

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA MATEMATIKY

Bakalářská práce

Optimalizace zásob a přepravní úlohy s omezeními

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jitka DYKOVÁ**
Osobní číslo: **A16B0218P**
Studijní program: **B1101 Matematika**
Studijní obor: **Matematika a finanční studia**
Téma práce: **Optimalizace zásob a přepravní úlohy s omezeními**
Zadávající katedra: **Katedra matematiky**

Zásady pro vypracování

Optimalizace zásob a dodávek (supply planning problem) je důležitou součástí distribuční logistiky. Obsahem práce je zpracování tematiky optimalizace zásob a příslušných přepravních úloh v oblasti matematických modelů a základních metod jejich řešení. Hlavním cílem je jejich aplikace na úlohy s dodatečnými omezeními založené na reálných nebo pseudoreálných datech.

Rozsah bakalářské práce: **cca 20 stran**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

- S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, ISBN-13:978-0521833783, 2005.
- A. Ben-Tal, L. El-Ghaoui, A. Nemirovski, Robust optimization. Princeton University Press, ISBN 978-0-691-14368-2, 2009.
- R. K. Cheung, W. B. Powell, Models and Algorithms for Distribution Problems with Uncertain Demands. Transportation Science 30:43-59, 1996.
- F. Maggioni, M. E. Bertocchi, F. A. Potra, Stochastic versus Robust Optimization for a Transportation Problem, Optimization Online, 2015.

Vedoucí bakalářské práce: **Doc. Ing. Roman Čada, Ph.D.**
Katedra matematiky

Datum zadání bakalářské práce: **1. října 2019**
Termín odevzdání bakalářské práce: **19. května 2020**

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka



Brandner

Doc. Ing. Marek Brandner, Ph.D.
vedoucí katedry

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně s použitím uvedených zdrojů.

V Plzni dne 20. 7. 2020

.....

Jitka Dyková

PODĚKOVÁNÍ

Tímto bych chtěla poděkovat vedoucímu práce panu doc. Ing. Romanovi Čadovi Ph.D. za odborné rady při psaní práce, jeho trpělivost a vstřícnost. Dále bych chtěla poděkovat svému partnerovi za podporu při vzniku této práce.

ABSTRAKT

Práce je věnována optimalizaci distribučních problémů. Z počátku jsou představeny základní úlohy a jejich způsoby řešení. Dále už se přechází na složitější a rozšířenější modely, kde je požadována přeprava různých produktů, od vícera dodavatelů k odběratelům na více dní. Daný problém je řešen deterministickým a stochastickým způsobem.

Je uvažována velikostně reálná situace, řešená pomocí softwaru. Cílem této práce je analýza dopravního problému a také zkoumání různých velikostí modelů v závislosti na době trvání výpočtu. Dále následuje modelování úloh v softwaru AMPL a porovnání řešení třemi vybranými řešiči CPLEX, Gurobi a Xpress .

ABSTRACT

In this thesis a distribution problem is discussed. At first some basic problems are introduced together with corresponding solution methods. Next, more and more complex models are being presented involving product delivery problems from multiple producers to multiple consumers during some predefined time etc. This example problem is solved using deterministic and also stochastic approach.

A software solution is presented involving real dataset size. This thesis aims to devlivery problem analysis and comparison of different model size and computational time. The solution has been implemented in AMPL and calculated by three solvers: CPLEX, Gurobi and Xpress.

OBSAH	
PROHLÁŠENÍ	i
PODĚKOVÁNÍ	ii
ABSTRAKT	iii
ABSTRACT	iii
ÚVOD	1
1 LINEÁRNÍ PROGRAMOVÁNÍ	2
1.1 Obecná formulace úlohy lineárního programování.....	2
1.2 Základní pojmy	3
2 GRAFICKÁ INTERPRETACE	4
2.1 Grafické metoda LP.....	4
2.2 Počet optimálních řešení.....	4
3 APLIKACE LP	6
3.1 Plánování výroby	6
3.2 Směšovací problém	6
3.3 Řezný problém	7
3.4 Dopravní úloha.....	7
4 SIMPLEXOVÁ METODA	7
4.1 Kanonický tvar	7
4.2 Bazické (základní) řešení.....	9
4.3 Postup simplexové metody	10
4.4 Počet řešení	12
4.5 Dualita lineárního programování	12
5 CELOČÍSELNÉ LINEÁRNÍ PROGRAMOVÁNÍ	13
5.1 Klasifikace úloh	13
5.2 Metody řešení.....	13
5.3 Totální unimodularita	14
6 DOPRAVNÍ ÚLOHA	15
6.1 Formulace dopravní úlohy obecně	15
6.2 Základní metody řešení	16
6.2.1 Metoda severozápadního rohu (SZ rohu).....	18
6.2.2 Indexová metoda.....	19

6.3 Dopravní úloha – cirkulace v sítích.....	24
6.3.1 Formulace úlohy	24
6.3.2 Celočíselnost dopravní úlohy	25
6.4 Multiperiodická úloha	25
7 STOCHASTICKÉ PROGRAMOVÁNÍ	26
8 ROBUSTNÍ PROGRAMOVÁNÍ.....	30
9 SLOŽITĚJŠÍ DOPRAVNÍ PROBLÉM.....	31
9.1 Software pro nalezení řešení.....	31
9.2 Implementace	32
9.3 Dopravní úloha v praxi	33
9.3.1 Deterministické řešení.....	34
9.3.2 Stochastické řešení – normální rozdělení.....	35
9.3.3 Stochastické řešení – rozdělení extrémálních hodnot.....	35
9.3.4 Praktické řešení.....	35
ZÁVĚR.....	39
LITERATURA	40

ÚVOD

Dopravní síť je významnou součástí našeho života. Už za války byla potřeba koordinace vzájemně souvisejících činností k dosažení daného cíle. Jedním z představitelů, který se zabýval myšlenkou optimalizačních postupů, byl matematik G. B. Dantzig. Ten je považován za zakladatele lineárního programování [1].

V prvních dvou kapitolách je popsána základní teorie lineárního programování, grafické znázornění pro lepší představu problému.

Kapitola 3 poukazuje na rozsah využití lineárního programování v různých odvětvích. V dalších částech je zpracováno celočíselné programování a simplexový algoritmus, který se aplikuje na méně náročné úlohy.

Od kapitoly 6 už se podrobněji zabýváme pouze jednou aplikací lineárního programování dopravní úlohou. Nejdříve jsou popsány jednoduché dvourozměrné modely a algoritmy řešení, vytvořené pouze pro dopravní úlohy. Na to navazují úlohy rozšířené o další rozměry například čas. Ty jsou řešeny pomocí implementace v softwaru AMPL a využitím řešičů pro získanou úlohu.

Kapitola 7 navrhuje stochastický přístup pohledu řešení problému pomocí normálního a extrémálního rozdělení. V 8. kapitole je okrajově zmíněno robustní programování jako další možný směr pokračování.

Protože cílem úlohy je implementace složitějšího optimalizačního problému, následuje proto porovnání doby trvání řešení pomocí deterministického a stochastického programování třemi vybranými řešiči.

1 LINEÁRNÍ PROGRAMOVÁNÍ

Lineární programování je odvětví operačního výzkumu. Je možné ho využít jako prostředek pro plánování budoucího vývoje daného problému, zabezpečující nalezení optimálního rozložení omezených zdrojů. Při jeho aplikaci na daný problém se využívá matematický model. V této části je čerpáno z [2], [3], [4], [5].

1.1 Obecná formulace úlohy lineárního programování

Jde o optimalizaci lineární funkce, zohledňující dané omezení problému. Matematický model se skládá z

- lineární účelové (kriteriální) funkce $z = f(x)$, která vyjadřuje cíl analýzy (1.1),
- lineárních omezujících podmínek, vyjadřující vliv činitelů, které musí řešení splňovat. Mohou být ve tvaru rovnic či nerovnic (1.2),
- podmínek nezápornosti (1.3).

Výsledný matematický model úlohy lineárního programování, dále LP, s n neznámými a m omezeními, kde $n, m \in \mathbb{N}$, při nalezení extrému definujeme

$$\text{opt } z = \mathbf{c}^T \mathbf{x}, \quad (1.1)$$

za podmínek

$$\mathbf{Ax} \leq \mathbf{b}, \quad (1.2)$$

$$\mathbf{x} \geq \mathbf{0}, \quad (1.3)$$

kde

opt = minimalizace či maximalizace funkce z ,

$\mathbf{c}^T = (c_1, c_2, c_3, c_4, \dots, c_n)$ je řádkový vektor cenových koeficientů účelové funkce $\mathbf{c} \in \mathbb{R}^n$,

$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)^T$ je sloupcový vektor proměnných modelu $\mathbf{x} \in \mathbb{R}^n$,

$\mathbf{b} = (b_1, b_2, b_3, \dots, b_m)^T$ je sloupcový vektor hodnot pravých stran podmínek

$\mathbf{b} \in \mathbb{R}^m$,

$\mathbf{0} = (0, 0, \dots, 0)^T$ je nulový sloupcový vektor

a \mathbf{A} matice koeficientů podmínek $\mathbf{A} \in \mathbb{R}^{m \times n}$.

Poznámka 1.1. Je-li definována úloha minimalizující účelovou funkci

$$\min z = f(x),$$

lze jednoduše přepsat pomocí maximalizace funkce jako

$$\max (-z) = -f(x).$$

1.2 Základní pojmy

Definice 1.2. Vektor $x \in \mathbb{R}^n$ splňující všechna omezení dané úlohy (1.2), (1.3) se nazývá **přípustné řešení**.

Definice 1.3. Vektor $x \in \mathbb{R}^n$, který nesplňuje alespoň jedno omezení (1.2), (1.3), nazýváme **nepřípustné řešení**.

Definice 1.4. Řešení x je **optimální** právě tehdy, když x je přípustné řešení a nabývá nejlepší hodnoty účelové funkce, tzn. kdy účelová funkce z je v požadovaném extrému.

Definice 1.5. Mějme množinu X . Ta je **konvexní** právě tehdy, když $\forall x_1, x_2 \in X$,
$$z = \alpha x_1 + \beta x_2 \in X \tag{1.4}$$

kde $\alpha, \beta \in \langle 0; 1 \rangle$; $\alpha + \beta = 1$

Poznámka 1.6. Mějme $a \in \mathbb{R}^n$; $\delta > 0$. **Otevřeným δ -okolím** bodu a rozumíme množinu bodů $x \in \mathbb{R}^n$ s euklidovskou vzdáleností od a menší než δ .

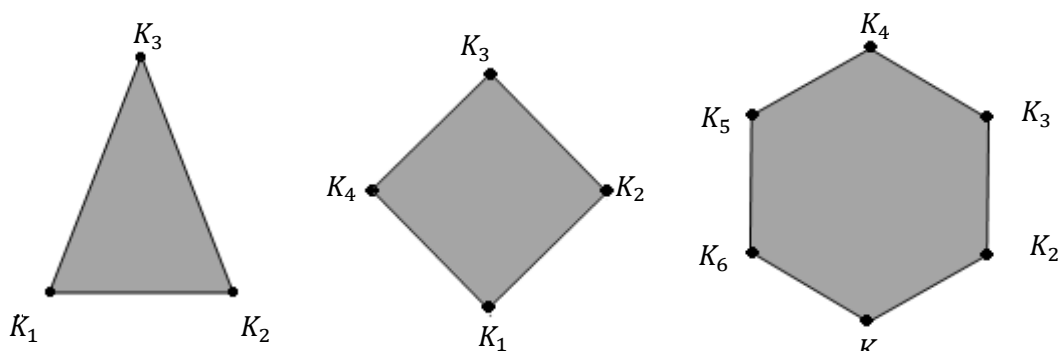
Definice 1.7. Neprázdná, uzavřená množina všech přípustných řešení množiny ve tvaru $Ax \leq b$ se nazývá **konvexní polyedr**.

Definice 1.8. Bod K se nazývá **krajní bod (KB)**, jestliže leží v konvexní množině a zároveň neleží na spojnici dvou jiných bodů z této množiny.

Poznámka 1.9. Pokud existuje KB lze ho nalézt pouze na „povrchu“ konvexní množiny.

Poznámka 1.10. Pokud je množina přípustných řešení konvexní polyedr a zároveň existuje optimální řešení, pak optimální řešení leží v jeho krajním bodu.

Na následujícím obrázku jsou uvedeny příklady konvexního polyedru s krajními body pro dimenzi \mathbb{R}^2 .



Obrázek 1: Konvexní polyedr se 3, 4 a 6 krajními body.

Poznámka 1.11. Konvexní polyedr v \mathbb{R}^3 může být například krychle či čtyřstěn.

2 GRAFICKÁ INTERPRETACE

Úlohu LP lze nejnázorněji na pochopení znázornit pomocí grafické metody. Jsme ji schopni zobrazit maximálně pro model v \mathbb{R}^3 . V modelu v \mathbb{R}^2 využijeme kartézské souřadnicové soustavy v rovině. Proměnné x_1, x_2 představují osy souřadnicového systému. Tato kapitola je zpracována dle [2].

2.1 Grafické metoda LP

Metoda je pouze ilustrační, jelikož ve skutečnosti bývá v modelu mnohem více proměnných.

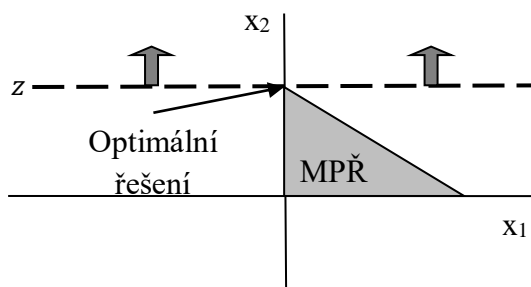
Postup řešení

- Omezující podmínky typu nerovnosti nejprve znázorníme jako přímky. Přímka rozdělí rovinu na dvě poloroviny. Dále zvolíme libovolný bod, který neleží na přímce, a dosadíme ho do nerovnosti. V případě, že nerovnost je splněna, hledáme právě tuhle polorovinu. Jinak hledáme polorovinu opačnou. Uvažujeme pouze I. kvadrant souřadnicového systému v souvislosti s nezáporností proměnných.
- Průnik omezujících podmínek, oblast kde jsou splněny všechny podmínky současně, se nazývá **množina přípustných řešení**, dále MPŘ.
- Následuje zakreslení účelové funkce pro libovolně zvolené $z(x)$. Jelikož jsou hladiny účelové funkce při různé volbě $z(x)$ rovnoběžné, stačí ji posunovat při maximalizaci ve směru růstu $z(x)$. Pro minimalizaci je posun ve směru klesání hodnoty $z(x)$.
- Optimum je krajní bod, který náleží průniku MPŘ a účelové funkce z .
- Zjištění hodnoty účelové funkce a následná interpretace výsledků.

2.2 Počet optimálních řešení

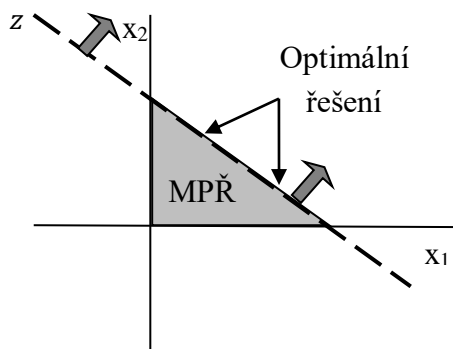
Obecně v úloze lineárního programování můžou nastat 3 situace v počtu řešení. Úloha může mít řešení jedno, nekonečně mnoho či žádné. Následující obrázky poukazují na situace počtu optimálních řešení v \mathbb{R}^2 , které mohou nastat.

- MPŘ tvoří konvexní polyedr.
Optimální řešení je právě jedno



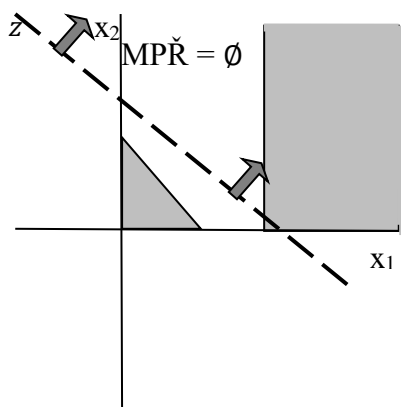
Obrázek 2: Příklad úlohy s jedním optimálním řešením

Optimálních řešení je nekonečno čili se jedná o **alternativní optimální řešení**. Je to speciální případ rovnoběžnosti omezující podmínky a hladin účelové funkce.



Obrázek 3: Příklad úlohy s nekonečně optimálními řešeními

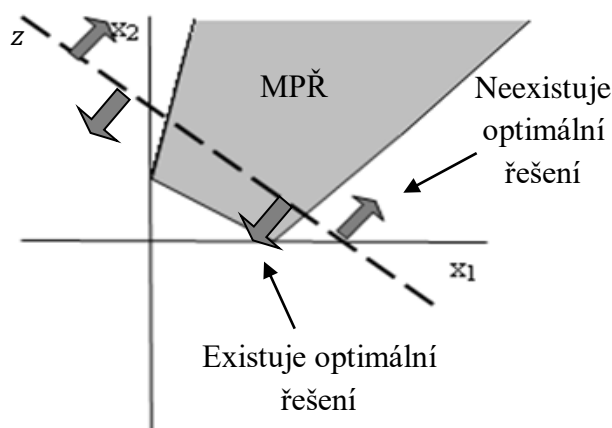
- MPŘ je prázdná
Optimální řešení neexistuje, řešení je nepřipustné.



Obrázek 4: Příklad prázdné MPŘ

- MPŘ je neomezená

Optimální řešení je právě jedno, či neexistuje. Zde záleží na posunu účelové funkce. Neexistence optimálního řešení vzniká v důsledku neomezenosti MPŘ.



Obrázek 5: Příklad znázornění otevřené MPŘ

3 APLIKACE LP

Uvedeme si následující oblasti, kde se lineární programování může uplatnit. Zpracováno podle [5].

3.1 Plánování výroby

V úlohách plánování výroby je potřeba určit množství produktů výroby tak, aby za daných omezujících podmínek byl zisk firmy maximální nebo náklady na výrobu produktů minimální. Proměnné v modelu zastupují jednotlivá množství vyrobených produktů. Mezi faktory ovlivňující plán výroby patří například kapacita surovin, hodinová práce dělníků, požadavky odběratelů a jiné.

3.2 Směšovací problém

Pod pojmem směšovací problém si můžeme představit kombinace látek, které mají požadované zastoupení ve výsledné směsi. Cílem analýzy problému může být minimalizace nákladů na vytvoření směsi. Mezi speciální případy směšovacích úloh lze zařadit úlohu optimalizace portfolia a nutriční problém.

Optimalizace portfolia

Cílem úlohy je optimální rozložení velikosti investic kapitálu do aktiv tak, aby byl očekávaný výnos maximální nebo vzniklé riziko minimální. Proměnné v modelu představují jednotlivé velikosti investic do aktiv. Omezující podmínkou mohou být množství finančních prostředků investora.

Nutriční problém (úloha o výživě)

Tato úloha řeší denní příjem potravy se správným obsahem denní výživy. Účelová funkce zabezpečuje minimální náklady na pořízení potravin či maximální podání předepsané látky. Omezujícími podmínkami může být kapacita, požadavek nebo přesné množství dané látky.

3.3 Řezný problém

Zde se řeší dělení celků materiálu (prken, trubek), aby vzniklý odpad byl minimální. Proměnné v modelu představují jednotlivé dělení větších celků na menší. Omezující podmínka je například počet nově vzniklých celků.

3.4 Dopravní úloha

Cílem je zajištění distribuce např. zboží, surovin z místa A (dodavatel) do místa B (odběratel) při minimálních nákladech. Proměnné jsou počty převezeného zboží. Omezující podmínky představují požadavky odběratele či kapacitu dodavatele. Danou úlohou se budeme podrobněji zabývat v kapitole 6.

4 SIMPLEXOVÁ METODA

V kapitole 2 jsme si ukázali, jak lze řešit jednoduché úlohy LP. V praxi se ale setkáme s mnohem složitějšími případy obsahující stovky proměnných. Pro výpočet se použije iterační metoda, která zkoumá krajní body MPŘ a hledá nejlepší z nich.

Formulace algoritmu řešení LP byla představena roku 1947. Její představitel byl americký matematik George Bernard Dantzig. K odvození algoritmu jsou použity texty [2], [6] a [7].

4.1 Kanonický tvar

Definice 4.1. Lineární model je v **kanonickém tvaru**, pokud

- jsou všechny omezující podmínky ve tvaru rovnice,
- všechny složky vektoru \mathbf{b} jsou nezáporné,
- matici \mathbf{A} má hodnost m ,

$$\text{opt } z = \mathbf{c}^T \mathbf{x}, \quad (4.1)$$

za podmínek

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (4.2)$$

$$\mathbf{x} \geq \mathbf{0}, \quad (4.3)$$

kde $\mathbf{c}^T = (c_1, c_2, c_3, c_4, \dots, c_n)$ je řádkový vektor cenových koeficientů účelové funkce $\mathbf{c} \in \mathbb{R}^n$,

$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)^T$ je sloupcový vektor proměnných modelu $\mathbf{x} \in \mathbb{R}^n$,

$\mathbf{b} = (b_1, b_2, b_3, \dots, b_m)^T$ je sloupcový vektor hodnot pravých stran podmínek $\mathbf{b} \in \mathbb{R}^m$,

$\mathbf{0} = (0, 0, \dots, 0)^T$ je nulový sloupcový vektor

a \mathbf{A} matice koeficientů podmínek $\mathbf{A} \in \mathbb{R}^{m \times n}$.

4.1.1. Vznik kanonického tvaru

Následujícími kroky upravujeme model (1.1) – (1.3).

- Pokud je zadání úlohy maximalizovat, přejdeme na další krok. V případě minimalizace musíme model převést na maximalizaci.

$$\min z = \sum_{j=1}^n c_j x_j \Leftrightarrow \max(-z) = \sum_{j=1}^n -c_j x_j \quad (4.4)$$

- Pokud má i -tá podmínka, složka vektoru \mathbf{b} zápornou hodnotu, vynásobíme výraz -1 .

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \Leftrightarrow \sum_{j=1}^n -a_{ij} x_j \leq -b_i \quad (4.5)$$

- Omezení tvaru nerovnosti převedeme na rovnost pomocí **přídavné proměnné** x , pro kterou platí $x \geq 0$. Přídavné proměnné nemají žádný vliv na účelovou funkci, proto je v účelové funkci jejich koeficient nulový.

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \Leftrightarrow \sum_{j=1}^n a_{ij} x_j + x = b_i \quad (4.6)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \Leftrightarrow \sum_{j=1}^n a_{ij} x_j - x = b_i \quad (4.7)$$

- V případě, že matice \mathbf{A} neobsahuje jednotkovou submatici řádu m , musíme do dané rovnice podmínky uměle doplnit jednotkový sloupcový vektor. Tato proměnná se nazývá **pomocná proměnná**. Celý proces se označuje jako M -metoda. Oproti přídavné proměnné, pomocné ovlivňují účelovou funkci. Jejich koeficientům proto přiřadíme vysokou hodnotu M .

Příklad 1.

Výchozí zadání		Převod na maximalizaci (4.4) a odstranění záporné pravé strany (4.5)	
	$\min 2x_1 - x_2$		$\max -2x_1 + x_2$
	$\text{z p. } x_1 - x_2 \geq -2$		$-x_1 + x_2 \leq 2$
	$x_1 + x_2 \leq 6$	\Rightarrow	$x_1 + x_2 \leq 6$
	$x_2 = 3$		$x_2 = 3$
	$x_1, x_2 \geq 0$		$x_1, x_2 \geq 0$
Převedení na rovnosti (4.6)		Doplnění jednotkové submatice	
\Rightarrow	$\max -2x_1 + x_2 + (0x_3 + 0x_4)$	\Rightarrow	$\max -2x_1 + x_2 + (0x_3 + 0x_4) + 1000x_5$
	$-x_1 + x_2 + x_3 = 2$		$-x_1 + x_2 + x_3 = 2$
	$x_1 + x_2 + x_4 = 6$		$x_1 + x_2 + x_4 = 6$
	$x_2 = 3$		$x_2 + x_5 = 3$
	$x_1, x_2, x_3, x_4 \geq 0$		$x_1, x_2, x_3, x_4, x_5 \geq 0$

Tabulka 1: Příklad tvorby kanonického tvaru.

4.2 Bazické (základní) řešení

V systému dvou proměnných se bazické řešení nacházelo v krajních bodech konvexního polyedru. Pro modely více proměnných by také šlo nalézt krajní body a v nich si vypočítat hodnotu účelové funkce a vybrat z nich optimální bod, ale je to zbytečně složité.

Definice 4.2. Je-li $\mathcal{B} = \{A_{j_1}, A_{j_2}, \dots, A_{j_m}\}$ množina m lineárně nezávislých sloupců matice A , pak vektor $x \in \mathbb{R}^n$, splňující podmínky

- $x_j = 0$ pro $A_j \notin \mathcal{B}$,
- x_k je k -tá složka vektoru $B^{-1}b$, kde $B = [A_{j_i}]_{i=1}^m$ pro $k = 1 \dots m$,

se nazývá **bazické řešení**, příslušné množiny \mathcal{B} .

Definice 4.3. **Bazické přípustné řešení BPŘ** je bazické řešení, pro které platí $x_j \geq 0, j = 1 \dots n$.

Pro představu budeme pokračovat v příkladu 1.

$$A = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ a } I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Když položíme $x_1, x_2 = 0$,

$$\begin{pmatrix} 1 & 0 & 0 & | & 2 \\ 0 & 1 & 0 & | & 6 \\ 0 & 0 & 1 & | & 3 \end{pmatrix} \Rightarrow \begin{matrix} x_3 = 2 \\ x_4 = 6 \\ x_5 = 3 \end{matrix} \Rightarrow \text{jedno z přípustných řešení je } x = [0, 0, 2, 6, 3].$$

4.3 Postup simplexové metody

- Převod modelu do kanonického tvaru a zároveň nalezení BPŘ.
- Nalezení jiného BPŘ s lepší hodnotou účelové funkce. To znamená vyměnit jeden ze sloupců z báze za jiný, který v bázi není, pomocí Gauss-Jordanovy metody transformací matice A . Výpočet probíhá v tabulce, kde je vepsán kanonický tvar daného modelu.

Koefficienty účelové funkce					Sloupec pravých stran
		c_1	...	c_n	
Báze	c_B	x_1	...	x_n	b
x_{B1}	c_{B1}	a_{11}	...	a_{1n}	b_1
...
x_{Bm}	c_{Bm}	a_{m1}	...	a_{mn}	b_m
		Pomocný řádek			Hodnota účelové funkce
Koefficienty účelové funkce bazických proměnných		$z_1 - c_1$			$HÚF$

Tabulka 2: Výchozí simplexová tabulka.

Sloupcové pravidlo (4.9) říká, že zavedeme do báze sloupec s s nejnižší zápornou hodnotou (maximalizační úloha), aby se zvýšila hodnota účelové funkce. Vybereme prvek za využití pomocného řádku, pro který platí:

$$z_j - c_j = \sum_{i=1}^m c_{B_i} a_{ij} - c_j \quad (4.8)$$

Naopak k nalezení prvku, který z báze vyjmeme, použijeme **pravidlo řádkové**.

Po určení sloupce s hledáme v jeho řádcích i prvek a_{is} , pro který platí:

$$\min \frac{b_i}{a_{is}} \quad (4.9)$$

kde a_{is} vybíráme pouze z prvků $a_{is} > 0$.

Na pozici, kde se protíná řádkové pravidlo se sloupcovým, vzniká určující prvek neboli pivot. Podle pivota přetransformujeme prvky v tabulce. V řádku $z_j - c_j$ u bazických proměnných se musí vyskytnout nula.

- Testování optimality se ověřuje v pomocném řádku $z_j - c_j$. Jestliže obsahuje pouze hodnoty ≥ 0 , je aktuální BPŘ optimálním řešením.

Poznámka 4.4. Může nastat situace kdy (4.8) a (4.9) splňuje současně více hodnot v sloupci/řádku. Zde volíme libovolně, který prvek vyjmemme a nahradíme.

Příklad 2:

$$\begin{aligned} \max z &= 8x_1 + 6x_2 + 2x_3 \\ \text{z p. } 4x_1 + 2x_2 + x_3 &\leq 36, \\ x_1 + 2x_2 - 2x_3 &\leq 12, \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Převedení na kanonický tvar

$$\begin{aligned} \max z &= 8x_1 + 6x_2 + 2x_3 + (0x_4 + 0x_5) \\ \text{z p. } 4x_1 + 2x_2 + x_3 + x_4 &= 36, \\ x_1 + 2x_2 - 2x_3 + x_5 &= 12, \\ x_1, x_2, x_3, x_4, x_5 &\geq 0. \end{aligned}$$

		8	6	2	0	0		
Báze	c_B	x_1	x_2	x_3	x_4	x_5	b	$\frac{b_i}{a_{is}}$
x_4	0	4	2	1	1	0	36	9
x_5	0	1	2	-2	0	1	12	12
$z_j - c_j$		-8	-6	-2	0	0	0	

BPŘ po 1. kroku $x^{(1)} = [0, 0, 0, 36, 12]$

x_1 za x_4		8	6	2	0	0		
Báze	c_B	x_1	x_2	x_3	x_4	x_5	b	$\frac{b_i}{a_{is}}$
x_1	8	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	9	18
x_5	0	0	$\frac{3}{2}$	$-\frac{9}{4}$	$-\frac{1}{4}$	1	3	2
$z_j - c_j$		0	-2	0	2	0	72	

BPŘ po 2. kroku $x^{(2)} = [9, 0, 0, 0, 3]$

x_2 za x_5		8	6	2	0	0		
Báze	c_B	x_1	x_2	x_3	x_4	x_5	b	$\frac{b_i}{a_{is}}$
x_1	8	1	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	8	8
x_2	6	0	1	$-\frac{3}{2}$	$-\frac{1}{6}$	$\frac{2}{3}$	2	$-\frac{4}{3}$
$z_j - c_j$		0	0	-3	$\frac{5}{3}$	$\frac{4}{3}$	76	

BPŘ po 3. kroku $x^{(3)} = [8, 2, 0, 0, 0]$

x_3 za x_1		8	6	2	0	0		
Báze	c_B	x_1	x_2	x_3	x_4	x_5	b	$\frac{b_i}{a_{is}}$
x_3	2	1	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	8	
x_2	6	$\frac{3}{2}$	1	0	$\frac{1}{3}$	$\frac{1}{6}$	14	
$z_j - c_j$		3	0	0	$\frac{8}{3}$	$\frac{1}{3}$	100	

$$x_{opt} = [0, 14, 8, 0, 0], HÚF = 100$$

4.4 Počet řešení

Při řešení lineárních úloh, mohou nastat možné situace.

- Neomezená MPŘ, optimální řešení nelze nalézt. Nastane situace, kdy pro všechny výrazy platí, že $\frac{b_i}{a_{is}} \leq 0$.
- Neexistence přípustného řešení. Z toho vyplývá, že nemůže existovat ani řešení optimální. To nastává v případě, že se v BPŘ na pozici pomocných proměnných objeví nenulová hodnota.
- Nekonečno optimálních řešení se projeví, jestliže má v pomocném řádku $z_j - c_j$ nebazická proměnná hodnotu 0.

4.5 Dualita lineárního programování

Teorie duality má velice důležitý význam. Ke každé primární úloze LP lze vytvořit duálně sdruženou úlohu.

Definice 4.5. Duální úloha LP k úloze primární, tj. úloze

$$\max z = \mathbf{c}^T \mathbf{x} \quad (4.10)$$

za podmínek

$$\mathbf{Ax} \leq \mathbf{b}, \quad (4.11)$$

$$\mathbf{x} \geq 0 \quad (4.12)$$

je úloha

$$\min g = \mathbf{b}^T \mathbf{y} \quad (4.13)$$

za podmínek

$$\mathbf{A}^T \mathbf{y} \geq \mathbf{c}, \quad (4.14)$$

$$\mathbf{y} \geq 0 \quad (4.15)$$

Poznámka 4.6. Složky \mathbf{y} se nazývají duální proměnné.

Věta 4.7. (Slabá věta o dualitě) Necht' \mathbf{x} a \mathbf{y} jsou přípustná řešení primární a duální úlohy, potom platí $\mathbf{c}^T \mathbf{x} \geq \mathbf{b}^T \mathbf{y}$.

Věta 4.8. (Silná věta o dualitě) Jestliže \mathbf{x} je optimálním řešením primární úlohy, poté existuje přípustné řešení duální úlohy \mathbf{y} takové, že $\mathbf{c}^T \mathbf{x} = \mathbf{b}^T \mathbf{y}$.

Poznámka 4.9. Existuje-li optimální řešení \mathbf{x} a \mathbf{y} , poté $z(\mathbf{x}) = g(\mathbf{y})$.

5 CELOČÍSELNÉ LINEÁRNÍ PROGRAMOVÁNÍ

V této kapitole uvažujeme stejný model, navíc přibude podmínka celočíselnosti řešení, tj. požadavek aby řešení nabývala celočíselných hodnot. V této kapitole je použit text [8].

$$\text{za podmínek} \quad \text{opt } z = \mathbf{c}^T \mathbf{x}, \quad (5.1)$$

$$\mathbf{Ax} \leq \mathbf{b}, \quad (5.2)$$

$$\mathbf{x} \geq 0, \quad (5.3)$$

$$\mathbf{x} \in \mathbb{Z} \quad (5.4)$$

5.1 Klasifikace úloh

Úlohy celočíselného programování se dělí na 3 typy:

- Ryze celočíselné, kde se celočíselnou požaduje ve všech proměnných.
- Částečně celočíselné, některé proměnné mohou nabývat i neceločíselných hodnot.
- Bivalentní $\{0; 1\}$, kde proměnné nabývají jen hodnot nula či jedna.

5.2 Metody řešení

Zde si uvedeme některé ze základních metod celočíselného řešení.

5.2.1 Grafická metoda

Grafická metoda je vhodná pro \mathbb{R}^2 . Postup řešení je stejný jako u grafické metody lineárního programování. Zde navíc zohledňujeme hledané celočíselné řešení. Jestliže řešení není celočíselné, mohli bychom uvažovat zaokrouhlení hodnoty, zde ale není zaručené, že daný bod je optimální či leží v MPR. Je proto lepší, využívat následující metodu větví a mezí.

5.2.2 Metoda větví a mezí

Princip této metody je dělení MPŘ na menší podmnožiny, dokud není nalezeno optimální řešení. K nalezení řešení se využívá stromový diagram složený z jednotlivých uzlů a větví. Větvící se uzly představují podmnožiny MPŘ. Postup pro maximalizaci

- Vyřešení optimálního řešení x_0 např. pomocí simplexové, grafické metody bez požadavku celočíselnosti řešení. Pokud by řešení bylo celočíselné, výpočet v tomto uzlu končí.
- V případě, kdy nějaká složka $x_i = d_i$ optimálního řešení x není celočíselná, rozdělíme množinu přípustných řešení na dvě podmnožiny přidáním podmínky $x_i \leq [d_i]$, respektive $[d_i] \leq x_i$. Tento proces se nazývá **větvění**, vytvoříme jím příslušné dva uzly.
- V každém uzlu s novým omezením nalezneme optimální řešení a horní mez. **Horní mezí** na množině se rozumí optimální hodnota účelové funkce dané úlohy bez ohledu na celočíselnost.

Poznámka 5.1. Neexistuje-li ve větvi přípustné řešení, nelze dále množinu větvit.

5.3 Totální unimodularita

Pro dané tvary matice A je získané řešení celočíselné. Uvedeme si podmínky, za kterých bude optimální řešení automaticky celočíselné [6] [9].

Definice 5.2. Matici A typu m/n nazveme **totálně unimodulární**, jestliže

- $a_{ij} \in \{0, 1, -1\}, i = 1, \dots, n, j = 1, \dots, m,$
- determinant každé čtvercové podmatice matice A je roven $-1, 0$ nebo 1 .

Věta 5.3. Necht' je dána lineární úloha ve tvaru $Ax = b, x \geq 0$, kde A je celočíselná matice s lineárně nezávislými řádky a vektor b je celočíselný.

Potom jsou následující podmínky ekvivalentní:

- Determinant každé báze B nabývá hodnot ∓ 1 .
- Extrémní body konvexního polyedru $Ax = b, x \geq 0$ jsou celočíselné pro všechny celočíselné vektory b
- Inverzní matice B^{-1} každé báze B je celočíselná.

Důkazy viz [9].

Věta 5.4. Matice A je **totálně unimodulární** právě tehdy, když součet prvků v každé Eulerovské čtvercové submatici je násobkem čtyř.

Poznámka 5.5. Jestliže je součet prvků v každém řádku a v každém sloupci matice A sudý, poté se nazývá **Eulerovská** matice.

Uvedeme postačující podmínky pro totální unimodulární matice.

Věta 5.6. Matice A je **totálně unimodulární**, jestliže jsou splněny obě podmínky:

- Každý sloupec obsahuje nejvýše dva nenulové prvky.
- Řádky matice A lze rozdělit na dvě části A_1 a A_2 tak, že dva nenulové prvky ve sloupci jsou ve stejné části řádků a mají odlišné znaménko. Pokud jsou dva nenulové prvky v A_1 a A_2 , znaménko je stejné.

6 DOPRAVNÍ ÚLOHA

Jak už bylo řečeno v kapitole 3, analýza dopravního problému hledá nejefektivnější dopravní plán přepravy zboží z pohledu přepravních nákladů, uspokojující všechny zákazníky a nepřesahující kapacity zdrojů. Dopravní úlohu budeme dále značit DÚ [2], [10], [11].

6.1 Formulace dopravní úlohy obecně

Cíl je stanovení počtu přepravovaných jednotek od dodavatele (zdroje) k odběrateli (určenému cíli) tak, aby byly minimalizovány přepravní náklady. Model dopravní úlohy lineárního programování lze sestavit následovně.

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (6.1)$$

za podmínek

$$\sum_{j=1}^n x_{ij} = a_i \quad i = 1, 2, \dots, m \quad (6.2)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad j = 1, 2, \dots, n \quad (6.3)$$

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n \quad (6.4)$$

kde

x_{ij} ... počet přepravovaných výrobků od i -tého dodavatele k j -tému odběrateli.

c_{ij} ... cenové koeficienty na přepravu jednotky výrobku od i -tého dodavatele k j -tému odběrateli,

a_i ... kapacita i -tého dodavatele,

b_j ... požadavek j -tého odběratele,

m ... počet dodavatelů,

n ... počet odběratelů.

Předpokládáme problém ve **vybilancovaném tvaru** (6.5). To znamená, že součet všech kapacit dodavatele a požadavků odběratele je shodný. V reálných případech tomu vždy není. Počet proměnných ve vybilancovaném tvaru je $m \cdot n$ a $m + n$ lineárně nezávislých podmínek. To znamená, proměnné x_{ij} bude nejvýše $m + n - 1$ hodnot kladných, ostatní proměnné budou nulové.

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (6.5)$$

Jakmile požadavky převažují kapacitu dodavatele či naopak, hovoříme o **nevybilancovaném tvaru**. Dopravní problém se musí převést na vybilancovaný tvar.

V případě, kdy je součet kapacit dodavatele vyšší než požadavky odběratele, doplníme model o **fiktivního odběratele** s požadavkem $\sum_{i=1}^m a_i - \sum_{j=1}^n b_j$. Při převisu požadavků odběratele doplníme model o **fiktivního dodavatele** s kapacitou $\sum_{j=1}^n b_j - \sum_{i=1}^m a_i$. V obou případech jsou cenové koeficienty c_{ij} rovny nule.

Poznámka 6.1. Při přechodu DÚ na vybilancovaný tvar je v podstatě převod modelu na kanonický tvar. Při doplnění fiktivního odběratele či dodavatele se přidávají další proměnné, dříve uvedené jako proměnné přídatné.

6.2 Základní metody řešení

DÚ řešíme pouze ve vybilancovaném tvaru. Možnost řešení DÚ by bylo simplexovým algoritmem, ale při převodu do kanonického tvaru by vektor proměnných x_{ij} obsahoval pouze dva jedničkové koeficienty na pozici i a $j + m$.

Ostatní koeficienty by byly nulové, proto je tato metoda k výpočtu komplikovaná. DÚ má své speciální metody řešení. Jako u simplexové metody, tak i při řešení dopravních úloh se využije přepisu do tabulky.

	Požadavky odběratele	b_1	b_2	...	b_n
Kapacita dodavatelů	odběratelé	O_1	O_2	...	O_n
	dodavatelé				
a_1	D_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1n} x_{1n}
a_2	D_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2n} x_{2n}
...
a_m	D_m	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mn} x_{mn}

Tabulka 3: Tabulkový zápis DÚ.

Nalezení řešení lze rozdělit do 3 kategorií

- Libovolné přípustné řešení

Zde se jedná o případ, libovolného přípustného řešení splňující omezující podmínky, kde zanedbáváme velikost nákladů. Dalším případem je, když jsou si všechny přepravní náklady c_{ij} rovny. V realitě tento případ skoro nenastane. K nalezení řešení používáme nejjednodušší metodu Severozápadního rohu.

- „Dobré“ přípustné řešení

K získání přibližného přípustného řešení, které nezaručuje však optimum, se využívají „heuristické“ metody. Mezi tyto metody patří například indexová a Vogelova aproximační metoda. Pomocí těchto metod lze řešení dopravního problému rychle a snadno nalézt.

- Optimální řešení

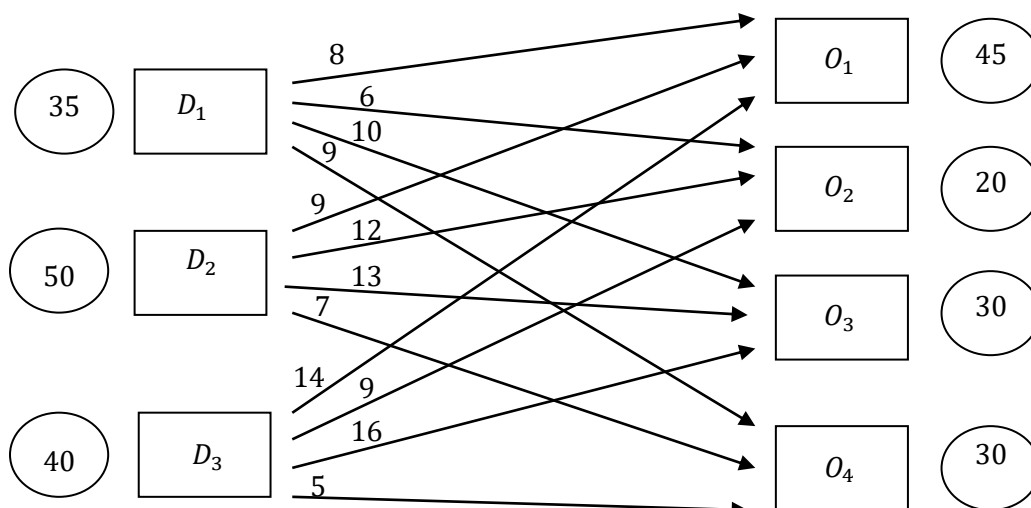
K nalezení optimálního řešení se využívá metoda modifikované distribuce MODI. Ta k nalezení přípustného řešení používá již zmíněnou metodu severozápadního rohu či nějakou heuristiku.

Na následujícím příkladu si vysvětlíme, jak metody fungují. Dále porovnáme nalezené řešení.

Příklad 3.

Je dána dopravní úloha s třemi dodavateli s kapacitami po řadě 35, 50, 40 jednotek výrobků a se čtyřmi odběrateli s požadavky 45, 20, 30 a 30 jednotek výrobků. Náklady na převoz jednotky jsou uvedeny v tabulce. Nalezněte řešení úlohy.

	Požadavky odběratele	45	20	30	30
Kapacita dodavatelů	odběratelé	O_1	O_2	O_3	O_4
	dodavatelé				
35	D_1	8	6	10	9
50	D_2	9	12	13	7
40	D_3	14	9	16	5



Obrázek 6: Grafické znázornění problému.

6.2.1 Metoda severozápadního rohu (SZ rohu)

Tato metoda patří mezi nejjednodušší pro nalezení libovolného přípustného řešení, ale zároveň nezohledňuje velikost nákladů. Název je odvozen z postupu řešení.

Algoritmus začíná v levém horním rohu tabulky. Uspokojíme požadavky odběratele. Jestliže se nevyprázdnila kapacita dodavatele, posuneme se směrem vpravo. V případě, kdy je kapacita prázdná, přecházíme směrem dolů k dalšímu dodavateli, který požadavek odběratele doplní.

	Požadavky odběratele	45-10 0	20 0	30-10 0	30 0
Kapacita dodavatelů	odběratelé	O_1	O_2	O_3	O_4
	dodavatelé				
35 0	D_1	8 35	6	10	9
50-40-20 0	D_2	9 10	12 20	13 20	7
40-30 0	D_3	14	9	16 10	5 30

Celkové náklady jsou:

$$35 \cdot 8 + 10 \cdot 9 + 20 \cdot 12 + 20 \cdot 13 + 10 \cdot 16 + 30 \cdot 5 = 1\,180 \text{ Kč.}$$

6.2.2 Indexová metoda

Indexová metoda oproti metodě SZ rohu uvažuje velikost nákladů na přepravu. Měla by proto poskytnout řešení s nižšími náklady, ale nemusí vrátit řešení optimální.

Postup začíná v buňce s minimální cenou, zde jsme do výše požadavků nebo volných kapacit zdroje. Pokračuje vždy v buňce s nejnižšími náklady, která má dostatek kapacit a požadavků. Jsou-li dvě buňky se stejnými náklady, libovolně zvolme jednu z nich.

	Požadavky odběratele	45-30 0	20-0	30-10 0	30 0
Kapacita dodavatelů	odběratelé	O_1	O_2	O_3	O_4
	dodavatelé				
35-15 0	D_1	8 15	6 20	10	9
50-20 0	D_2	9 30	12	13 20	7
40-10 0	D_3	14	9	16 10	5 30

3. krok 4. krok 2. krok 5. krok 6. krok 1. krok

Celkové náklady jsou:

$$15 \cdot 8 + 20 \cdot 6 + 30 \cdot 9 + 20 \cdot 13 + 10 \cdot 16 + 30 \cdot 5 = 1\,080 \text{ Kč.}$$

6.2.3 Vogelova aproximační metoda (VAM)

K výpočtu zavedeme pojem difference.

Definice 6.2. Difference představuje rozdíl dvou nejnižších neškrtnutých hodnot nákladů v každém sloupci a řádku.

Postup metody:

- určení diferencí ve všech řádcích a sloupcích nákladů c_{ij} ,
- nalezení sloupce či řádku s nejvyšší hodnotou difference a v něm najít minimální cenu c_{ij} . Na této pozici provést uspokojení odběratele v závislosti na kapacitě dodavatele
- přepočítání diferencí zbývajících řádků a sloupců a vrátit se na krok 2.

Výpočet končí vyčerpáním zdroje a uspokojením zákazníka.

	Požadavky odběratele	45 0	20 10 0	30 5 0	30 0	diference
Kapacita dodavatelů	odběratelé dodavatelé	O_1	O_2	O_3	O_4	
35 25 0	D_1	8	6	10	9	2
50 5 0	D_2	9	12	13	7	2 3 4
40 10 0	D_3	14	9	16	5	4 5
diference		1	3 6	3	2	

4. krok
3. krok
2. krok
5. krok
6. krok
1. krok

Celkové náklady jsou:

$$10 \cdot 6 + 25 \cdot 10 + 45 \cdot 9 + 5 \cdot 13 + 10 \cdot 9 + 30 \cdot 5 = 1\,020 \text{ Kč.}$$

6.2.4 Metoda modifikované distribuce (MODI)

V této části si ukážeme, jak jednoduše lze použít k nalezení řešení simplexový algoritmus. Metoda je též známá pod názvem *network simplex*. Pro připomenutí po nalezení řádku obsahujícího pivota se ostatní řádky musely pomocí eliminačních úprav vynulovat. U řešení dopravní úlohy si vystačíme jen s operacemi sčítání a odčítání [12].

Postup metody

Krok 1: Převod na vybilancovaný model pokud model není. Uvažujeme minimalizaci modelu.

Krok 2: Použijte libovolnou metodu z podkapitoly 6.2 k nalezení přípustného řešení.

Krok 3: Položme $u_1 = 0$ a pro alokované buňky tj. buňky, které jsou součástí řešení, použijeme vzorec $u_i + v_j = c_{ij}$ kde u_i jsou řádky,

v_j sloupce

a c_{ij} náklady na přepravu.

Krok 4: Pokud platí pro nealokované buňky, že $u_i + v_j - c_{ij} \leq 0$ tak je toto řešení optimální. V opačném případě nalezneme nejvyšší kladnou hodnotu $\bar{c}_{ij} = u_i + v_j - c_{ij}$ a ta bude novým pivotem. Nalezneme nové přípustné řešení.

Krok 5: Opakujeme krok 3 a 4 dokud není nalezeno optimální řešení.

Vezmeme v úvahu příklad 3. Použijeme výchozí řešení dle metody SZ rohu.

		45	20	30	30
		v_1	v_2	v_3	v_4
35	u_1	8	6	10	9
50	u_2	35	12	13	7
40	u_3	10	20	20	5
		14	9	16	30
				10	

Položme $u_1 = 0$ a pro alokované buňky použijeme vzorec $u_i + v_j = c_{ij}$

$$u_1 + v_1 = 8 \rightarrow v_1 = 8 \qquad u_2 + v_3 = 13 \rightarrow v_3 = 12$$

$$u_2 + v_1 = 9 \rightarrow u_2 = 1 \qquad u_3 + v_3 = 16 \rightarrow u_3 = 4$$

$$u_2 + v_2 = 12 \rightarrow v_2 = 11 \qquad u_3 + v_4 = 5 \rightarrow v_4 = 1$$

Pro nealokované proměnné použijeme $\bar{c}_{ij} = u_i + v_j - c_{ij}$

$$\bar{c}_{12} = 0 + 11 - 6 = 5$$

$$\bar{c}_{24} = 1 + 1 - 7 = -5$$

$$\bar{c}_{13} = 0 + 12 - 10 = 2$$

$$\bar{c}_{31} = 4 + 8 - 14 = -2$$

$$\bar{c}_{14} = 0 + 1 - 9 = -8$$

$$\bar{c}_{32} = 4 + 11 - 9 = 6$$

Vybereme \bar{c}_{32} , náklady se sníží o 6 Kč za jednotku. Vznikne uzavřená cesta zahrnující vstupní proměnnou x_{32} a další základní proměnné $(3,2) - (3,3) -$

(2,3) – (2,2) taková, že alokuje do nealokované buňky co nejvíce množství s největším poklesem nákladů.

		45	20	30	30
		$v_1 = 8$	$v_2 = 11$	$v_3 = 12$	$v_4 = 1$
35	$u_1 = 0$	8 35	6	10	9
50	$u_2 = 1$	9 10	12 20	13 20	7
40	$u_3 = 4$	14	9	16 10	5 30

Protože $x_{22} = 20$ a $x_{33} = 10$ jsou liché buňky, pivot sníží hodnotu x_{33} , x_{22} o 10 a hodnotu x_{23} , x_{32} o 10 zvýší.

		45	20	30	30
		v_1	v_2	v_3	v_4
35	u_1	8 35	6	10	9
50	u_2	9 10	12 10	13 30	7
40	u_3	14	9 10	16	5 30

Opakujeme krok 3 a 4.

Položme $u_1 = 0$

$$u_1 + v_1 = 8 \rightarrow v_1 = 8$$

$$u_2 + v_3 = 13 \rightarrow v_3 = 12$$

$$u_2 + v_1 = 9 \rightarrow u_2 = 1$$

$$u_3 + v_2 = 9 \rightarrow u_3 = -2$$

$$u_2 + v_2 = 12 \rightarrow v_2 = 11$$

$$u_3 + v_4 = 5 \rightarrow v_4 = 7$$

$$\bar{c}_{12} = 0 + 11 - 6 = 5$$

$$\bar{c}_{24} = 1 + 7 - 7 = 1$$

$$\bar{c}_{13} = 0 + 12 - 10 = 2$$

$$\bar{c}_{31} = -2 + 8 - 14 = -8$$

$$\bar{c}_{14} = 0 + 7 - 9 = -2$$

$$\bar{c}_{33} = -2 + 12 - 16 = -6$$

Vstupní proměnná bude x_{12} a další základní proměnné (1,2) – (2,2) – (2,1) – (1,1). Liché buňky jsou x_{11} , x_{22} .

		45	20	30	30
		$v_1 = 8$	$v_2 = 11$	$v_3 = 12$	$v_4 = 7$
35	$u_1 = 0$	8 35	6	10	9
50	$u_2 = 1$	9 10	12 10	13 30	7
40	$u_3 = -2$	14	9 10	16	5 30

Pivot sníží hodnotu x_{22}, x_{11} o 10 a hodnotu x_{12}, x_{21} o 10 zvýší.

		45	20	30	30
		$v_1 = 8$	$v_2 = 6$	$v_3 = 12$	$v_4 = 2$
35	$u_1 = 0$	8	6	10	9
		25	10		
50	$u_2 = 1$	9	12	13	7
		20		30	
40	$u_3 = 3$	14	9	16	5
			10		30

přepočít u_i, v_j

$$u_1 = 0$$

$$u_1 + v_1 = 8 \rightarrow v_1 = 8$$

$$u_2 + v_3 = 13 \rightarrow v_3 = 12$$

$$u_1 + v_2 = 6 \rightarrow v_2 = 6$$

$$u_3 + v_2 = 9 \rightarrow u_3 = 3$$

$$u_2 + v_1 = 9 \rightarrow u_2 = 1$$

$$u_3 + v_4 = 5 \rightarrow v_4 = 2$$

Při výpočtu \bar{c}_{ij} je jediná pozitivní $\bar{c}_{13} = 2$. Vstupní proměnná bude x_{13} uzavřenou cestu tvoří $(1, 3) - (2, 3) - (2, 1) - (1, 1)$. Hodnoty se budou měnit o 25 jednotek.

		45	20	30	30
		$v_1 = 6$	$v_2 = 6$	$v_3 = 10$	$v_4 = 2$
35	$u_1 = 0$	8	6	10	9
			10	25	
50	$u_2 = 3$	9	12	13	7
		45		5	
40	$u_3 = 3$	14	9	16	5
			10		30

$$u_1 = 0$$

$$u_1 + v_2 = 6 \rightarrow v_2 = 6$$

$$u_2 + v_3 = 13 \rightarrow u_2 = 3$$

$$u_1 + v_3 = 10 \rightarrow v_3 = 10$$

$$u_3 + v_2 = 9 \rightarrow u_3 = 3$$

$$u_2 + v_1 = 9 \rightarrow v_1 = 6$$

$$u_3 + v_4 = 5 \rightarrow v_4 = 2$$

Po výpočtu \bar{c}_{ij} platí pro všechny hodnoty, že $\bar{c}_{ij} \leq 0$. Dané řešení je optimální.

Celkové náklady jsou:

$$10 \cdot 6 + 25 \cdot 10 + 45 \cdot 9 + 5 \cdot 13 + 10 \cdot 9 + 30 \cdot 5 = 1\,020 \text{ Kč.}$$

6.3 Dopravní úloha – cirkulace v sítích

Úloha nejlevnější cirkulace představuje zobecnění dopravní úlohy uvedené v části 6.1. S využitím teorie grafů, lze řadu typů optimalizačních úloh řešit pomocí jejich grafické reprezentace. Více se lze dozvědět v [13].

6.3.1 Formulace úlohy

Uvažujme dopravní síť $G = (V, H)$ s ohodnocenými hranami h_{ij} , ve které převážíme zboží při minimálních nákladech.

kde

V jsou vrcholy představující místo,

H jsou hrany, které daná místa spojují,

h_{ij} představují cenu přepravy.

U každého vrcholu V předpokládáme **rovnováhu toku**. To znamená že, množství výrobků, které je na toto místo dopraveno, tak ho také stejné množství opustí.

Dopravní modely lze rozlišit dle počtu stupňů a rozměru.

Počet stupňů v modelu představuje počet vrcholů V , kterými se výrobek přepravuje od dodavatele k odběrateli. V případě přímé realizace přepravy se jedná o **jednostupňovou** úlohu. Když je při cestě k odběrateli ještě mezisklad, je tato úloha **dvoustupňová**.

Počtem rozměrů se rozumí složitost přepravy. **Dvourozměrná** úloha představuje přepravu mezi počátečním a cílovým místem. V **třírozměrné** úloze se navíc objevuje dopravní prostředek pro přepravu zboží.

Definice 6.2. Necht' u_{ij} je kapacita $v_i v_j$,

x_{ij} je množství převážených produktů přes $v_i v_j$,

b_i je požadavek na produkt v v_i ,

c_{ij} jsou náklady na přepravovanou jednotku mezi vrcholy $v_i v_j$,

Poté problém může být formulován

$$\min \sum c_{ij} x_{ij} \quad \forall (i, j) \in H \quad (6.6)$$

za podmínek

$$\sum_j x_{ij} - \sum_j x_{ji} = b_i \quad \forall v_i, v_j \in V \quad (6.7)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in H \quad (6.8)$$

Maticově lze psát

$$\min \mathbf{CX} \quad (6.9)$$

za podmínek

$$\mathbf{AX} = \mathbf{B} \quad (6.10)$$

$$0 \leq \mathbf{X} \leq \mathbf{U} \quad (6.11)$$

kde matice \mathbf{A} je matice sousednosti sítě $\mathbf{G} = (\mathbf{V}, \mathbf{H})$.

Věta 6.3. (O stromech).

Sloupce matice \mathbf{A} tvoří bázi právě tehdy, když tvoří kostru grafu původního grafu \mathbf{G} .

Metoda Out-of-kilter

Tato metoda řeší problém toku sítě s minimálními náklady. Na začátku máme k dispozici sadu toků a duálních proměnných. Po každé iteraci se zkoumá optimalita hran v síti. Hrany, které splňují podmínky, náleží *kilteru*. Ostatní se musí do *kilteru* dostat pomocí změny toků a duálních proměnných. Algoritmus končí, jestliže všechny hrany náleží *kilteru*.

6.3.2 Celočíslnost dopravní úlohy

Definice 6.4. Necht' $\vec{\mathbf{G}}$ je orientovaný a neobsahující smyčky. Matice $\mathbf{A}(\vec{\mathbf{G}})$ velikosti n/m dána

$$a_{ij} = \begin{cases} 1, & \text{když } v_i \text{ počátečním vrcholem hrany } h_j, \\ -1, & \text{když } v_i \text{ koncovým vrcholem hrany } h_j, \\ 0 & \text{jinak,} \end{cases}$$

se nazývá **incidenční matice** orientovaného grafu $\vec{\mathbf{G}}$.

Věta 6.5. Každá incidenční matice orientovaného grafu $\vec{\mathbf{G}}$ je totálně unimodulární.

Důsledek 6.6. Každá dopravní úloha obsahuje unimodulární matici, a proto je výsledné řešení vždy celočíselné [6].

6.4 Multiperiodická úloha

Oproti modelu u podkapitoly 6.1 řešíme problém požadavků n odběratelů na každé období T . Tímto problémem se zabýval Bellmore, Eklof a Nemhauser (B.E.N) [14]. Předpokladem je, že A_i, c_{ij}, o_j jsou funkce času. Převoz od m dodavatelů k odběratelům lze uskutečnit ve stejném období jako požadavek na poptávku. Tomu odpovídají náklady $c_{ijt}r_{ijt}$, nebo v některém případě mohou vzniknout náklady na skladování zásob. Necht' m představuje stav zásob z období

$(t - 1)$ do období t skladu j , d_j jsou náklady na skladování zásob, s_{jt} představuje kapacitu skladu mezi časy $(t - 1)$ a t . Poté lze problém formulovat jako:

$$\min \sum_{i,j} \sum_{t=1}^T c_{ijt} r_{ijt} + \sum_{j=1}^n \sum_{t=2}^T d_{jt} m_{jt} \quad (6.12)$$

za podmínky

$$\sum_{j=1}^n r_{ijt} \leq A_{it}, \forall i, t \quad (6.13)$$

$$\sum_{i=1}^m r_{ijt} + m_{jt} - m_{j(t+1)} \geq o_{jt}, \forall j, t \quad (6.14)$$

$$r_{ij} \geq 0 \quad (6.15)$$

$$0 \leq m_{jt} \leq s_{jt}. \quad (6.16)$$

7 STOCHASTICKÉ PROGRAMOVÁNÍ

Stochastické programování se zabývá situacemi, kdy jsou parametry problému popsány pomocí náhodných proměnných. Nejčastěji se objevuje při řešení reálných situacích, kdy je těžké určit přesné hodnoty parametrů nabízeného a požadovaného množství produktů. Základní myšlenkou je převod pravděpodobnostního problému na deterministický. V této části se využije myšlenky pravděpodobnostního omezení [11], [15]. Model obecně bude definován následovně:

$$\max \sum_{j=1}^n c_j x_j \quad (7.1)$$

$$P \left\{ \sum_{j=1}^n a_{ij} x_j \leq b_i \right\} \geq 1 - \alpha_i \quad i = 1, 2, \dots, m, x_j \geq 0 \forall j. \quad (7.2)$$

Pravděpodobnostní omezení vyplývá z omezení $\sum_{j=1}^n a_{ij} x_j \leq b_i$, které je realizováno s minimální pravděpodobností z $(1 - \alpha_i, 0 < \alpha_i < 1)$. Obecně předpokládáme, že c_j, a_{ij} a b_i jsou náhodné proměnné. Skutečnost, že c_j je náhodná proměnná, může být nahrazena střední hodnotou. V následujícím textu jsou uvedeny případy, kdy a_{ij} a b_i jsou náhodné proměnné. Dále se předpokládá, že parametry jsou z normálního rozdělení a jsou použity statistické veličiny průměr a výběrový rozptyl.

Případ 1.

Předpokladem je, že a_{ij} je z normálního rozdělení se střední hodnotou $E\{a_{ij}\}$ a rozptylem $Var\{a_{ij}\}$. Dále kovariance a_{ij} a $a_{i'j'}$ je $Cov\{a_{ij}, a_{i'j'}\}$.

Pro i -té omezení platí, že

$$P\left\{\sum_{j=1}^n a_{ij} x_j \leq b_i\right\} \geq 1 - \alpha_i. \quad (7.3)$$

Dále definujeme

$$h_i = \sum_{j=1}^n a_{ij} x_j \quad (7.4)$$

Potom h_i má normální rozdělení s parametry

$$E\{h_i\} = \sum_{j=1}^n E\{a_{ij}\} x_j \text{ a } Var\{h_i\} = \mathbf{X}^T \mathbf{D}_i \mathbf{X} \quad (7.5)$$

kde $\mathbf{X} = (x_1, x_2, \dots, x_n)^T$

$$\mathbf{D}_i = \begin{pmatrix} Var\{a_{i1}\} & \dots & Cov\{a_{i1}, a_{in}\} \\ \vdots & & \vdots \\ Cov\{a_{in}, a_{i1}\} & \dots & Var\{a_{in}\} \end{pmatrix}, \quad (7.6)$$

\mathbf{D}_i je kovarianční matice náhodného vektoru a_{ij} .

Nyní

$$P\{h_i \leq b_i\} = P\left\{\frac{h_i - E\{h_i\}}{\sqrt{Var\{h_i\}}} \leq \frac{b_i - E\{h_i\}}{\sqrt{Var\{h_i\}}}\right\} \geq 1 - \alpha_i \quad (7.7)$$

kde $(h_i - E\{h_i\}) / \sqrt{Var\{h_i\}}$ je normované normální rozdělení s $E\{h_i\} = 0, Var\{h_i\} = 1$, tzn. $h_i \sim N(0,1)$. To znamená, že

$$P\{h_i \leq b_i\} = \Phi\left(\frac{b_i - E\{h_i\}}{\sqrt{Var\{h_i\}}}\right), \quad (7.8)$$

kde Φ je distribuční funkce normovaného normálního rozdělení.

Nechť K_{α_i} je kvantil normálního rozdělení a platí pro něj

$$\Phi(K_{\alpha_i}) = 1 - \alpha_i \quad (7.9)$$

Poté $P\{h_i \leq b_i\} \geq 1 - \alpha_i$ je realizováno právě tehdy, když

$$\frac{b_i - E\{h_i\}}{\sqrt{Var\{h_i\}}} \geq K_{\alpha_i} \quad (7.10)$$

To poskytuje následující nelineární omezení, které je ekvivalentní s původním stochastickým omezením.

$$\sum_{j=1}^n E\{a_{ij}\} x_j + K_{\alpha_i} \sqrt{\mathbf{X}^T \mathbf{D}_i \mathbf{X}} \leq b_i \quad (7.11)$$

Pro speciální případ, kdy jsou hodnoty nezávislé, $Cov\{a_{ij}, a_{i'j'}\} = 0$, kde $i \neq j$ a poslední omezení se redukuje na

$$\sum_{j=1}^n E\{a_{ij}\} x_j + K_{\alpha_i} \sqrt{\sum_{j=1}^n Var\{a_{ij}\} x_j^2} \leq b_i \quad (7.12)$$

Toto omezení můžeme nyní vyjádřit pomocí separabilního programování a použít substituci

$$y_i = \sqrt{\sum_{j=1}^n Var\{a_{ij}\} x_j^2} \text{ pro } \forall i. \quad (7.13)$$

Z toho vyplývá, že původní omezení je ekvivalentní s

$$\sum_{j=1}^n E\{a_{ij}\} x_j + K_{\alpha_i} y_i \leq b_i \quad (7.14)$$

a

$$\sum_{j=1}^n Var\{a_{ij}\} x_j^2 - y_i^2 = 0, \quad (7.15)$$

kde $y_i \geq 0$.

Případ 2.

V tomto případě je pouze b_i z normálního rozdělení se střední hodnotou $E\{b_i\}$ a rozptylem $Var\{b_i\}$. Postup je velmi podobný případu 1. Uvažujme stochastické omezení ve tvaru

$$P \left\{ b_i \geq \sum_{j=1}^n a_{ij} x_j \right\} \geq \alpha_i \quad (7.16)$$

Dále jak v případě 1,

$$P \left\{ \frac{b_i - E\{b_i\}}{\sqrt{Var\{b_i\}}} \geq \frac{\sum_{j=1}^n a_{ij} x_j - E\{b_i\}}{\sqrt{Var\{b_i\}}} \right\} \geq \alpha_i \quad (7.17)$$

To je splněno pouze pokud

$$\frac{\sum_{j=1}^n a_{ij} x_j - E\{b_i\}}{\sqrt{Var\{b_i\}}} \leq K_{\alpha_i} \quad (7.18)$$

Z toho vyplývá, že stochastické omezení je ekvivalentní s deterministickým lineárním omezením

$$\sum_{j=1}^n a_{ij} x_j \leq E\{b_i\} + K_{\alpha_i} \sqrt{Var\{b_i\}} \quad (7.19)$$

Poznámka 7.1. V případě 2 lze model s pravděpodobnostními omezeními převést na ekvivalentní lineární problém.

Rozdělení **extremálních hodnot** má také dva parametry. α , která znázorňuje místo a parametr měřítka β . Rozdělení vzniká jako omezení pro maximum a minimum vzorku nezávislých a identicky distribuovaných náhodných proměnných *iid.*, v případě velkého sběru náhodných dat. Funkce hustoty extrémálního rozdělení pro maxima se známými parametry α, β je dána:

$$f(x) = \frac{1}{\beta} e^{-\left(\frac{x-\alpha}{\beta}\right)} e^{-e^{-\left(\frac{x-\alpha}{\beta}\right)}}; -\infty < x < \infty; \alpha, \beta > 0 \quad (7.20)$$

Předpokládejme, že a_i jsou nezávislé náhodné proměnné s parametry α_i, β_i , kde $\gamma_i \in (0; 1)$. Odvození podle [16].

$$P(\sum_{j=1}^n x_{ij} \leq b_i) \geq 1 - \gamma_i, i=1, 2, \dots, m \quad (7.21)$$

S funkcí hustoty

$$f(b_i) = \frac{1}{\beta_i} e^{-\left(\frac{b_i-\alpha_i}{\beta_i}\right)} e^{-e^{-\left(\frac{b_i-\alpha_i}{\beta_i}\right)}}; -\infty < b_i < \infty; \alpha_i, \beta_i > 0 \quad (7.22)$$

Pravděpodobnostní omezení lze psát jako

$$\int_{\sum_{j=1}^n x_{ij}}^{\infty} f(b_i) d(b_i) \geq 1 - \gamma_i \quad (7.23)$$

integrál (7.23) lze napsat ve tvaru:

$$\int_{\sum_{j=1}^n x_{ij}}^{\infty} \frac{1}{\beta_i} e^{-\left(\frac{b_i-\alpha_i}{\beta_i}\right)} e^{-e^{-\left(\frac{b_i-\alpha_i}{\beta_i}\right)}} d(b_i) \geq 1 - \gamma_i \quad (7.24)$$

Nechť $z = e^{-\left(\frac{b_i-\alpha_i}{\beta_i}\right)}$

Potom lze výraz (7.24) přepsat do tvaru:

$$\int_0^e e^{-z} d(z) \geq 1 - \gamma_i \quad (7.25)$$

Po zintegrování (7.25) dostáváme

$$- [e^{-z}]_0^e \geq 1 - \gamma_i \quad (7.26)$$

Po uspořádání získáme

$$e^{-e} \leq \gamma_i \quad (7.27)$$

Dvakrát zlogaritmuje celý výraz (7.27).

$$-\left(\frac{\sum_{j=1}^n x_{ij} - \alpha_i}{\beta_i}\right) \geq [\ln\{-\ln(\gamma_i)\}] \quad (7.28)$$

Dalšími zjednodušujícími úpravami dosáhneme

$$\sum_{j=1}^n x_{ij} - \alpha_i \leq -\beta_i [\ln\{-\ln(\gamma_i)\}] \quad (7.29)$$

Zde lze pravděpodobnostní výraz (7.25) transformovat na deterministický jako

$$\sum_{j=1}^n x_{ij} \leq \alpha_i - \beta_i [\ln\{-\ln(\gamma_i)\}] \quad (7.30)$$

8 ROBUSTNÍ PROGRAMOVÁNÍ

V této kapitole se zaměříme na modifikovaný postup lineárního programování, známý jako robustní programování. Tento postup je imunní proti jakýmkoliv stochastickým vychýlením oproti původnímu deterministickému modelu [17], [18]. Pro připomenutí definujeme lineární model jako:

$$\min z = \sum_{j=1}^n c_j x_j \quad (8.1)$$

za podmínek

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m \quad (8.2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n \quad (8.3)$$

Problémem v původním modelu je, že malá změna nákladů může způsobit velkou změnu výsledného řešení, tzn. úloha je špatně podmíněna. Z tohoto důvodu se v robustním modelu do účelové funkce doplní odchylka δ . Model robustního programování vypadá:

$$\min z = \sum_{j=1}^n (c_j + \delta_j^c) x_j \quad (8.4)$$

za podmínek:

$$\sum_{j=1}^n (a_{ij} + \delta_{ij}^a) x_j \leq b_i + \delta_i^b, \quad i = 1, 2, \dots, m \quad (8.5)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n \quad (8.6)$$

odpovídající deterministická hodnota z původního lineárního modelu pak leží v intervalu $\langle a_{ij} - \delta_{ij}^a; a_{ij} + \delta_{ij}^a \rangle$. Odchylka δ je náhodná proměnná, která zaznamenává nejistotu, odchylku měření a podobně.

Robustní programování lze použít v zde zkoumaném problému distribuce zboží. Proměnná δ může představovat náklady na dopravu, například náhlá změna trasy transportu z důvodu zhoršení dopravní situace.

9 SLOŽITĚJŠÍ DOPRAVNÍ PROBLÉM

V jedné z předchozích částí jsme se zabývali pouze přepravou jednoho druhu produktu od dodavatele k odběrateli. Tyto úlohy jsou poměrně rozsahově malé. Většina reálných případů mají stovky až tisíce omezení. Mohou přibýt nová omezení například:

- Několik druhů produktů se vyrábí ve více výrobnách a jsou přepravovány do několika skladů.
- Výroba má být naplánována na dané období dopředu.
- Vysoký požadavek na dodání produktů a vznik problému s přepravou produktu.

9.1 Software pro nalezení řešení

V případě komplikovanější úlohy, obsahující někdy až tisíce proměnných, je možné využít software.

V našem případě se využije modelovací jazyk AMPL. Pomocí tohoto systému lze řešit rozsáhlé optimalizační úlohy. Podporuje rychlý vývoj a spolehlivost výsledků. Používá algebraickou reprezentaci k popisování optimalizačních modelů, proto je s ním dobré pracovat [19].

Umožňuje přístup k několika řešitelům:

- Lineární CPLEX, Gurobi, Xpress.
- Nelineární CONOPT, MINOS, SNOPT, Bonmin, Couenne.

CPLEX patří mezi nejznámější a nejrozšířenější. Řeší lineární a kvadratickou optimalizaci, dále smíšené celočíselné programování. Využívá algoritmy pro základní a duální simplex, lineární tok sítě, heuristiku, řezné problémy a další. CPLEX je vhodný na řešení velmi velkých optimalizačních problémů v řádech milionu proměnných a omezení.

Gurobi využívá výhod procesorů o více jádrech a proto je oproti CPLEX rychlejší. Řeší například lineární a smíšené celočíselné lineární programování, dále konvexní a nekonvexní kvadratické programování, kvadraticky omezené programování a další.

XPRESS řeší lineární a kvadratickou optimalizaci včetně jejich celočíselných verzí.

9.2 Implementace

Výše uvedené modely byly implementovány v modelovacím jazyce AMPL. Tato sekce popíše programový kód algoritmu pro řešení zde uvedeného problému v tomto jazyce.

Za prvé se definují proměnné pro počty jednotlivých entit. Veškeré počty musí být kladné. To se v AMPL definuje takto:

```
param I > 0;  
param J > 0;  
param P > 0;  
param T > 0;
```

Pro jednotlivé výroby, sklady a produkty se vytvoří lineární vektor, který obsahuje pro každý prvek danou hodnotu. Také je založen vektor pro každý den distribuce produktů.

```
set ORIG = 1..I;  
set DEST = 1..J;  
set PROD = 1..P;  
set PERIOD = 1..T;
```

Dále je potřeba definovat cesty mezi výrobny a sklady:

```
set LINKS = {ORIG, DEST};
```

a také podmínky nezápornosti počátečních a průběžných zásob a požadavků.

```
param supply {ORIG, PROD, PERIOD} >= 0;  
param demand {DEST, PROD, PERIOD} >= 0;  
param init {DEST, PROD} >= 0;  
var zasoby {DEST, PROD, 0..T} >= 0;
```

Poslední podmínky jsou na nezápornost ceny cesty a distribuce.

```
param cost {LINKS} >= 0;  
var Trans {LINKS, PROD, PERIOD} >= 0;
```


Účelová funkce byla zapsána v AMPL jak je uvedeno níže. Jedná se o přepis výrazu (8.1).

```
minimize Total_Cost:
    sum {t in PERIOD, p in PROD, (i,j) in LINKS} cost[i,j] *
    Trans[i,j,p,t];
```

Model je omezen podmínkami, které jsou následující:

```
subject to Supply {t in PERIOD, p in PROD, i in ORIG}:
    sum {(i,j) in LINKS} Trans[i,j,p,t] = supply[i,p,t];

subject to Demand {t in PERIOD, p in PROD, j in DEST}:
    sum {(i,j) in LINKS} Trans[i,j,p,t] >= demand[j,p,t] -
    zasoby[j,p,t-1];

subject to Zasoby {t in PERIOD, p in PROD, j in DEST}:
    zasoby[j,p,t] = zasoby[j,p,t-1] - demand[j,p,t] + sum
    {(i,j) in LINKS} Trans[i,j,p,t];

subject to Init {p in PROD, j in DEST}:
    zasoby[j,p,0] = init[j,p];
```

Tyto podmínky zajišťují aby:

- Bylo rozvezeno vše, co bylo vyrobeno.
- Bylo rozvezeno alespoň to, co je poptáváno bez zásob z předcházejícího dne.
- Zásoby byly rovny zásobám z předchozího dne se zohledněním přepravovaného zboží a poptávky

Poslední podmínka je tzv. *okrajová*, tedy definuje zásoby prvního dne na skladu pevnými hodnotami.

9.3 Dopravní úloha v praxi

Tento problém je inspirován skutečným případem plánování rozvozu zboží. Kde je znám počáteční stav zásob. Objednávky jsou dány jen na pár dní předem, poté už se využívá předpověď objednávek odvozených z předešlých let v daném období. Musí být dopředu na 1-2 dny naplánované převozy, aby se pro ně zajistila přeprava. Zboží, které není poptáváno, zůstává na skladu jako zásoba. Vznikají tím další náklady na uskladnění. V realitě také může nastat situace, kdy požadované zboží není dodáno do dané lhůty, a tak vznikají náklady za penále.

Úkolem je zjistit, zda lze jednoduše vyřešit tento problém přes deterministický a stochastický model, dále porovnat získané řešení.

Data byla definována počtem

- výroben I ,
- skladů J ,
- produktů P ,
- dnů T ,

kde $I, J, P, T > 0$.

Dále známe stav skladu na počátku dne, denní produkci a ceny jednotlivých převozů. Předpoklady jsou neomezenost kapacity skladů a převozu. Pracujeme s vyváženým modelem.

9.3.1 Deterministické řešení

Poté sestavíme deterministický model jako:

$$\min \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I c_{ij} \sum_{p=1}^P r_{pijt} \quad (9.1)$$

za podmínek

$$\left(\begin{array}{l} o_{pjt} \leq m_{pjt} \\ m_{pjt} = m_{pj(t-1)} + \sum_{i=1}^I r_{pij(t-1)} \quad \forall p, j, t = 2, \dots, T + 1 \end{array} \right) \quad (9.2)$$

$$\sum_{j=1}^J r_{pijt} = p_{pi(t-1)} \quad \forall p, i, t = 1, \dots, T + 1 \quad (9.3)$$

$$m_{pjt}; r_{pijt} \geq 0 \quad (9.4)$$

kde

o_{pjt} je požadavek na odběr produktu p ve skladu j v den t (odběry na počátku dne),

m_{pjt} představuje počet kusů produktu p ve skladu j na počátku dne t ,

p_{pit} produkce komodit ve výrobně i v den (lze rozvést až do $(t + 1)$ – ního dne),

$\sum_{p=1}^P r_{pijt}$ počet přepravených druhů p z i do j v den t .

9.3.2 Stochastické řešení – normální rozdělení

Zde se předpokládá, že požadavek na odběr $o_{pjt} \sim N(E\{o_{pjt}\}, Var\{o_{pjt}\})$.

Kde v podmínce (9.6) je použit vzorec (7.19). Potom lze stochastické model zapsat ve tvaru:

$$\min \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I c_{ij} \sum_{p=1}^P r_{pijt} \quad (9.5)$$

za podmínek

$$E\{o_{pjt}\} + K_{\alpha_i} \sqrt{Var\{o_{pjt}\}} \leq m_{pjt} \quad (9.6)$$

$$\left(m_{pjt} = m_{pj(t-1)} + \sum_{i=1}^I r_{pij(t-1)} \quad \forall p, j, t = 2, \dots, T+1 \right) \quad (9.7)$$

$$\sum_{j=1}^J r_{pijt} = p_{pi(t-1)} \quad \forall p, i, t = 1, \dots, T+1 \quad (9.8)$$

$$m_{pjt}; r_{pijt} \geq 0$$

9.3.3 Stochastické řešení – rozdělení extrémálních hodnot

Oproti předchozímu modelu nastala změna v (9.10), kde se liší kvantilová funkce daných rozdělení.

$$\min \sum_{t=1}^T \sum_{j=1}^J \sum_{i=1}^I c_{ij} \sum_{p=1}^P r_{pijt} \quad (9.9)$$

za podmínek

$$\alpha_i - \beta_i [\ln\{-\ln(\gamma_i)\}] \leq m_{pjt} \quad (9.10)$$

$$\left(m_{pjt} = m_{pj(t-1)} + \sum_{i=1}^I r_{pij(t-1)} \quad \forall p, j, t = 2, \dots, T+1 \right) \quad (9.11)$$

$$\sum_{j=1}^J r_{pijt} = p_{pi(t-1)} \quad \forall p, i, t = 1, \dots, T+1 \quad (9.8)$$

$$m_{pjt}; r_{pijt} \geq 0$$

9.3.4 Praktické řešení

Daný problém byl spuštěn na PC s procesorem Intel Xeon CPU E5-2630 v 2.6GHz a 16GB RAM pamětí pod Windows 10. AMPL byl spuštěn ve verzi 3.5.0.201802211250, řešiče byly ve verzích CPLEX 12.9.0.0, Gurobi 9.0.0 a XPRESS 8.6.0 (34.01.02).

Jelikož získaná deterministická data ze společnosti nebyla úplná, neexistence některého produktu ve výrobě či skladu, chybějící náklady na převoz a jiné, byla připravena data, která jsou na modelování vyhovující. Zároveň byla zachována velikost jednotlivých objektů. Výroben a skladů jsou řádově desítky, druhů produktů stovky a období je na 1 měsíc. Data pro stochastická rozdělení byla domodelována z parametrů stochastických modelů za pomoci programovacího jazyka MATLAB R2019a. Byly zafixovány hodnoty pro počet výroben na 10 a na 30 dní. Měnily se počty skladů v rozmezí 10 - 90 po kroku 20 a počty produktů v rozmezí 100 - 900 s krokem 200.

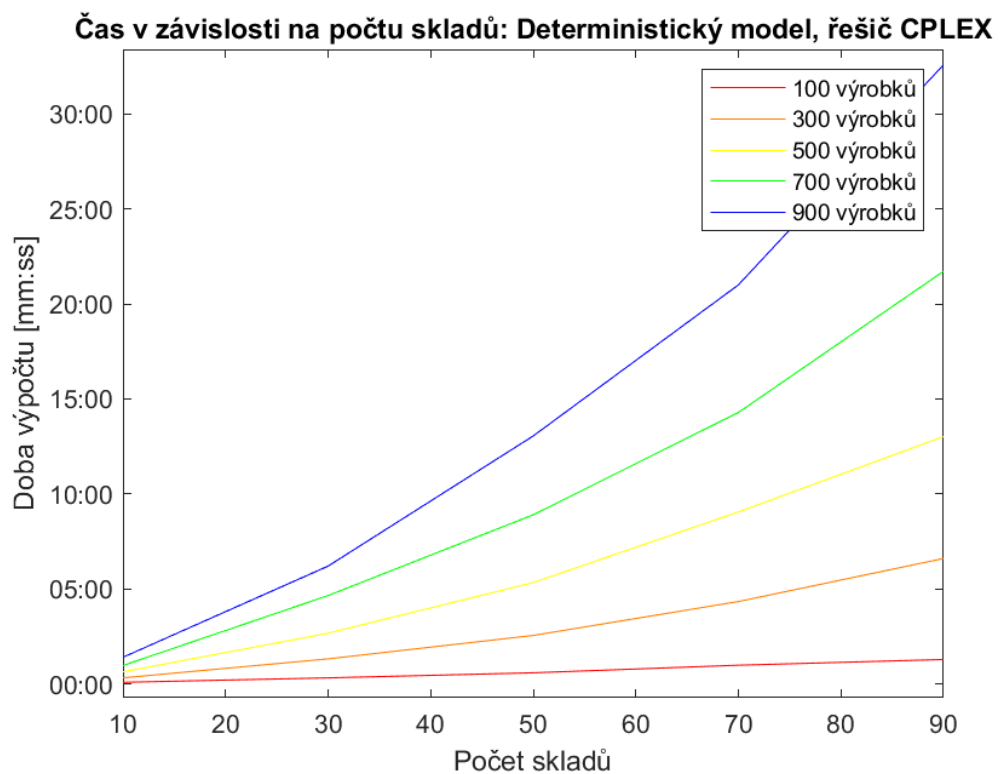
Deterministické a stochastické přístupy byly řešeny třemi řešiči CPLEX, Gurobi a XPRESS. Řešiče byly vždy spuštěny se svým standardním nastavením. Při každém spuštění bylo ve všech případech nalezeno optimální řešení. Následující tabulka znázorňuje dobu výpočtů v sekundách.

Produkty x sklady	Deterministické			Stochastické normální			Stochastické extrémní		
	CPLEX	GUROBI	XPRESS	CPLEX	GUROBI	XPRESS	CPLEX	GUROBI	XPRESS
100x10	5	3	5	4	2	3	5	2	3
100x30	19	5	12	19	5	11	22	5	10
100x50	35	9	21	38	10	21	54	9	21
100x70	59	15	34	68	14	33	111	13	35
100x90	77	17	49	103	18	51	163	18	48
300x10	19	5	9	18	5	9	16	5	9
300x30	79	17	39	79	18	39	99	17	39
300x50	153	30	76	173	31	77	261	34	85
300x70	260	51	140	320	47	126	531	55	145
300x90	396	70	190	521	69	203	521	69	203
500x10	38	9	16	32	9	16	33	8	17
500x30	160	30	69	180	32	70	210	30	69
500x50	320	56	136	380	388	148	507	195	445
500x70	543	101	239	693	93	231	988	82	232
500x90	781	124	328	1054	109	318	1731	119	313
700x10	58	15	23	54	13	24	52	13	23
700x30	279	46	101	290	44	101	332	42	99
700x50	534	86	225	719	173	208	901	82	219
700x70	857	125	340	1082	130	342	1908	136	370
700x90	1303	199	492	2203	571	521	3228	170	493
900x10	84	16	35	81	17	35	77	32	144
900x30	372	62	142	436	61	141	510	60	140
900x50	783	112	294	1176	145	340	1587	141	321
900x70	1260	211	493	1931	222	496	2610	188	453
900x90	1955	450	756	3085	302	730	4435	314	764

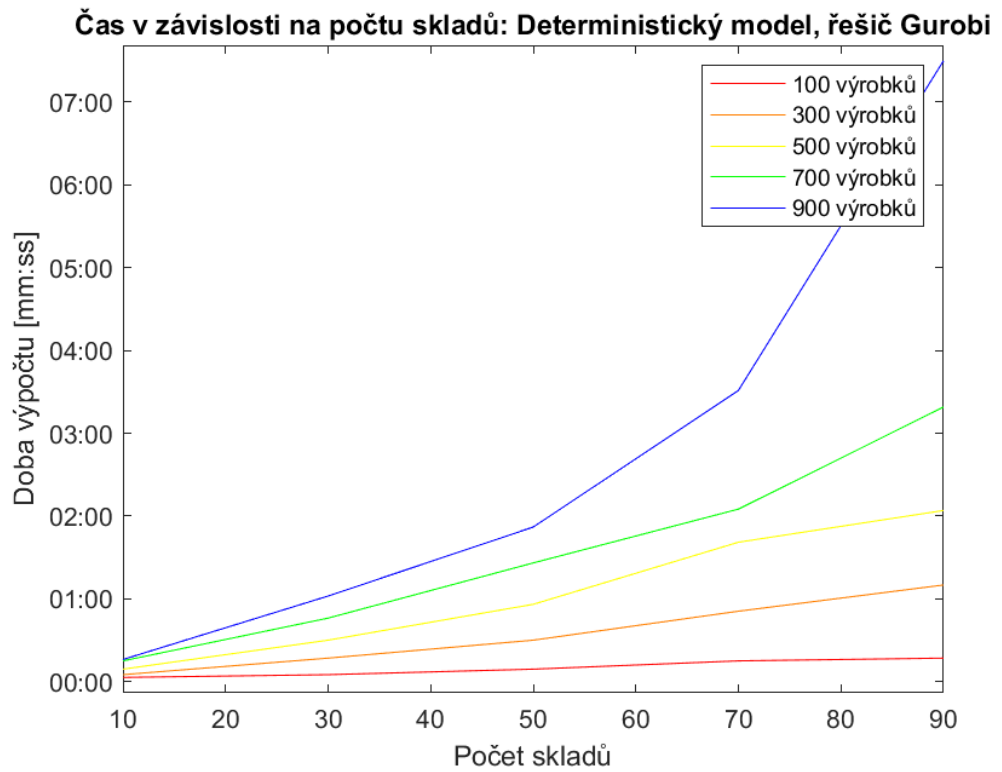
Tabulka 4: Výsledné řešení problému

Z výsledků vyplývá, že s narůstajícím počtem skladů a produktů roste také délka běhu programu. Z testovaných řešičů si nejlépe vedl Gurobi, který ve většině případů našel řešení rychleji než CPLEX (přibližně 6x) nebo XPRESS (přibližně 2.3x). Ze všech nejhůře dopadl řešič CPLEX, který byl přibližně 2.7x pomalejší než XPRESS.

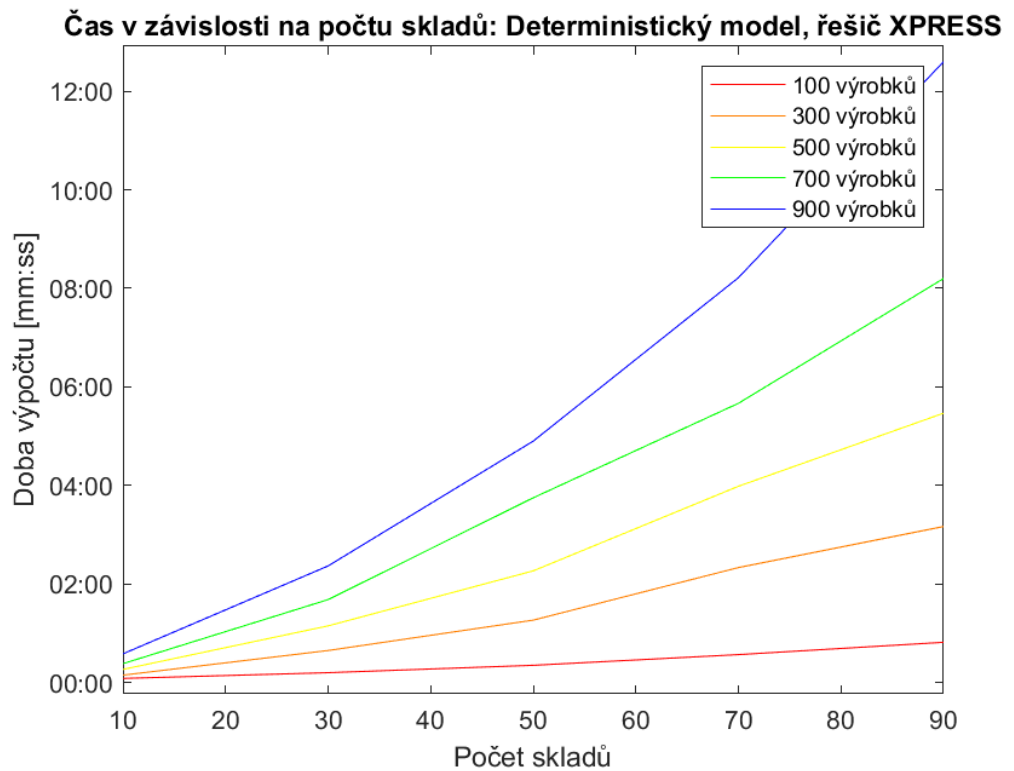
Na ukázkou je na obrázcích u deterministického přístupu znázorněn růst času vlivem zvětšování počtu skladů a produktů. Z počátku se pohybujeme v řádech jednotek sekund, zatímco pro 90 skladů (odběratelů) požadujících 900 různých druhů produktů se může doba výpočtu protáhnout do minut či hodin. U stochastického rozdělení tomu tak nebylo. Ne vždy platilo, že s nárůstem počtu skladů a výrobků se zvýší délka běhu algoritmu oproti předchozímu.



Obrázek 6: Doba běhu řešiče CPLEX nad deterministickými daty vzhledem k počtu skladů.



Obrázek 7: Doba běhu řešiče Gurobi nad deterministickými daty vzhledem k počtu skladů.



Obrázek 8: Doba běhu řešiče XPRESS nad deterministickými daty vzhledem k počtu skladů.

ZÁVĚR

Cílem práce bylo naimplementovat v programovacím jazyce AMPL dopravní úlohu velikostně odpovídající reálnému problému. K tomu se použila deterministická a stochastická data vygenerována pro normální a extrémní rozdělení pomocí programovacího jazyka MATLAB. Všechny úlohy po ukončení algoritmu došly k optimálnímu řešení. Další zkoumání v podobě časové náročnosti této úlohy různými přístupy a řešiči dosáhlo následujícího poznatku: k doporučení na efektivní řešení praktické přepravní úlohy by byl řešič Gurobi, který vyřešil největší daný problém v řádově jednotkách sekund, oproti tomu nejpomaleji získal řešení CPLEX, který běžel déle než hodinu.

LITERATURA

- [1] DANTZIG G. B., THAPA M. N. *Linear Programming*. New York: Springer-Verlag, 1997. ISBN 978-0-387-9433-1.
- [2] PLEVNÝ M., ŽIŽKA M. *Modelování a optimalizace v manažerském rozhodování*. 2. vydání – 2. dotisk. Plzeň ZČU, 2010. ISBN 978-80-7043-933-3.
- [3] JABLONSKÝ J. *Operační výzkum – kvantitativní modely pro ekonomické rozhodování*. 3. vydání. Praha: Professional Publishing, 2007. ISBN 978-80-86946-44-3.
- [4] LAGOVÁ M., JABLONSKÝ J. *Lineární modely*. Praha VŠE, 2014. ISBN 978-80-245-2020-9.
- [5] BOYD S. a VANDENBERGHE L. *Convex Optimization*. Cambridge University Press, 2005. ISBN-13-978-0521833783.
- [6] Přednáška z předmětu Teorie grafů, optimalizace a složitost 2, Západočeská univerzita v Plzni, [online]. [cit. 20. 7. 2020]. Dostupné z http://najada.fav.zcu.cz/~ryjacek/KMA_TGD2.htm
- [7] FERNÁNDEZ GONZÁLEZ V. ZELAIA JAUREGI A. (2011) "*Ikerkuntza Operatiboa. Programazio Lineala*", kapitola 2. [cit. 2020-07-11] Dostupné z: https://ocw.ehu.eus/pluginfile.php/8160/mod_resource/content/1/2_Simplex.pdf
- [8] FERNÁNDEZ GONZÁLEZ V. ZELAIA JAUREGI A. (2011) "*Ikerkuntza Operatiboa. Programazio Lineala*", kapitola 6. [cit. 2020-07-11] Dostupné z: https://ocw.ehu.eus/pluginfile.php/8164/mod_resource/content/1/6_Integer.pdf
- [9] LAWLER E. *Combinatorial Optimization Networks and Matroids*. Mineola: Dover publications, 2001. ISBN 0-486-41453-1.
- [10] CHEUNG R. K., POWELL W. B., *Models and Algorithms for Distribution Problems with Uncertain Demands*. Transportation Science 30:43-59, 1996.

- [11] TAHA H. A. *Operations Research: An Introduction*. 4. ed. New York: Macmillan Publishing Company, 1989. ISBN: 0024189405.
- [12] WINSTON W. L. *Operations Research: Applications and Algorithms*. 3rd ed. Belmont: Duxbury press, 1994. ISBN 0534209718.
- [13] FOULDS, L. R. *Graph Theory Applications* [online]. New York, NY: Springer New York, 1992 [cit. 2020-07-13]. Universitext. ISBN 978-0-387-97599-3.
- [14] BOOKBINDER J. H. a SETHI S. P. The dynamic transportation problem: a survey. *Naval Research Logistics Quarterly* 1980.
- [15] MAGGIONI F., BERTOCCHI M. E. a POTRA F. A. *Stochastic versus Robust Optimization for a Transportation problem*. *Optimization Online*, 2015.
- [16] MAHAPATRA D. R., ROY S. K., BISWAL M. P. *Multi-choice stochastic transportation problem involving extreme value distribution*, *Applied Mathematical Modelling*, 2013. ISSN 0307-904X. Dostupné z: <https://doi.org/10.1016/j.apm.2012.04.024>
- [17] BEN-TAL A., EL-GHAOUY L. a NEMIROVSKI A. *Robust optimization*. Princeton University Press, 2009. ISBN 978-0-691-14368-2.
- [18] HLAVATÝ R. – BROŽOVÁ H. *Robust optimization approach in transportation problem*. In 35th International conference Mathematical methods in economics, MME 2017, Conference proceedings 13. 09. 2017, Hradec Králové. Hradec Králové: University of Hradec Králové, 2017. s. 225-230.
- [19] AMPL - streamlined modeling for real optimization [online]. [cit. 20. 7. 2020]. Dostupné z: <https://ampl.com/products/ampl/>