Nyní již víme, jak řešit úlohy lineárního programování pomocí simplexové metody, jaká jsou pravidla pro výběr klíčového řádku a sloupce a jak postupovat v případě, kdy výchozí bazický přípustný bod nevidíme na první pohled. Existují však další varianty simplexové metody a jiné metody, jak úlohy lineárního programování řešit. Vybrané metody stručně okomentujeme v následující části práce.

3.4 Alternativní metody řešení úloh lineárního programování

Jak jsme již viděli v historické poznámce, s rozvojem výpočetní techniky se začal klást důraz na složitost algoritmů využívaných k řešení úloh lineárního programování. Složitostí můžeme rozumět jak časovou, tak paměťovou náročnost na řešení úlohy, a to v závislosti na velikosti vstupu (označíme n). Jestliže např. časová náročnost roste dvakrát rychleji, než velikost vstupu, řekneme, že složitost algoritmu je $2n$. Složitosti můžeme také rozdělit na dvě třídy – polynomiální a nepolynomiální. Polynomiální algoritmy si definujeme tak, že jejich složitost roste s n polynomiálně, tj. můžeme ji zapsat jako n^k pro nějaké k . Nepolynomiální algoritmy jsou pak ty, pro které žádná takové k neexistuje. Složitost nepolynomiálních algoritmů roste rychleji, než polynomiálně, např. exponenciálně, tj. k^n . Pro výpočet úlohy, která má exponenciální složitost s $k = 3$ je složitost pro $n = 3$ rovna 27, pro $n = 5$ již 243. Nepolynomiální algoritmy jsou ve všeobecnosti velmi pomalé, při výpočtech se jim proto snažíme vyhýbat. Příklad simplexové metody, ve které pro nalezení minima funkce 3 proměnných omezené 3 nerovnostními omezeními potřebujeme 2^3 kroků simplexového algoritmu lze najít např. v [47]. Více k tématu složitosti lze najít např. v [34], o průměrném čase trvání simplexové metody se lze dočíst např. v [36] či [39].

Vzhledem k tomu, že dosud není známo, zda existuje varianta simplexové metody s polynomiální výpočetní složitostí, se začaly objevovat pokusy najít algoritmus, jehož výpočetní složitost bude polynomiální. Zároveň bylo dokázáno, že pro většinu praktických

úloh lineárního programování s m omezeními stačí řádově m (maximálně $2m$ až $3m$) iterací. Myšlenkou, ze které se později zrodila tzv. *revidovaná simplexová metoda*, bylo zjednodušit simplexový algoritmus tak, aby se v každém kroku nemusela přepočítávat a ukládat celá tabulka. V různých variantách revidované metody tak ukládáme a aktualizujeme např. pouze aktualizovaný Δ -řádek, příslušný klíčový sloupec a sloupec x_B . Více o různých variantách revidované simplexové metody najde čtenář např. v [9] a [47]. Poslední variantou simplexové metody, kterou zde zmíníme, je tzv. *duální simplexová metoda*. Jejím podrobnému odvození se zde věnovat nebudeme, pojem duální úloha a základní myšlenku duální simplexové metody však uvedeme v následující kapitole.

Jedním z alternativních způsobů řešení lineárních programů je *metoda elipsoidů*, jež v roce 1979 publikoval L. Kachian. Tato metoda ovšem není pro praktické úlohy zdaleka efektivnější než metoda simplexová. Další metodou s polynomiální složitostí je Karmarkarova *metoda vnitřních bodů* zveřejněná v roce 1984. Přesněji řečeno, Karmarkarova metoda je jen jednou z metod souhrnně označených jako *metody snižování potenciálu*. Tyto metody se spolu s dalšími druhy metod (např. *metod sledování centrální cesty*) řadí do velké skupiny metod vnitřních bodů.

Rozdíl mezi postupem těchto tří metod je, že simplexová metoda, jak již víme, postupuje po vrcholech mnohostěnu, metoda elipsoidů se až na poslední krok pohybuje vně a metoda vnitřních bodů pracuje uvnitř mnohostěnu. Pro bližší seznámení s alternativními metodami je možné využít např. [22], [27], [29] a [43].

Kapitola 4

Dualita v lineárním programování

Ještě před tím, než se podíváme na úlohy lineárního programování motivované reálnými problémy, si v této kapitole vysvětlíme, co je duální úloha, jak ji odvodit a jaká je její praktická interpretace. Ukážeme si také, jak při řešení úloh lineárního programování využít tzv. *duální simplexovou metodu*.

Zkusme se nejprve nad úlohami, které jsme již viděli, zamyslet trochu jinak. Vezměme v úvahu např. následující minimalizační úlohu, která se podobá úlohám řešeným grafickou metodou v Kapitole 1, tj.

$$\begin{array}{ll} \min & 3x_1 + 4x_2 \\ \text{za podmínek} & 2x_1 + x_2 \geq 8 \\ & x_1 + 2x_2 \geq 3 \\ & 2x_1 + 3x_2 \geq 7 \\ & x_1, x_2 \geq 0. \end{array}$$

Tento příklad bychom mohli vyřešit např. grafickou metodou, kterou bychom získali řešení $x^* = [4, 0]$ s hodnotou účelové funkce $f^* = 12$. Můžeme minimální hodnotu účelové funkce také nějak odhadnout? Nyní si vyzkoušíme, jak tuto hodnotu odhadnout s využitím zadaných nerovností. S využitím třetího omezení a omezení na znaménko můžeme např. získat odhad minima následujícím způsobem

$$3x_1 + 4x_2 \geq 2x_1 + 3x_2 \geq 7.$$

Hodnota účelové funkce (f^*) na dané množině tak nemůže být menší než 7. Nyní zkusíme využít kombinaci prvního a druhého omezení a nezápornosti proměnných, čímž dostaneme

$$3x_1 + 4x_2 \geq (2x_1 + x_2) + (x_1 + 2x_2) \geq 8 + 3 = 11,$$

f^* tedy nemůže být menší než 11. Ke stejnému závěru bychom mohli dojít i s využitím prvního omezení (vyděleného 2) a třetího omezení, neboť

$$3x_1 + 4x_2 = (2x_1 + 3x_2) + (x_1 + x_2) \geq (2x_1 + 3x_2) + (x_1 + \frac{1}{2}x_2) \geq 7 + 4 = 11.$$

Můžeme ale získat nějaký lepší odhad? Využitím kombinací zadaných omezení jsme se vlastně snažili odvodit nerovnost ve tvaru

$$d_1x_1 + d_2x_2 \geq h, \tag{4.1}$$

kde $d_1 \leq 3$, $d_2 \leq 4$ a h je největší možné číslo. S využitím omezení na znaménko $x_1, x_2 \geq 0$ pak můžeme psát

$$3x_1 + 4x_2 \geq d_1x_1 + d_2x_2 \geq h.$$

Číslo h je tedy nějaká dolní hranice minimální hodnoty účelové funkce na dané přípustné množině. Nyní zbývá jen najít příslušné koeficienty d_1 a d_2 , díky kterým h získáme. Tyto koeficienty najdeme s využitím všech omezení původní úlohy, které zkombinujeme pomocí nezáporných koeficientů (označených y_1, y_2, y_3) do tvaru nerovnosti (4.1). Můžeme psát

$$d_1x_1 + d_2x_2 = (2y_1 + y_2 + 2y_3)x_1 + (y_1 + 2y_2 + 3y_3)x_2 \geq h. \quad (4.2)$$

Nezápornost koeficientů y_1, y_2 a y_3 je nutná kvůli zachování znaménka této nerovnosti. Úpravou (4.2) dostaneme

$$y_1(2x_1 + x_2) + y_2(x_1 + 2x_2) + y_3(2x_1 + 3x_2) \geq 8y_1 + 3y_2 + 7y_3 = h.$$

Naším cílem je najít co největší h tak, aby koeficienty d_1 a d_2 splňovaly zadaná omezení. Tímto postupem jsme se dostali k jiné úloze lineárního programování, jejíž zadání je

$$\left. \begin{array}{l} \max \quad 8y_1 + 3y_2 + 7y_3 \\ \text{za podmíněk} \quad d_1 = 2y_1 + y_2 + 2y_3 \leq 3 \\ \quad \quad \quad d_2 = y_1 + 2y_2 + 3y_3 \leq 4 \\ \quad \quad \quad y_1, y_2, y_3 \geq 0. \end{array} \right\} \quad (4.3)$$

Úlohu (4.3) nazýváme *duální úlohou* k původní úloze, kterou v tomto případě označíme *primární úloha*. Označení primární a duální úloha závisí pouze na tom, která úloha je brána jako původní. V literatuře se tak často setkáme se souhrnným označením *duálně sdružené úlohy* (viz např. [25]) či *dvojice vzájemně duálních úloh* (viz [32]). Abychom získali nejlepší dolní odhad minima hodnoty účelové funkce primární úlohy, musíme vyřešit duální úlohu, která primární úlohu zdola ohraničuje. Optimální hodnotu účelové funkce nazýváme *hodnota úlohy*. Označme f^* hodnotu primární úlohy a ϕ^* hodnotu úlohy duální. Z odvození duální úlohy pak vyplývá vztah

$$f^* \geq \phi^*.$$

Provedeme-li podobnou úvahu o hledání horního odhadu hodnoty účelové funkce pro úlohu (4.3) (tj. budeme-li hledat duální úlohu k duální úloze), dostaneme opět primární úlohu. Obecný tvar duální úlohy pro minimalizační úlohu lineárního programování v obecném tvaru, tj. úlohu

$$\begin{array}{l} \min \quad \langle c, x \rangle \\ \text{za podmíněk} \quad \langle a_i, x \rangle \geq b_i, \quad i = 1, \dots, k \\ \quad \quad \quad \langle a_i, x \rangle = b_i, \quad i = k + 1, \dots, m \\ \quad \quad \quad x_i \geq 0, \quad i = 1, \dots, s, \end{array}$$

je maximalizační úloha lineárního programování ve tvaru

$$\begin{array}{l} \max \quad \langle b, y \rangle \\ \text{za podmíněk} \quad \langle a_i, y \rangle \geq c_i, \quad i = 1, \dots, s \\ \quad \quad \quad \langle a_i, y \rangle = c_i, \quad i = s + 1, \dots, n \\ \quad \quad \quad y_i \geq 0, \quad i = 1, \dots, k. \end{array}$$

Pro primární úlohu v kanonickém tvaru je duální úloha v základním tvaru. Pro vztahy řešení primární a duální úlohy platí několik užitečných tvrzení, která si nyní uvedeme. Již víme, že námi odvozená duální úloha zdola ohraničovala úlohu primární. Toto tvrzení je předmětem tzv. *slabé věty o dualitě*.

Věta 4.1. *Slabá věta o dualitě.*

Pro každý přípustný bod primární úlohy a každý přípustný bod duální úlohy platí $c^T x \geq b^T y$. Zejména, je-li primární úloha neohraničená (zdola), pak je duální úloha nepřípustná. Naopak, je-li duální úloha neohraničená (shora), pak je primární úloha nepřípustná.

Důkaz. Důkaz lze najít např. v [22]. □

Z této věty plyne následující důsledek, jenž dává do souvislosti řešení primární a duální úlohy. Pro tento důsledek lze v literatuře (např. v [44]) najít označení *certifikát optimality*.

Důsledek 4.2. *Jestliže pro nějaký přípustný bod primární úlohy x^* a nějaký přípustný bod duální úlohy y^* platí rovnost $c^T x^* = b^T y^*$, pak jsou x^* a y^* řešeními příslušných úloh.*

Důkaz. Důkaz lze najít např. v [32]. □

Pro libovolnou primární úlohu lineárního programování je duální úloha též úlohou lineárního programování. V obou úlohách mohou nastat tři možnosti, které již známe z výše uvedené teorie. Úlohy mohou mít řešení, mohou být neohraničené či nepřípustné. Celkem tedy máme devět možností, ke kterým může při řešení primární a duální úlohy dojít. Některé z těchto možností jsme slabou větou o dualitě již vyloučili (např. nelze, aby jedna úloha byla neohraničená a druhá měla řešení). Dále lze dokázat, že nemůže nastat situace, ve které je jedna úloha nepřípustná a druhá má řešení. Vzhledem k tomu, že obě úlohy jsou úlohami lineárního programování, jsou výsledky, které jsme získali, symetrické. Po vyloučení všech situací, jež nemohou nastat, získáváme pro primární a duální úlohy pouze čtyři možnosti. Ty jsou obsahem tzv. *silné věty o dualitě*.

Věta 4.3. *Silná věta o dualitě.*

Pro primární úlohu lineárního programování a příslušnou duální úlohu může nastat právě jedna z následujících možností.

1. *Obě úlohy jsou nepřípustné (tj. $X = Y = \emptyset$).*
2. *Primární úloha je neohraničená a duální úloha je nepřípustná.*
3. *Primární úloha je nepřípustná a duální úloha je neohraničená.*
4. *Obě úlohy jsou přípustné a ohraničené. Pak existují řešení x^* a y^* , která splňují $c^T x^* = b^T y^*$, tj. hodnota primární a duální úlohy je stejná.*

Důkaz. Důkazy jednotlivých tvrzení lze najít např. v [32]. □

Výsledky, které jsme o řešitelnosti primární a duální úlohy získali, uvedeme nyní pro přehlednost v Tabulce 4.1. Nadpis NP značí nepřípustnost úlohy (tj. $X = \emptyset$, případně $Y = \emptyset$), PaO označuje přípustné a ohraničené úlohy a NO znamená neohraničenost (tj. $f^* = -\infty$ pro primární úlohu a $\phi^* = \infty$ pro úlohu duální).

\ primární úloha	NP	PaO	NO
duální úloha			
NP	✓	✗	✓
PaO	✗	✓	✗
NO	✓	✗	✗

Tabulka 4.1: Možnosti, které mohou pro primární a duální úlohu nastat

Nyní si ukážeme, proč je pro nás uvedená teorie duality důležitá. První, co nás může napadnout, je využít duální úlohu k nalezení řešení primární úlohy. Vzhledem k rovnosti $f^* = \phi^*$ si můžeme vybrat, kterou z úloh budeme řešit, neboť řešením jedné z nich obdržíme zároveň řešení druhé. Při řešení primární úlohy získáme řešení duální úlohy (s opačným znaménkem) v Δ -řádku poslední simplexové tabulky, a naopak při řešení duální úlohy najdeme v Δ -řádku poslední tabulky řešení primární úlohy (s opačným znaménkem). Toto je důležitý výsledek, protože řešení duální úlohy může být výrazně jednodušší, a tedy i rychlejší, než řešení úlohy primární. Např. pro primární úlohu, která má 9 proměnných a 99 omezení, bude v simplexové tabulce 100 řádků. Pro příslušnou duální úlohu však bude v tabulce 10 řádků a 99 sloupců. Pomocí duálních problémů můžeme také spočítat citlivost řešení primární úlohy na změny jejích koeficientů (A , b a c), čímž se zabývá tzv. *citlivostní analýza*.

Pro praktické příklady je velmi důležitá i interpretace duální úlohy, jejíž řešení nám navíc poskytuje další informace o původní úloze. Pro větší názornost si nyní uvedeme příklad primární úlohy, ke které sestrojíme úlohu duální a vysvětlíme si její význam.

Příklad 4.1. Interpretace duální úlohy.

Představme si firmu, která prodává dva výrobky, jež si označíme y_1 a y_2 . Zisk z prodeje těchto výrobků je 30 Kč a 40 Kč. Firma k výrobě využívá tři zdroje (označené A , B a C), které má k dispozici pouze v omezeném množství. Tato omezenost vyplývá např. z faktu, že si firma objednala zdroje v nějakém množství a nemůže snadno získat další. K výrobě jednotky y_1 potřebuje firma 10 kusů A a 20 kusů B . K výrobě jednotky y_2 musí firma mít 4 kusy A , 25 kusů B a 6 kusů C . Maximální dostupná množství pro jednotlivé zdroje jsou postupně 1000, 3000 a 600. Tato firma chce vyrábět nezáporná množství svých výrobků, tj. nechce je od nikoho nakupovat. Všechny údaje z tohoto odstavce jsou pro větší přehlednost uvedeny v Tabulce 4.2.

\ zdroje	A	B	C	zisk (Kč)
výrobky				
y_1	10	20	0	30
y_2	4	25	6	40
omezení	1000	3000	600	

Tabulka 4.2: Zadání Příkladu 4.1

Problém, kterému firma čelí, je tedy následující maximalizační úloha.

$$\begin{array}{ll} \max & 30y_1 + 40y_2 \\ \text{za podmíněk} & 10y_1 + 4y_2 \leq 1000 \\ & 20y_1 + 25y_2 \leq 3000 \\ & 6y_2 \leq 600 \\ & y_1, y_2 \geq 0 \end{array}$$

Tuto úlohu bychom mohli vyřešit simplexovou metodou, kterou bychom získali $y_1^* = 25$ a $y_2^* = 100$. Maximální zisk, kterého může tato firma dosáhnout, je $f^* = 4750$. Ověření správnosti tohoto výsledku ponecháme na čtenáři. Nyní se na úlohu podívejme z druhé strany. Představme si, že v odvětví existuje jiná firma, kterou si pro přehlednost označíme firma 2. Vedení této firmy by rádo rozšířilo svoji výrobu. Kvůli omezenosti ale nemůže získat zdroje potřebné k výrobě. Vedení firmy 2 tak osloví původní firmu s nabídkou od ní její výrobní zdroje odkoupit. Jakou cenu má firma 2 nabídnout, aby byl tento obchod proveden? Existuje vůbec taková cena?

Firma 2 chce najít takové ceny, aby se původní firmě vyplatilo nabídku přijmout. Zároveň chce ale minimalizovat své náklady na nákup těchto zdrojů. Vzhledem k tomu, že se firmy pohybují ve stejném odvětví, mají dobrý přehled o zdrojích potřebných k vyrobení produktů y_1 a y_2 . Vedení firmy 2 ví, že k výrobě jednotky y_1 potřebuje původní firma 10 kusů A a 20 kusů B a že zisk z prodeje je 30 Kč. Musí tedy přijít s takovou nabídkou, která je pro původní firmu minimálně stejně dobrá, jako když vyrábí a prodává svoje produkty sama. Označme si ceny, které je firma 2 ochotna zaplatit za zdroje, x_1 , x_2 a x_3 . Aby původní firma nabídku přijala, musí za zdroje, díky kterým by vyrobila a prodala y_1 , získat alespoň 30 Kč, tj. musí platit

$$10x_1 + 20x_2 \geq 30.$$

Pro výrobek y_2 získáme s využitím stejné úvahy nerovnost

$$4x_1 + 25x_2 + 6x_3 \geq 40.$$

Nákladová funkce, kterou chce firma 2 minimalizovat, je součtem výdajů na všechny zdroje, tj.

$$1000x_1 + 3000x_2 + 600x_3.$$

Cílem firmy 2 je nákup zdrojů od původní firmy, jistě tedy platí $x_1, x_2, x_3 \geq 0$. Firma 2 řeší minimalizační úlohu

$$\begin{array}{ll} \min & 1000x_1 + 3000x_2 + 600x_3 \\ \text{za podmíněk} & 10x_1 + 20x_2 \geq 30 \\ & 4x_1 + 25x_2 + 6x_3 \geq 40 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Tato úloha je duální úlohou k úloze, kterou řeší původní firma. Z výše uvedené teorie víme, že když existuje řešení primární úlohy, jistě existuje i řešení úlohy duální. Tím

je bod $x^* = [0, 3/2, 5/12]$. Souřadnice tohoto bodu najdeme v posledním kroku řešení primární úlohy v Δ -řádku. Stejně tak při řešení úlohy duální zároveň získáme řešení úlohy primární, a to opět v Δ -řádku. Vzhledem k tomu, že již známe hodnotu úlohy $f^* = 4750$, nás nepřekvapí, že pro řešení duální úlohy platí $\phi^* = 4750$.

Výsledné hodnoty x_1 , x_2 a x_3 představují jak maximální cenu, za kterou je firma 2 ochotná zdroje nakoupit, tak minimální cenu, za kterou je původní firma ochotná zdroje prodat. Jedná se tedy o jakousi jednotkovou cenu daných zdrojů. Maximální zisk, který může původní firma získat, tak můžeme spočítat nejen přes princip maximalizace zisku za prodej jejích výrobků, ale i vypočtením hodnoty jejích výrobních zdrojů, neboli řešením duální úlohy. Těmto cenám se říká tzv. *stínové ceny*, neboť představují ceny vstupů do výroby. Stínové ceny tak udávají citlivost zisku na množství dostupných zdrojů.

△

Lze si všimnout, že primární úlohu jsme s využitím Tabulky 4.2 získali čtením jednotlivých sloupců, duální úlohu pak s využitím jednotlivých řádků. Tato tabulka svým uspořádáním připomíná tzv. *Tuckerovu tabulku*, kterou můžeme využít pro přehledné zapísání primární a duální úlohy.

Nyní jsme si ukázali, jaký význam má duální úloha a její koeficienty. Duální i primární úlohu jsme schopni vyřešit nám již dobře známou simplexovou metodou. Jak ale postupovat v případě, kdy máme úlohu, ve které jsou na pravé straně záporná čísla? Je vždy nutné využít metodu velkého M nebo dvoufázovou metodu, abychom v takovém případě našli výchozí bazický přípustný bod dané primární úlohy? Nebo lze i takovou úlohu vyřešit s využitím „klasické“ simplexové metody? Na tyto otázky si nyní odpovíme.

Při řešení úloh „klasickou“ simplexovou metodou jsme vždy potřebovali najít výchozí bazický přípustný bod tak, aby v simplexové tabulce nebyly ve sloupci označeném x_B záporné prvky. Bod, pro který jsou ve sloupci x_B záporné prvky, sice není přípustný pro primární úlohu, může ovšem být přípustný pro úlohu duální. Bod je tzv. *duálně přípustný*, když jsou všechny prvky na nebazických pozicích v Δ -řádku nekladné. Na této myšlence je postavena tzv. *duální simplexová metoda*, kterou si nyní ukážeme.

4.1 Duální simplexová metoda

Duální simplexová metoda, stejně jako klasická simplexová metoda, je algoritmus, který generuje bazické přípustné body. Duální simplexová metoda ale vychází z bodu, který je duálně přípustný, což znamená, že není přípustný pro zadanou primární úlohu, ale je přípustný pro úlohu k ní duální. Při postupu duální simplexové metody generujeme duálně přípustné body, výsledkem algoritmu je pak bod, který je přípustný jak duálně, tak primárně. Algoritmus vlastně postupuje tak, jako bychom klasickou simplexovou metodou řešili duální úlohu, kterou ovšem při použití duální simplexové metody nemusíme sestavovat.

V případě využití duální simplexové metody postupujeme obdobně, jako při využití klasické simplexové metody. Výpočet opět pro přehlednost uspořádáme do tabulky. Hlavním rozdílem těchto dvou metod je vybírání klíčového řádku a sloupce. Duální simplexovou metodu si nyní algoritmicky popíšeme. Podrobnější odvození najde čtenář např. v [3], [15] či [36]. Stejně jako v případě klasické simplexové metody existuje i pro duální metodu více verzí. Opět také může dojít k zacyklení či k situaci, kdy nemůžeme snadno najít vý-

chozí bod. V duální simplexové metodě pro nalezení výchozího duálně přípustného bodu nepřidáváme umělé proměnné, ale umělá omezení. Tuto metodu ve stručnosti popíšeme v následující části textu. Nejprve uvedeme popis algoritmu duální simplexové metody a ilustrační příklad.

1. Úlohu převedeme do kanonického tvaru a doplníme ji tak, aby v matici A byly všechny sloupce jednotkové matice, proměnné odpovídající těmto sloupcům jsou v bázi. Platí tedy $x_B = b$. V případě, že pro některou složku vektoru b platí $b_i < 0$, využijeme duální simplexovou metodu. V opačném případě postupujeme nám již známým způsobem, tj. využijeme klasickou simplexovou metodu.
2. Najdeme výchozí bazický duálně přípustný bod a sestavíme si simplexovou tabulku včetně Δ -řádku.
3. V tomto kroku vybíráme klíčový řádek. Při výběru mohou nastat dvě situace.
 - (a) Jestliže pro nějaké $j \in B$ platí $x_B < 0$, vybereme tento index j . Řádek s tímto indexem je *klíčovým řádkem*, příslušná proměnná tedy opouští bázi. Jestliže je takových indexů více, vybereme ten, který má ve sloupci x_B nejmenší číslo.
 - (b) Naopak, jestliže jsou všechny prvky ve sloupci x_B nezáporné, našli jsme primárně přípustný bod, který je řešením úlohy. Jestliže pro nějaký bazový index j je ve sloupci x_B nula, pak je nalezený bod degenerovaný. Algoritmus v tomto případě končí.
4. Při výběru klíčového sloupce mohou nastat dvě situace.
 - (a) Všechny prvky v klíčovém řádku pro nebazické proměnné jsou nezáporná čísla. Primární úloha je v tomto případě nepřípustná a algoritmus končí.
 - (b) Existuje alespoň jedno záporné číslo λ pro nebazické proměnné v klíčovém řádku. Pak pro všechna $\lambda < 0$ pro nebazické proměnné v klíčovém řádku spočítáme podíly Δ/λ . Z těch poté vybereme minimum. Nalezený sloupec je *klíčovým sloupcem*.
5. S využitím prvku v průniku klíčového řádku a klíčového sloupce (tj. pivota) provedeme Gaussovu eliminaci.
6. Vrátime se ke kroku 3 a algoritmus opakujeme, dokud nedojdeme k jednomu ze závěrů popsaných výše. V případě, že najdeme řešení úlohy, odečteme hodnoty jednotlivých bazických proměnných pro primární úlohu ve sloupci x_B , na nebazické pozice doplníme nuly. Řešení duální úlohy odečteme z Δ -řádku.

Nevýhodou duální simplexové metody oproti klasické metodě je, že postupem přes duálně přípustné body získáme bazický přípustný bod primární úlohy až v posledním kroku metody. Metoda tak musí vždy proběhnout celá. V případě klasické metody se můžeme např. spokojit i s neoptimálním výsledkem, který dostaneme v průběhu výpočtu, nemusíme tak vždy projít celý algoritmus až do konce. Duální simplexovou metodu si nyní ukážeme na příkladu.

Příklad 4.2. Duální simplexová metoda.

Duální simplexovou metodou vyřešte následující minimalizační úlohu.

$$\begin{aligned} \min \quad & x_1 + 4x_3 + 3x_4 + 2x_5 \\ \text{za podmíněk} \quad & x_1 + x_2 \geq 12 \\ & x_1 + 2x_3 + x_4 \geq 20 \\ & 2x_2 + x_3 + 3x_4 + 5x_5 \geq 26 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Řešení. Začneme převodem úlohy na kanonický tvar. Všechna omezení si nejprve vynásobíme číslem -1 a poté přičteme doplňkové proměnné (označené s_1, s_2 a s_3), čímž dostaneme rovnostní omezení ve tvaru

$$\begin{aligned} -x_1 - x_2 + s_1 &= -12, \\ -x_1 - 2x_3 - x_4 + s_2 &= -20, \\ -2x_2 - x_3 - 3x_4 - 5x_5 + s_3 &= -26. \end{aligned}$$

K úloze ještě přidáme omezení na znaménko doplňkových proměnných, tj. $s_1, s_2, s_3 \geq 0$. Úloha, kterou řešíme, je tedy ve tvaru

$$\begin{aligned} \min \quad & x_1 + 4x_3 + 3x_4 + 2x_5 \\ \text{za podmíněk} \quad & -x_1 - x_2 + s_1 = -12 \\ & -x_1 - 2x_3 - x_4 + s_2 = -20 \\ & -2x_2 - x_3 - 3x_4 - 5x_5 + s_3 = -26 \\ & x_1, x_2, x_3, x_4, x_5, s_1, s_2, s_3 \geq 0. \end{aligned}$$

Nyní již sestavíme výchozí simplexovou tabulku. Vidíme, že v této tabulce jsou pro nebazické pozice v Δ -řádku pouze nekladná čísla. Výchozí bod $[-12, -20, -26]$ je tedy duálně přípustný. Výchozí bázi tvoří doplňkové proměnné.

V Tabulce 4.3 vidíme, že ve sloupci x_B jsou tři záporná čísla. Vybereme tak to nejmenší z nich, které přísluší proměnné s_3 , která v tomto kroku opustí bázi. Nyní pro všechna záporná čísla příslušná nebazickým proměnným ve třetím řádku vypočteme podíly Δ/λ . Minimum z těchto podílů je číslo 0 ve druhém sloupci, do báze tedy vstoupí proměnná x_2 . Nyní s využitím pivota (číslo -2) provedeme, stejně jako pro klasickou simplexovou metodu, Gaussovu eliminaci. Výsledek vidíme ve druhé simplexové tabulce. Ve druhém kroku postupujeme obdobně. Bod, který jsme získali prvním krokem metody stále není primárně přípustný. Jediné záporné číslo ve sloupci x_B je nyní ve druhém řádku, z báze tedy vystoupí s_2 . Vypočtením podílů Δ/λ najdeme proměnnou, která do báze vstoupí, tou je x_1 . Opět provedeme Gaussovu eliminaci, čímž získáme poslední tabulku. V této tabulce již ve sloupci x_B není žádné záporné číslo. Našli jsme tedy bod, který je jak primárně přípustný, tak duálně přípustný (v Δ -řádku jsou pro nebazické proměnné nekladná čísla). Výsledný bod je tedy řešením primární úlohy. V původních proměnných je to bod $x^* = [20, 13, 0, 0, 0]$, hodnota účelové funkce v tomto bodě je $f^* = 20$.

c	B	x_1	x_2	x_3	x_4	x_5	s_1	s_2	s_3	x_B
0	s_1	-1	-1	0	0	0	1	0	0	-12
0	s_2	-1	0	-2	-1	0	0	1	0	-20
0	s_3	0	(-2)	-1	-3	-5	0	0	1	-26 →
	c	1	0	4	3	2	0	0	0	
	z	0	0	0	0	0	0	0	0	
	Δ	-1	0	-4	-3	-2	0	0	0	0
	Δ/λ	-	0/-2 ↑	-4/-1	-3/-3	-2/-5	-	-	-	
	s_1	-1	0	1/2	3/2	5/2	1	0	-1/2	1
	s_2	(-1)	0	-2	-1	0	0	1	0	-20 →
	x_2	0	1	1/2	3/2	5/2	0	0	-1/2	13
	Δ	-1	0	-4	-3	-2	0	0	0	0
	Δ/λ	1 ↑	-	2	3	-	-	-	-	
	s_1	0	0	5/2	5/2	5/2	1	-1	-1/2	21
	x_1	1	0	2	1	0	0	-1	0	20
	x_2	0	1	1/2	3/2	5/2	0	0	-1/2	13
	Δ	0	0	-2	-2	-2	0	-1	0	20

Tabulka 4.3: Duální simplexová metoda pro Příklad 4.2

Řešení duální úlohy (s opačným znaménkem) najdeme v Δ -řádku. Čtenář se snadno přesvědčí, že řešením úlohy

$$\begin{aligned}
 &\max && 12y_1 + 20y_2 + 26y_3 \\
 &\text{za podmíněk} && y_1 + y_2 \leq 1 \\
 &&& y_1 + 2y_3 \leq 0 \\
 &&& 2y_2 + y_3 \leq 4 \\
 &&& y_2 + 3y_3 \leq 3 \\
 &&& 5y_3 \leq 2 \\
 &&& y_1, y_2, y_3 \geq 0,
 \end{aligned}$$

kteřá je duální k úloze, jež jsme právě vyřešili, je bod $y^* = [0, 1, 0]$. Hodnota úlohy je pak $\phi^* = 20$. Souřadnice tohoto bodu s opačnými znaménky jsou v posledním Δ -řádku v Tabulce 4.3 ve sloupcích příslušných doplňkovým proměnným. Kdybychom řešili duální úlohu v kanonickém tvaru (tj. přidali bychom 5 doplňkových proměnných s_1, \dots, s_5), dostali bychom ve výsledku pro tyto umělé proměnné hodnoty 0, 0, 2, 2, 2, což odpovídá hodnotám v Δ -řádku ve sloupcích pro původní proměnné primární úlohy vynásobeným číslem -1 . Tuto nesrovnalost znamének jsme získali díky způsobu výpočtu Δ -řádku. V případě, že bychom při jeho výpočtu využili místo $z_j - c_j$ rozdíl $c_j - z_j$, by tato nesrovnalost nevznikla. Všechny prvky v Δ -řádku by tímto způsobem měly opačná znaménka. Výchozí

duálně přípustný bod by tak musel mít v Δ -řádku všechny prvky kladné.

△

4.1.1 Výchozí duálně přípustný bazický bod

V předchozím příkladu jsme pro výchozí simplexovou tabulku měli bod, který byl duálně přípustný. K této situaci ovšem nemusí dojít vždy. V této části si ukážeme způsob, jakým můžeme výchozí duálně přípustný bazický bod najít. Na rozdíl od metody velkého M pro klasickou simplexovou metodu, ve které jsme do úlohy přidali umělé proměnné, přidáme v případě duální simplexové metody tzv. *umělá omezení*. Metoda, kterou si nyní ukážeme, se tak nazývá *metoda umělých omezení*. Tuto metodu využijeme v případě, že jsou pro výchozí bod na některých nebazických pozicích kladná čísla (bod tedy není duálně přípustný).

Při využití metody umělých omezení původní úlohu rozšíříme o omezení ve tvaru

$$\sum_{j \notin B} x_j \leq M,$$

kde M je opět velké číslo, se kterým počítáme jako s parametrem. Pokud by M nebylo dostatečně velké, mohli bychom touto úpravou ztratit některé body z původní přípustné množiny. Toto omezení pak přičtením nové doplňkové proměnné (omezené na znaménko) převedeme do tvaru rovnosti. Tuto doplňkovou proměnnou si pro přehlednost označme s_{n+1} . Proměnnou s_{n+1} přidáme do báze. Poté vybereme nebazický sloupec, pro který je v Δ -řádku kladné číslo, a provedeme Gaussovu eliminaci pomocí pivotu, kterým je číslo ve vybraném sloupci a řádku, který odpovídá přidanému umělému omezení. Pokud je v Δ -řádku více kladných čísel v nebazických sloupcích, vybereme např. to největší z nich. Tímto krokem nahradí proměnnou s_{n+1} v bázi proměnná, která způsobuje duální nepřípustnost.

Jakmile se dostaneme do situace, ve které je nalezený bod duálně přípustný, pokračujeme s duální simplexovou metodou pro úlohu s umělým omezením. Pokud při řešení úlohy s umělým omezením zjistíme, že nemá přípustný bod, pak nemá přípustný bod ani původní úloha. V případě, že najdeme přípustný bod úlohy s umělým omezením, mohou pro původní úlohu nastat následující situace.

1. Proměnná s_{n+1} je v bázi. V tomto případě získáme řešení původní úlohy z řešení úlohy s umělým omezením vynecháním složky příslušné k s_{n+1} . Hodnota úlohy s umělým omezením v tomto případě nezávisí na M .
2. Proměnná s_{n+1} není v bázi a hodnota řešení úlohy s umělým omezením závisí na M . Pak je původní úloha neohraničená.
3. Proměnná s_{n+1} není v bázi a hodnota řešení úlohy s umělým omezením nezávisí na M . V tomto případě má původní úloha nekonečně mnoho řešení, která získáme volbou M tak, aby platilo $x_B \geq 0$.

Metodu umělých omezení si nyní ukážeme na čtyřech příkladech, které ilustrují všechny výše popsané situace.

Příklad 4.3. Metoda umělých omezení – optimální řešení.

Duální simplexovou metodou s využitím metody umělých omezení vyřešte následující minimalizační úlohu.

$$\begin{aligned} \min \quad & -x_1 - 6x_2 \\ \text{za podmínek} \quad & x_1 + x_2 \leq 20 \\ & x_1 + x_2/2 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Řešení. Úlohu nejprve převedeme do kanonického tvaru vynásobením druhého omezení číslem -1 a přidáním dvou umělých proměnných. Tím dostaneme úlohu ve tvaru

$$\begin{aligned} \min \quad & -x_1 - 6x_2 \\ \text{za podmínek} \quad & x_1 + x_2 + s_1 = 20 \\ & -x_1 - x_2/2 + s_2 = -1 \\ & x_1, x_2, s_1, s_2 \geq 0. \end{aligned}$$

Pro tuto úlohu sestavíme výchozí simplexovou tabulku, která je znázorněna v Tabulce 4.4.

c_B	B	x_1	x_2	s_1	s_2	x_B
0	s_1	1	1	1	0	20
0	s_2	-1	-1/2	0	1	-1
	c	-1	-6	0	0	
	z	0	0	0	0	
	Δ	1	6	0	0	0

Tabulka 4.4: Výchozí simplexová tabulka pro Příklad 4.3

V této tabulce vidíme, že jsou pro nebazické proměnné x_1 a x_2 v Δ -řádku kladná čísla. Tento bod tak není duálně přípustný. Vzhledem k tomu, že ve sloupci x_B je záporné číslo, není tento bod ani primárně přípustný. Musíme tedy využít metodu umělých omezení. K úloze přidáme omezení

$$x_1 + x_2 \leq M,$$

které přidáním umělé proměnné s_3 upravíme na rovnost

$$x_1 + x_2 + s_3 = M.$$

Proměnnou s_3 pak zavedeme do báze a sestavíme novou výchozí simplexovou Tabulku 4.5.

Nyní zahájíme metodu umělých omezení. Z Δ -řádku vybereme největší kladné číslo, které odpovídá nebazickým proměnným. Tím je číslo 6 ve druhém sloupci, do báze tedy na místo s_3 vstoupí proměnná x_2 . Gaussovou eliminací získáme další simplexovou tabulku, ve které již pro nebazické proměnné nejsou v Δ -řádku kladná čísla. Nalezený bod je tak je duálně přípustný.

c_B	B	x_1	x_2	s_1	s_2	s_3	x_B
0	s_1	1	1	1	0	0	20
0	s_2	-1	-1/2	0	1	0	-1
0	s_3	1	①	0	0	1	$M \rightarrow$
	c	-1	-6	0	0	0	
	z	0	0	0	0	0	
	Δ	1	6↑	0	0	0	0
	s_1	0	0	1	0	①(-1)	$20 - M \rightarrow$
	s_2	-1/2	0	0	1	1/2	$-1 + M/2$
	x_2	1	1	0	0	1	M
	Δ	-5	0	0	0	-6	$-6M$
	Δ/λ	-	-	-	-	6↑	
	s_3	0	0	-1	0	1	$M - 20$
	s_2	-1/2	0	1/2	1	0	9
	x_2	1	1	1	0	0	20
	Δ	-5	0	-6	0	0	-120

Tabulka 4.5: Metoda umělých omezení pro Příklad 4.3

Nyní již můžeme pokračovat duální simplexovou metodou, jejíž postup známe z předchozího příkladu. V prvním řádku je ve sloupci x_B hodnota $20 - M < 0$. Pro nebazické proměnné, pro které je v prvním řádku záporná hodnota, vypočítáme podíl Δ/λ . Jediná proměnná, která v tomto kroku může vstoupit do báze je s_3 . Po provedení eliminace se dostáváme k poslední simplexové tabulce, ve které jsou ve sloupci x_B pouze kladná čísla. Našli jsme tedy bod, který je primárně i duálně přípustný. Umělá proměnná s_3 je v bázi a hodnota účelové funkce ve výsledném bodě nezávisí na M . Vynecháním umělé proměnné tak získáme řešení úlohy, kterým je v původních proměnných bod $x^* = [0, 20]$. Hodnota úlohy je $f^* = -120$.

△

Příklad 4.4. Metoda umělých omezení – neohraničená úloha.

Duální simplexovou metodou s využitím metody umělých omezení vyřešte následující minimalizační úlohu.

$$\begin{aligned} \min \quad & 4x_1 - 5x_2 \\ \text{za podmínek} \quad & 2x_1 + 2x_2 \geq 4 \\ & x_1 - x_2 \geq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Řešení. Stejně jako v předchozím příkladu si úlohu nejprve převedeme do kanonického

tvary, čímž získáme omezení ve tvaru

$$\begin{aligned} -2x_1 - 2x_2 + s_1 &= -4, \\ -x_1 + x_2 + s_2 &= -3, \\ x_1, x_2, s_1, s_2 &\geq 0. \end{aligned}$$

Výchozí simplexová tabulka pro tuto úlohu je zobrazena v Tabulce 4.6.

c_B	B	x_1	x_2	s_1	s_2	x_B
0	s_1	-2	-2	1	0	-4
0	s_2	-1	1	0	1	-3
	c	4	-5	0	0	
	z	0	0	0	0	
	Δ	-4	5	0	0	0

Tabulka 4.6: Výchozí simplexová tabulka pro Příklad 4.4

V Tabulce 4.6 je v Δ -řádku kladné číslo pro nebazickou proměnnou x_2 . Musíme tedy přidat umělé omezení, které rovnou doplníme do tvaru rovnosti. Tím získáme omezení

$$x_1 + x_2 + s_3 = M,$$

proměnnou s_3 zavedeme do báze a sestavíme novou tabulku. Ta je i s dalším postupem metody v Tabulce 4.7.

c_B	B	x_1	x_2	s_1	s_2	s_3	x_B
0	s_1	-2	-2	1	0	0	-4
0	s_2	-1	1	0	1	0	-3
0	s_3	1	①	0	0	1	$M \rightarrow$
	c	4	-5	0	0		
	z	0	0	0	0		
	Δ	-4	$5 \uparrow$	0	0	0	0
	s_1	0	0	1	0	2	$-4 + 2M$
	s_2	①-2	0	0	1	-1	$-3 - M \rightarrow$
	x_2	1	1	0	0	1	M
	Δ	-9	0	0	0	-5	$-5M$
	Δ/λ	$9/2 \uparrow$	-	-	-	5	
	s_1	0	0	1	0	2	$-4 + 2M$
	x_1	1	0	0	-1/2	1/2	$3/2 + M/2$
	x_2	0	1	0	1/2	1/2	$-3/2 + M/2$
	Δ	0	0	0	-9/2	-1/2	$27/2 - M/2$

Tabulka 4.7: Duální simplexová metoda s využitím umělého omezení pro Příklad 4.4

V prvním kroku musíme do báze na místo umělé proměnné s_3 zavést proměnnou x_2 , a to kvůli kladnému číslu v Δ -řádku. Poté provedeme Gaussovu eliminaci, čímž se dostaneme do situace, kdy jsou pro nebazické proměnné v Δ -řádku pouze nekladná čísla. Našli jsme tak duálně přípustný bod a můžeme pokračovat s duální simplexovou metodou. Jediné záporné číslo ve sloupci x_B je ve druhém řádku. Z báze proto vystoupí proměnná s_2 . Nejmenší podíl v řádku Δ/λ najdeme v prvním sloupci, do báze tak vstupuje proměnná x_1 . Po jednom kroku duální metody se dostaneme do situace, ve které je bod již primárně i duálně přípustný. V bázi však není proměnná s_3 , hodnota účelové funkce navíc závisí na M . Původní úloha je tedy neohrazená. △

Příklad 4.5. Metoda umělých omezení – nepřípustná úloha.

Duální simplexovou metodou s využitím metody umělých omezení vyřešte následující minimalizační úlohu.

$$\begin{aligned} \min \quad & -2x_1 - x_2 \\ \text{za podmínek} \quad & x_1 + x_2 \leq 2 \\ & -3x_1 + x_2 \geq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Řešení. Stejně jako v předchozím příkladu si úlohu nejprve převedeme do kanonického tvaru, čímž získáme omezení ve tvaru

$$\begin{aligned} x_1 + x_2 + s_1 &= 2, \\ -3x_1 + x_2 + s_2 &= -3, \\ x_1, x_2, s_1, s_2 &\geq 0. \end{aligned}$$

Výchozí simplexová tabulka pro tuto úlohu je zobrazena v Tabulce 4.8.

c_B	B	x_1	x_2	s_1	s_2	x_B
0	s_1	1	1	1	0	2
0	s_2	3	-1	0	1	-3
	c	-2	-1	0	0	
	z	0	0	0	0	
	Δ	2	1	0	0	0

Tabulka 4.8: Výchozí simplexová tabulka pro Příklad 4.5

V tabulce je opět v Δ -řádku kladné číslo pro nebazické proměnné, tentokrát pro x_1 a x_2 . Stejně jako v předchozích příkladech přidáme umělé omezení, které rovnou doplníme do tvaru rovnosti. Máme tedy rovnost

$$x_1 + x_2 + s_3 = M.$$

Proměnnou s_3 zavedeme do báze a sestavíme novou tabulku. Ta je i s dalším postupem metody v Tabulce 4.9.

c_B	B	x_1	x_2	s_1	s_2	s_3	x_B
0	s_1	1	1	1	0	0	2
0	s_2	3	-1	0	1	0	-3
0	s_3	①	1	0	0	1	$M \rightarrow$
	c	-2	-1	0	0		
	z	0	0	0	0		
	Δ	2 ↑	1	0	0	0	0
	s_1	0	0	1	0	-1	$2 - M$
	s_2	0	④	0	1	-3	$-3 - 3M \rightarrow$
	x_1	1	1	0	0	1	M
	Δ	0	-1	0	0	-2	$-2M$
	Δ/λ	-	1/4 ↑	-	-	2/3	
	s_1	0	0	1	0	①	$2 - M \rightarrow$
	x_2	0	1	0	-1/4	3/4	$3/4 + 3M/4$
	x_1	1	0	0	1/4	1/4	$-3/4 + M/4$
	Δ	0	0	0	-1/4	-5/4	$3/4 - 5M/4$
	Δ/λ	-	-	-	-	5/4 ↑	
	s_3	0	0	-1	0	1	$-2 + M$
	x_2	0	1	3/4	-1/4	0	9/4
	x_1	1	0	1/4	1/4	0	-1/4
	Δ	0	0	-5/4	-1/4	0	-7/4

Tabulka 4.9: Duální simplexová metoda s využitím umělého omezení pro Příklad 4.5

V prvním kroku musíme do báze na místo s_3 zavést proměnnou x_1 , neboť v prvním sloupci je v Δ -řádku největší kladné číslo. Bod, který získáme Gaussovou eliminací podle pivota, je již duálně přípustný, neboť jsou v Δ -řádku pouze nekladná čísla. Můžeme tedy pokračovat s duální simplexovou metodou. Nejmenší záporné číslo ve sloupci x_B je ve druhém řádku. Z báze tedy vystoupí proměnná s_2 . Nejmenší podíl v řádku Δ/λ najdeme ve druhém sloupci, do báze tak vstupuje proměnná x_2 . Po tomto kroku duální metody se dostaneme do situace, ve které je v x_B jediné záporné číslo. Z báze v tomto kroku vystoupí proměnná s_1 . Jediná proměnná, jež nyní může vstoupit do báze, je s_3 .

Dalším krokem duální metody se dostaneme do situace, ve které můžeme vybrat klíčový řádek (bázi by mohla opustit proměnná x_1), nelze však již vybrat klíčový sloupec. Pro úlohu s umělým omezením tedy nejsme schopni najít primárně přípustný bod. Původní úloha je tak nepřípustná.

△

Příklad 4.6. Metoda umělých omezení – nekonečně-mnoho řešení.

Duální simplexovou metodou s využitím metody umělých omezení vyřešte následující minimalizační úlohu.

$$\begin{aligned} \min \quad & 2x_1 - 2x_2 \\ \text{za podmínek} \quad & 2x_1 + x_2 \geq 3 \\ & x_1 - x_2 \geq -3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Řešení. Převodem do kanonického tvaru tak, abychom v matici A měli jednotkovou matici, dostaneme omezení

$$\begin{aligned} -2x_1 - x_2 + s_1 &= -3, \\ -x_1 + x_2 + s_2 &= 3, \\ s_1, s_2 &\geq 0. \end{aligned}$$

Z koeficientů účelové funkce si všimneme, že podobně jako v předchozích příkladech bude v Δ -řádku pro proměnnou x_2 kladné číslo. Bod, který jsme tímto způsobem získali, tak není duálně přípustný. K úloze tedy přidáme omezení

$$x_1 + x_2 + s_3 = M$$

již upravené do tvaru rovnosti. Nyní si sestavíme simplexovou tabulku pro úlohu s umělým omezením. Ta je zobrazena v Tabulce 4.10.

c_B	B	x_1	x_2	s_1	s_2	s_3	x_B
0	s_1	-2	-1	1	0	0	-3
0	s_2	-1	1	0	1	0	3
0	s_3	1	①	0	0	1	$M \rightarrow$
	c	2	-2	0	0		
	z	0	0	0	0		
	Δ	-2	$2 \uparrow$	0	0	0	0
	s_1	-1	0	1	0	1	$-3 + M$
	s_2	②	0	0	1	-1	$3 - M \rightarrow$
	x_2	1	1	0	0	1	M
	Δ	-4	0	0	0	-2	$-2M$
	Δ/λ	$2 \uparrow$	-	-	-	2	
	s_1	0	0	1	0	-1/2	$-9/2 + 3M/2$
	x_1	1	0	0	-1/2	1/2	$-3/2 + M/2$
	x_2	0	1	0	1/2	1/2	$3/2 + M/2$
	Δ	0	0	0	-2	0	-6

Tabulka 4.10: Duální simplexová metoda s využitím umělého omezení pro Příklad 4.6

Po prvním kroku metody umělého omezení získáme duálně přípustný bod, můžeme tak pokračovat s duální simplexovou metodou. Ve druhém kroku dostaneme v řádku Δ/λ dvě stejná čísla. Do báze v tomto kroku zavedeme např. proměnnou x_1 , a to na místo proměnné s_2 . Tato volba byla náhodná, čtenář se však může přesvědčit, že kdybychom v tomto kroku zavedli do báze proměnnou s_3 , dostali bychom se v dalším kroku do situace, ve které najdeme řešení. Přesněji, našli bychom bod $[0, 3]$ (v původních proměnných), který je jak duálně, tak primárně přípustný. Jedná se tedy o řešení.

V případě, že do báze zavedeme x_1 se v dalším kroku též dostaneme do situace, ve které najdeme primárně přípustný bod. V poslední tabulce vidíme, že hodnota účelové funkce nezáleží na M a že proměnná s_3 není v bázi. Řešením je v tomto případě bod $x^* = [-3/2 + M/2, 3/2 + M/2]$, jehož souřadnice obsahují parametr M . To znamená, že řešení je nekonečně mnoho, ve všech je však hodnota účelové funkce rovna -6 . Volbou M dostatečně velkého (zde alespoň rovného 3, a to kvůli nezápornosti všech složek bodu x^*) vždy dostaneme přípustný bod, a tudíž i další řešení. Jedná se o polopřímka řešení vycházející z bodu $[0, 3]$, který je jediným řešením a zároveň bazickým přípustným bodem.

△

Nyní již víme, jak řešit úlohy lineárního programování simplexovou i duální simplexovou metodou, co je primární a duální úloha a jaká je jejich interpretace. V následující části práce si ukážeme praktické úlohy lineárního programování.

Kapitola 5

Příklady

V této kapitole se podíváme na vybrané typy úloh lineárního programování a jejich aplikace. Kapitola je rozdělena do šesti částí, v každé z nich si ukážeme jeden typ úlohy a příklad. Tyto příklady jsou komplexnější než motivační příklady, které jsme viděli v Kapitole 1. I v těchto příkladech však využíváme jistá zjednodušení reality. K výpočtu úloh byl využit doplněk Řešitel v Excelu, k práci je přiložen soubor obsahující zadání a řešení všech příkladů z této kapitoly. Stručný návod na aktivaci Řešitele a ukázkou jeho využití najde čtenář v příloze. K řešení úloh lineárního programování je možné využít celé řady softwarů, např. LINDO, LiPS či volně dostupné online aplikace. Jako první si ukážeme problém jídelníčku, který je jednou z typických úloh lineárního programování. Problém jídelníčku je, stejně jako další problémy, které uvidíme, motivován minimalizací nákladů. Dalšími problémy, se kterými se v kapitole setkáme, jsou vytvoření optimálního zásobovacího programu, vytvoření pracovního rozvrhu, řezný problém, investiční problém a výrobní problém.

5.1 Problém jídelníčku

Motivací této úlohy je sestavit co nejlevnější jídelníček tak, abychom z potravin získali všechny potřebné živiny, dostatek energie a vitamínů. Tento problém můžeme obecně naformulovat následujícím způsobem.

Máme n potravin, u kterých známe složení. Zajímá nás splnění požadavků na m živin. Obsah i -té živiny v j -té potravine označíme a_{ij} . Známe také ceny jednotlivých potravin p_j . Tyto ceny obvykle uvádíme normované na určitou jednotku potravin. Při sestavování jídelníčku musíme mít i minimální množství b_i jednotlivých živin, které chceme přijímat. Naším úkolem je najít optimální složení jídelníčku, tj. najít optimální množství x_j jednotlivých potravin.

Úloha lineárního programování, kterou v tomto případě řešíme, je ve tvaru

$$\begin{aligned} \min \quad & \sum_{j=1}^n p_j x_j \\ \text{za podmíněk} \quad & \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i \in \{1, \dots, m\}, \\ & x_j \geq 0, \quad j \in \{1, \dots, n\}. \end{aligned}$$

V případě, že známe i maximální množství c_i jednotlivých přijímaných živin, můžeme v úloze doplnit také podmínky ve tvaru

$$\sum_{j=1}^n a_{ij}x_j \leq c_i, \quad i \in \{1, \dots, m\}.$$

Jednoduchou ilustraci tohoto problému lze najít téměř ve všech titulech věnovaných lineárnímu programování. Tyto úlohy, viz např. [15], [21], [28], jsou vždy velmi jednoduché, jídelníček sestavujeme např. z 5 potravin. V této práci si uvedeme příklad jídelníčku, který skládáme ze 30 běžných potravin, u kterých sledujeme 7 výživových hodnot. Motivací tohoto příkladu tak je najít co nejlevnější osobní jídelníček studentky, která potřebuje ušetřit a zároveň mít dostatek sil na psaní diplomové práce.

Příklad 5.1. Optimální jídelníček.

Na začátku tohoto příkladu je nutné zmínit, že jídelníček je sestaven pro studentku. Potřebné výživové hodnoty tedy nejsou univerzálně platné. Při sestavování jídelníčku je důležité si stanovit, jaké výživové hodnoty budeme sledovat a kolik bychom jich měli denně přijímat. Pro tento příklad jde o 7 hodnot, a to celkovou energetickou hodnotu (v kJ), množství tuku (g), sacharidů (g), bílkovin (g), vlákniny (g), vitamínu C (mg) a obsahu vody (ml).

Otázka kolik čeho jíst je při sestavování jídelníčku velmi důležitá, mírně však přesahuje obsah této práce. Pro reálnost hodnot však byly pro zjištění těchto informací využity [24], [33] a [48]. Pro vypočtení doporučených hodnot pro celkovou energii byla využita tzv. *Harrisova–Benedictova rovnice*. Z této rovnice na základě věku, výšky, váhy a pohlaví zjistíme, kolik je tzv. *bazální metabolismus*, který udává výdej energie v klidovém stavu. Tento údaj poté upravíme podle typu činností, které daný člověk vykonává. Pro studentku je tak potřebné přijmout přibližně 8 060 kJ energie denně. Z výživových doporučení a obsahu energie v jednotlivých typech živin pak vypočítáme, kolik z toho by měla přijmout ve formě tuku, sacharidů, bílkovin atd. Tímto postupem zjistíme, že doporučený denní příjem pro studentku je 70 gramů tuku, 210 gramů sacharidů, 100 gramů bílkovin, 25 gramů vlákniny a 80 miligramů vitamínu C. Je také nutné, aby vypila alespoň 2 litry vody.

Dalším krokem při sestavování jídelníčku je výběr potravin, které budeme uvažovat. Pro tento příklad jsou potraviny rozděleny do 6 kategorií, a to nápoje, mléčné výrobky, maso, ovoce a zelenina, přílohy a sladkosti. V každé kategorii je vybráno několik potravin, které jsou běžně dostupné a tvoří velkou část obvyklých jídelníčků. U těchto potravin jsou uvedeny také jejich výživové hodnoty, a to z [49], [50] a [51], odkud jsou převzaty též ceny jednotlivých výrobků. Výživové hodnoty i ceny všech potravin, s výjimkou vajec, jsou uvedeny pro 100 gramů potraviny. Hodnoty u vejce jsou přepočteny pro jeden kus, vejce váží přibližně 70 gramů. Seznam vybraných potravin, jejich ceny a výživové hodnoty jsou uvedeny v Tabulce 5.1. V posledním řádku tabulky jsou uvedeny doporučené denní dávky jednotlivých živin.

Nyní již známe všechny potřebné informace a můžeme tak sestavit a vyřešit konkrétní problém jídelníčku. Po vyřešení se ještě pozastavíme nad situací, kdy studentka uvažuje i některá maximální omezení a případem, ve kterém chce do svého jídelníčku přidat alespoň jednu potravinu z vybrané kategorie. Vzhledem k velikosti této úlohy zde nebudeme uvádět průběh simplexové metody, ale pouze výsledek zadané úlohy.

surovina (100 g)	energie (kJ)	tuk (g)	sacharidy (g)	bílkoviny (g)	vláknina (g)	vitamín C (mg)	voda (ml)	cena (Kč)
neperlivá voda	0	0	0	0	0	0	100	0,8
Coca-Cola	184,8	0	11,2	0	0	0	100	2,79
Poděbradka cit.	75,6	0	4,2	0	0	0	100	1,06
bílý jogurt	280	3,1	5	4,9	0	0	0	6,5
polotučné mléko	192	1,5	4,8	3,2	0	0	0	1,99
eidam (30 %)	1215	19	0,3	30	0	0	0	20,9
cottage	497	7	3	11	0	0	0	18,9
kuřecí prsa	486	2,1	0	17,8	0	0	0	14,9
tuňák	461	1	0	24	0	0	0	29,9
vejce	512	8,5	0,53	11,2	0	0	0	2,99
kuřecí šunka	702	11	0	18,2	0	0	0	24,9
losos	902	15,9	0	19,8	0	0	0	34,9
jablko	218,4	0,2	14	0,3	2,4	4,6	0	2,99
banán	369,6	0,3	23	1,1	2,6	8,7	0	2,99
kiwi	252	0,5	15	1,1	3	92,7	0	4,99
mrkev	172,2	0,2	10	0,9	2,8	5,9	0	1,99
rajče	71,4	0,2	3,9	0,9	1,2	13,7	0	5,99
okurka	63	0,1	3,6	0,7	0,5	2,8	0	6
kukuřice (konz.)	309	1,4	11	2,7	4	0	0	7
hrách (konz.)	308	0,7	7,4	5,5	7,8	0	0	6
brambory	336	0,4	19,2	2	2,2	19,7	0	3,2
těstoviny	1603	1,3	78	10,2	3,6	0	0	1,99
ovesné vločky	1571	3	60,3	14,2	2	0	0	4,9
rýže	1489	2,9	79	8,6	0,4	0	0	3,99
kuskus	1563	2,1	73	12,9	4,2	0	0	6,99
chléb kmínový	950	1	46,6	7,6	2,7	0	0	2,99
čočka	1306	1,6	39	24	8	0	0	3,99
Tatranka	2110	28	54	5,8	4,6	0	0	19,8
Jojo medvídci	1350	0,3	74	5	0,67	0	0	24,9
Milka mléčná	2213	29	59	6,3	1,8	0	0	29,9
denní příjem	8060	70	210	100	25	80	2000	

Tabulka 5.1: Výživové hodnoty

Řešení. Označme množství potravin v optimálním jídelníčku postupně x_1, \dots, x_{30} . Ceny těchto surovin označíme p_i . Jednotlivé výživové hodnoty označíme podle pořadí jim příslušných sloupců v Tabulce 5.1.

Problém, který studentka řeší, můžeme zapsat jako

$$\begin{aligned} \min \quad & \sum_{i=1}^{30} p_i x_i \\ \text{za podmíněk} \quad & \sum_{i=1}^{30} a_{i1} x_i \geq 8060, \quad \sum_{i=1}^{30} a_{i2} x_i \geq 70, \quad \sum_{i=1}^{30} a_{i3} x_i \geq 210, \quad \sum_{i=1}^{30} a_{i4} x_i \geq 100, \\ & \sum_{i=1}^{30} a_{i5} x_i \geq 25, \quad \sum_{i=1}^{30} a_{i6} x_i \geq 80, \quad \sum_{i=1}^{30} a_{i7} x_i \geq 2000, \\ & x_i \geq 0, \quad i \in \{1, \dots, 30\}. \end{aligned}$$

Řešením této úlohy je situace, ve které studentka pije pouze vodu (2 litry) a jí 7,23 vejce, 86 gramů kiwi a 622,5 gramů těstovin. Za den tak za svoje stravování zaplatí 54,32 Kč. Označme si tento scénář jako jídelníček 1. Výživové hodnoty tohoto jídelníčku jsou zobrazeny v Tabulce 5.2.

surovina (100 g)	energie (kJ)	tuk (g)	sacharidy (g)	bílkoviny (g)	vláknina (g)	vitamín C (mg)	voda (ml)	cena (Kč)
denní příjem	8060	70	210	100	25	80	2000	
jídelníček 1	13899,6	70	502,35	145,45	25	80	2000	54,32

Tabulka 5.2: Výživové hodnoty pro jídelníček 1

V Tabulce 5.2 vidíme, že jídelníček 1 obsahuje téměř 14 000 kJ energie, což je oproti požadovaným 8 060 o dost více. Studentka sice chce ušetřit, nechce ale přijímat více energie než potřebuje a ztloustnout. Také není spokojená s vysokým příjmem sacharidů. Rozhodneme se tedy přidat k problému další podmínky, a to maximální denní příjem energie na 9 000 kJ a sacharidů na 280 g. Nová omezení jsou ve tvaru

$$\sum_{i=1}^{30} a_{i1} x_i \leq 9000, \quad \sum_{i=1}^{30} a_{i3} x_i \leq 280.$$

Vyřešíme-li úlohu s přidáním omezení, získáme jídelníček, který označíme číslem 2. Jídelníček 2 se skládá z následujících množství jednotlivých potravin: 2 litry vody, 7,59 vejce, 81 gramů kiwi, 82 gramů mrkve, 143,6 gramu těstovin a 188,7 gramu čočky. Tento jídelníček musí být dražší než jídelníček 1, růst ceny není ovšem tak dramatický, jedná se pouze o 45 haléřů, viz Tabulka 5.3. S tímto jídelníčkem je studentka již spokojenější, nepřijímá totiž zbytečně moc energie, všechny výživové hodnoty (až na bílkoviny) jsou v podstatě na své doporučené hodnotě a navíc je strava pestřejší.

Přesto se studentce zdá, že přijímá až moc energie, pro její běžný život by podle výpočtu mělo stačit 8 060 kJ. Rozhodne se tedy změnit maximální denní příjem energie na 8 500. Zároveň je právě uprostřed psaní diplomové práce a ráda by si občas dopřála nějakou sladkost, a to alespoň 35 gramů denně. V úloze tedy změníme maximální denní příjem energie na 8 500 a přidáme omezení ve tvaru

$$x_{28} + x_{29} + x_{30} \geq 0,35.$$

Vyřešíme-li úlohu s těmito omezeními, získáme následující stravovací plán: 2 litry vody, 6,53 vejce, 73 gramů kiwi, 206,39 gramu mrkve, 133,77 gramu těstovin, 132,5 gramu čočky a 35 gramů Tatranky. Přidáním dalších omezení se stravovací plán opět zdražil, za den by tak studentku stál 58,18 Kč. Na to je ale ochotná přistoupit, neboť si denně může dát téměř celou Tatranku a ostatní živiny přijímá v dostatečném a přiměřeném množství, viz Tabulka 5.3. S tímto jídelníčkem je tak spokojená.

surovina (100 g)	energie (kJ)	tuk (g)	sacharidy (g)	bílkoviny (g)	vláknina (g)	vitamín C (mg)	voda (ml)	cena (Kč)
denní příjem	8060	70	210	100	25	80	2000	
jídelníček 1	13899,6	70	502,35	145,45	25	80	2000	54,32
jídelníček 2	9000	70	210	146,61	25	80	2000	54,77
jídelníček 3	8500	70	210	123,35	25	80	2000	58,18

Tabulka 5.3: Výživové hodnoty pro jídelníčky

△

Tento příklad je samozřejmě pouze ilustrativním problémem jídelníčku. I s tímto počtem surovin jsme ale schopni poskládat jídelníček tak, abychom měli dostatek všech sledovaných výživových hodnot. Tento příklad také předpokládá dokonalou dělitelnost všech potravin, neboť jde o problém lineárního programování. Je snadné zkonsumovat přesně 73 gramů kiwi, méně snadná může být konzumace necelého počtu vajec. V běžných obchodech navíc nemůžeme nakoupit námi vypočítaná množství surovin. Kdybychom tento jídelníček však chtěli dodržovat dlouhodoběji, např. 10 dní v kuse, nakoupili bychom požadovaná množství bez problémů. Navzdory těmto problémům je ale příklad velmi užitečný. Snadno jsme totiž zjistili, jaká přibližná množství zaručí studentce dostatek energie a dalších požadovaných živin a za jakou cenu se dá takový optimální jídelníček nakoupit.

Obecně se problémy, jako je tento, označují přívláskem *směšovací*. Směšovací problémy mají v praxi široké využití, a to jak při přípravě jídelníčku či míchání krmných směsí, tak např. při míchání oceli nebo v petrochemickém průmyslu. Další zajímavou aplikací je vytváření strategií reklamních kampaní. U každé z těchto strategií známe dosah a také cenu, snažíme se tak oslovit určitý počet lidí (např. napříč regiony, demografickými skupinami, atp.) s využitím co nejmenšího rozpočtu. V [26] najdeme směšovací úlohu motivovanou hledáním optimální léčby s využitím léků různého složení. Kromě omezení na minimální obsah složek se můžeme setkat také s omezeními na maximální obsah, na povinný obsah některé složky či celočíselný požadavek na některé proměnné. S prvními dvěma zmíněnými dodatečnými jsme se setkali při sestavování jídelníčku pro studentku. Požadavek celočíselnosti vyžaduje jiné metody řešení, neboť již nejde o úlohu lineárního programování. Stručně ho však okomentujeme v následující části textu.

5.2 Minimalizace nákladů na zásobování

V této části si ukážeme další úlohu motivovanou nalezením minimálních nákladů. Dosud jsme se setkali se dvěma úlohami se stejnou motivací, a to v Kapitole 4 při interpretaci

duální úlohy a při hledání optimálního jídelníčku. Úlohy tohoto typu, společně s úlohami o maximalizaci zisku, hrají v ekonomii významnou roli, neboť se dotýkají všech ekonomických subjektů. Nyní zde uvedeme příklad inspirovaný využitím lineárního programování pro plánování zásobování v poválečném Berlíně z [1].

Příklad 5.2. Letecké zásobování Wu-chanu.

Na konci ledna 2020 byla kvůli šíření koronaviru pro čínské město Wu-chan vyhlášena karanténa. Jedním ze způsobů, jak do Wu-chanu dovézt potřebné zdravotnické pomůcky, jídlo a další nezbytné věci, je letecká doprava. Představme si, že jsme v roli poradců státu, kteří plánují dopravit do Wu-chanu ve 4 obdobích postupně 200 000, 300 000, 300 000 a 400 000 tun zboží. Kapacita letadel, které máme k dispozici je 2 000 tun. Naším cílem je najít plán zásobování, který minimalizuje náklady. Operace, kterou plánujeme, je však poměrně náročná, neboť existuje několik opatření, na které musíme brát ohledy. Dále se musíme potýkat s omezeným množstvím pilotů i letadel. Všechna nařízení a omezení si nyní rozebereme podrobněji.

1. V prvním období máme k dispozici 330 pilotů. Každé letadlo musí obsluhovat 3 piloti. Na začátku operace máme 110 letadel. V průběhu času můžeme najímat a cvičit nové piloty i zakoupit nová letadla, o jejich počtech se však musíme rozhodnout vždy o jedno období dopředu.
2. V každém období přijdeme o 20 % pilotů, a to z důvodu nakažení koronavirem, i letadel, která zůstanou v Číně, neboť piloti jsou v karanténě. Přijdeme o ně vždy až po doručení zásob. Podle nařízení týkajících se letecké bezpečnosti pak piloti, kteří se vrátí zpět, musí alespoň 1 období odpočívat.
3. Piloti, kteří v daném období neletí do Wu-chanu, mohou být v záloze, odpočívat (pokud minulé období letěli) nebo trénovat nové piloty. Pilot, který se ujme tréninku, je v jednom období schopen vytrénovat 19 pilotů (včetně něho tedy 20 pilotů) schopných pracovat v následujícím období.
4. V každém období můžeme nakoupit nová letadla. Jejich ceny jsou společně s náklady na různé aktivity pilotů uvedeny v Tabulce 5.4. Náklady na letadla, která v daném období nevyužijeme, jsou nulové. Všechny ceny jsou uvedeny ve stejných měnových jednotkách.

náklady	období 1	období 2	období 3	období 4
nové letadlo	400	375	350	325
odpočívající pilot	5	4,9	4,8	4,7
pilot v záloze	7	6,9	6,8	6,7
trénující pilot	14,3	11,9	9,8	9,3

Tabulka 5.4: Náklady pro jednotlivá období

Při plánování zásobování Wu-chanu nebereme v potaz cenu materiálu, který vezeme. Zajímá nás pouze nalezení strategie, která minimalizuje náklady na samotný převoz. S využitím těchto omezení formulujeme úlohu, jejímž výsledkem bude nalezení minimálních nákladů na zásobovací akci.

Řešení. Pro přehlednost si nejprve zavedeme značení. Označme a_i množství zboží, které musíme v i -tém období dopravit do Wu-chanu ve 100 000 tunách, pro celou úlohu platí $i \in \{1, 2, 3, 4\}$. Zkratkou ns_i označíme počet letadel, která v i -tém období nepoužijeme, ks_i pak označuje počet nově zakoupených letadel v daném období. Pro piloty zavedeme následující značení: op_i pro odpočívající piloty, zp_i pro piloty v záloze a np_i pro piloty, kteří se v i -tém období věnují tréninku (a to jak trenéři, tak nově trénovaní). Počet trenérů je pak $np_i/20$. O problémech spojených s neceločíselností tohoto výrazu se zmíníme po vyřešení úlohy.

Ze zadání víme, že pro množství převezeného zboží musí platit

$$a_1 = 2, \quad a_2 = 3, \quad a_3 = 3, \quad a_4 = 4.$$

Nyní si zformulujeme podmínky, které musí splňovat počty letadel. Pro převezení 100 000 tun zboží potřebujeme 50 letadel. V každém období tak pro převoz využijeme $50a_i$ letadel. Zbytek letadel, která máme v daném období k dispozici, nevyužijeme. Pro první období tak můžeme psát

$$50a_1 + ns_1 = 110,$$

neboť na začátku operace máme k dispozici 110 letadel. Pro následující období je situace složitější. Z letadel, která jsme v předchozím období využili, nám jich zůstane 80%. Letadla ve 2. období mohou mít opět 2 funkce, mohou být nasazeny do akce, nebo mohou být nevyužitá. Počet těchto letadel získáme součtem 80% letadel použitých v minulém období (tj. $40a_1$), počtem nově zakoupených letadel v minulém období (ks_1) a letadel, která jsme v minulém období nevyužili (ns_1). Celkově tedy dostaneme

$$50a_2 + ns_2 = 40a_1 + ks_1 + ns_1.$$

Pro období 3 a 4 postupujeme obdobně, čímž získáme rovnosti

$$50a_3 + ns_3 = 40a_2 + ks_2 + ns_2,$$

$$50a_4 + ns_4 = 40a_3 + ks_3 + ns_3.$$

Nyní se zaměříme na podmínky, které musí splňovat počty pilotů. K obsluze jednoho letadla potřebujeme vždy 3 piloty. Počet pilotů, kteří v daném období letí do Wu-chanu, je tak třikrát větší, než počet letadel, tj. $150a_i$. Další role, kterou mohou piloti mít, je trénování nových pilotů. Abychom pro následující období vytrénovali np_i pilotů, potřebujeme $np_i/20$ pilotů trenérů. Piloti mohou mít ještě 2 činnosti, a to odpočinek po absolvování cesty do Wu-chanu, nebo být v záloze. Pro piloty v 1. období tak musí platit rovnost

$$150a_1 + 0,05np_1 + zp_1 = 330,$$

neboť na začátku můžeme využít 330 pilotů. Ve druhém období můžeme využít piloty, kteří byli v předchozím období v záloze nebo trénovali. V dalších obdobích pak budeme

moci navíc využít piloty, kteří v předchozím období odpočívali. Pro piloty ve 2. období pak platí rovnost

$$150a_2 + 0,05np_2 + zp_2 = zp_1 + np_1.$$

Pro piloty v následujících obdobích získáme obdobným postupem rovnosti

$$150a_3 + 0,05np_3 + zp_3 = zp_2 + np_2 + op_2,$$

$$150a_4 + 0,05np_4 + zp_4 = zp_3 + np_3 + op_3.$$

Poslední podmínky se týkají pilotů, kteří odpočívají po výkonu. Počet pilotů, kteří ve 2. období odpočívají, můžeme zapsat jako $120a_1$, neboť 20 % pilotů, kteří v 1. období letěli, muselo zůstat v karanténě ve Wu-chanu. Pro odpočívající piloty tedy platí rovnosti

$$op_1 = 0,$$

$$op_2 = 120a_1,$$

$$op_3 = 120a_2,$$

$$op_4 = 120a_3.$$

Všechny proměnné jsou navíc omezené na znaménko, tj. platí

$$a_i \geq 0, ns_i \geq 0, ks_i \geq 0, op_i \geq 0, zp_i \geq 0, np_i \geq 0.$$

Nyní již máme všechny podmínky odpovídající požadavkům uvedeným v zadání. Funkce, kterou minimalizujeme, je formálně funkcí 24 proměnných, 8 z nich v ní však vystupuje s nulovým koeficientem (všechna a_i a ns_i), neboť náklady na tyto proměnné jsou nulové. Účelovou funkci tak s využitím nákladů z Tabulky 5.4 zapíšeme jako

$$400ks_1 + 375ks_2 + 350ks_3 + 325ks_4 + 5op_1 + 4,9op_2 + 4,8op_3 + 4,7op_4 + \\ + 7zp_1 + 6,9zp_2 + 6,8zp_3 + 6,7zp_4 + 14,3np_1 + 11,9np_2 + 9,8np_3 + 9,3np_4.$$

Všechna omezení jsou ve tvaru rovností a všechny proměnné jsou omezené na znaménko, máme tedy úlohu v kanonickém tvaru. Omezení upravíme tak, aby jednotlivé proměnné byly na levé straně. Pro přehlednost si omezení, která jsme tímto způsobem získali, uvedeme v Tabulce 5.5. V této tabulce jsou zobrazeny i koeficienty účelové funkce.

Vzhledem k tomu, že úlohu řešíme jako problém lineárního programování, nemáme zaručeno, že získáme celočíselné výsledky. Výsledky této úlohy (zaokrouhlené na 2 desetinná místa) jsou zobrazeny v Tabulce 5.6.

Hodnota účelové funkce v bodě z Tabulky 5.6 je 79 380,16. V bodě, ve kterém se realizují minimální náklady, musíme v 1. období nakoupit 60 nových letadel, ve 2. období 30 letadel a ve 3. období 80 letadel. Ve všech obdobích, s výjimkou prvního, využijeme k zásobování všechna dostupná letadla. Počty odpočívajících pilotů jsou jasně dány počty pilotů, kteří pracovali v minulém období.

náklad				letadla navíc				nová letadla				odpočívající piloti				piloti v záloze				trénující piloti				omezení
a_1	a_2	a_3	a_4	ns_1	ns_2	ns_3	ns_4	ks_1	ks_2	ks_3	ks_4	op_1	op_2	op_3	op_4	zp_1	zp_2	zp_3	zp_4	np_1	np_2	np_3	np_4	
0	0	0	0	0	0	0	0	400	375	350	325	5	4,9	4,8	4,7	7	6,9	6,8	6,7	14,3	11,9	9,8	9,3	min
1																								2
	1																							3
		1																						3
			1																					4
50				1																				110
-40	50			-1	1			-1																0
	-40	50			-1	1			-1															0
		-40	50			-1	1			-1														0
150																								330
	150																			0,05				0
		150																		-1	0,05			0
			150																	-1	-1	0,05		0
																				-1	-1	-1	0,05	0
																				1	1	1	0,05	0
																				-1	-1	-1	0,05	0
																								0
-120												1												0
	-120												1											0
		-120												1										0
			-120												1									0

Tabulka 5.5: Omezení pro Příklad 5.2

období \ proměnná	proměnná					
	a_i	ns_i	ks_i	op_i	zp_i	np_i
1	2	10	60	0	7,31	453,79
2	3	0	30	240	0	222
3	3	0	80	360	0	240
4	4	0	0	360	0	0

Tabulka 5.6: Výsledky Příkladu 5.2

V případě pilotů v záloze a trénujících pilotů již pro první období dostaneme celočíselný výsledek. Jeho prostým zaokrouhlením můžeme získat představu o optimálním celočíselném bodě, tato strategie ale nezafunguje vždy. Bod, který tímto způsobem získáme, také nemusí splňovat všechna omezení. Řešení problémů celočíselného programování metodami pro lineární programování může být i tak užitečné, neboť jsou zpravidla rychlejší a výsledek, který získáme, nám může dát dostatečně přesnou informaci o celočíselném výsledku. Hodnota účelové funkce pro celočíselné hodnoty navíc nemůže být menší než hodnota, kterou jsme našli. Zaokrouhlíme-li počet pilotů v záloze, získáme pro 1. období výsledek 7. U trénujících pilotů je situace obtížnější. Pro 453,79 pilotů potřebujeme $453,79/20 = 22,6895$ trenérů. Jestliže zaokrouhlíme počet trenérů v 1. období na 23, budeme mít ve 2. období k dispozici 460 nově vytrénovaných pilotů. Tímto zásahem se změní hodnoty některých omezení, což ovlivní např. i počty pilotů v záloze (v 1. období jich vycvičíme více, než je optimální, počty pilotů v záloze tak vzrostou). V případě, že bychom chtěli získat celočíselný výsledek, ale využít pouze simplexovou metodu, bychom nyní mohli problém přepočítat pro období 2-4, výsledky pro 2. období vhodně zaokrouhlit a pokračovat pro období 3 a 4. Na závěr bychom museli ověřit, že bod, který tímto způsobem získáme, leží v přípustné množině.

I přesto, že jsme řešením tohoto problému simplexovou metodou získali pouze přibližný výsledek, jsme si ukázali další zajímavou aplikaci lineárního programování. Vzhledem ke tvaru omezení této úlohy bychom ji mohli rozdělit na dvě podúlohy – jednu pro letadla, druhou pro piloty. Výsledek bychom poté získali součtem hodnot těchto úloh.

△

Příklad, který jsme právě viděli, je zajímavý i tím, že optimalizujeme na několik období, která jsou na sobě závislá. Příklad bychom mohli též rozdělit na 4 samostatné příklady, které by se týkaly jednotlivých období. Jedná se tak vlastně o problém tzv. *dynamického programování*, více viz např. [5] a [14].

I s příklady jako je tento se můžeme setkat i v běžném životě. Aktuální problém, který při začínající epidemii koronaviru řeší běžné domácnosti, je jestli a jak se dobře zásobit. Řeší tak problém podobný tomu, který jsme si zde ukázali, na rozdíl od příkladu však dochází k růstu cen. Ztrátu mezi obdobími si můžeme motivovat např. tím, že část potravin se v průběhu času kazí. Do hry by navíc mohla vstupovat množstevní omezení na jeden nákup atd. Další aplikací lineárního programování, kterou si zde uvedeme, je vytváření pracovních rozvrhů.

5.3 Pracovní rozvrhy

V této části si ukážeme další možnost využití lineárního programování v běžném životě, kterou je sestavování rozvrhů či plánování pracovních směn. V provozech, jako jsou kina, restaurace, továrny či některá oddělení nemocnic je nutné naplánovat směny tak, aby byl vždy přítomen dostatečný počet pracovníků. Pracovníci mohou být homogenní, tj. všichni jsou schopni dělat stejnou práci (s touto možností se můžeme setkat např. v kinech, kde všichni brigádníci dělají všechny úkony), nebo se mohou ve svých schopnostech lišit (např. sestry a lékaři v nemocnicích). Při plánování je též nutné dodržovat předpisy, např. o maximálním počtu odpracovaných hodin, atp. V následujícím příkladu si ukážeme, jak vytvořit pracovní rozvrh v případě nemocnice. V tomto příkladu ilustrujeme jak situaci homogenních pracovníků, tak případ, kdy máme dvě různé skupiny zaměstnanců.

Příklad 5.3. Rozdělení lékařů na směny v nemocnici

Představme si nemocnici, ve které na pohotovosti pracuje 60 lékařů. Lékaři pracují v třísměnném provozu, směny jsou osmihodinové. Pro jednoduchost předpokládejme, že směny začínají vždy o půlnoci (směna 1), v 8:00 (směna 2) a v 16:00 (směna 3). Vedení nemocnice se chystá vytvořit optimální plán směn tak, aby využilo co nejméně lékařů. Z dlouhodobého pozorování vedení zná minimální počet zaměstnanců, kteří musí být přítomni na jednotlivých směnách. Ten je uveden v Tabulce 5.7.

směna \ den	PO	ÚT	ST	ČT	PÁ	SO	NE
1	9	3	4	4	3	10	12
2	7	9	14	8	7	15	15
3	7	11	10	9	11	12	13

Tabulka 5.7: Zadání Příkladu 5.3

Lékaři si u vedení nemocnice za pomoci odborů vydobyli následující podmínky:

1. Pracovat vždy 4 po sobě jdoucí dny v týdnu a poté 3 po sobě jdoucí dny nepracovat.
2. Ve čtyřech po sobě následujících dnech pracovat vždy na stejné směně (např. 4x za sebou mít směnu 1).

Sestavme a vyřešme lineární úlohu, kterou řeší vedení nemocnice. Vedení chce najít dlouhodobý rozvrh, podle kterého se budou směny řídit, ne pouze rozvrh na jeden týden. Vedení tak při řešení počítá s tím, že na pondělní směně pracují lékaři, kteří začali čtyřdenní směnu minulý týden. Jejich cílem je minimalizovat množství zaměstnanců, kteří pokryjí potřebné minimální počty. Tuto úlohu bychom mohli formulovat i tak, že chceme minimalizovat náklady na platy zaměstnanců, nechceme tedy, aby v době, kdy potřebujeme 12 zaměstnanců jich na pracovišti bylo např. 20.

Ve druhé části příkladu se podíváme na situaci, ve které jsou zaměstnanci dvou různých typů. Vzhledem k propukající epidemii hlásí nemocnice mírný nárůst počtů pacientů, a tedy i počtů lékařů potřebných na jednotlivých směnách. Tyto počty jsou uvedeny v Tabulce 5.8.

směna \ den	PO	ÚT	ST	ČT	PÁ	SO	NE
1	9	3	3	5	5	15	12
2	7	9	15	8	7	15	15
3	7	11	10	9	12	12	13

Tabulka 5.8: Zadání druhé části Příkladu 5.3

Nárůst požadavků na lékaře není prozatím nijak závratný, jejich maximální počet se však snížil na 45, neboť někteří jsou kvůli pobytu v rizikových oblastech nyní uzavřeni v domácí karanténě. Místní univerzita však nemocnici nabídla studenty lékařské fakulty, kteří mohou vypomáhat. Studentů je k dispozici 100. Studenty, na rozdíl od stálých zaměstnanců, najímá nemocnice pouze na jednotlivé směny, a mohou pracovat bez dalších omezení.

Jediné bezpečnostní omezení, které nemocnice vyžaduje, je, aby studenti na všech směnách tvořili maximálně jednu třetinu ze všech potřebných lékařů. V této části se již nebudeme snažit minimalizovat celkový počet zaměstnanců, ale celkové náklady na ně. Mzdu všech lékařů i studentů uvádíme ve stejných jednotkách. Mzda jednoho zaměstnance je 42 (za 4 směny), studenti dostanou 8 za práci na směně 1 nebo směně 2 a 14 za práci na směně 3. Sestavme úlohu, která odpovídá druhé části příkladu a vyřešme ji.

Řešení. Pro přehlednost si nejprve zavedeme značení. Značení bude stejné pro první i druhou část příkladu. Označme dny v týdnu čísla 1-7 a směny čísla 1, 2 a 3. Dále označme N_{ij} počet lékařů, kteří začínají svou práci v i -tém dni na j -té směně. Dále označme D_{ij} počty potřebných zaměstnanců v i -tém dni na j -té směně. Z tabulky víme, že např. $D_{13} = 7$. V první části příkladu chceme minimalizovat počet lékařů, kteří budou v daném týdnu pracovat. Účelová funkce je tedy

$$\sum_{i=1}^7 \sum_{j=1}^3 N_{ij}.$$

Prvním omezením, které máme, je omezení na maximální počet lékařů, které můžeme využít, tj.

$$\sum_{i=1}^7 \sum_{j=1}^3 N_{ij} \leq 60.$$

Nyní sestavíme omezení pro jednotlivé směny. Lékaři, kteří začnou 1. směnou v pondělí, budou pracovat na 1. směně i v úterý, ve středu a ve čtvrtek. Ve čtvrtek tak pracují zaměstnanci, kteří začali v pondělí, v úterý, ve středu a ve čtvrtek. Pro první směnu ve čtvrtek tak máme omezení ve tvaru

$$N_{11} + N_{21} + N_{31} + N_{41} \geq D_{41} = 4.$$

Stejnou úvahou dostaneme omezení pro ostatní čtvrteční směny. Pro jednotlivé dny tímto

způsobem získáme 21 omezení, která pro $j \in \{1, 2, 3\}$ jsou

$$\begin{aligned} N_{1j} + N_{5j} + N_{6j} + N_{7j} &\geq D_{1j}, \\ N_{1j} + N_{2j} + N_{6j} + N_{7j} &\geq D_{2j}, \\ N_{1j} + N_{2j} + N_{3j} + N_{7j} &\geq D_{3j}, \\ N_{1j} + N_{2j} + N_{3j} + N_{4j} &\geq D_{4j}, \\ N_{2j} + N_{3j} + N_{4j} + N_{5j} &\geq D_{5j}, \\ N_{3j} + N_{4j} + N_{5j} + N_{6j} &\geq D_{6j}, \\ N_{4j} + N_{5j} + N_{6j} + N_{7j} &\geq D_{7j}. \end{aligned}$$

Posledním omezením, které uvažujeme, je nezápornost všech N_{ij} . Řešení této úlohy je zobrazeno v Tabulce 5.9.

směna \ den	PO	ÚT	ST	ČT	PÁ	SO	NE
1	0	0	1	3	6	0	3
2	0	0	7	6	0	2	7
3	0	3	3	5	0	4	4

Tabulka 5.9: Řešení první části Příkladu 5.3

směna \ den	PO	ÚT	ST	ČT	PÁ	SO	NE
1	9	3	4	4	10	10	12
2	9	9	14	13	13	15	15
3	8	11	10	11	11	12	13

Tabulka 5.10: Řešení první části Příkladu 5.3 – celkové počty

Z Tabulky 5.9 vidíme například, že v pondělí na směny nenastupují žádní noví lékaři, neboť kapacitu pokryjí lékaři, kteří začaly své směny v pátek, sobotu a neděli. Sečtením všech čísel v této tabulce získáme celkový počet lékařů, který na daný týden potřebujeme, jenž je 54. Celočíslnost výsledku, který jsme řešením úlohy simplexovou metodou získali, je způsobena přesným tvarem zadání, nejedná se o žádné pravidlo. V Tabulce 5.10 vidíme celkové počty lékařů, kteří jsou přítomni na jednotlivých směnách. Porovnáním této tabulky s Tabulkou 5.7 čtenář snadno ověří, že na všech směnách je dostatek lékařů.

Druhou část úlohy řešíme obdobně. Označme A_{ij} počet studentů vypomáhajících v i -tém dni na j -té směně. Účelovou funkci, jejíž minimum hledáme v této části příkladu, tak zapíšeme jako

$$42 \sum_{i=1}^7 \sum_{j=1}^3 N_{ij} + 8 \sum_{i=1}^7 (A_{i1} + A_{i2}) + 14 \sum_{i=1}^7 A_{i3}.$$

Omezení na maximální počet lékařů z první části příkladu se nyní změní na

$$\sum_{i=1}^7 \sum_{j=1}^3 N_{ij} \leq 45.$$

Změní se též omezení na minimální počty zaměstnanců, která jsou nyní pro $j \in \{1, 2, 3\}$ ve tvaru

$$N_{1j} + N_{5j} + N_{6j} + N_{7j} + A_{1j} \geq D_{1j},$$

$$N_{1j} + N_{2j} + N_{6j} + N_{7j} + A_{2j} \geq D_{2j},$$

$$N_{1j} + N_{2j} + N_{3j} + N_{7j} + A_{3j} \geq D_{3j},$$

$$N_{1j} + N_{2j} + N_{3j} + N_{4j} + A_{4j} \geq D_{4j},$$

$$N_{2j} + N_{3j} + N_{4j} + N_{5j} + A_{5j} \geq D_{5j},$$

$$N_{3j} + N_{4j} + N_{5j} + N_{6j} + A_{6j} \geq D_{6j},$$

$$N_{4j} + N_{5j} + N_{6j} + N_{7j} + A_{7j} \geq D_{7j}.$$

Kromě těchto omezení musíme k úloze přidat též omezení na maximální počet studentů, které můžeme v krizové situaci zaměstnat, tj.

$$\sum_{i=1}^7 \sum_{j=1}^3 A_{ij} \leq 100.$$

Omezení, která jsou dána bezpečnostním nařízením nemocnice o maximálním poměru studentů na počet potřebných lékařů, jsou pro $i \in \{1, \dots, 7\}$ a $j \in \{1, 2, 3\}$ ve tvaru

$$A_{ij} \leq D_{ij}/3, \quad \text{neboli} \quad 3A_{ij} \leq D_{ij}.$$

Tímto k úloze přidáme dalších 21 omezení. I v této části úlohy požadujeme nezápornost všech proměnných.

směna \ den	den						
	PO	ÚT	ST	ČT	PÁ	SO	NE
1	0	0	2	2	3	3	0
2	0	0	5	3	0	2	5
3	0	7	0	2	3	4	0
1	3	0	1	1	0	5	4
2	0	2	5	0	0	5	5
3	0	0	3	0	0	3	4

Tabulka 5.11: Řešení druhé části Příkladu 5.3

Řešení druhé části úlohy je zobrazeno v Tabulce 5.11. V první části tabulky jsou počty lékařů, kteří začínají pracovat v daný den a pracují 4 dny, ve druhé pak počty studentů, kteří pracují pouze na konkrétní směně. V Tabulce 5.12 jsou zobrazeny počty lékařů

směna \ den	PO	ÚT	ST	ČT	PÁ	SO	NE
1	6 + 3	3 + 0	2 + 1	4 + 1	7 + 0	10 + 5	8 + 4
2	7 + 0	7 + 2	10 + 5	8 + 0	8 + 0	10 + 5	10 + 5
3	7 + 0	11 + 0	7 + 3	9 + 0	12 + 0	9 + 3	9 + 4

Tabulka 5.12: Řešení druhé části Příkladu 5.3 – celkové počty

(první sčítanec) a studentů (druhý sčítanec), kteří jsou přítomni na dané směně. Při tomto sestavování pracovního rozvrhu využijeme 41 zaměstnanců. V průběhu celého týdne také 41x využijeme pomoc studentů. Vzhledem k tomu, že na práci studentů nemáme žádné speciální požadavky (studenti mohou pracovat každý den v týdnu, více směn za den, atp.) se v krajním případě může jednat o jakýkoli počet studentů větší než 5, neboť z tabulky vidíme, že nejvyšší počet studentů pracujících najednou je právě 5. Studenti vypomůžou 31x na směnách 1 a 2 a 10x na směně 3. Minimum nákladů, kterého dosáhneme touto strategií, je 2110. Celočíslnost výsledku není pravidlem, je opět dána konkrétním tvarem zadání.

△

V tomto příkladu jsme si ukázali, jak vytvořit rozvrh zaměstnancům na pohotovosti. Tento příklad, ve kterém jsme využili jak homogenní, tak nehomogenní pracovní sílu, byl opět jistým zjednodušením reality. Využili jsme v něm však několik zajímavých požadavků a omezení. Více o podobných problémech čtenář najde např. v [2] či [30]. Podobný rozvrhový problém bychom mohli motivovat i snahou sestavit např. co nejvýkonnější sportovní tým. Řešením problému bychom pak z konkrétních charakteristik jednotlivých hráčů určili skupiny, které spolu budou na hřišti, či pořadí, v jakém se hráči budou střídat. Další aplikací lineárního programování, již si zde uvedeme, je optimální dělení materiálu. Úlohy tohoto typu najdeme v literatuře obvykle pod názvem řezné problémy.

5.4 Řezný problém

Řezný problém je obvykle motivován tak, že máme k dispozici nějaký materiál určité velikosti, který chceme rozdělit do kusů jiné velikosti. Můžeme si to představit například tak, že výrobce vyrábí dřevěná prkna či ocelové trubky v několika délkách a následně je řeže podle požadavků zákazníků. V řezných problémech pak máme např. omezenou nabídku původního materiálu, či požadovaná množství materiálů různých velikostí, které chceme po řezání získat. Pro úlohu vždy musíme sestavit též tzv. *řezný plán*, který udává způsoby, jakými je možné materiál rozdělit v závislosti na požadovaných velikostech nařezaných kusů. Účelová funkce pak může vyjadřovat např. množství materiálu, který po řezání zbude, či náklady na jednotlivé kusy materiálu, které musíme před řezáním zakoupit. V obou těchto případech bychom hledali minimum účelové funkce.

Označme s_i , $i \in \{1, \dots, m\}$, nabídku i -té délky materiálu, kterou máme k dispozici. Položme počet různých délek řezů rovný p . Proměnná d_k , $k \in \{1, \dots, p\}$, označuje poptávku po k -té délce řezu. Při řezání máme k dispozici n řezných plánů. Proměnnou x_j označíme, kolikrát využijeme j -tý řezný plán. Koeficient a_{kj} označuje, kolik kusů požadované délky

řezu k získáme využitím j -tého řezného plánu. Proměnné z_j označují zbytky materiálu z jednotlivých řezných plánů. Označme T_i množinu všech řezných plánů j , které využívají materiál i . S využitím zavedeného značení můžeme řezný problém obecně naformulovat například následujícím způsobem

$$\begin{aligned} \min \quad & \sum_{j=1}^n z_j \\ \text{za podmínku} \quad & \sum_{j=1}^n a_{ij}x_j \geq d_k, \quad k \in \{1, \dots, p\}, \\ & \sum_{j \in T_i} x_j \leq s_i, \quad i \in \{1, \dots, m\}, \\ & x_j \geq 0, \quad j \in \{1, \dots, n\}. \end{aligned}$$

K problému můžeme, stejně jako u problému jídelníčku, dodat i omezení na maximální výrobu jednotlivých délek řezu, označeným b_k , $k \in \{1, \dots, p\}$. Takové omezení by bylo ve tvaru

$$\sum_{j=1}^n a_{ij}x_j \leq b_k, \quad k \in \{1, \dots, p\}.$$

Formulaci řezného problému najdeme ve většině literatury věnované lineárnímu programování, např. v [15], [23] či [28]. Tyto problémy jsou většinou motivovány potřebou nařezat trubky, prkna nebo papírové role. V následujícím příkladu si ukážeme aplikaci motivovanou potřebou nastříhat látku na několik různých délek.

Příklad 5.4. Stříhání látek.

V Brně a dalších českých městech se v reakci na nedostatek roušek při epidemii objevili jak dobrovolníci, kteří roušky šijí, tak dárci látek, kteří jsou schopni poskytnout potřebný materiál. Lidé, kteří mají k dispozici různé kusy bavlny, je mohou doručit přímo lidem, kteří roušky šijí. Vzniklo ale i několik skupin a míst, kam mohou dárci látek látku poslat, aniž by se starali o jejich doručení konkrétním lidem. Lidé, kteří tyto skupiny organizují, mají za úkol doručené látky nastříhat tak, aby se každému, kdo chce šít, dostalo požadované množství kusů, ze kterých může roušky vyrobit. Navíc existuje několik způsobů šití roušek, přičemž každý z nich vyžaduje jiné rozměry použité látky. Dobrovolníci, kterým budeme látku posílat, se přihlásili o následující 4 délky jednotlivých kusů: 20 cm, 21,5 cm, 16 cm a 30 cm. Šířka všech požadovaných kusů je 20 cm.

Předpokládejme, že organizátoři skupiny již látky nastříhali na 20 cm široké pásy. V závislosti na doručených látkách jsou pásy ve 4 různých délkách – 70 cm, 45 cm, 38 cm a 64 cm. Organizátoři mají k dispozici postupně 130, 25, 48 a 110 kusů těchto pásů. Tyto pásy nyní potřebují rozstříhat na délky, které požadují jednotliví dobrovolníci, kteří budou roušky šít. Organizátoři znají celkové poptávky po jednotlivých typech nastříhaných látek, které jsou postupně 300 ks, 250 ks, 175 ks a 125 ks.

Organizátoři akce nyní řeší, jak z látek, které mají k dispozici, nastříhat materiál na jednotlivé roušky. Jejich cílem je minimalizovat zbytky látek, které se již nebudou dít použít. Organizátoři samozřejmě chtějí pokrýt poptávku po látkách na roušky. Vzhledem k náročnosti operace mají však jeden dodatečný požadavek – z každého pruhu látky vystříhat maximálně 2 různě dlouhé typy kusů. Nemůže se tedy stát, že pás dlouhý

70 cm rozstříhnou na 3 kusy, jeden dlouhý 20 cm, druhý 21,5 cm a třetí 16 cm. Formulujme tuto úlohu jako problém lineárního programování a najděme řešení, které minimalizuje celkový zbytek látek.

Organizátory také zajímá, jaké maximální množství látek na roušky jednotlivých rozměrů jsou schopni z dostupných látek vyrobit tak, že dodrží poptávková omezení po jednotlivých rozměrech. Zodpovězme i na tuto otázku.

Řešení. Než přistoupíme k samotné matematické formulaci úlohy, musíme si ujasnit, jaké jsou možnosti nastříhání látek o délkách 70 cm, 45 cm, 38 cm a 64 cm na látky o délkách 20 cm, 21,5 cm, 16 cm a 30 cm. Tyto stříhy navíc musí splnit omezení, které organizátoři vyžadují – z jednoho kusu tak můžeme vyrobit maximálně 2 různé typy délek. Tyto možnosti, neboli řezné plány, si nyní postupně vypíšeme. Začneme s řeznými plány pro 70 cm dlouhý pruh látky. Nejprve si uvědomíme, jak můžeme tento pruh látky nastříhat na jednotlivé typy délek. Z tohoto pruhu můžeme získat 3 látky o délce 20 cm (se zbytkem 10 cm), 3 látky o délce 21,5 cm (zbytek 5,5 cm), 4 látky o délce 16 cm (zbytek 6 cm) nebo 2 látky o délce 30 cm (zbytek 10 cm). V dalším kroku chceme získat dostupné řezné plány, ve kterých látku o délce 70 cm rozdělíme na výrobu 2 různých délek látek na roušky. Jednou z těchto možností je například ustříhnout 3 kusy dlouhé 16 cm a jeden kus dlouhý 21,5 cm, z čehož získáme pouze 0,5 cm zbytku. Z důvodu přehlednosti si nyní uvedeme všechny řezné plány (označené čísly 1-12) pro látku dlouhou 70 cm v Tabulce 5.13.

řezné plány	1	2	3	4	5	6	7	8	9	10	11	12
20 cm	3					1	2	2	2	1		
21,5 cm		3			1		1			2	2	
16 cm			4		3	3			1		1	2
30 cm				2				1				1
zbytek	10	5,5	6	10	0,5	2	8,5	0	14	7	11	8

Tabulka 5.13: Řezné plány pro 70 cm dlouhý pruh látky

Prvních pět řezných plánů, které jsme si popsali v předchozím odstavci, najdeme v Tabulce 5.13 pod čísly 1-5. V tabulce vidíme, že možných řezných plánů pro pruh látky dlouhý 70 cm je 12. Tabulku čteme tak, že například řezný plánem číslo 10 vyrobíme z daného pruhu látky 1 kus dlouhý 20 cm a 2 kusy dlouhé 21,5 cm se zbytkem 7 cm.

Řezné plány si obdobně sestavíme i pro látky délek 45, 38 a 64 cm. Vždy si nejprve vypíšeme plány, ze kterých stříháme látky pouze jedné délky a poté plány na stříhání látek dvou různých délek. Tímto postupem získáme sedm řezných plánů pro látku o délce 45 cm, šest řezných plánů pro látku dlouhou 38 cm a jedenáct řezných plánů pro 64 cm dlouhé látky. Všechny dostupné řezné plány pro tuto úlohu jsou zobrazeny v Tabulce 5.14. Pod čísly 1-12 jsou plány pro látku o délce 70 cm, čísla 13-19 označují plány pro látku o délce 45 cm, čísla 20-25 plány pro 38 cm dlouhé látky a čísla 26-36 plány pro látky o délce 64 cm. Při sestavování této tabulky byly záměrně vynechány některé plány, které by nebyly využity, neboť ze zbytku látky by z nich bylo možné vytvořit další roušku. Takový plán je například nastříhání 70 cm dlouhého kusu na 3 kusy dlouhé 16 cm a vyhození zbytku, atd. Vyřazení těchto plánů úlohu mírně zjednodušuje, je však opodstatněné tím, že organizátoři

provádí tuto akci pouze jednou a nechtějí si nechat kusy látek na nějaké jiné použití. V případě, že by se jednalo o opakovaný problém, by mohlo být výhodné využít např. plány, ve kterých z 70 cm dlouhé látky ustříhneme 20 cm dlouhý kus a 50 cm si necháme, čímž vlastně získáme plán beze zbytku.

řezné plány	1	2	3	4	5	6	7	8	9	10	11	12
20 cm	3					1	2	2	2	1		
21,5 cm		3			1		1			2	2	
16 cm			4		3	3			1		1	2
30 cm				2				1				1
zbytek	10	5,5	6	10	0,5	2	8,5	0	14	7	11	8
Řezné plány	13	14	15	16	17	18	19	20	21	22	23	24
20 cm	2				1	1		1				1
21,5 cm		2			1		1		1			
16 cm			2			1	1			2		1
30 cm				1							1	
zbytek	5	2	13	15	3,5	9	7,5	18	16,5	6	8	2
Řezné plány	25	26	27	28	29	30	31	32	33	34	35	36
20 cm		3				2	2	1		1		
21,5 cm	1		2			1		2	2		1	
16 cm	1			4			1		1	2	2	2
30 cm					2							1
zbytek	0,5	4	21	0	4	2,5	8	1	5	12	10,5	2

Tabulka 5.14: Řezné plány pro Příklad 5.4

Nyní víme, jaké jsou všechny řezné plány, které můžeme využít. Můžeme tak již matematicky naformulovat problém, který organizátoři řeší. Označme x_j počet využití plánu j . Poptávková omezení, která chceme splnit, získáme pro jednotlivé kusy látek z řádků Tabulky 5.14. Tato omezení jsou postupně

$$3x_1 + x_6 + 2x_7 + 2x_8 + 2x_9 + x_{10} + 2x_{13} + x_{17} + x_{18} + x_{20} + x_{24} + 3x_{26} + 2x_{30} + 2x_{31} + x_{32} + x_{34} \geq 300, \quad (5.1)$$

$$3x_2 + x_5 + x_7 + 2x_{10} + 2x_{11} + 2x_{14} + x_{17} + x_{19} + x_{21} + x_{25} + 2x_{27} + x_{30} + 2x_{32} + 2x_{33} + x_{35} \geq 250, \quad (5.2)$$

$$4x_3 + 3x_5 + 3x_6 + x_9 + x_{11} + 2x_{12} + 2x_{15} + x_{18} + x_{19} + 2x_{22} + x_{24} + x_{25} + 4x_{28} + x_{31} + x_{33} + 2x_{34} + 2x_{35} + 2x_{36} \geq 175, \quad (5.3)$$

$$2x_4 + x_8 + x_{12} + x_{16} + x_{23} + 2x_{29} + x_{36} \geq 125. \quad (5.4)$$

Látky, ze kterých materiál na roušky stříháme, jsou k dispozici v omezeném množství. Omezení na dostupnost zdrojů můžeme formulovat s využitím proměnných x_j následujícím

způsobem.

$$\sum_{j=1}^{12} x_j \leq 130, \quad \sum_{j=13}^{19} x_j \leq 25, \quad \sum_{j=20}^{25} x_j \leq 48, \quad \sum_{j=26}^{36} x_j \leq 110$$

První z uvedených omezení říká, že plány 1-12, které se týkají látky dlouhé 70 cm můžeme využít maximálně 130x, neboť máme k dispozici pouze 130 kusů dané látky. Další omezení mají stejný význam pro plány 13-19, které se týkají látky dlouhé 45 cm, atd. Poslední omezení, které k formulaci přidáme, je omezení na nezápornost všech proměnných x_j . Zbytky z provedení jednotlivých plánů (v centimetrech látky) jsou uvedeny v Tabulce 5.14. Účelová funkce je součtem všech těchto zbytků pro provedené plány. Označíme-li zbytky, které získáme provedením jednotlivých řezných plánů, písmeny z_j , můžeme účelovou funkci zapsat ve tvaru

$$\sum_{j=1}^{36} z_j x_j.$$

Nyní již můžeme úlohu vyřešit simplexovou metodou. Tou získáme řešení, ve kterém využijeme pouze plány označené čísly 5, 8, 14, 25, 28 a 32, ostatní plány nevyužijeme (tj. ostatní x_i jsou rovny nule). Pro využití plánů získáme následující počty

$$x_5 = 5, \quad x_8 = 125, \quad x_{14} = 16,5, \quad x_{25} = 48, \quad x_{28} = 28, \quad x_{32} = 82.$$

Tato čísla nám říkají, kolikrát daný plán využijeme. První, čeho si můžeme všimnout, je, že výsledný bod obsahuje pouze 6 nenulových proměnných, a je tedy degenerovaný. Setkáme se tak s praktickým problémem, který vede na degenerovanou úlohu. Dále vidíme, že všechny hodnoty, až na proměnnou x_{14} , jsou celočíselné. Zkontrolujeme-li splnění omezení, uvidíme, že všechna poptávková omezení, kromě omezení (5.1), se realizují jako rovnosti. Totéž platí pro nabídková omezení, s výjimkou omezení číslo 2 pro látku o délce 45 cm. To znamená, že při této strategii nevyužijeme všechny dostupné látky o délce 45 cm, v nalezeném bodě jich využijeme pouze 16,5. Plán x_{14} tak můžeme udělat 17x, neboť máme k dispozici látku dlouhou 45 cm. Zaokrouhlením x_{14} na hodnotu 17 získáme v tomto případě řešení, které bychom obdrželi s využitím metod pro celočíselné programování. Celkové počty nastříhaných dílků jsou v případě provedení optimálního plánu následující: 332 ks látek dlouhých 20 cm, 250 ks látek dlouhých 21,5 cm, 175 látek o délce 16 cm a 125 kusů délky 30 cm. Z těchto hodnot vidíme, že v případě roušky dlouhé 20 cm vyrobíme při optimálním střihání více než požadovaný počet. Poptávky po ostatních délkách látek jsou splněny přesně. Provedením tohoto plánu tak získáme látky na 882 roušek. V optimálním případě nám ze střihů zbude pouze 141,5 cm zbytků látky a 8,5 nevyužitých látek o délce 45 cm. Pro zajímavost zde uvedeme též výsledek celočíselného problému, který se liší pouze tím, že bychom získali 251 látek dlouhých 21,5 cm, vyrobili látky na 883 roušek, vytvořili 142,5 cm zbytků látky a nevyužili jen 8 látek o délce 45 cm.

Ve druhé části příkladu bychom chtěli najít maximální množství látek na roušky, které můžeme získat z dostupných látek. V této části řešíme problém, který má stejná omezení, neboť organizátoři chtějí splnit všechny poptávky a množství dostupných látek jsou stále stejná. Jediné, co se změní, je účelová funkce. Označme a_{ij} koeficienty z Tabulky 5.14,

prázdné buňky odpovídají koeficientu nula. Účelovou funkci, jež chceme nyní maximalizovat, pak můžeme zapsat ve tvaru

$$\sum_{i=1}^4 \sum_{j=1}^{36} a_{ij} x_j.$$

Řešením této úlohy jsou následující počty látek jednotlivých délek: 300 ks, 250 ks, 237,2 ks a 125 ks. Těchto počtů dosáhneme využitím plánů označených čísly 5, 8, 14, 25, 28, 29 a 32, a to v následujícím počtu opakování

$$x_5 = 14,4, \quad x_8 = 115,6, \quad x_{14} = 25, \quad x_{25} = 48, \quad x_{28} = 36,5, \quad x_{29} = 4,7, \quad x_{32} = 68,8.$$

Čtenář si jistě všimne, že i v tomto případě se jedná o degenerovaný bod. Zbytek je v tomto případě necelých 170 cm látky. V tomto případě jsme schopni vyrobit látku na 912,2 roušek. Vzhledem k tomu, že plány nejde využít neceločíselně-krát, by se toto množství zavedením omezení na celočíselnost snížilo, a to na 912. I řešení získané pomocí simplexové metody nám ale dává dostatečně přesnou informaci o tom, jaké řezné plány bychom měli využít, a kolik roušek jsme schopni z dostupných látek vyrobit.

Organizátoři se z jasných důvodů nakonec rozhodnou nezvolit plán, který minimalizuje zbytek, ale plán, který umožní vyrobit co největší počet roušek. Na problém šití roušek a jejich rozvozu lidem, kteří je potřebují, se podíváme v následující kapitole v Příkladu 6.8.

△

V uvedeném řezném problému jsme viděli, že nejobtížnější část příkladu nemusí být jeho matematická formulace či řešení simplexovou metodou. Nejobtížnější částí mnoha řezných problémů je formulace jednotlivých řezných plánů, které můžeme využít. V uvedené úloze jsme měli pouze 4 délky látek, ze kterých jsme chtěli stříhat 4 konkrétní délky, a to navíc s omezením, že z jedné látky můžeme vystříhat kusy maximálně dvou různých délek. Délky dostupných látek navíc nebyly v poměru k rozměrům, kterých jsme chtěli docílit, nijak velké. I přesto jsme v úloze měli 36 dostupných řezných plánů. Ve většině praktických úloh je tak počet řezných plánů velmi velký. Řezné problémy mohou být i vícedimenzionální. V případě stříhání roušek bychom např. mohli požadovat jak různou délku, tak různou šířku látek. Formulace dvourozměrného řezného problému by z hlediska matematiky byla stále stejná. O mnoho složitější by ale bylo najít jednotlivé řezné plány, a to i v případě stříhu pravidelných tvarů. Příklad dvourozměrného řezného problému čtenář najde např. v [15]. Speciálním typem dvourozměrného řezného problému je tzv. *gilotinový problém*, ve kterém hledáme způsob, jak z obdélníku za pomoci rovnoběžných a kolmých řezů vyřezat obdélníky zadané velikosti s co nejmenším odpadem. Tento problém je obvykle motivován řezáním tabulek skla, čtenář se o něm dozví více např. v [16].

5.5 Investice – tvorba portfolia

Lineární programování můžeme využít též při plánování investic neboli tvorbě portfolia. Obecně bychom problém sestavení portfolia formulovali například následujícím způsobem. Představme si, že máme n investičních příležitostí, jejichž výnos R_j , $j \in \{1, \dots, n\}$, známe. Výnos R_j je obecně náhodná veličina, při tvorbě portfolia se budeme zajímat např. o její

střední hodnotu (ozn. $\mathbf{E}(R_j)$). Maximální hodnotu prostředků, které jsme ochotni investovat, označíme písmenem M . Označíme-li x_j množství peněz, které investujeme do j -té investice, můžeme střední hodnotu zisku, též označovanou jako očekávaný výnos, zapsat ve tvaru

$$\mathbf{E}(\pi) = \sum_{j=1}^n x_j \mathbf{E}(R_j).$$

V případě, že by nás zajímal pouze očekávaný výnos, bychom problém vyřešili snadno, a sice investovali bychom všechny dostupné prostředky do investice s největším výnosem. Velmi výnosné investice s sebou však obvykle nesou velkou míru rizika (označeného s_j), které můžeme definovat např. pomocí střední hodnoty absolutní odchylky od průměru jako

$$s_j = \mathbf{E}|x_j R_j - x_j \mathbf{E}(R_j)|.$$

Při vytváření portfolia musíme brát v úvahu oba cíle, kterými jsou maximalizace zisku a minimalizace rizika. Jedním ze způsobů, které uvádí např. [43], je využití parametru μ , jehož přidáním do účelové funkce nastavíme, jak důležitý je pro nás výnos oproti riziku. Účelová funkce může být např. ve tvaru

$$\mu \sum_{j=1}^n x_j \mathbf{E}(R_j) - \sum_{j=1}^n s_j.$$

Pro velké hodnoty μ je pro nás očekávaný výnos mnohem důležitější než riziko, pro malé hodnoty μ je pro nás na druhou stranu důležité podstoupit malé riziko. K úloze bychom samozřejmě přidali omezení na maximální hodnotu investovaných prostředků, tj.

$$\sum_{j=1}^n x_j \leq M.$$

V literatuře se též můžeme setkat s přístupem, ve kterém x_j označuje podíl celkové částky, kterou investujeme do j -té investice. V tomto případě je předchozí omezení ve tvaru

$$\sum_{j=1}^n x_j \leq 1, \quad \text{případně} \quad \sum_{j=1}^n x_j = 1.$$

Rovnostní omezení můžeme využít např. v případě, kdy je jedna z investičních možností bez rizika, což si můžeme představit např. jako uložení prostředků do banky. Tuto obecnou formulaci problému sestavování portfolia si nyní ukážeme na příkladu, ve kterém budeme pro jednoduchost předpokládat, že výnos není náhodná veličina. V tomto příkladu budeme risk měřit tzv. *mírou defaultu*, tj. pravděpodobností, že klient, kterému půjčíme peníze, nám je nebude schopen splatit. V příkladu, který si ukážeme, budeme s rizikem pracovat třemi různými způsoby.

Příklad 5.5. Bankovní rozhodování o spotřebitelských úvěrech.

V tomto příkladu se podíváme na problém, před kterým stojí vedení banky. Banka má momentálně k dispozici 50 milionů Kč, které chce buď investovat do státních dluhopisů nebo vypůjčit klientům, jejichž poptávka po spotřebitelských úvěrech je v současné době velmi vysoká. Investování do státních dluhopisů je v podstatě bezrizikové, úvěrování je

však pro banku mnohem výnosnější. Vedení banky chce maximalizovat zisk, ovšem za předpokladu, že nebudou muset podstoupit příliš velké riziko.

Banka svoje klienty řadí podle jejich charakteristik do osmi skupin označených písmeny A-H. V kategorii označené písmenem A jsou pouze nejlepší klienti, se kterými má banka již zkušenost, a ví, že s pravděpodobností 0,999 svůj úvěr splatí i s úroky. Pravděpodobnost, že klient nebude schopen svoji půjčku splatit, zde označíme mírou defaultu. Ze zkušenosti zná vedení též pravděpodobnost toho, že klient, který není schopen splácet, vrátí bance polovinu hodnoty svého úvěru (bez úroků). Banka zároveň ví, že klienti ve skupině A snadno získají úvěr i u jiné banky, neboť jsou téměř bezriziková. Musí jim tak nabídnout nízké úroky, aby si vybrali její produkt. Banka má již vypracovanou strategii, podle které nabídne jednotlivým kategoriím úrokové sazby. Stupnici s úroky, mírou defaultu a pravděpodobností navrácení poloviny úvěru v případě defaultu pro jednotlivé skupiny klientů zobrazuje Tabulka 5.15.

kategorie	A	B	C	D	E	F	G	H	2T repo
úrok (%)	2,9	3,9	4,9	6,9	7,5	10,9	12,9	16,9	1
míra defaultu	0,001	0,009	0,04	0,085	0,1	0,12	0,18	0,2	0
pravd. splacení	0,999	0,991	0,96	0,915	0,9	0,88	0,82	0,8	1
pravd. polovina	0,75	0,75	0,5	0,5	0,5	0,3	0,3	0,25	0

Tabulka 5.15: Rozdělení klientů podle rizikivosti

V posledním řádku Tabulky 5.15 vidíme, jaká je pravděpodobnost, že klienti, kteří nebudou schopni splácet, vrátí bance polovinu vypůjčené částky bez úroků. Banka tak z těchto klientů ztratí pouze polovinu hodnoty poskytnutého úvěru a úroky. Kromě osmi kategorií klientů je v tabulce též sloupec označený 2T repo, což je úroková sazba, za kterou si mohou banky na 2 týdny uložit peníze u ČNB. Tento způsob investování je bezrizikový, pravděpodobnost splacení je tak rovna jedné. Stejně jako úrokové sazby pro klienty je i 2T repo sazba roční, v Tabulce 5.15 je uvedena sazba platná od 27. 3. 2020. Tato sazba se v průběhu roku může změnit, v příkladu však považujeme sazbu za neměnnou po celý rok. Banka by tedy své přebytečné prostředky uložila u ČNB na celý rok.

Vedení banky chce maximalizovat svůj zisk. Vzhledem ke striktním pravidlům bankovního systému však nemůže vypůjčit své peníze úplně libovolně. Banka musí kvůli rizikivosti alespoň třetinu hodnoty svých úvěrů vypůjčit klientům z kategorie A a B. Ze stejného důvodu může banka do kategorií F, G a H vypůjčit maximálně polovinu toho, co zapůjčí do kategorií A, B a C. Banka si též váží svých klientů z kategorie A, se kterými má předběžně domluvené úvěry o hodnotě 800 000 Kč. Klientům v kategorii A chce tak vedení nabídnou úvěry minimálně v této hodnotě.

Vzhledem k tomu, že se vedení rozhoduje o hodnotách úvěrů pro jednotlivé kategorie, počítá se ztrátou jako s částkou, kterou jednotlivé kategorie nesplatí, tj. se součinem hodnoty úvěrů v dané kategorii a míry defaultu, od kterého odečte hodnotu částky, již splatí část nesplácejících klientů. Vedení banky chce uvážit tři situace, které se liší pojetím rizika. Ve všech situacích se však zajímá o úrokový výnos z prvního roku splácení. V první z nich chce maximalizovat výnos (z úroků a návratů částí úvěrů od klientů v defaultu) s tím, že omezí maximální hodnotu, kterou jsou ochotni ztratit, na částku 2 500 000 Kč. Ve druhé situaci se pak chce podívat na zisk jako rozdíl výnosů a ztráty. Ve třetí situaci, kterou chce

vedení otestovat, je hodnota výnosu dvakrát důležitější než riziko. Jaké pojetí rizika přinese bance větší zisk? Jak se situace změní, jestliže se vedení rozhodne kvůli solidaritě a dobré reputaci vypůjčit alespoň 5 000 000 Kč lidem z kategorií F, G a H? Sestavme a vyřešme úlohu lineárního programování, před kterou stojí vedení banky.

Řešení. Označme x_1, \dots, x_8 prostředky investované do kategorií A-H. Celkovou sumu, kterou banka investuje do bezrizikových dluhopisů za 2T repo sazbu poté vypočítáme jako zbývající částku, kterou neinvestuje do úvěrů. Striktní pravidla bankovního systému, kterým banka čelí, můžeme zapsat jako

$$x_1 + x_2 \geq 1/3 \sum_{i=1}^8 x_i,$$

$$x_6 + x_7 + x_8 \leq (x_1 + x_2 + x_3)/2.$$

Banka dále čelí omezení na celkovou výši svých úvěrů, kterým je

$$\sum_{i=1}^8 x_i \leq 50 \cdot 10^6.$$

Poslední omezení, jež chce banka splnit, je omezení o úvěrech pro klienty z kategorie A, které musí být alespoň ve výši 800 000, tj. musí platit

$$x_1 \geq 800\,000.$$

Pro všechna x_i platí též omezení nezápornosti. Všechna uvedená omezení jsou společná pro situace, které banka uvažuje.

Jednotlivé situace se však liší ve formulaci účelové funkce či dalších omezeních. V první situaci se banka zajímá pouze o výnos z úroků a navrácenou částku od některých nesplácejících klientů. Označme r_i úrokové míry jednotlivých kategorií, p_i pravděpodobnosti splacení úvěru a h_i pravděpodobnost, že klient nezvládne splácet, ale vrátí bance polovinu zapůjčené částky. Úrokový výnos za první rok splacení, který chce banka maximalizovat, pak v této situaci můžeme zapsat ve tvaru

$$\sum_{i=1}^8 p_i \cdot x_i \cdot r_i / 100 + \sum_{i=1}^8 (1 - p_i) \cdot h_i \cdot x_i / 2,$$

kde první suma představuje úrokový výnos a druhá součet navrácené částky od klientů, kteří svůj úvěr nejsou schopni splácet. V této situaci je banka ochotná podstoupit maximálně riziko ztráty 2,5 milionu. Z tohoto důvodu k úloze přidáme omezení

$$\sum_{i=1}^8 (1 - p_i) \cdot x_i - \sum_{i=1}^8 (1 - p_i) \cdot h_i \cdot x_i / 2 \leq 2,5 \cdot 10^6.$$

První suma v tomto omezení reprezentuje částku, kterou ztratíme kvůli nesplácejícím klientům, druhá pak částku, kterou nám někteří z nich vrátí. Nyní již máme všechna potřebná omezení a účelovou funkci. Vypočteme-li tuto úlohu lineárního programování, získáme výsledek, který je po zaokrouhlení na dvě desetinná místa uveden v Tabulce 5.16. V řádku označeném vrácení jsou uvedeny hodnoty, které bance vrátí část defaultních klientů.

kategorie	A	B	C	D	E	F	G	H	2T repo
x_i	800000	22967521,37	9565811,97	0	0	0	0	16666666,67	0
úroky	23176,8	887671,73	449975,79	0	0	0	0	2253333,33	0
vrácení	300	77515,38	95658,12	0	0	0	0	416666,67	0
ztráta	500	129192,31	286974,36	0	0	0	0	2083333,33	0

Tabulka 5.16: Výsledky pro první situaci v Příkladu 5.5

Banka se v tomto případě soustředí na klienty z kategorií A, B, C a H. Očekávaný výnos, který banka tímto způsobem získá, dostaneme sečtením řádků pro úroky a vrácení z Tabulky 5.16. Tento výnos je po zaokrouhlení 4 204 298 Kč. Z tabulky zároveň vidíme, že v tomto případě banka čelí velké ztrátě, která je přesně 2 500 000 Kč, což je maximální povolená hodnota. Tento výsledek by nás neměl překvapit, neboť čím větší ztrátu jsme ochotni podstoupit, tím více můžeme vydělat na úrocích. Výše ztráty je však velmi důležitá při určování zisku, o který se ve výsledku vždy zajímáme. Zisk banky dostaneme odečtením ztráty od výnosu. V této situaci tak banka získá přibližně 1 704 298 Kč.

Nyní se podíváme na situaci, ve které chce vedení banky maximalizovat rovnou ziskovou funkci, jež je tvořená příjmy z úroků, příjmy z navracení poloviny částky některými nesplácejícími a ztrátou z nesplácení. Účelovou funkci pak můžeme s využitím zadaného značení zapsat ve tvaru

$$\sum_{i=1}^8 p_i \cdot x_i \cdot r_i / 100 + \sum_{i=1}^8 (1 - p_i) \cdot h_i \cdot x_i / 2 - \sum_{i=1}^8 (1 - p_i) \cdot x_i.$$

K této úloze nepřidáváme žádné další podmínky. Její výsledek najde čtenář v Tabulce 5.17.

kategorie	A	B	C	D	E	F	G	H	2T repo
x_i	800000	49200000	0	0	0	0	0	0	0
úroky	23176,8	1901530,8	0	0	0	0	0	0	0
vrácení	300	166050	0	0	0	0	0	0	0
ztráta	500	276750	0	0	0	0	0	0	0

Tabulka 5.17: Výsledky pro druhou situaci v Příkladu 5.5

Z Tabulky 5.17 vidíme, že nejvýhodnější strategií je v tomto případě vypůjčit všechny prostředky, až na 800 000 vyhrazených pro kategorii A, do kategorie B, což je dáno tím, že klienti poměrně spolehlivě splácí, zároveň jim však můžeme dát znatelně vyšší úrok než klientům z kategorie A. Výnos banky je v tomto případě roven 2 091 057,6 Kč. Při ztrátě 277 250 Kč tak banka celkově získá 1 813 807,6 Kč, což je o téměř 110 000 Kč více, než v předchozí situaci. Vedení banky tak prozatím upřednostňuje strategii úvěrování pouze nejlepších klientů. Samozřejmě je však zajímavá i výsledek třetí situace.

Ve třetím případě chce vedení najít strategii, při které maximalizuje stejnou účelovou funkci jako ve 2. případě s jedinou výjimkou, kterou je vynásobení výnosů koeficientem $\mu = 2$, podobně jako jsme viděli v obecném popisu úlohy. Maximalizujeme tedy funkci

$$2 \left(\sum_{i=1}^8 p_i \cdot x_i \cdot r_i / 100 + \sum_{i=1}^8 (1 - p_i) \cdot h_i \cdot x_i / 2 \right) - \sum_{i=1}^8 (1 - p_i) \cdot x_i$$

na množině zadané omezeními společnými pro všechny případy. V této situaci získáme výsledek zobrazený v Tabulce 5.18.

kategorie	A	B	C	D	E	F	G	H	2T repo
x_i	800000	15866666,67	16666666,67	0	0	0	0	16666666,67	0
úroky	23176,8	613230,8	784000	0	0	0	0	2253333,33	0
vrácení	300	53550	166666,67	0	0	0	0	416666,67	0
ztráta	500	89250	500000	0	0	0	0	2083333,33	0

Tabulka 5.18: Výsledky pro třetí situaci v Příkladu 5.5

Z Tabulky 5.18 vidíme, že se banka, stejně jako v první situaci, zaměří pouze na klienty z kategorií A, B, C a H. Jediný rozdíl je v rozdělení úvěrů mezi kategorie B a C, v tomto případě se, oproti první situaci, banka ve větší míře soustředí na klienty z kategorie C. Nalezené maximum účelové funkce je nyní rovno 5 948 765,2 Kč. Opět se ale zaměříme spíše na zisk, který je 1 637 840,93 Kč, což je nejméně ze všech uvažovaných situací. Vedení se tak rozhodne využít strategii 2, ve které maximalizuje přímo ziskovou funkci.

Nyní se podíváme na to, jak se změni optimální rozhodnutí banky v jednotlivých situacích, jestliže k úloze přidáme omezení

$$x_6 + x_7 + x_8 \geq 5 \cdot 10^6.$$

Z Tabulek 5.16 a 5.18 vidíme, že v první a třetí situaci je optimální vypůjčit klientům z kategorie H více než 16 milionů. Nalezená řešení tak jistě splní i nově přidané omezení, které je nijak neovlivní. Ve druhé situaci je tomu ale jinak. Z Tabulky 5.17 vidíme, že banka půjčuje pouze klientům z kategorií A a B. Přidáním dalšího omezení tak jistě změním výsledek, který je pro novou formulaci úlohy zobrazen v Tabulce 5.19.

kategorie	A	B	C	D	E	F	G	H	2T repo
x_i	800000	44200000	0	0	0	0	0	5000000	0
úroky	23176,8	1708285,8	0	0	0	0	0	676000	0
vrácení	300	149175	0	0	0	0	0	125000	0
ztráta	500	248625	0	0	0	0	0	625000	0

Tabulka 5.19: Výsledky pro změnu ve druhé situaci v Příkladu 5.5

Zisk banky je v tomto případě 1 807 812,6 Kč, i s přidáním dalšího omezení je tedy druhý způsob pojetí rizika tím nejvýhodnějším. Vedení banky se tak rozhodne využít při svém rozhodování model, ve kterém maximalizuje přímo ziskovou funkci, a to nezávisle na tom, zda využije poslední uvažované omezení či ne. Čtenář si mohl všimnout, že ve všech situacích, které jsme viděli, vypůjčila banka klientům z kategorie A pouze přísliběných 800 000 Kč. I přes vysokou pravděpodobnost splácení těchto klientů se bance nevyplatí jim půjčit více, a to z důvodu velmi nízkých úrokových příjmů.

△

Příklad na tvorbu portfolia, který jsme právě viděli, obsahoval několik výrazných zjednodušení reality. Banky se totiž běžně rozhodují o každém klientovi zvlášť tím způsobem, že mu podle jeho charakteristik dají hodnocení, podle kterého určí nabízenou úrokovou sazbu, na kterou klient buď přistoupí, nebo ne. V každém případě je však bankovní sektor poměrně regulovaný a banky musí splňovat podmínky pro hodnoty jednotlivých rizik svých investic. Příklady takových omezení jsme z tohoto důvodu v úloze využili. Úloha byla sice ilustrativní, v případě obecného rozhodování o investicích bychom však úlohu o tvorbě portfolia mohli formulovat obdobně.

Problém tvorby portfolia je často formulován též jako úloha kvadratického programování, jež nám dovolí využít nelineární účelovou funkci. Riziko tak můžeme definovat např. jako rozptyl. I přesto však můžeme lineární programování ve finančních aplikacích využít, a to nejen při tvorbě portfolia, ale např. i při oceňování opcí. Další příklady využití lineárního programování ve financích najde čtenář např. v [15] a [43]. V poslední části kapitoly si ukážeme další příklad motivovaný minimalizací nákladů, kterým je výrobní problém.

5.6 Plánování zásob a výroby

V Příkladu 1.1 jsme viděli typický výrobní problém, ve kterém je cílem výrobce najít optimální výrobní plán tak, aby maximalizoval zisk z prodeje svých výrobků. Výrobní problém je posledním typem úloh lineárního programování, který si v této kapitole ukážeme. V této části textu se na výrobní problém podíváme z jiného úhlu pohledu. V příkladu, který si uvedeme, chceme optimálně naplánovat výrobu tak, abychom byli vždy schopni uspokojit poptávky našich zákazníků. Zároveň jsme limitováni výrobními kapacitami. V důsledku toho musíme vždy výrobu plánovat dopředu. Tento problém bychom mohli též zařadit do kategorie nazývané plánování zásob. Obecný problém plánování zásob můžeme motivovat např. následujícím způsobem.

Výrobce, který čelí předem známé poptávce po svých produktech, chce naplánovat roční výrobu tak, aby pokryl poptávku s co nejmenšími náklady. Poptávka se ovšem v jednotlivých měsících mění, stejně jako dostupné výrobní kapacity. Do nákladů můžeme počítat jak náklady na samotnou výrobu, tak náklady na skladování výrobků, které jsou součástí zásob. Dalším nákladem může být též náklad na změnu objemu výroby, s tímto přístupem se setkáme např. v [28]. Náklady na změnu výroby jsou podstatné, neboť výroba rozdílného množství každý měsíc vyžaduje změnu počtu pracovníků, které docílíme např. přijetím sezónních pracovníků, a celkovou reorganizací výroby.

Matematicky můžeme problém, ve kterém výrobce hledá optimální roční plán výroby a zásob za situace, ve které nemá na začátku žádné zásoby, zformulovat například následujícím způsobem. Označme d_i poptávku v i -tém měsíci, x_i vyrobené množství v i -tém měsíci a z_i zásoby, které máme na skladě na konci i -tého měsíce, přičemž $i \in \{1, \dots, 12\}$. Dále označme p_i cenu skladování zásob v i -tém měsíci, s_i maximální výrobní kapacity v i -tém měsíci a c cenu změny výroby oproti minulému měsíci o jednotku. Problém nalezení

optimálního plánu zásob a výroby pak můžeme zapsat např. ve tvaru

$$\begin{aligned} \min \quad & c \sum_{i=1}^{12} |x_i - x_{i-1}| + \sum_{i=1}^{12} p_i z_i \\ \text{za podmíněk} \quad & x_i \leq s_i, \quad i \in \{1, \dots, 12\}, \\ & x_i + z_{i-1} - z_i = d_i, \quad i \in \{1, \dots, 12\}, \\ & x_0 = z_0 = 0, \\ & x_i, s_i, d_i, z_i \geq 0, \quad i \in \{1, \dots, 12\}. \end{aligned}$$

První uvedené omezení nám říká, že můžeme vyrobit maximálně tolik, kolik jsou naše výrobní kapacity. Druhé dává do souvislosti množství zásob a výroby, kterými musíme pokrýt poptávku. Dané omezení je rovnostní, neboť vše, co vyrobíme, případně máme v zásobách, nad poptávku, uchováme v zásobách do dalšího období. K úloze jsou doplněny též podmínky na nulovost počátečních zásob a výroby v nultém měsíci a podmínky nezápornosti všech proměnných. Čtenář si jistě všimne, že účelová funkce v tomto zápisu obsahuje absolutní hodnotu. Tuto funkci však můžeme, jak jsme již viděli v Příkladu 1.3, snadno převést na funkci lineární. V následujícím příkladu se podíváme na problém výrobce, kterému nevádí přenastavování výroby. Naopak ale čelí různým výrobním nákladům v jednotlivých obdobích.

Příklad 5.6. Půlroční plán výrobce pálenek.

Představme si výrobce, který produkuje čtyři druhy ovocných pálenek. Jeho cílem je naplánovat výrobu na následujících šest měsíců tak, aby minimalizoval náklady na výrobu a skladování zásob. Na začátku roku nemá výrobce žádné zásoby. Ze zkušenosti s podnikáním zná výrobce poptávky, kterým v jednotlivých měsících čelí, i výrobní kapacity, jež může v daném měsíci využít. Náklady na výrobu jednotlivých pálenek se v průběhu půlroku mění, neboť se mění cena ovoce potřebného pro jejich výrobu. Maximální produkční možnosti a poptávky po jednotlivých druzích pálenek (označených P_1 , P_2 , P_3 a P_4) v hektolitrech jsou uvedeny v Tabulce 5.20, ve které sloupce představují jednotlivé měsíce. Produkční možnosti se podobně jako ceny mění, a to z důvodu lišících se dostupných množství ovoce. V poslední části tabulky vidíme náklady na výrobu jednotlivých pálenek a cenu za skladování (ozn. S), jež je pro všechny pálenky stejná. Všechny náklady jsou uvedeny ve sto tisících Kč.

Výrobce chce naplánovat výrobu tak, aby vždy uspokojil poptávky a minimalizoval při tom náklady. Náklady na reorganizaci výroby však neuvažuje. Pomozme nyní výrobcovi najít plán optimální výroby a tvorby zásob, který splní zadané požadavky.

Ve druhé části příkladu se podíváme na situaci, do které se výrobce dostal po dvou měsících výroby podle optimálního plánu z první části příkladu. Vzhledem k pandemii, která na začátku roku nečekaně zasáhla celý svět, se výrazně změnila očekávaná poptávka po všech druzích pálenky. Náklady na výrobu a skladování však zůstaly stejné. Výrobce chce nyní optimální plán pro zbývající čtyři měsíce upravit tak, aby opět pokryl očekávané poptávky po svých výrobcích. Jeho maximální výrobní kapacity jsou ovšem nyní určeny optimálním výrobním plánem z první části úlohy, neboť podle něho výrobce nakoupil zásoby a přijal zaměstnance. Zároveň chce zachovat alespoň 90 % celkového naplánovaného měsíčního využití výrobních kapacit, neboť nechce propouštět zaměstnance. Zachování

max. výroba	1	2	3	4	5	6
P_1	100	100	110	120	140	60
P_2	120	100	80	80	60	120
P_3	80	90	100	110	100	90
P_4	40	50	90	90	120	110
poptávka	1	2	3	4	5	6
P_1	80	110	50	130	40	80
P_2	60	40	180	90	50	130
P_3	40	100	80	60	170	60
P_4	40	40	40	20	150	90
náklady	1	2	3	4	5	6
P_1	1	1,2	0,9	0,8	0,9	1
P_2	1,10	1	0,8	0,95	1,1	1,2
P_3	1	0,9	0,8	0,9	1	1,1
P_4	0,8	0,8	0,75	0,7	0,85	1
S	0,5	0,5	0,7	0,9	0,8	0,6

Tabulka 5.20: Zadání první části Příkladu 5.6

výroby chce dosáhnout výrobou dezinfekce (označené D), které je v současné době nedostatek, a k jejíž výrobě může využít líh a další již zakoupené suroviny. Náklady na výrobu dezinfekce jsou spolu s novými poptávkami uvedeny v Tabulce 5.21. Dezinfekci může výrobce vyrábět s využitím výrobních kapacit pro všechny druhy pálenek. Dezinfekci výrobce nikdy neskládá, neboť ji chce co nejdříve rozvést na potřebná místa. Zásoby, které máme k dispozici na začátku třetího měsíce, jsou zásoby ze druhého měsíce při provozování původního optimálního výrobního plánu. Najděme nyní nový optimální výrobní plán tak, aby byla splněna všechna omezení. Kolik dezinfekce je výrobce schopen v jednotlivých měsících vyrobit? Jak by se situace změnila, kdyby chtěl výrobce zachovat 90 % celkového (ne měsíčního) využití kapacit podle původního plánu?

poptávka	3	4	5	6
P_1	40	100	180	40
P_2	120	45	40	95
P_3	40	130	120	30
P_4	110	10	40	45
náklady	3	4	5	6
D	0,6	0,7	0,75	0,8

Tabulka 5.21: Zadání druhé části Příkladu 5.6

Řešení. Označme x_{ij} vyráběné množství i -té pálenky v j -tém měsíci, kde $i \in \{1, 2, 3, 4\}$, $j \in \{1, \dots, 6\}$, p_{ij} náklady na výrobu i -té pálenky v j -tém měsíci a S_j příslušné náklady na skladování. Dále označme d_{ij} poptávku po i -té pálence v j -tém měsíci a s_{ij} maximální objem výroby i -té pálenky v j -tém měsíci. Zásoby i -té pálenky na konci j -tého měsíce označíme z_{ij} . V celém příkladu uvažujeme množství pálenek v hektolitrech a náklady ve sto tisících Kč. S využitím zavedeného značení můžeme účelovou (nákladovou) funkci, kterou chce výrobce minimalizovat, zapsat např. ve tvaru

$$\sum_{j=1}^6 \sum_{i=1}^4 p_{ij} x_{ij} + \sum_{j=1}^6 S_j \sum_{i=1}^4 z_{ij}.$$

Omezení, kterým výrobce čelí při vytváření svého plánu, můžeme rozdělit do dvou kategorií, a to výrobní a poptávková omezení. Výrobní kapacity pro jednotlivé pálenky jsou omezeny, což vyjádříme skupinou omezení ve tvaru

$$x_{ij} \leq s_{ij}, \quad i \in \{1, 2, 3, 4\}, \quad j \in \{1, \dots, 6\}.$$

Ze zadání víme, že výrobce nemá na začátku žádné zásoby, proměnné z_{i0} tedy položíme rovny nule. Pokrytí poptávky v následujících šesti měsících výrobce zajistí jak výrobou, tak zásobami z předchozích měsíců, příslušná omezení jsou ve tvaru

$$x_{ij} + z_{ij-1} \geq d_{ij}, \quad i \in \{1, 2, 3, 4\}, \quad j \in \{1, \dots, 6\}, \quad (5.5)$$

přičemž v prvním měsíci výrobce poptávku naplní pouze výrobou. V případě využití tohoto omezení bychom k problému museli přidat omezení pro vztah mezi zásobami a výrobou, tj. rovnost

$$z_{ij} = x_{ij} + z_{ij-1} - d_{ij},$$

neboť do zásob jde vše, co v daném období neprodáme, z_{ij} tak vyrovná obě strany omezení (5.5). Poptávková omezení, která k úloze přidáme, jsou tak ve tvaru

$$x_{ij} + z_{ij-1} - z_{ij} = d_{ij}, \quad i \in \{1, 2, 3, 4\}, \quad j \in \{2, \dots, 6\}.$$

Nesmíme zapomenout ani na omezení na znaménko všech proměnných, tj. na

$$x_{ij}, z_{ij} \geq 0.$$

Nyní již můžeme příklad vyřešit simplexovou metodou. Řešení, které dostaneme, je zobrazeno v Tabulce 5.22.

V Tabulce 5.22 jsou kromě optimálních množství výroby a zásob uvedeny též celkové počty vyrobených a uskladněných hektolitrů pálenek pro každý měsíc a náklady na výrobu a skladování v daném měsíci. Čtenář snadno ověří, že celkové náklady na výrobu jsou 1 842 a náklady na skladování 232. Náklady na provedení optimálního výrobního plánu jsou 2 074. Z tabulky vidíme, že po šesti měsících výroby nezbudou výrobcům žádné zásoby i přesto, že jsme tento výsledek explicitně nepožadovali. Vzhledem k tomu, že nám v příkladu další zásoby nepřinesou žádný budoucí zisk, je to ovšem zcela přirozený výsledek. Celočíslnost výsledku je dána konkrétním tvarem zadání. Výrobce je s nalezeným plánem spokojen a začne se jím řídit. Mimo jiné oznámí svůj plán zaměstnancům, kteří ve

výroba	1	2	3	4	5	6
P_1	90	100	60	120	60	60
P_2	110	100	80	80	60	120
P_3	50	90	100	110	100	60
P_4	40	40	40	50	120	90
celkem	290	330	280	360	340	330
náklady	293	333	228	306	322	360
zásoby	1	2	3	4	5	6
P_1	10	0	10	0	20	0
P_2	50	110	10	0	10	0
P_3	10	0	20	70	0	0
P_4	0	0	0	30	0	0
celkem	70	110	40	100	30	0
náklady	35	55	28	90	24	0

Tabulka 5.22: Řešení první části Příkladu 5.6

výrobě pracují. V průběhu druhého měsíce začne být výroba pálenek ovlivňována světovou pandemií. Vyřešme nyní druhou část příkladu se změněnými poptávkami.

Označme d_j vyrobené množství dezinfekce a c_j náklady na její výrobu v j -tém měsíci, přičemž v této části příkladu uvažujeme $j \in \{3, 4, 5, 6\}$. Účelová funkce, která nyní obsahuje i náklady na výrobu dezinfekce, je ve tvaru

$$\sum_{j=3}^6 \sum_{i=1}^4 p_{ij} x_{ij} + \sum_{j=3}^6 c_j d_j + \sum_{j=3}^6 S_j \sum_{i=1}^4 z_{ij}.$$

Konkrétní poptávková omezení, kterým výrobce čelí, se změní, neboť se změnila poptávaná množství. Jejich tvar je ovšem stejný, tj.

$$x_{ij} + z_{ij-1} - z_{ij} = d_{ij}, \quad i \in \{1, 2, 3, 4\}, \quad j \in \{3, \dots, 6\}.$$

Proměnné z_{i2} nabývají hodnot ze druhé části Tabulky 5.22, tj. $z_{12} = 0$, $z_{22} = 110$, $z_{32} = 0$ a $z_{42} = 0$, neboť výrobce z počátku dodržuje nalezený plán z první části úlohy. Výrobce je v této situaci ovšem rozdílně limitován svou kapacitou výroby, neboť má sjednanou práci a suroviny pouze na výrobu množství z Tabulky 5.22. Koefficienty s_{ij} , které najdeme v první části dané tabulky, se tak v této části příkladu liší od původních, nabídková omezení jsou ve tvaru

$$x_{ij} + d_j \leq s_{ij}, \quad i \in \{1, 2, 3, 4\}, \quad j \in \{3, 4, 5, 6\}.$$

Vzhledem k tomu, že výrobce navíc požaduje zachování výroby alespoň na 90 % celkové měsíční plánované výroby, musíme k úloze přidat čtyři omezení ve tvaru

$$\sum_{i=1}^4 x_{ij} + d_j \geq 0,9 \sum_{i=1}^4 s_{ij}, \quad j \in \{3, 4, 5, 6\}.$$

K úloze opět přidáme podmínky nezápornosti proměnných $x_{ij}, z_{ij}, d_j \geq 0$. Vyřešíme-li zformulovanou úlohu simplexovou metodou, získáme řešení, jež je zobrazeno v Tabulce 5.23.

výroba	3	4	5	6	zásoby	3	4	5	6
P_1	40	120	60	40	P_1	0	20	0	0
P_2	10	60	60	95	P_2	0	20	0	0
P_3	80	110	100	30	P_3	40	20	0	0
P_4	40	10	95	90	P_4	0	0	55	0
d	82	24	0	42	d	0	0	0	0
celkem	252	324	306	297	celkem	40	60	55	0
náklady	187,2	275,8	300,75	310,6	náklady	28	54	44	0

Tabulka 5.23: Řešení druhé části Příkladu 5.6

Z Tabulky 5.23 vidíme nový optimální výrobní plán na následující čtyři měsíce. Výrobce v něm vyrábí jak čtyři druhy pálenek, tak dezinfekci, a to v množství, které mu pomůže zachovat aspoň 90 % výroby. Čtenář se snadno přesvědčí, že výrobce vyrábí množství dezinfekce, které mu umožní splnit podmínky na zachování 90 % výroby jako rovnosti. Například pro třetí měsíc byla celková původní výroba 280, nyní je 252, což přesně odpovídá devadesáti procentům. Jediný měsíc, ve kterém tomu tak není, je pátý. V tom výrobce nevyrobí žádnou dezinfekci, neboť výrobní kapacity zaměstná na více než 90 % i výrobou pálenek.

Celkové náklady na skladování zásob jsou nyní 126, náklady na výrobu (včetně dezinfekce) jsou 1 074,35. Celkové náklady na čtyřměsíční výrobu jsou 1 200,35, což je méně, než v původním případě, kdy byly náklady na měsíce 3-6 celkem 1 358. Pokles nákladů je způsoben celkovým menším množstvím výroby. V původním případě však výrobce za tyto čtyři měsíce vyrobil dohromady 1 310 jednotek pálenek, v současné situaci vyrobí kvůli snížené poptávce pouze 1 040 (což je přibližně 80 % původní výroby). Výrobce na produkci dezinfekce, na rozdíl od produkce a prodeje pálenek, nijak nevydělá, neboť jím vyrobených 148 hektolitrů poskytne darem státu. Dodatečná výroba mu však kromě dobrého pocitu pomůže se zachováním zaměstnanosti.

V případě, že by chtěl výrobce zaměstnat alespoň 90 % výrobních kapacit a nezáleželo by mu na rozdělení v měsících, jistě nedostaneme výsledek s vyššími náklady, než v případě, který jsme právě vyřešili. Mohl by totiž výrobu naplánovat tak, aby vyráběl více v měsících s nižšími náklady. Úloha by nyní místo čtyř výrobních omezení obsahovala jediné ve tvaru

$$\sum_{i=1}^4 \sum_{j=3}^6 (x_{ij} + d_j) \geq 0,9 \sum_{i=1}^4 \sum_{j=3}^6 s_{ij}.$$

△

V této kapitole jsme viděli šest aplikací lineárního programování, s dalšími zajímavými využitími se čtenář může setkat například v teorii her, viz [43] a [42], či ve fyzice, viz např. [22]. Kromě problémů motivovaných ekonomickými potřebami můžeme najít i jinak

motivované problémy, jako je např. problém vybarvení mapy co nejmenším počtem barev (viz [5]) a další. K dalšímu studiu aplikací lineárního programování lze kromě již uvedené literatury doporučit např. [1], [23], [9], [15] a [26]. V následující kapitole se podíváme na další typ úlohy lineárního programování, a to na dopravní problém.

Kapitola 6

Dopravní problém

V této kapitole se podíváme na jeden ze speciálních případů úlohy lineárního programování, kterým je tzv. *dopravní problém*. S dopravním problémem se setkáme téměř ve všech publikacích věnovaných lineárnímu programování, např. v [15], [26], [32] či [40]. Dříve než si vysvětlíme, čím je tento problém speciální a jak ho obecně formulujeme, se podíváme na motivační příklad.

Příklad 6.1. Motivační příklad dopravního problému.

Představme si, že máme dvě pole, ze kterých dovážíme obilí třem odběratelům, kteří ho dále prodávají. Kvalitu obilí z jednotlivých polí nejsme schopni rozlišit. Na jednom poli vypěstujeme 25 tun a na druhém 15 tun obilí. Odběratelé jsou s námi dohodnutí na dodávce 9, 18 a 13 tun. Náklady na dopravu jedné tuny obilí z obou polí ke všem odběratelům jsou zobrazeny v Tabulce 6.1.

pole \ odběratel	odběratel			nabídka
	O_1	O_2	O_3	
P_1	5	4	9	25
P_2	8	10	6	15
poptávka	9	18	13	40

Tabulka 6.1: Zadání Příkladu 6.1

V tabulce vidíme též nabídku obilí ze dvou polí a poptávku jednotlivých odběratelů. Naším cílem je najít plán přepravy, který minimalizuje celkové dopravní náklady. Formulujme tuto úlohu jako úlohu lineárního programování a vyřešme ji simplexovou metodou.

Řešení. Označme x_{ij} množství obilí, které dopravíme z i -tého pole j -tému odběrateli. Dále označme a_i , $i \in \{1, 2\}$ nabídky obilí z našich polí a b_j , $j \in \{1, 2, 3\}$ poptávky jednotlivých odběratelů. Účelovou funkci, jejíž minimum hledáme, můžeme zapsat jako

$$5x_{11} + 4x_{12} + 9x_{13} + 8x_{21} + 10x_{22} + 6x_{23}.$$

V posledním řádku a posledním sloupci Tabulky 6.1 vidíme, že naší nabídkou jsme schopni přesně pokrýt poptávky všech odběratelů, tj. že platí rovnost

$$\sum_{i=1}^2 a_i = \sum_{j=1}^3 b_j.$$

Dopravní problém s touto vlastností označujeme jako tzv. *vyrovnaný*. V opačném případě bychom o problému hovořili jako o tzv. *nevyrovnaném*. V nevyrovnaném problému dochází k převisu na straně poptávky, případně nabídky. V našem příkladu je součet poptávaného i nabízeného množství roven 40.

Maximální množství, které můžeme dovézt z pole 1, je 25. Vzhledem k tomu, že jde o vyrovnaný problém, je množství, které z pole 1 odvezeme, právě rovno 25. Z této podmínky dostáváme omezení ve tvaru

$$x_{11} + x_{12} + x_{13} = 25.$$

Pro druhé pole obdobně získáme podmínku

$$x_{21} + x_{22} + x_{23} = 15.$$

Podobná omezení platí i pro stranu poptávky. Odběratelé chtějí získat dohodnutá množství, můžeme tedy psát

$$\begin{aligned} x_{11} + x_{21} &= 9, \\ x_{12} + x_{22} &= 18, \\ x_{13} + x_{23} &= 13. \end{aligned}$$

Poslední podmínky v této úloze jsou podmínky nezápornosti všech proměnných, tj. $x_{ij} \geq 0$.

Nyní můžeme přejít k samotnému řešení úlohy, která je již v kanonickém tvaru. Matice A však neobsahuje žádný sloupec jednotkové matice, musíme tak využít jednu z metod představených v Části 3.3. Výchozí bazický přípustný bod najdeme např. s využitím dvoufázové metody. Přidáním 5 umělých proměnných w_1, \dots, w_5 , bychom pro první fázi získali následující výchozí simplexovou tabulku.

c_B	B	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	w_1	w_2	w_3	w_4	w_5	x_B	$\frac{x_B}{\lambda}$
1	w_1	1	1	1	0	0	0	1	0	0	0	0	25	
1	w_2	0	0	0	1	1	1	0	1	0	0	0	15	
1	w_3	1	0	0	1	0	0	0	0	1	0	0	9	
1	w_4	0	1	0	0	1	0	0	0	0	1	0	15	
1	w_5	0	0	1	0	0	1	0	0	0	0	1	13	
	c	0	0	0	0	0	0	1	1	1	1	1		
	z_k	2	2	2	2	2	2	1	1	1	1	1	80	
	Δ	2	2	2	2	2	2	0	0	0	0	0	80	

Tabulka 6.2: První fáze dvoufázové metody pro Příklad 6.1

Již na první pohled vidíme, že tabulka má „speciální“ strukturu – všechny koeficienty v omezeních jsou buď jedničky nebo nuly, jejich uspořádání navíc splňuje určitou strukturu. Právě kvůli této speciální struktuře (a také kvůli častému využití tohoto typu úloh v praxi) vznikly algoritmy, které jsou při řešení dopravních úloh efektivnější než simplexová metoda. Přesto bychom mohli úlohu vyřešit i dvoufázovou metodou, podrobnější

řešení však necháme na čtenáři. Po šesti krocích první fáze bychom se dostali do situace, ve které je v bázi umělá proměnná w_5 s nulovou hodnotou. S podobnou situací jsme se již setkali při řešení v Příkladu 3.15. Ve druhé fázi bychom získali řešení, kterým je bod $x^* = [x_{11}^*, x_{12}^*, x_{13}^*, x_{21}^*, x_{22}^*, x_{23}^*] = [7, 18, 0, 2, 0, 13]$. Splnění všech podmínek úlohy snadno ověříme dosazením tohoto bodu do výše uvedených omezení.

△

V předchozím příkladu jsme viděli jednoduchý dopravní problém. Obecně můžeme dopravní úlohu zformulovat následujícím způsobem.

$$\left. \begin{array}{l} \min \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{za podmíněk} \quad \sum_{j=1}^n x_{ij} = a_i > 0, \quad i \in \{1, \dots, m\} \\ \quad \quad \quad \sum_{i=1}^m x_{ij} = b_j > 0, \quad j \in \{1, \dots, n\} \\ \quad \quad \quad x_{ij} \geq 0 \end{array} \right\} \quad (6.1)$$

Indexem $i \in \{1, \dots, m\}$ označíme jednotlivé dodavatele a indexem $j \in \{1, \dots, n\}$ odběratele. Kapacity jednotlivých dodavatelů jsou a_i , poptávky odběratelů jsou označeny b_j . Proměnné x_{ij} označují množství přepravené od i -tého dodavatele k j -tému odběrateli, c_{ij} jsou náklady na přepravení jednotkového množství zboží. Je nutné zde zdůraznit, že v dopravní úloze předpokládáme, že zboží je tzv. *homogenní*, tj. je naprosto stejné kvality a nikdo není schopen rozlišit, od jakého dodavatele zboží pochází.

V dopravní úloze také předpokládáme, že náklady jsou lineární funkcí množství zboží. Tento předpoklad je ovšem značným zjednodušením reality, neboť dopravní náklady jsou po částech konstantní. Příkladem může být převoz praček. Pro převoz jedné pračky musíme vypravit dodávku, což nás stojí určité náklady. Druhou pračku do dodávky umístíme snadno, náklady na její převoz jsou tak téměř nulové. V případě, že se do dodávky vejde 20 praček, jsou náklady na všechny tyto pračky, kromě první, blízké nule. Náklady na převoz 21. pračky však nulové rozhodně nejsou, neboť v případě převozu 21 praček potřebujeme vypravit dvě dodávky.

Problém (6.1) má mn proměnných a $m+n$ omezení. Jedno omezení je však lineárně závislé, neboť sečteme-li všech m nabídkových omezení, získáme stejné omezení, jako když sečteme všech n poptávkových omezení. Takto formulovaný problém je tak vyrovnaný, neboť pro něj platí rovnost poptávek a nabídek. Vzhledem k tomu, že jedno z $m+n$ omezení můžeme vyjádřit jako lineární kombinaci zbývajících $m+n-1$ omezení, nemá matice A pro vyrovnaný problém plnou hodnotu.

Problém, ve kterém by došlo k převisu nabídky, by byl ve tvaru nerovnosti. Místo znaménka $=$ by měl v první skupině omezení znaménko \leq . Problém, ve kterém by došlo k převisu na straně poptávky, by naopak měl znaménko \leq ve druhé skupině omezení. V následující části textu se budeme věnovat pouze vyrovnaným dopravním problémům, neboť nevyrovnané problémy můžeme na vyrovnané převést přičtením tzv. *fiktivního odběratele* či *dodavatele*, čímž získáme ve všech omezeních rovnosti. Postup při řešení nevyrovnaného problému si ukážeme v Příkladu 6.6.

Jak již víme, dopravní problém je jedním ze speciálních typů úloh lineárního programování. V Kapitole 1 jsme se setkali s přiřazovací úlohou, která má též speciální strukturu. Přiřazovací úloha je vlastně speciálním případem dopravního problému, v němž jsou všechny kapacity dodavatelů a poptávky odběratelů rovny jedné.

Výhodou dopravního problému je, že již ze zadání úlohy poznáme, zda má či nemá řešení, a to díky platnosti následující ekvivalence.

Věta 6.1. *Pro dopravní úlohu (6.1) jsou následující podmínky ekvivalentní.*

1. Úloha má bazický přípustný bod.
2. Úloha má řešení.
3. Jedná se o vyrovnaný dopravní problém, tj. platí rovnost

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

Důkaz. Důkaz této věty lze najít např. v [32]. □

Nyní již víme, jaká je obecná formulace dopravního problému i jak snadno zjistit, zda má problém řešení. V následující části textu se podíváme na způsoby, jak toto řešení najít. Nejprve se ovšem seznámíme se třemi metodami, pomocí kterých můžeme najít výchozí bazický přípustný bod úlohy. Řešení dopravního problému i hledání výchozího bazického přípustného bodu je pro přehlednost organizováno v tabulkách podobných Tabulce 6.3. Sloupce této tabulky znázorňují jednotlivé odběratele, řádky jednotlivé prodejce. Při řešení dopravních úloh se běžně setkáme s vyplňováním pouze nenulových buněk tabulky, buňky, pro které platí $x_{ij} = 0$, při výpočtech ponecháme prázdné. Prázdné buňky tabulky tak značí nebazické proměnné. S využitím podobných tabulek si nyní ukážeme, jak najít výchozí bazický přípustný bod úlohy.

	O_1	O_2	O_3	nabídka
P_1	c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}	a_1
P_2	c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}	a_2
P_3	c_{31} x_{31}	c_{32} x_{32}	c_{33} x_{33}	a_3
poptávka	b_1	b_2	b_3	$b_1 + b_2 + b_3$

Tabulka 6.3: Obecná tabulka pro dopravní úlohu

6.1 Výchozí bazický přípustný bod

V této části si představíme tři nejběžnější metody pro hledání výchozího bazického přípustného bodu, a to *metodu severozápadního rohu*, *metodu minimální cesty* a *Vogelovu aproximační metodu*. Metody si vždy nejprve popíšeme a poté je ilustrujeme na příkladu. Podrobnější teoretický popis a další metody lze najít např. v [9].

6.1.1 Metoda severozápadního rohu

Metoda severozápadního rohu představuje velmi jednoduchý a intuitivní způsob, jakým můžeme najít výchozí bazický přípustný bod. Jak naznačuje název, v této metodě začneme v severozápadním rohu tabulky, neboli v buňce vlevo nahoře. Do této buňky napíšeme minimum z hodnot nabídky v příslušném řádku a poptávky v příslušném sloupci. V Tabulce 6.3 by se jednalo o minimum z čísel a_1 a b_1 . Tímto krokem jsme vyčerpali buď nabídku nebo poptávku, druhou z těchto hodnot jsme změnili. Tyto změny musíme mít při dalším postupu metody na paměti. Přiřazením čísla $k = \min\{a_1, b_1\}$ jsme poptávku změnili na $b_1 - k$ a nabídku na $a_1 - k$, jedno z těchto čísel je jistě nulové. Tento krok můžeme chápat tak, že dodavatel prodal k kusů zboží příslušnému odběrateli. Poté pokračujeme s vyplňováním jedné ze sousedních buněk, a to buňky ve stejném řádku, pokud se minimum nacházelo v daném sloupci, v opačném případě buňky ve stejném sloupci (pouze o řádek níže). Do vybrané buňky opět vepíšeme hodnotu minima příslušné upravené poptávky a nabídky. Algoritmicky bychom metodu mohli popsat následujícím způsobem.

1. Do buňky v severozápadním rohu (tj. buňky s nejmenšími možnými indexy) přiřadíme největší možné číslo tak, aby bylo splněno příslušné nabídkové i poptávkové omezení, tj. vybereme minimum z těchto dvou čísel. Přepočítáme poptávkové a nabídkové omezení, která jsme tímto krokem změnili.
2. Do nejbližší dostupné sousední buňky přiřadíme největší možné číslo tak, že opět dodržujeme zadaná omezení. Řazení blízkosti sousedních buněk je následující: buňka vedle, buňka pod, buňka diagonálně vpravo dole. Omezení opět přepočítáme.
3. Krok číslo 2 opakujeme, dokud nedojdeme k buňce v jihovýchodním rohu (tj. buňce s největšími možnými indexy). Pro vyrovnanou dopravní úlohu v tomto kroku vždy dojde ke splnění všech poptávkových a nabídkových omezení úlohy.

Tuto metodu si nyní ukážeme na Příkladu 6.2.

Příklad 6.2. Metoda severozápadního rohu.

Metodou severozápadního rohu najděte výchozí bazický přípustný bod pro dopravní úlohu zadanou Tabulkou 6.4. V tabulce jsou písmenem P označení jednotliví prodejci, písmenem O odběratelé. Matematickou formulaci této úlohy můžeme v souladu s (6.1) zapsat ve tvaru

$$\begin{aligned} \min & \quad 6x_{11} + 8x_{12} + 10x_{13} + 7x_{21} + 11x_{22} + 11x_{23} + 4x_{31} + 5x_{32} + 12x_{33} \\ \text{za podmíněk} & \quad x_{11} + x_{12} + x_{13} = 15, \quad x_{21} + x_{22} + x_{23} = 17, \quad x_{31} + x_{32} + x_{33} = 18, \\ & \quad x_{11} + x_{21} + x_{31} = 20, \quad x_{12} + x_{22} + x_{32} = 10, \quad x_{13} + x_{23} + x_{33} = 20, \\ & \quad x_{ij} \geq 0, \quad i, j \in \{1, 2, 3\}. \end{aligned}$$

prodejce \ odběratel	O_1	O_2	O_3	nabídka
	P_1	6	8	10
P_2	7	11	11	17
P_3	4	5	12	18
poptávka	20	10	20	50

Tabulka 6.4: Zadání Příkladů 6.2, 6.3 a 6.4

Řešení. K zápisu řešení využijeme tabulku podobnou Tabulce 6.3, která je pro tento příklad zobrazena v Tabulce 6.5.

	O_1	O_2	O_3	nabídka		O_1	O_2	O_3	nabídka
P_1	6	8	10	15	P_1	6	8	10	15
P_2	7	11	11	17	P_2	7	11	11	17
P_3	4	5	12	18	P_3	4	5	12	18
poptávka	20	10	20	50	poptávka	20	10	20	50

Tabulka 6.5: Zadání Příkladu 6.2

Tabulka 6.6: První krok Příkladu 6.2

Při řešení začínáme s tabulkou, ve které jsou buňky pro x_{ij} prázdné, tj. Tabulkou 6.5. Metodu zahájíme tím, že do políčka, které je v severozápadním rohu (tj. x_{11}), napíšeme minimum z nabídky a poptávky. V našem příkladu jde o číslo 15, neboť odběratel 1 poptává 20, prodejce 1 však nabízí pouze 15. První krok metody je zobrazen v Tabulce 6.6. Vyčerpání nabídky prodejce 1 vyznačíme v dalším kroku hvězdičkou u P_1 . Dále pokračujeme k jedné ze sousedních buněk, kterými jsou buňky pro x_{12} , x_{21} či x_{22} . V našem příkladu již prodejce 1 prodal všechno své zboží odběrateli 1, v prvním řádku tak již nemáme žádné dostupné nabízené zboží. V celém řádku bychom mohli doplnit nuly. Odběratel 1 již má 15 ks zboží, jeho nová poptávka je tak pouze 5.

	O_1	O_2	O_3	nabídka		O_1^*	O_2	O_3	nabídka
P_1^*	6	8	10	15	P_1^*	6	8	10	15
P_2	7	11	11	17	P_2	7	11	11	17
P_3	4	5	12	18	P_3	4	5	12	18
poptávka	20	10	20	50	poptávka	20	10	20	50

Tabulka 6.7: Druhý krok Příkladu 6.2

Tabulka 6.8: Třetí krok Příkladu 6.2

V dalším kroku musíme přejít ke druhému řádku, a to do buňky pro proměnnou x_{21} .

Do něj napíšeme minimum z čísel 17 (nabídka) a 5 (nová poptávka odběratele 1), pro x_{21} tak platí $x_{21} = 5$. Poptávka odběratele 1 je po tomto kroku naplněna, což v další tabulce vyznačíme hvězdičkou u O_1 . Ve zbylých buňkách prvního sloupce si můžeme představit nuly. Tento krok je zobrazen v Tabulce 6.7. V dalších krocích postupujeme obdobně. Vzhledem k tomu, že dodavatel 2 zatím prodal pouze 5 ks svého zboží, je jeho nová nabídka rovna 12. Odběratel 2 poptává 10 ks. Do buňky pro proměnnou x_{22} napíšeme 10, neboť jde o hledané minimum z čísel 10 a 12. Tímto krokem, zobrazeným v Tabulce 6.8, jsme vyčerpali poptávku odběratele 2, nabídka prodejce 2 se snížila na 2 ks.

	O_1^*	O_2^*	O_3	nabídka	
P_1^*	15	6	8	10	15
P_2	5	7	11	11	17
P_3		4	5	12	18
poptávka	20	10	20	50	

	O_1	O_2	O_3	nabídka	
P_1	15	6	8	10	15
P_2	5	7	11	11	17
P_3		4	5	12	18
poptávka	20	10	20	50	

Tabulka 6.9: Čtvrtý krok Příkladu 6.2

Tabulka 6.10: Řešení Příkladu 6.2

V dalším kroku (Tabulka 6.9) se přesuneme k políčku x_{23} , do kterého napíšeme číslo 2, jež je minimem z nabídky (2) a poptávky (20). Tímto krokem vyčerpáme nabídku prodejce 2. V posledním kroku jsou již splněny poptávky odběratelů 1 a 2 a vyčerpány nabídky prodejců 1 a 2. Do políčka x_{33} tak vepíšeme minimum z čísel 18 (nabídka P_3) a 18 (upravená poptávka O_3). Vidíme, že nyní jsou všechna omezení pro nabídky i poptávky splněna, našli jsme tak výchozí bazický přípustný bod. Ověření splnění všech rovností uvedených v matematické formulaci příkladu ponecháme na čtenáři. Nalezený bod je zobrazen v Tabulce 6.10. Hodnota účelové funkce v tomto bodě je rovna součtu nákladů na jednotlivé převozy, celkové náklady nalezené dopravní strategie tedy jsou

$$15 \cdot 6 + 5 \cdot 7 + 10 \cdot 11 + 2 \cdot 11 + 18 \cdot 12 = 473.$$

△

6.1.2 Metoda minimální ceny

Druhou metodou, kterou si zde ukážeme, je metoda minimální ceny. Tato metoda, jak již napovídá název, postupuje po buňkách s minimální cenou. Metoda je opět velmi intuitivní. Algoritmicky ji můžeme popsat následujícím způsobem.

1. Najdeme buňku, ve které jsou nejmenší dopravní náklady c_{ij} a přiřadíme do ní největší možnou hodnotu tak, aby bylo splněno jak nabídkové, tak poptávkové omezení. V případě rovnosti c_{ij} u více buněk si vybereme náhodně.
2. Tento postup opakujeme, dokud čísla v tabulce nesplňují všechna omezení.

Při aplikování metody minimální ceny se již při hledání výchozího bazického přípustného bodu snažíme najít bod, ve kterém je hodnota účelové funkce blízká optimální hodnotě. Ve většině případů získáme využitím této metody bazický přípustný bod s menší hodnotou účelové funkce než s využitím metody severozápadního rohu, která koeficienty účelové funkce nebere v potaz. Při následném řešení dopravní úlohy s výchozím bazickým přípustným bodem, který je blíž optimu, pak využijeme méně iterací. Tuto metodu si nyní ukážeme na stejném příkladu, jako metodu severozápadního rohu.

Příklad 6.3. Metoda minimální ceny.

Metodou minimální ceny najdete výchozí bazický přípustný bod pro dopravní úlohu zadanou Tabulkou 6.4.

Řešení. Stejně jako v Příkladu 6.2 začneme s prázdnou tabulkou (tj. Tabulkou 6.5), do které postupně doplníme hodnoty x_{ij} . Metodu zahájíme nalezením nejmenších dopravních nákladů. Z Tabulky 6.11 vidíme, že nejmenší cena (4) se nachází v buňce x_{31} . Maximální nabídka ve třetím řádku je 18, poptávka v prvním sloupci je 20. Do buňky x_{31} napíšeme minimum z těchto hodnot, tj. 18. Tímto krokem jsme vyčerpali veškerou nabídku prodejce P_3 , ve 3. řádku jsou tak ostatní buňky prázdné, což ve druhém kroku označíme hvězdičkou u P_3 . První krok metody je zobrazen v Tabulce 6.11.

	O_1	O_2	O_3	nabídka		O_1	O_2	O_3	nabídka
P_1	6	8	10	15	P_1	2	8	10	15
P_2	7	11	11	17	P_2	7	11	11	17
P_3	4	5	12	18	P_3^*	4	5	12	18
poptávka	20	10	20	50	poptávka	20	10	20	50

Tabulka 6.11: První krok Příkladu 6.3

Tabulka 6.12: Druhý krok Příkladu 6.3

Jak vidíme z Tabulky 6.12, nejmenší cena mimo třetí řádek je v buňce x_{11} . Vzhledem ke zbylé poptávce odběratele 1, která je nyní pouze 2, zde doplníme číslo 2. Poptávka odběratele O_1 je po tomto kroku metody vyčerpána, ve zbývajících buňkách v prvním sloupci již nemohou být nenulové hodnoty.

V dalším kroku se nejnižší „nevyčerpaná“ cena nachází v buňce x_{12} , což vidíme z Tabulky 6.13. Prodejce P_1 má nyní k dispozici 13, odběratel O_2 poptává 10, do buňky tedy vepíšeme číslo 10. Tímto krokem jsme uspokojili poptávku odběratele O_2 . Jediné dvě buňky, které nyní můžeme doplnit, jsou x_{13} a x_{23} . Nejnižší cenu nyní obsahuje buňka x_{13} , doplníme do ní tak minimum z čísel 3 (zbývajících nabídku P_1) a 20 (poptávka O_3). Tento krok je znázorněn v Tabulce 6.14.

V posledním kroku vepíšeme do buňky x_{23} číslo 17, které zaručí splnění omezení prodejce P_2 a odběratele O_3 . Výsledná tabulka s výchozím bazickým přípustným bodem nalezeným metodou minimální ceny je zobrazena v Tabulce 6.15. Čísla v tabulce splňují všechna omezení, jedná se tedy opravdu o výchozí bazický přípustný bod. Účelová funkce v něm nabývá hodnotu 381, což je výrazné zlepšení oproti bodu, který jsme našli metodou severozápadního rohu.

	O_1^*	O_2	O_3	nabídka
P_1	2 6	10 8	10	15
P_2	7	11	11	17
P_3^*	18 4	5	12	18
poptávka	20	10	20	50

	O_1^*	O_2^*	O_3	nabídka
P_1	2 6	10 8	3 10	15
P_2	7	11	11	17
P_3^*	18 4	5	12	18
poptávka	20	10	20	50

Tabulka 6.13: Třetí krok Příkladu 6.3

Tabulka 6.14: Čtvrtý krok Příkladu 6.3

	O_1	O_2	O_3	nabídka
P_1	2 6	10 8	3 10	15
P_2	7	11	17 11	17
P_3	18 4	5	12	18
poptávka	20	10	20	50

Tabulka 6.15: Řešení Příkladu 6.3

△

6.1.3 Vogelova aproximační metoda (VAM)

Poslední metodou, kterou si zde uvedeme, je Vogelova aproximační metoda (též označovaná zkratkou VAM). Stejně jako metoda minimální ceny se i Vogelova aproximační metoda snaží najít co nejlepší výchozí bazický přípustný bod. V metodě minimální ceny můžeme dojít do situace, kdy musíme obsadit i buňky s velmi vysokými cenami. Tomu se snaží Vogelova aproximační metoda vyhnout. Metodu bychom mohli označit i za tzv. *penalizační metodu*, neboť její postup spočívá v přidělení *penalizace* každému dodavateli a odběrateli. Tyto penalizace vypočítáme jako absolutní hodnotu rozdílu dvou nejmenších čísel v každém sloupci a řádku. Penalizace nám tak říká, o kolik jsou náklady na zvolení druhé nejlepší cesty větší než náklady na zvolení optimální cesty. Algoritmický popis Vogelovy aproximační metody je následující.

1. Vypočítáme penalizace pro všechny dostupné sloupce a řádky v tabulce, ve kterých jsou alespoň dvě dostupné buňky. Penalizace je absolutní hodnota rozdílu dvou nejmenších čísel v každém řádku a sloupci.
2. Vybereme řádek nebo sloupec s největší penalizací.
3. Do buňky s nejmenší cenou ve vybraném řádku nebo sloupci přiřadíme největší možné číslo splňující poptávkové i nabídkové omezení.

4. Opakujeme kroky 1-3 dokud nenajdeme bod, který splní všechna zadaná omezení.

Tato metoda je výpočetně mírně náročnější než předchozí metody, obvykle však dává nejlepší výsledky. V příkladu zadaném Tabulkou 6.4 k tomuto výsledku ovšem nedojde. V některých případech velmi malých úloh se může stát, že pomocí Vogelovy aproximační metody získáme přímo řešení úlohy. I tuto metodu si nyní ukážeme na příkladu.

Příklad 6.4. Vogelova aproximační metoda.

Vogelovou aproximační metodou najděte výchozí bazický přípustný bod pro dopravní úlohu zadanou Tabulkou 6.4.

Řešení. Stejně jako u předchozích metod začneme i zde s prázdnou tabulkou (tj. Tabulkou 6.5). V prvním kroku metody si vypočítáme všechny penalizace. Postup jejich výpočtu je naznačen v Tabulce 6.16. Z Tabulky 6.16 vidíme, že největší penalizace se nachází ve

	O_1	O_2	O_3	nabídka	penalizace
P_1	6	8	10	15	$ 8 - 6 = 2$
P_2	7 17	11	11	17	$ 11 - 7 = 4$
P_3	4	5	12	18	$ 5 - 4 = 1$
poptávka	20	10	20	50	
penalizace	$ 6 - 4 = 2$	$ 8 - 5 = 3$	$ 11 - 10 = 1$		

Tabulka 6.16: První krok Vogelovy metody pro Příklad 6.4

druhém řádku. Do buňky s nejmenší cenou v tomto řádku tak doplníme minimum z čísel 17 a 20, tj. 17. Řádky a sloupce, do kterých již nemůžeme z důvodu vyčerpání nabídky či poptávky doplnit kladná čísla, opět označíme hvězdičkou. Nyní musíme přepočítat penalizace ve všech řádcích a sloupcích, ve kterých není vyčerpána nabídka, případně poptávka.

	O_1	O_2	O_3	nabídka	penalizace
P_1	6	8	10	15	$ 8 - 6 = 2$
P_2^*	7 17	11	11	17	
P_3	4	5 10	12	18	$ 5 - 4 = 1$
poptávka	20	10	20	50	
penalizace	$ 6 - 4 = 2$	$ 8 - 5 = 3$	$ 12 - 10 = 2$		

Tabulka 6.17: Druhý krok Vogelovy metody pro Příklad 6.4

Druhý krok Vogelovy aproximační metody je zobrazen v Tabulce 6.17. Ve druhém kroku jsou hodnoty penalizací stejné jako v prvním kroku, s výjimkou druhého řádku,

pro který již penalizace nepočítáme, neboť je v něm vyčerpaná nabídka. Kvůli vyčerpání nabídky ve druhém řádku se změní i penalizace ve třetím sloupci, neboť buňka s náklady 11 již není dostupná. Nejvyšší penalizací je nyní číslo 3 ve druhém sloupci, který vybereme. Do buňky s nejmenšími náklady v tomto sloupci, tj. x_{32} napíšeme číslo 10, což je minimum z čísel 10 a 18. V dalších krocích postupujeme obdobně. Po přepočítání penalizací nyní dostaneme Tabulku 6.18, ve které je největší penalizace ve třetím řádku. Do buňky x_{31} doplníme číslo 3, které je minimum zbývajících poptávek odběratele O_1 a zbývajících nabídky prodejce P_3 .

	O_1	O_2^*	O_3	nabídka	penalizace
P_1	6	8	10	15	$ 10 - 6 = 4$
P_2^*	7 17	11	11	17	
P_3	4 3	5 10	12	18	$ 12 - 4 = 8$
poptávka	20	10	20	50	
penalizace	$ 6 - 4 = 2$		$ 12 - 10 = 2$		

Tabulka 6.18: Třetí krok Vogelovy metody pro Příklad 6.4

Po přepočtení penalizací se dostaneme k Tabulce 6.19. V této tabulce je již pouze jedna penalizace, neboť poptávky odběratelů O_1 a O_2 jsou již naplněny, stejně tak jako nabídka prodejce P_2 . V posledním sloupci tak vybereme buňku s nejnižšími náklady, kterou je x_{13} . Do této buňky vepíšeme číslo 15, což je minimum z čísel 15 a 20.

	O_1^*	O_2^*	O_3	nabídka	penalizace
P_1	6	8	10 15	15	
P_2^*	7 17	11	11	17	
P_3	4 3	5 10	12	18	
poptávka	20	10	20	50	
penalizace			$ 12 - 10 = 2$		

Tabulka 6.19: Čtvrtý krok Vogelovy metody pro Příklad 6.4

V posledním kroku se dostaneme do situace, ve které není možné spočítat žádnou penalizaci. Nyní máme pouze jednu dostupnou buňku x_{33} , kterou můžeme doplnit. Do této buňky vepíšeme číslo 5, čímž zajistíme splnění všech omezení. Výsledek, který jsme získali pomocí Vogelovy aproximační metody, je zobrazen v Tabulce 6.20. Hodnota účelové funkce v nalezeném výchozím bazickém přípustném bodě je rovna 391, což je více než pro metodu minimální ceny, ale méně než pro metodu severozápadního rohu.

	O_1	O_2	O_3	nabídka
P_1	6	8	10	15
P_2	7	11	11	17
P_3	4	5	12	18
poptávka	20	10	20	50

Tabulka 6.20: Řešení Příkladu 6.4

△

6.2 Modifikovaná distribuční metoda

Nyní již víme, jak najít výchozí bazický přípustný bod. Při postupu řešení bychom tak mohli využít klasickou simplexovou metodu. Vzhledem ke speciálnímu tvaru úlohy ale existují i efektivnější metody. Jednou z těchto metod je tzv. *modifikovaná distribuční metoda (MODI)*, kterou si nyní ukážeme. Tuto metodu lze v literatuře najít též pod názvem *dopravní simplexová metoda* (viz např. [9]) či *U-V metoda* (viz např. [32]). Základem této metody jsou následující dva kroky. V prvním kroku ověříme, zda je nalezený bazický přípustný bod optimální či ne, neboli zda je nalezené schéma přepravy to nejlevnější možné. V případě, že bazický přípustný bod není optimální, pak ve druhém kroku najdeme nový bazický přípustný bod, ve kterém jsou celkové náklady nižší. Tuto metodu si nyní popíšeme podrobněji, uvedeme její algoritmický popis a ukážeme si ji na příkladu. Při řešení dopravních úloh využijeme duální úlohu, která je pro problém (6.1) ve tvaru

$$\begin{aligned} \max \quad & \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \\ \text{za podmínek} \quad & u_i + v_j \leq c_{ij} \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}. \end{aligned}$$

Jestliže máme nedegenerovaný bazický přípustný bod a jestliže pro x_{ij} v úloze (6.1) platí ostrá nerovnost $x_{ij} > 0$ (tj. daná buňka je v bázi), duální omezení $u_i + v_j \leq c_{ij}$ se realizuje jako rovnost $u_i + v_j = c_{ij}$. Tato rovnost plyne z Lagrangeova principu, konkrétně z tzv. *podmínky komplementarity*, jež je ve tvaru $(c_{ij} - u_i - v_j)x_{ij} = 0$. Podrobnější odůvodnění a vysvětlení čtenář najde např. v [3]. Celkově máme soustavu $n + m - 1$ rovnic (neboť jedno omezení primární úlohy jsme kvůli lineární závislosti odebrali) pro $n + m$ neznámých. Její řešení tedy není jednoznačné. Postupujeme tak, že jednu z duálních proměnných položíme rovnu nule, tj. např. $u_1 = 0$, a poté dopočítáme hodnoty ostatních duálních proměnných. Pro buňky, ve kterých je $x_{ij} = 0$ (tj. buňky nejsou v bázi), poté vypočítáme tzv. *náklady příležitosti* jako rozdíl

$$s_{ij} = c_{ij} - u_i - v_j.$$

Náklady příležitosti, někdy též označované jako tzv. *redukované cenové koeficienty*, nám říkají, jaká je cena za přeřazení jednotkového množství do dané buňky. Tyto rozdíly mají podobný význam jako prvky v Δ -řádku v simplexové tabulce, liší se akorát svým znaménkem. Zatímco v simplexové metodě do báze mohly vstoupit proměnné s nezáporným číslem v Δ -řádku, v případě MODI metody mohou do báze vstoupit proměnné s nekladnou hodnotou s_{ij} . Pokud tedy existují indexy i a j , pro které je $s_{ij} \leq 0$, můžeme jimi určenou proměnnou zavést do báze a snížit (v případě $s_{ij} = 0$ nezvýšit) tak hodnotu účelové funkce původní úlohy. V opačném případě jsme našli řešení dopravní úlohy, neboť jsou splněna všechna primární omezení (všechna nabídková i poptávková omezení jsou splněna) i duální omezení (všechna s_{ij} jsou kladná).

V případě, že do báze může vstoupit více proměnných, můžeme využít pravidlo podobné Dantzigovu pravidlu maximálního relativního ocenění. V dopravním problému však vybereme tu proměnnou, pro kterou je s_{ij} minimální. Uvažujme nyní případ, ve kterém chceme do báze zavést proměnnou x_{kl} , neboť platí $s_{kl} \leq 0$. Do této buňky chceme přidělit co největší číslo, které označíme θ . Nyní si vysvětlíme, jak tuto hodnotu získat.

V dalším kroku MODI, podobně jako v klasické simplexové metodě, musíme určit proměnnou, která bázi opustí. V případě, že jsme do buňky pro x_{kl} přiřadili množství θ , musíme množství v ostatních buňkách upravit tak, aby stále platila nabídková i poptávková omezení. V řádku k tak musíme od některé z bazických buněk odečíst θ . Tímto ovšem narušíme některá sloupcová poptávková omezení, ostatní buňky musíme také upravit tak, aby byla zachována přípustnost daného bodu. V tabulce tedy musíme najít uzavřenou cestu, která začíná i končí v buňce x_{kl} , tak, že v každém řádku a sloupci, jež jsou součástí této cesty, jednou přičteme a jednou odečteme θ . Tato cesta prochází pouze přes x_{kl} a některé bazické buňky a má vždy sudý počet vrcholů. Výsledná cesta může sama sebe protínat. Pro nede degenerovaný bod je navíc určena jednoznačně. Jelikož musíme dodržet podmínky nezápornosti x_{ij} , získáme hodnotu θ jako minimum z prvků x_{ij} , od kterých θ odčítáme. Tímto vynulujeme jednu z původních bazických proměnných, neboli určíme proměnnou, která v daném kroku bázi opustí.

Tímto postupem získáme nový bazický přípustný bod. Pro tento bod poté vypočítáme hodnoty s_{ij} , podle kterých určíme, zda se jedná o řešení či nikoliv. Postup, který jsme si nyní podrobně popsali můžeme popsat následujícím algoritmem.

1. Jedním ze způsobů uvedených v Části 6.1 najdeme výchozí bazický přípustný bod.
2. Jednu z proměnných u_i a v_j položíme rovnu nule. Pro usnadnění výpočtu vynulujeme proměnnou v řádku nebo sloupci, ve kterém je nejvíce obsazených buněk. Poté ze vztahu $u_i + v_j = c_{ij}$ vypočítáme hodnoty duálních proměnných u_i a v_j .
3. Pro nebazické proměnné vypočítáme s využitím vztahu $s_{ij} = c_{ij} - u_i - v_j$ náklady příležitosti. Pokud pro všechna s_{ij} platí $s_{ij} > 0$, našli jsme řešení dopravní úlohy a algoritmus končí.
4. Najdeme $s_{kl} = \min\{s_{ij} | s_{ij} \leq 0\}$. Pokud je $s_{kl} = 0$, existuje alternativní řešení úlohy se stejnou hodnotou účelové funkce, které můžeme najít pokračováním v algoritmu. Pokud je $s_{kl} < 0$, příslušná proměnná vstoupí do báze.

5. Do buňky dané indexy k a l přiřadíme největší možnou hodnotu tak, aby byla splněna všechna poptávková i nabídková omezení a omezení na znaménko pro všechna x_{ij} . Proměnná, kterou v tomto kroku vynulujeme, vystupuje z báze. V tomto kroku také přepočítáme celou tabulku.
6. Vrátime se ke kroku 2 a algoritmus opakujeme, dokud nenajdeme řešení.

Tuto metodu si nyní ukážeme na dopravním problému zadaném Tabulkou 6.4.

Příklad 6.5. Modifikovaná distribuční metoda.

Pro výchozí bazický přípustný bod nalezený metodou minimální ceny pro dopravní problém zadaný Tabulkou 6.4 najděte modifikovanou distribuční metodou řešení.

Řešení. Pro inicializaci metody využijeme Tabulku 6.15, ve které je výchozí bazický přípustný bod nalezený metodou minimální ceny. K této tabulce doplníme jeden sloupec a jeden řádek. Do nových buněk tabulky pak zapíšeme hodnoty duálních proměnných u_i a v_j . Proměnnou u_1 položíme rovnu nule, neboť v řádku, který jí přísluší, je nejvíce obsazených buněk. Do poslední buňky ve sloupci a řádku pro hodnoty duálních proměnných vepíšeme

	O_1	O_2	O_3	nabídka	u_i
P_1	6 2	8 10	10 3	15	$u_1 = 0$
P_2	7	11	11 17	17	$u_2 = 1$
P_3	4 18	5	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 8$	$v_3 = 10$		381

Tabulka 6.21: První krok modifikované distribuční metody pro Příklad 6.5

hodnotu účelové funkce. Hodnoty ostatních proměnných získáme s využitím vztahu $u_i + v_j = c_{ij}$. Řešíme tedy následující soustavu rovnic

$$u_1 + v_1 = 6$$

$$u_1 + v_2 = 8$$

$$u_1 + v_3 = 10$$

$$u_2 + v_3 = 11$$

$$u_3 + v_1 = 4.$$

Z této soustavy získáme hodnoty duálních proměnných $u_1 = 0$, $u_2 = 1$, $u_3 = -2$, $v_1 = 6$, $v_2 = 8$ a $v_3 = 10$. Tyto hodnoty poté zapíšeme do výchozí tabulky. Nyní již můžeme vypočítat rozdíly $s_{ij} = c_{ij} - u_i - v_j$ pro proměnné, které nejsou pro výchozí bod v bázi

(neboli pro neobsazené buňky tabulky). Tímto krokem získáme náklady příležitosti

$$\begin{aligned} s_{21} &= c_{21} - u_2 - v_1 = 7 - 1 - 6 = 0, \\ s_{22} &= c_{22} - u_2 - v_2 = 11 - 1 - 8 = 2, \\ s_{32} &= c_{32} - u_3 - v_2 = 5 - (-2) - 8 = -1, \\ s_{33} &= c_{33} - u_3 - v_3 = 12 - (-2) - 10 = 4. \end{aligned}$$

Vypočtené náklady příležitosti nám říkají, jak by se změnily celkové náklady, kdybychom do prázdné buňky přesunuli jednotku zboží. V případě buňky x_{22} by např. náklady vzrostly o 2 na jednotku přesunutého zboží. Pouze v případě zahrnutí buňky x_{32} máme možnost celkové náklady snížit. Bod, který jsme našli metodou minimální ceny, tak není řešením, neboť celkové náklady nejsou minimální. V tomto kroku do báze vstupuje proměnná x_{32} . Do této buňky chceme alokovat největší možné množství zboží, neboť za každou jednotku, kterou do buňky přesuneme, snížíme celkové náklady o 1. Množství, které do buňky přiřadíme, označíme θ . Postup výběru proměnné, která z báze vystoupí, je zachycen v Tabulce 6.22. Přiřadíme-li do buňky x_{32} hodnotu θ , musíme upravit hodnotu v buňce

	O_1	O_2	O_3	nabídka	u_i
P_1	$2 + \theta$	$10 - \theta$	3	15	$u_1 = 0$
P_2			17	17	$u_2 = 1$
P_3	$18 - \theta$	θ		18	$u_3 = -2$
poptávka	20	10	20	50	
v_i	$v_1 = 6$	$v_2 = 8$	$v_3 = 10$		$381 - \theta$

Tabulka 6.22: Pokračování modifikované distribuční metody pro Příklad 6.5

x_{31} tak, aby bylo zachováno nabídkové omezení prodejce P_3 . Od čísla 18 v dané buňce musíme nyní odečíst θ . Nyní musíme změnit hodnotu v buňce x_{11} tak, abychom zachovali poptávkové omezení odběratele O_1 , což zajistíme přičtením θ . Tímto krokem jsme porušili omezení prodejce P_1 , od buňky x_{12} tedy odečteme θ , abychom bylo omezení opět splněno. Tímto krokem jsme zároveň splnili i poptávkové omezení odběratele O_2 . Výsledná uzavřená cesta je tedy $x_{32} \rightarrow x_{31} \rightarrow x_{11} \rightarrow x_{12} \rightarrow x_{32}$. V Tabulce 6.22 jsou všechna omezení splněna.

Maximální hodnota, kterou může θ nabýt, je 10, neboť pro $\theta > 10$ by nebylo splněno omezení na znaménko pro proměnnou x_{12} . Tato proměnná nyní vystupuje z báze. Pro hodnotu $\theta = 10$ získáme Tabulku 6.23. Hodnota účelové funkce v nalezeném bodě je rovna 371, což je o 10 menší než ve výchozím bodě. Tento výsledek odpovídá tomu, že jsme do buňky x_{32} přiřadili množství 10 a na jednotku množství ušetřili 1. Pro tento bod nyní přepočítáme hodnoty duálních proměnných. Opět položíme $u_1 = 0$ a řešíme následující

	O_1	O_2	O_3	nabídka	u_i
P_1	6 12	8	10 3	15	$u_1 = 0$
P_2	7	11	11 17	17	$u_2 = 1$
P_3	4 8	5 10	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_i	$v_1 = 6$	$v_2 = 7$	$v_3 = 10$		371

Tabulka 6.23: Druhý krok modifikované distribuční metody pro Příklad 6.5

soustavu rovnic

$$u_1 + v_1 = 6$$

$$u_1 + v_3 = 10$$

$$u_2 + v_3 = 11$$

$$u_3 + v_1 = 4$$

$$u_3 + v_2 = 5.$$

Výsledky této soustavy jsou zobrazeny v Tabulce 6.23. Pro nebazické proměnné potom najdeme náklady příležitosti vyřešením soustavy

$$s_{12} = c_{12} - u_1 - v_2 = 8 - 0 - 7 = 1,$$

$$s_{21} = c_{21} - u_2 - v_1 = 7 - 1 - 6 = 0,$$

$$s_{22} = c_{22} - u_2 - v_2 = 11 - 1 - 7 = 3,$$

$$s_{33} = c_{33} - u_3 - v_3 = 12 - (-2) - 10 = 4.$$

Žádný z vypočtených rozdílů není v tomto kroku záporný, což znamená, že jsme našli řešení úlohy s minimálními dopravními náklady. Rozdíl s_{21} je nulový. Pro úlohu tak existuje alternativní řešení. To můžeme modifikovanou distribuční metodou najít stejně, jako další bazický přípustný bod. Do báze by v tomto kroku vstoupila proměnná x_{21} . Do příslušné buňky bychom opět přiřadili hodnotu θ . Nyní bychom museli ověřit splnění omezení a doplnit ostatní buňky tak, aby byla všechna omezení splněna. Tímto krokem bychom získali Tabulku 6.24. Z hodnot v Tabulce 6.24 vidíme, že největší možná hodnota θ je rovna 12. Čtenář si snadno vypočítá hodnoty, které by doplněním této hodnoty do tabulky získal. Hodnota účelové funkce je opět rovna 371. Našli jsme tak dvě řešení úlohy, které minimalizují celkové přepravní náklady.

△

Posloupnost bodů, které prochází modifikovaná distribuční metoda, je stejná, jako posloupnost, kterou bychom získali při řešení problému simplexovou metodou s využitím Dantzigova pravidla. Modifikovaná distribuční metoda není jedinou metodou pro řešení

	O_1	O_2	O_3	nabídka	u_i
P_1	6 12- θ	8	10 3+ θ	15	$u_1 = 0$
P_2	7 θ	11	11 17- θ	17	$u_2 = 1$
P_3	4 8	5 10	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_i	$v_1 = 6$	$v_2 = 7$	$v_3 = 10$		371

Tabulka 6.24: Nalezení alternativního řešení pro Příklad 6.5

dopravního problému. V literatuře, např. v [40], lze najít další metodu, kterou je tzv. *stepping stone method*.

V motivačním Příkladu 6.1 i v Příkladu 6.5 jsme získali celočíselná řešení, aniž bychom v úloze celočíselnost vyžadovali. Tento výsledek není náhodný, naopak, jedná se o důležitou vlastnost dopravního problému, která je předmětem následující věty.

Věta 6.2. *Jestliže jsou všechny koeficienty a_i a b_j v dopravní úloze (6.1) celočíselné, pak jsou všechny bazické přípustné body také celočíselné, a to nezávisle na celočíselnosti koeficientů c_{ij} .*

Důkaz. Důkaz této věty lze najít např. v [32]. □

Tato věta nám zaručuje i celočíselnost libovolného přiřazovacího problému s celočíselnými pravými stranami (viz např. Příklad 1.4), který je obecně pouze speciálním případem dopravního problému. Modifikovanou distribuční metodu si nyní ukážeme na příkladu nevyrovnaného dopravního problému.

Příklad 6.6. Nevyrovnaný dopravní problém.

Vyřešte dopravní problém zadaný Tabulkou 6.25.

prodejce \ odběratel	odběratel			nabídka
	O_1	O_2	O_3	
P_1	6	7	5	17
P_2	4	10	9	18
P_3	11	6	12	20
poptávka	15	15	30	

Tabulka 6.25: Zadání Příkladu 6.6

Řešení. Z Tabulky 6.25 vidíme, že úloha je nevyrovnaná, neboť součet poptávek jednotlivých odběratelů (= 60) převyšuje součet nabídek prodejců (= 55). Prvním krokem při řešení této úlohy tak musí být zavedení tzv. *fiktivního prodejce*, jehož nabídka problém

vyrovná. Náklady na převoz od fiktivního prodejce budou vždy rovny nule, neboť množství v těchto buňkách ve skutečnosti nikam nepřevážíme. Proměnné v řádku fiktivního prodejce jsou tak *doplňkové proměnné*, díky kterým úlohu převedeme na kanonický tvar. Přidání umělých proměnných neovlivní výsledek úlohy. Řešení původní (nevyrovnané) úlohy pak získáme vynecháním přidaného řádku, některá poptávková omezení však nemusí být vzhledem k nedostatečné nabídce splněna. V případě nedostatečné poptávky bychom postupovali obdobně, k úloze bychom v tomto případě přidali tzv. *fiktivního odběratele*, jehož poptávka by úlohu vyrovnala. Řešení bychom poté získali vynecháním sloupce pro fiktivního odběratele, celková nabídka by v tomto případě zůstala nevyčerpána.

Tabulka pro vyrovnanou úlohu s fiktivním prodejcem je zobrazena v Tabulce 6.26. Výchozí bazický přípustný bod najdeme např. metodou minimální ceny. Tou postupujeme

	O_1	O_2	O_3	nabídka
P_1	6	7	5	17
P_2	4	10	9	18
P_3	11	6	12	20
FP	0	0	0	5
poptávka	15	15	30	60

Tabulka 6.26: Výchozí tabulka pro Příklad 6.6

následujícím způsobem. Jako první do buňky x_{41} vepíšeme číslo 5, což je nabídka fiktivního prodejce. Toto minimum bychom mohli vepsat i do ostatních buněk s nulovými náklady, buňka byla vybrána náhodně. Dostupná buňka s nejmenšími náklady je nyní x_{21} . Do této buňky přiřadíme 10, neboť jde o minimum z upravené poptávky odběratele O_1 a nabídky prodejce P_2 . V dalším kroku dáme do buňky x_{13} hodnotu 17, čímž vyčerpáme nabídku prodejce P_1 . Další vybranou buňkou je x_{32} , do které přiřadíme číslo 15, neboť jsme limitováni poptávkou odběratele O_2 . V následujícím kroku vepíšeme číslo 8 do buňky x_{32} , čímž zajistíme splnění všech nabídkových omezení s výjimkou omezení prodejce P_3 . V posledním kroku přiřadíme do buňky x_{33} číslo 5. Tímto postupem jsme našli bod, který splňuje všechna omezení. Celkové náklady v tomto bodě jsou 347. Nalezený výchozí bazický přípustný bod je zobrazen v Tabulce 6.27.

Nyní metodou MODI ověříme, zda se jedná o řešení úlohy, či zda můžeme dopravní náklady zmenšit. V Tabulce 6.28 jsou uvedeny hodnoty duálních proměnných v případě, že

	O_1	O_2	O_3	nabídka
P_1	6	7	5	17
P_2	4	10	9	18
P_3	11	6	12	20
FP	0	0	0	5
poptávka	15	15	30	60

Tabulka 6.27: Výchozí bazický přípustný bod pro Příklad 6.6

v_3 položíme rovno nule. Při výpočtu těchto hodnot jsme řešili následující soustavu rovnic

$$\begin{aligned}
 u_1 + v_3 &= 5 \\
 u_2 + v_1 &= 4 \\
 u_2 + v_3 &= 9 \\
 u_3 + v_2 &= 6 \\
 u_3 + v_3 &= 12 \\
 u_4 + v_1 &= 0.
 \end{aligned}$$

	O_1	O_2	O_3	nabídka	u_i
P_1	6	7	5	17	$u_1 = 5$
P_2	4	10	9	18	$u_2 = 9$
P_3	11	6	12	20	$u_3 = 12$
FP	0	0	0	5	$u_4 = 5$
poptávka	15	15	30	60	
v_i	$v_1 = -5$	$v_2 = -6$	$v_3 = 0$		347

Tabulka 6.28: První krok modifikované distribuční metody pro Příklad 6.6

Pro nebazické proměnné potom s využitím vztahu $s_{ij} = c_{ij} - u_i - v_j$ dostaneme

$$s_{11} = 6, \quad s_{12} = 8, \quad s_{22} = 7, \quad s_{31} = 4, \quad s_{42} = 1, \quad s_{43} = -5.$$

Podrobnější řešení této části příkladu ponecháme na čtenáři. Z výsledků pro nebazické proměnné vidíme, že zapojením proměnné x_{43} do báze získáme menší celkové náklady.

	O_1	O_2	O_3	nabídka	u_i
P_1	6	7	5	17	$u_1 = 5$
P_2	$10+\theta$	10	$8-\theta$	18	$u_2 = 9$
P_3	11	6	12	20	$u_3 = 12$
FP	0	0	0	5	$u_4 = 5$
poptávka	15	15	30	60	
v_i	$v_1 = -5$	$v_2 = -6$	$v_3 = 0$		$347 - 5\theta$

Tabulka 6.29: Pokračování modifikované distribuční metody pro Příklad 6.6

Do buňky x_{43} tak přiřadíme hodnotu θ a zjistíme, jak se změní ostatní buňky v tabulce. Řešení této části příkladu je uvedeno v Tabulce 6.29.

Z Tabulky 6.29 vidíme, že přiřadíme-li do buňky x_{43} hodnotu θ , musíme pozměnit buňky x_{41} , x_{21} a x_{23} . Tímto krokem opět dosáhneme splnění všech omezení. Maximální hodnota θ je rovna 5, z báze v tomto kroku odchází proměnná x_{41} . Dosazení této hodnoty do tabulky ponecháme na čtenáři. V dalším kroku bychom opět vypočítali hodnoty duálních proměnných a následně rozdíly s_{ij} pro nebazické proměnné. V tomto kroku bychom získali pouze kladné rozdíly, což znamená, že bod nalezený jedním krokem metody MODI je optimální. Minimální náklady na přepravu jsou 322, což můžeme zjistit jak vynásobením převážných hodnot a nákladů na jednotku, tak odečtením hodnoty 25 od celkových nákladů ve výchozím bazickém přípustném bodě. V novém bodě jsou náklady o 25 menší, neboť jsme do buňky, která přinese pokles nákladů o 5 na jednotku přesunuli množství 5.

Výsledek původní úlohy získáme vynecháním řádku pro fiktivního prodejce. Řešení nevyrovnané úlohy je zobrazeno v Tabulce 6.30. V této tabulce vidíme, že poptávkové omezení odběratele O_3 není kvůli nedostatečné nabídce splněno.

	O_1	O_2	O_3	nabídka
P_1	6	7	5	17
P_2	15	10	3	18
P_3	11	6	12	20
poptávka	15	15	30	60 55

Tabulka 6.30: Řešení Příkladu 6.6



Podobně jako ostatní úlohy lineárního programování, může i dopravní problém být degenerovaný. I u dopravního problému dojde k degenerovanosti v případě, že je některá z bazických proměnných rovna nule. V případě dopravního problému poznáme degenerovaný problém tak, že jsou všechna omezení splněna obsazením méně než $n + m - 1$ buněk v tabulce. Podrobnější teoretický popis degenerovanosti dopravního problému lze najít např. v [32]. V případě degenerované úlohy se můžeme dostat do situace, kdy nejsme schopni vypočítat hodnoty duálních proměnných a rozdílů s_{ij} . Tuto situaci můžeme vyřešit tak, že do jedné z prázdných buněk vepíšeme hodnotu nula, neboli zavedeme do báze novou proměnnou s nulovou hodnotou. Poté již můžeme příslušné hodnoty vypočítat a pokračovat v MODI metodě. Degenerovaný dopravní problém si ukážeme v Příkladu 6.7. I v případě dopravního problému může u degenerovaných úloh dojít k zacyklení. Obdobně jako v případě zacyklení při využití simplexové metody zde můžeme využít náhodné pravidlo či perturbační metodu.

Příklad 6.7. Degenerovaný dopravní problém.

Metodou minimální ceny najděte výchozí bazický přípustný bod dopravního problému zadaného Tabulkou 6.31 a vyřešte jej.

		odběratel			nabídka
		O_1	O_2	O_3	
prodejce	P_1	3	11	9	10
	P_2	4	6	8	30
	P_3	8	13	7	20
poptávka		10	25	25	60

Tabulka 6.31: Zadání Příkladu 6.7

Řešení. Při řešení tohoto problému postupujeme obdobně jako v předchozích problémech. Nejprve metodou minimální ceny najdeme výchozí bazický přípustný bod zobrazený v Tabulce 6.32. Posloupnost buněk, které doplňujeme, je x_{11} , x_{22} , x_{33} , x_{23} . Ověření správnosti tohoto výsledku ponecháme na čtenáři. Stejný výchozí bazický přípustný bod bychom získali i metodou severozápadního rohu a Vogelovou aproximační metodou.

	O_1	O_2	O_3	nabídka
P_1	3 10	11	9	10
P_2	4	6 25	8 5	30
P_3	8	13	7 20	20
poptávka	10	25	25	60

Tabulka 6.32: Výchozí bazický přípustný bod pro Příklad 6.7

Chceme-li pro nalezený bod zahájit modifikační distribuční metodu, získáme následující soustavu rovnic

$$u_1 + v_1 = 3$$

$$u_2 + v_2 = 6$$

$$u_2 + v_3 = 8$$

$$u_3 + v_3 = 7.$$

Položíme-li u_2 rovno nule, dostaneme $u_3 = -1$, $v_2 = 6$ a $v_3 = 8$. Kvůli degenerovanosti nalezeného bodu (v Tabulce 6.32 jsou vyplněny pouze 4 buňky, což je méně než $3+3-1$) nejsme schopni dopočítat hodnoty u_1 a v_1 . Do jedné z nebazických buněk v tabulce tak doplníme hodnotu 0, a to např. do buňky x_{21} . V tomto kroku můžeme vybrat jakoukoli nebazickou proměnnou z prvního řádku a prvního sloupce, neboť tyto proměnné nám umožní soustavu vyřešit. Jedno z pravidel doporučuje vybrat buňku s nejnižšími náklady. Doplníme-li proměnnou do buňky x_{21} , která je shodou okolností buňkou s nejnižšími náklady, soustava bude navíc obsahovat rovnici $u_2 + v_1 = 4$. S využitím této rovnice již můžeme dopočítat hodnoty všech duálních proměnných, máme tedy $u_1 = -1$, $v_1 = 4$. Tento výsledek je zobrazen v Tabulce 6.33.

	O_1	O_2	O_3	nabídka	u_i
P_1	10	11	9	10	-1
P_2	0	25	5	30	0
P_3		13	7	20	-1
poptávka	10	25	25	60	
v_j	4	6	8		360

Tabulka 6.33: Řešení Příkladu 6.7

Pro nebazické proměnné dostaneme hodnoty

$$s_{12} = 11 - (-1) - 6 = 6,$$

$$s_{13} = 9 - (-1) - 8 = 2,$$

$$s_{31} = 8 - (-1) - 4 = 5,$$

$$s_{32} = 13 - (-1) - 6 = 8,$$

ze kterých vidíme, že výchozí bazický bod je zároveň řešením úlohy. Minimální dopravní náklady jsou 360. Ke stejnému výsledku bychom dospěli i zavedením x_{12} či jiné původně nebazické proměnné do báze.

△

Na závěr této kapitoly si ukážeme poslední příklad dopravního problému, který navazuje na Příklad 5.4.

Příklad 6.8. Rozvoz materiálu na roušky.

Organizátoři, kteří v Příkladu 5.4 řešili, jak nastříhat látky na požadované velikosti, nyní stojí před dalším problémem. Potřebují z 912 látek co nejrychleji ušít roušky a rozvézt je lidem, kteří je nutně potřebují. Předpokládejme pro jednoduchost, že dobrovolníci, kteří budou roušky šít, si poradí s různými tvary látek, neboli, že všechny kusy látek nyní považujeme za stejné. Organizátoři mají po Brně rozmístěných 10 míst, ve kterých šijí roušky. Tato místa se liší jak rychlostí šití a následné dopravy k potřebným, tak maximálním množstvím roušek, které mohou ušít. V těchto místech nyní potřebují organizátoři ušít roušky a rozvézt je do 18 míst, jako jsou např. domovy důchodců. Organizátoři chtějí provést úkol co nejefektivněji a dopravit do všech míst roušky co nejrychleji. Jejich cílem je najít takový dopravní plán, který minimalizuje celkové časové náklady na šití a následný rozvoz. Všechny potřebné údaje jsou uvedeny v Tabulce 6.34. Domovy důchodců a další zájemci o roušky jsou označeni písmenem D , organizátoři písmenem O . Pomozme organizátorům s vyřešením tohoto dopravního problému.

Řešení. I přesto, že je tento dopravní problém větší než problémy, které jsme již řešili, můžeme ho metodami vysvětlenými v této kapitole snadno vyřešit. Výchozí bazický přípustný bod najdeme například metodou minimální ceny. Její postup si nyní ukážeme.

Buňky budeme pro větší přehlednost označovat symboly O_iD_j . Metodu minimální ceny zahájíme buňkou O_3D_{12} s cenou rovnou jedné, do které doplníme číslo 48, čímž naplníme poptávku dobrovolníka D_{12} . Další buňkou s cenou jedna, kterou vyplníme, je O_4D_9 , do které vepíšeme číslo 29, čímž naplníme poptávku D_9 . Pokračujeme dalšími buňkami s cenou 1, a to s buňkou O_5D_{17} , kterou vyplníme číslem 28, čímž naplníme poptávku D_{17} , a buňkou $O_{10}D_8$, do které doplníme 19, čímž vyčerpáme nabídku O_{10} .

Další buňky, které vyplníme, jsou ty s cenou rovnou 2. Postupně jde o buňky O_8D_{13} , jejímž doplněním číslem 105 vyčerpáme nabídku O_8 , buňku O_3D_{14} , do které doplníme číslo 45, čímž naplníme poptávku D_{14} a buňku O_5D_{18} , do níž vepíšeme 30, čímž naplníme poptávku D_{18} . V dalších krocích postupujeme obdobně, a to přes buňku O_9D_3 , kterou doplníme číslem 24 a vyčerpáme tak poptávku D_3 a buňku O_4D_{10} , do níž vepíšeme číslo 14 a splníme tím poptávku D_{10} .

Pokračujeme s dostupnými buňkami, které mají dopravní náklady rovny 3. Do buňky O_6D_8 doplníme číslo 59, čímž splníme poptávku D_8 , do buňky O_5D_1 v dalším kroku vepíšeme 40, čímž naplníme poptávku dobrovolníka D_1 . Nejmenší dostupné náklady jsou nyní rovny 4. Postupně doplňujeme buňku O_4D_{13} , do které vepíšeme 42, čímž vyčerpáme nabídku O_4 , buňku O_9D_4 , do které doplníme 41 a vyčerpáme tak nabídku O_9 , buňku O_7D_2 , kterou vyplníme číslem 32, čímž vyčerpáme nabídku O_7 , a buňku O_1D_{11} , jejímž doplněním číslem 21 naplníme poptávku dobrovolníka D_{11} .

Nyní jsou nejnižší dostupné náklady rovny 5, a to v buňce O_5D_{16} , kterou vyplníme číslem 22, čímž vyčerpáme nabídku O_5 , a v buňce O_6D_7 , do které vepíšeme 9 a vyčerpáme tak nabídku O_6 . Následně najdeme dostupné buňky s nejnižšími náklady, které jsou nyní 6. Těmi jsou buňky: O_2D_{15} , jež doplníme číslem 76, čímž naplníme poptávku D_{15} , O_1D_6 , kterou vyplníme číslem 42 a naplníme tak poptávku D_6 , O_2D_7 , jejímž doplněním číslem 27 pokryjeme poptávku D_7 , a O_2D_{13} , do které vepíšeme 18, čímž naplníme poptávku D_{13} .

V dalším kroku jsou nejnižší dostupné náklady 7, a to v buňce O_3D_{16} , do které dopíšeme 41 a naplníme tak poptávku D_{16} . Buňka s nejnižšími dostupnými náklady 8 je nyní O_3D_5 , do které vepíšeme 6, čímž vyčerpáme nabídku O_3 .

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	nabídka
O_1	11	13	17	14	9	6	12	8	9	3	4	11	31	12	7	8	10	14	153
O_2	16	18	14	10	10	11	6	5	6	19	8	17	6	5	6	9	11	12	125
O_3	21	24	13	10	8	8	9	6	13	14	6	1	28	2	17	7	3	26	140
O_4	6	7	5	6	11	22	31	4	1	2	18	14	4	19	7	7	8	8	85
O_5	3	6	5	7	17	8	12	27	12	16	6	7	9	15	6	5	1	2	120
O_6	5	25	6	19	18	24	5	3	11	25	22	1	6	17	8	19	3	4	68
O_7	13	4	16	4	7	11	8	21	4	2	14	19	6	7	15	6	5	4	32
O_8	5	14	5	6	27	6	25	5	2	6	17	6	2	6	6	19	2	8	105
O_9	11	7	2	4	4	30	28	17	26	8	7	24	25	13	7	18	7	9	65
O_{10}	6	17	6	7	11	22	21	1	18	3	6	5	12	7	6	26	3	4	19
poptávka	40	53	24	90	30	42	36	78	29	14	21	48	165	45	76	63	28	30	912

Tabulka 6.34: Zadání Příkladu 6.8

V následujícím kroku doplníme číslem 24 buňku O_1D_5 s náklady 9, čímž naplníme poptávku D_5 . Nyní doplněním čísla 4 do buňky O_2D_4 s náklady 10 vyčerpáme nabídku O_2 . Nejnižší dostupné náklady 13 se v tomto kroku nachází v buňce O_1D_2 , doplněním 21 do této buňky pokryjeme poptávku D_2 . V posledním kroku metody doplníme číslo 45 do buňky O_1D_4 s náklady 14. Tímto krokem jsme splnili všechny poptávky a vyčerpali všechny nabídky. Nalezený bod je tak bazickým přípustným bodem. Tento bod je zobrazen v Tabulce 6.35. Bod je nedegenerovaný, neboť je v tabulce vyplněno $27 = 18 + 10 - 1$ buněk. Splnění všech omezení můžeme s využitím tabulky snadno ověřit. Celkové časové náklady v tomto bodě jsou 4 028.

I přes rozměry této úlohy jsme byli schopni rychle nalézt výchozí bazický přípustný bod metodou minimální ceny. V porovnání se sestavováním simplexových tabulek se jedná o opravdové usnadnění.

Při řešení úlohy modifikovanou distribuční metodou musíme v každém kroku vypočítat hodnoty všech duálních proměnných, kterých je 28 a hodnoty všech rozdíků s_{ij} , kterých je v tomto příkladu 153. Pro ilustraci zde pouze stručně popíšeme první krok metody a bod, který získáme. Další postup modifikované distribuční metody zde však neuvádíme.

V prvním kroku nejprve vypočteme hodnoty duálních proměnných. V případě, že položíme u_1 rovno nule, jsou tyto hodnoty

$$u_1 = 0, \quad u_2 = -4, \quad u_3 = -1, \quad u_4 = -6, \quad u_5 = -3, \quad u_6 = -5, \quad u_7 = -9,$$

$$u_8 = -8, \quad u_9 = -10, \quad u_{10} = -7, \quad v_1 = 6, \quad v_2 = 13, \quad v_3 = 12, \quad v_4 = 14,$$

$$v_5 = 9, \quad v_6 = 6, \quad v_7 = 10, \quad v_8 = 8, \quad v_9 = 7, \quad v_{10} = 8, \quad v_{11} = 4,$$

$$v_{12} = 2, \quad v_{13} = 10, \quad v_{14} = 3, \quad v_{15} = 10, \quad v_{16} = 8, \quad v_{17} = 4, \quad v_{18} = 5.$$

Nyní musíme vypočítat hodnoty nákladů příležitosti $s_{i,j}$, podrobnější řešení této části ponecháme na čtenáři. Při výpočtu získáme 12 záporných hodnot, které jsou

$$s_{1,10} = -5, \quad s_{1,15} = -3, \quad s_{3,4} = -3, \quad s_{3,8} = -1, \quad s_{4,3} = -1, \quad s_{4,4} = -2,$$

$$s_{5,2} = -4, \quad s_{5,3} = -4, \quad s_{5,4} = -4, \quad s_{5,15} = -1, \quad s_{6,3} = -1, \quad s_{7,4} = -1.$$

Výchozí bazický přípustný bod tak není optimální. Největší záporná hodnota je $s_{1,10} = -5$. Obdobně jako v předchozích příkladech musíme nyní najít uzavřenou cestu, která je nyní

$$O_1D_{10} \rightarrow O_4D_{10} \rightarrow O_4D_{13} \rightarrow O_2D_{13} \rightarrow O_2D_4 \rightarrow O_1D_4 \rightarrow O_1D_{10}.$$

Do buňky O_1D_{10} můžeme alokovat maximálně hodnotu 14, z báze v tomto kroku odchází proměnná O_4D_{10} . Přepočteme-li nyní tabulku, získáme bod, který je zobrazen v Tabulce 6.36. Tímto krokem jsme snížili celkový čas o $14 \cdot 5 = 70$ na hodnotu 3 958, o čemž se čtenář snadno přesvědčí v příslušné tabulce.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	nábfika
O_1	11	13	17	14	9	6	12	8	9	3	4	11	31	12	7	8	10	14	153
O_2	16	18	14	10	10	11	6	5	6	19	8	17	6	5	6	9	11	12	125
O_3	21	24	13	10	8	8	9	6	13	14	6	1	28	2	17	7	3	26	140
O_4	6	7	5	6	11	22	31	4	1	2	18	14	4	19	7	7	8	8	85
O_5	3	6	5	7	17	8	12	27	12	16	6	7	9	15	6	5	1	2	120
O_6	5	25	6	19	18	24	5	3	11	25	22	1	6	17	8	22	28	30	68
O_7	13	4	16	4	7	11	8	21	4	2	14	19	6	7	15	6	5	4	32
O_8	5	14	5	6	27	6	25	5	2	6	17	6	2	6	6	19	2	8	105
O_9	11	7	2	4	4	30	28	17	26	8	7	24	25	13	7	18	7	9	65
O_{10}	6	17	6	7	11	22	21	1	18	3	6	5	12	7	6	26	3	4	19
popřávka	40	53	24	90	30	42	36	78	29	14	21	48	165	45	76	63	28	30	912

Tabulka 6.35: Výchozí bazický přípustný bod pro Příklad 6.8

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	nabídka
O_1	11	13 21	17	14 31	9 24	6 42	12	8	9	3 14	4 21	11	31	12	7	8	10	14	153
O_2	16	18	14	10 18	10	11	6 27	5	6	19	8	17	6 4	5	6 76	9	11	12	125
O_3	21	24	13	10	8 6	8	9	6	13	14	6	1	28	2	17	7	3	26	140
O_4	6	7	5	6	11	22	31	4	1	2	18	14	4 56	19	7	7	8	8	85
O_5	3 40	6	5	7	17	8	12	27	12	16	6	7	9	15	6	5	1	2	120
O_6	5	25	6	19	18	24	5 9	3 59	11	25	22	1	6	17	8	19	3	4	68
O_7	13	4 32	16	4	7	11	8	21	4	2	14	19	6	7	15	6	5	4	32
O_8	5	14	5	6	27	6	25	5	2	6	17	6	2 105	6	6	19	2	8	105
O_9	11	7	2 24	4 41	4	30	28	17	26	8	7	24	25	13	7	18	7	9	65
O_{10}	6	17	6	7	11	22	21	1 19	18	3	6	5	12	7	6	26	3	4	19
poptávka	40	53	24	90	30	42	36	78	29	14	21	48	165	45	76	63	28	30	912

Tabulka 6.36: První krok modifikované distribuční metody pro Příklad 6.8

Vzhledem k velikosti úlohy zde nyní uvedeme řešení, které bychom získali po několika krocích modifikované distribuční metody. Tím je bod, v němž jsou celkové časové náklady rovny 3 683. Výsledný bod uvedený v Tabulce 6.37 má 15 bazických proměnných se stejnými hodnotami jako výchozí bod. Tento bod má dále 7 stejných bazických proměnných jako výchozí bod, které se ovšem liší svými hodnotami (O_2D_{13} , O_2D_{15} , O_3D_5 , O_3D_{16} , O_4D_{13} , O_5D_{16} a O_5D_{17}), a 5 nových proměnných, které ve výchozím bodě nejsou v bázi (O_1D_{10} , O_1D_{15} , O_1D_{16} , O_4D_4 a O_5D_2 , které v bázi nahradily proměnné O_1D_2 , O_1D_4 , O_1D_5 , O_2D_4 a O_4D_{10}). Tento bod je, stejně jako výchozí bod, nedegenerovaný.

△

V uvedeném příkladu jsme si ukázali, jak postupovat, když považujeme nastříhané látky za homogenní. Kdybychom tento předpoklad upustili, získali bychom čtyři dopravní problémy, jeden pro každý druh nastříhané látky. Postup řešení by však byl opět stejný.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	nabídka
O_1	11	13	17	14	9	6	12	8	9	3	4	11	31	12	7	8	10	14	153
O_2	16	18	14	10	10	11	6	5	6	14	8	17	6	5	6	9	11	12	125
O_3	21	24	13	10	8	8	9	6	13	14	6	1	28	2	17	7	3	26	140
O_4	6	7	5	6	11	22	31	4	1	2	18	14	4	19	7	7	8	8	85
O_5	3	6	5	7	17	8	12	27	12	16	6	7	9	15	6	5	1	2	120
O_6	5	25	6	19	18	24	5	3	11	25	22	1	6	17	8	18	11	30	68
O_7	13	4	16	4	7	11	8	21	4	2	14	19	6	7	15	6	5	4	32
O_8	5	14	5	6	27	6	25	5	2	6	17	6	2	6	6	19	2	8	105
O_9	11	7	2	4	4	30	28	17	26	8	7	24	25	13	7	18	7	9	65
O_{10}	6	17	6	7	11	22	21	1	18	3	6	5	12	7	6	26	3	4	19
poptávka	40	53	24	90	30	42	36	78	29	14	21	48	165	45	76	63	28	30	912

Tabulka 6.37: Řešení Příkladu 6.8

Příloha

K diplomové práci je přiložen soubor aplikace Excel s řešenými příklady z Kapitoly 5. Nyní si stručně popíšeme, jak se softwarem Excel pracovat při řešení úloh lineárního programování. Excel nabízí k řešení úloh lineárního programování doplněk Řešitel, který lze do Excelu zavést následujícím způsobem. Nejprve na pásu karet vybereme záložku *Soubor*, dále zvolíme položku *Možnosti* v levém dolním rohu obrazovky. Po kliknutí se nám otevře nové okno, ve kterém vybereme *Doplňky*. V posledním řádku okna nyní v rozevíracím seznamu *Spravovat*: vybereme možnost *Doplňky Excelu*, volbu potvrdíme kliknutím na tlačítko *Přejít*, čímž získáme seznam dostupných doplňků Excelu. V tomto seznamu si vybereme doplněk *Řešitel* (označený též *Solver*). Po aktivaci doplňku najdeme ikonu *Řešitel* ve skupině *Analýza* na kartě *Data*. Podrobnější návod aktivace Řešitele najde čtenář např. v [53].

Jakmile máme aktivovaný doplněk Řešitel, můžeme řešit úlohy lineárního programování. Práci s doplňkem si nyní ilustrujeme na Příkladu 1.1. Zadání příkladu můžeme do Excelu zadat např. způsobem zobrazeným na Obrázku 6.1.

	A	B	C	D	E	F	G	H
1		papír (ks)	kroužková vazba (ks)	práce (min)	zisk (Kč)		počet vyrobených kusů	zisk
2	diář:	18	1	30	100		0	0
3	stolní kalendář:	28	2	20	180		0	0
4	max. omezení:	1000	80	900				
5	využití zdrojů:	0	0	0		celkem:	0	

Obrázek 6.1: Zápis zadání Příkladu 1.1 v Excelu

Buňky podbarvené žlutě označují množství jednotlivých zdrojů (ve sloupcích) potřebná k výrobě jednotlivých produktů (v řádcích). V řádku označeném *max. omezení* vidíme maximální dostupné počty jednotlivých zdrojů, řádek označený *využití zdrojů* obsahuje počet využitých zdrojů při daném výrobním plánu. Buňky podbarvené zelenou barvou jsou proměnné, které označují počty vyrobených diářů a stolních kalendářů. V oranžově podbarvené buňce se nachází hodnota účelové funkce, která vznikne jako součet zisku z prodeje jednotlivých výrobků, v syntaxi Excelu můžeme hodnotu buňky zadefinovat např. jako $= G2 \cdot E2 + G3 \cdot E3$. Při zadávání příkladu je nutné vyplnit buňky obsahující proměnné i účelovou funkci. Běžná volba je doplnění buněk jednotlivých proměnných nulami. Další buňky jsou dopočítány podle zadaných vzorců.

Nyní již můžeme přejít k samotnému řešení příkladu. Na pásu karet si na záložce *Data* vybereme *Řešitel*, čímž dostaneme okno nazvané *Parametry Řešitele*. V tomto okně postupně vybereme buňku, v níž se nachází hodnota účelové funkce (v našem příkladu jde o buňku G5) a označíme, zda hledáme její maximum či minimum. Buňky můžeme

vybírat jak vepsáním jejich polohy, tak kliknutím na ně. Dále vybereme buňky, ve kterých jsou proměnné modelu (zde buňky G2 a G3), a pomocí tlačítka *Přidat* zadáme omezující podmínky pro jednotlivé zdroje (zde $B5 \leq B4$, $C5 \leq C6$ a $D5 \leq D4$). V okně si též můžeme vybrat, zda nastavíme podmínky nezápornosti (tj. omezíme všechny proměnné na znaménko) a jakou metodu řešení využijeme. Zadání našeho příkladu je zobrazeno na Obrázku 6.2.

Parametry Řešitele

Účelová funkce:

Hledat: Max Min Hodnota:

Proměnné modelu:

Omezující podmínky:

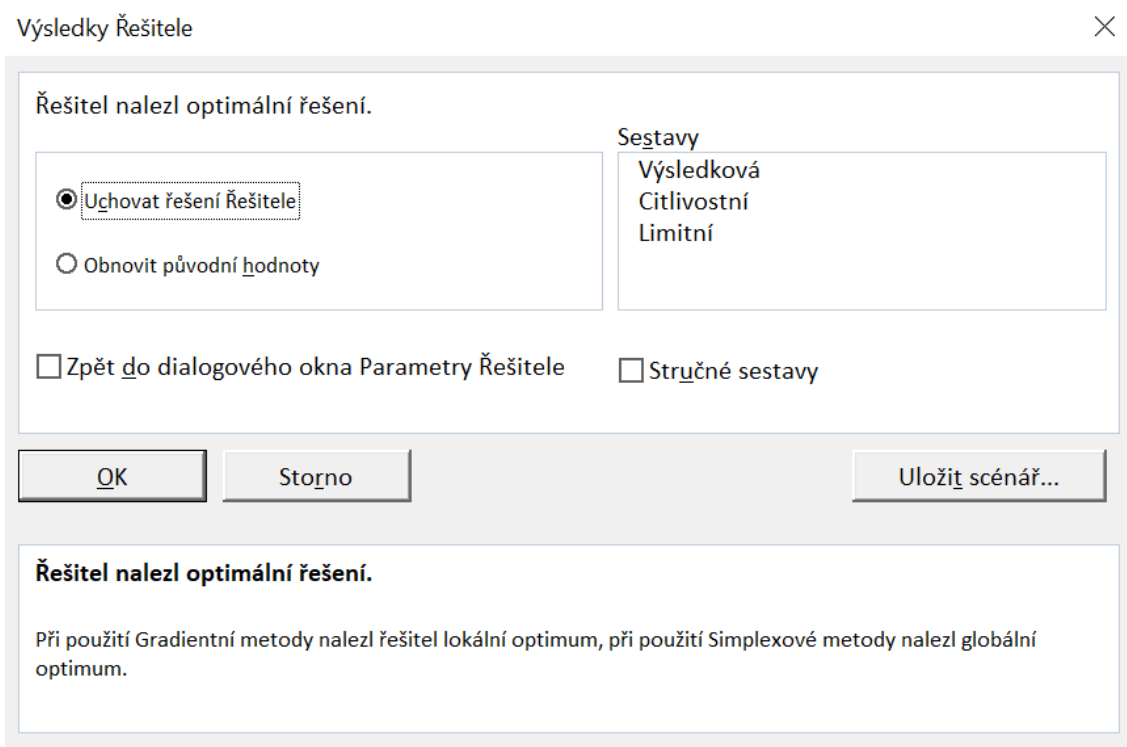
Nastavit podmínky nezápornosti

Vyberte metodu řešení:

Metoda řešení
 Simplexovou metodu zvolte pro lineární optimalizační problémy, Gradientní metodu pro hladké nelineární problémy a Evoluční algoritmus pro nehladké nelineární problémy.

Obrázek 6.2: Zápis zadání Příkladu 1.1 v Řešiteli

Po kliknutí na tlačítko *Řešit* se objeví okno *Výsledky Řešitele*, na kterém můžeme specifikovat, co si přejeme zobrazit. Toto okno je zobrazeno na Obrázku 6.3. V případě, že zvolíme pouze přednastavenou možnost *Uchovat řešení Řešitele*, získáme řešení našeho problému v námi zadané tabulce. Řešení příkladu (zaokrouhlené na jedno desetinné místo) je zobrazeno na Obrázku 6.4. Z výsledků vidíme, že se firma maximalizující zisk zaměří pouze na výrobu stolních kalendářů, a to v největším možném množství, které vyhoví všem zadaným podmínkám.



Obrázek 6.3: Zpráva o výsledcích Řešitele Příkladu 1.1

	A	B	C	D	E	F	G	H
1		papír (ks)	kroužková vazba (ks)	práce (min)	zisk (Kč)		počet vyrobených kusů	zisk
2	diář:	18	1	30	100		0	0
3	stolní kalendář:	28	2	20	180		35,7	6428,6
4	max. omezení:	1000	80	900				
5	využití zdrojů:	1000	71,4	714,3		celkem:	6428,6	

Obrázek 6.4: Výsledek Příkladu 1.1

V přiloženém excelovém souboru najde čtenář zadání i řešení všech příkladů z Kapitoly 5 ve formě ilustrované zde na Příkladu 1.1.

Seznam použité literatury

- [1] D. Alevras, M.W. Padberg, *Linear Optimization and Extensions: Problems and Solutions*, Springer, New York, 2001.
- [2] O. Y. M. Al-Rawi, T. Mukherjee, *Application of Linear Programming in Optimizing Labour Scheduling*, *Journal of Mathematical Finance* **9** (2019) 272-285.
- [3] M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, *Linear Programming and Network Flows*, 4. vydání, John Wiley, Hoboken, 2010.
- [4] A. Ben-Tal, A. Nemirovski, *Optimization III - Lecture Notes*, Georgia Institute of Technology, 2020 (elektronicky). Citováno 12. 3. 2020. Dostupné z: https://www2.isye.gatech.edu/~nemirovs/OPTIII_LectureNotes2020.pdf
- [5] D. Bertsimas, J.N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, 1997.
- [6] G. B. Dantzig, *Lineárne programovanie a jeho rozvoj*, Slovenské vydavateľstvo technickej literatúry, Bratislava, 1966.
- [7] G. B. Dantzig, *The Diet Problem*, *Dantzig, Interfaces* **20**, (1990), č. 4, 43–47.
- [8] G. B. Dantzig, *Linear Programming*, *Operations Research* **50** (2002), č. 1, 42–47.
- [9] G. B. Dantzig, M. N. Thapa, *Linear Programming*, Springer, New York, 1997.
- [10] J. C. Dickson, F. P. Frederick, *A decision rule for improved efficiency in solving linear programming problems with the simplex algorithm*, *Communications Of The Acm.* **3**, 509-512.
- [11] C. van Dooren, *A Review of the Use of Linear Programming to Optimize Diets, Nutritiously, Economically and Environmentally*, *Frontiers In Nutrition* **5** (2018).
- [12] R. J. Duffin, *On Fourier's Analysis of Linear Inequality Systems, in: Pivoting And Extension*, Springer Berlin, 1974, str. 71-95.
- [13] D. Eigen, *Pivot Rules for the Simplex Method*, 2011 (elektronicky). Citováno 4. 3. 2020. Dostupné z: <https://pdfs.semanticscholar.org/9f11/f1fa16c966be86da9356c0b52d367599d131.pdf>
- [14] H. A. Eiselt, C. L. Sandblom, *Integer Programming and Network Models*, Springer, 2000.

- [15] H. A. Eiselt, C. L. Sandblom, *Linear Programming and its Applications*, Springer, New York, 2007.
- [16] F. Furini, E. Malaguti, D. Thomopulos, *Modeling Two-Dimensional Guillotine Cutting Problems via Integer Programming*, *Inform Journal On Computing* **28** (2016) 736-751.
- [17] S. G. Garille, S. I. Gass, *Stigler's Diet Problem Revisited*, *Operations Research* **49** (2001) 1-13.
- [18] J. F. Grcar, *How ordinary elimination became Gaussian elimination*, *Historia Mathematica* **38** (2011), č. 2, 163-218.
- [19] I. Griva, S. Nash, A. Sofer, *Linear and Nonlinear Optimization*, 2. vydání, Society for Industrial and Applied Mathematics, Philadelphia, 2009.
- [20] J. A. J. Hall, K. I. M. McKinnon, *The simplest examples where the simplex method cycles and conditions where EXPAND fails to prevent cycling*, *Math. Program., Ser. B* **100** (2004) 133–150.
- [21] G. H. Hurlbert, *Linear Optimization: The Simplex Workbook*, Springer, New York, 2010.
- [22] E. K. P. Chong, S. H. Zak, *An Introduction to Optimization*, 3. vydání, Wiley-Interscience, Hoboken, 2008.
- [23] V. Chvátal, *Linear Programming*, W.H. Freeman, New York, 1983.
- [24] P. Jež, *Kolik mám denně přijmout bílkovin, tuků a sacharidů, obecně energie?*, 2017 (elektronicky). Citováno 12. 3. 2020. Dostupné z: <http://www.faktorplus.cz/rs/vyziva-a-dopluky-stravy/kolik-mam-denne-prijmout-bilkovin-tuku-sacharidu-obecne-energie.html>
- [25] J. Klapka, J. Dvořák, P. Popela, *Metody operačního výzkumu*, 2. vydání, VUTIUM, Brno, 2001.
- [26] B. Kolman, R. E. Beck, *Elementary Linear Programming with Applications*, 2. vydání, Academic Press, San Diego, 1995.
- [27] P. Kolman, *Metoda vnitřního bodu*, Univerzita Karlova, 2011 (elektronicky). Citováno 1. 3. 2020. Dostupné z: <https://kam.mff.cuni.cz/~kolman/vnitri.pdf>
- [28] J. Matoušek, *Lineární programování*, Univerzita Karlova, 2006 (elektronicky). Dostupné z: <https://iti.mff.cuni.cz/series/2006/311.pdf>
- [29] A. Nemirovski, *Introduction to Linear Optimization*, Georgia Institute of Technology, 2012 (elektronicky). Citováno 25. 2. 2020. Dostupné z https://www2.isye.gatech.edu/~nemirovs/OPTI_LectureNotes.pdf

- [30] U. S. Pawar and D. B. Hanchate *Literature Review on Personnel Scheduling*, International Journal of Computer Engineering and Technology **5** (2013).
- [31] A. Phillips, *The Tableau Economique as a Simple Leontief Model*, The Quarterly Journal Of Economics **69** (1955).
- [32] J. Plesník, J. Dupačová, M. Vlach, *Lineárne programovanie*, Alfa, vydavateľstvo technickej a ekonomickej literatúry, Bratislava, 1990.
- [33] V. Pourová, A. Jakešová, *O výživě*, Pointa, Praha, 2019.
- [34] P. Savický, *Výpočetní složitost I*, Univerzita Karlova, 2016 (elektronicky). Citováno 21. 2. 2020. Dostupné z <http://www.cs.cas.cz/~savicky/vyuka/vyysl/vyysl2016zs2.pdf>
- [35] I. V. Semoushin, *Linear Programming: Computer Interactive Course*, Gerhard Mercator Universität Duisburg, 2002.
- [36] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, 1986.
- [37] A. Schrijver, *On the History of the Transportation and Maximum Flow Problems*, Documenta Mathematica, Extra Volume ISMP (2012) 169–180.
- [38] G. Sierksma, Y. Zwols, *Linear and Integer Optimization, Theory and Practice*, 3. vydání, 2015.
- [39] D. Spielman, S. H. Teng, *Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time*, Proceedings Of The Thirty-Third Annual Acm Symposium On Theory Of Computing - Stoc '01, ACM Press, New York, USA, 2001, str. 296-305.
- [40] B. W. Taylor, *Introduction to management science*, 10. vydání, Prentice Hall, Upper Saddle River, 2010.
- [41] T. Terlaky, S. Zhang, *Pivot rules for linear programming: A survey on recent theoretical development*, Annals of Operations Research **46-47** (1993), no. 1, 203.
- [42] P. R. Thie, G. E. Keough, *An Introduction to Linear Programming and Game Theory*, 3. vydání, Wiley, 2008.
- [43] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, 3. vydání, Springer, Boston, 2008.
- [44] B. Van Roy, K. Mason, *Formulation and Analysis of Linear Programs*, 2016 (elektronicky). Citováno 4. 3. 2020. Dostupné z: <https://web.stanford.edu/class/engr62/notes/notes.pdf>
- [45] A. M. Vershik, *L. V. Kantorovich and Linear Programming*, 2001 (elektronicky). Citováno 11. 2. 2020. Dostupné z: http://www.mathsoc.spb.ru/pantheon/kantorov/versh_e.pdf

- [46] A. M. Vershik, *Long History of the Monge–Kantorovich Transportation Problem*, *The Mathematical Intelligencer* **35** (2013) 1-9.
- [47] P. Ženčák, *Lineární programování*, Univerzita Palackého v Olomouci, 2013.
- [48] *Doporučené denní dávky vitamínů a minerálů*, 2017 (elektronicky). Citováno 12. 3. 2020. Dostupné z: <https://www.naturis.cz/blog/view/doporucene-denni-davky-vitaminu-a-mineralu/>
- [49] *Kalorické tabulky*, 2020 (elektronicky). Citováno 12. 3. 2020. Dostupné z: <http://www.kaloricke-tabulky.cz/nastroje/vyzivove-vyhledavani-potravin>
- [50] *Fit Výživa Rob*, 2015 (elektronicky). Citováno 12. 3. 2020. Dostupné z: <https://fitvyzivarob.cz/stranka/556772-nutricni-hodnoty-priloh/>
- [51] *Obchod Tesco online*, 2020 (elektronicky). Citováno 12. 3. 2020. Dostupné z: <https://itesco.cz/>
- [52] *Stigler's Nutrition Model* (elektronicky). Citováno 11. 2. 2020. Dostupné z: https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_diet.html
- [53] *Zavedení doplňku Řešitel v Excelu* (elektronicky). Citováno 19. 5. 2020. Dostupné z: <https://is.muni.cz/go/koso5t>

