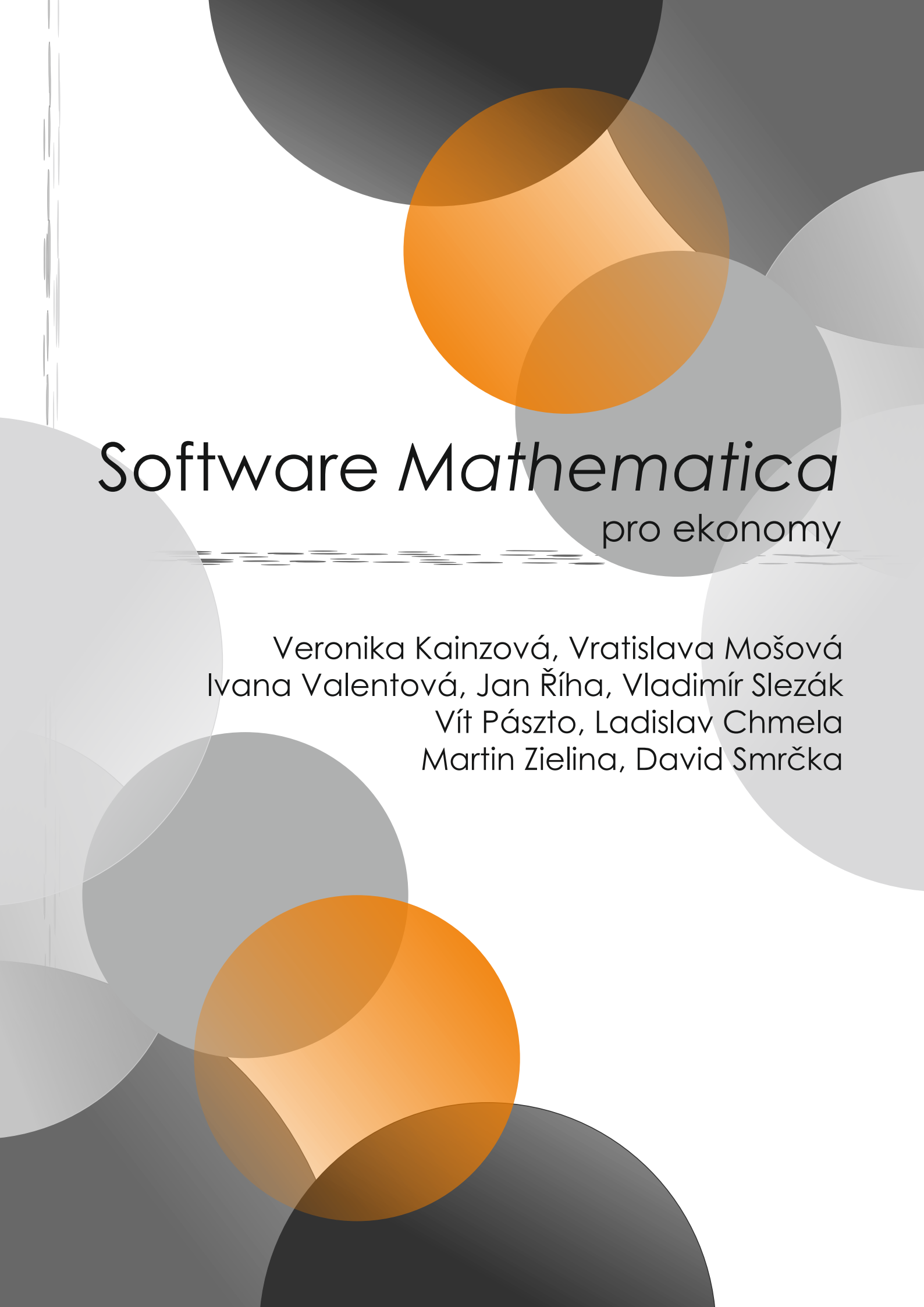




Software *Mathematica*

pro ekonomy

Veronika Kainzová, Vratislava Mošová
Ivana Valentová, Jan Říha, Vladimír Slezák
Vít Pászto, Ladislav Chmela
Martin Zielina, David Smrčka



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Moderní přístup k aplikaci matematických dovedností v přírodovědných a ekonomických oborech
reg. č.: CZ.1.07/2.2.00/28.0168

Software Mathematica pro ekonomy

Veronika Kainzová, Vratislava Mošová, Ivana Valentová, Jan Říha, Vladimír Slezák, Vít Pászto,
Ladislav Chmela, Martin Zielina, David Smrčka

Olomouc 2014

Oponenti: RNDr. Vladimíra Mádrová, CSc.
Mgr. Vladimír Vaněk, Ph.D.

Mgr. Veronika Kainzová, Ph.D., a kolektiv

Software Mathematica pro ekonomy

Publikace určena pro studenty Přírodovědecké fakulty UP a Moravské vysoké školy Olomouc.

Zpracováno v rámci řešení projektu Evropského sociálního fondu a Ministerstva školství, mládeže a tělovýchovy České republiky

Moderní přístup k aplikaci matematických dovedností v přírodovědných a ekonomických oborech, reg. č. CZ.1.07/2.2.00/28.0168.

Výkonný redaktor

Odpovědná redaktorka

Technická redakce autor

Návrh a grafická úprava obálky Mgr. Zdeněk Pucholt

Vydala a vytiskla Univerzita Palackého v Olomouci

Křížkovského 8, 771 46 Olomouc

<http://www.upol.cz/vup>

email: vup@upol.cz

Olomouc 2014

Neprodejné

Obsah

1. Řešení rozhodovacích problémů pomocí SW *Mathematica* 2

- 1.1 Teoretický úvod 2
- 1.2 Modelový příklad s komentářem - výběr televizoru 3
- 1.3 Modelový příklad - výběr nejvýhodnějšího dopravního prostředku 9
- 1.4 Modelový příklad - výběr nejlepšího města k bydlení 13
- 1.5 Modelový příklad - výběr nejlepšího dárku pro kamarádku 17
- 1.6 Modelový příklad - výběr brokera pro retailovou investiční činnost 21
- 1.7 Modelový příklad - výběr pračky 24

2. Import dat z MS Excel a tvorba grafů 26

- 2.1 Import tabulky MS Excel 27
- 2.2 Tvorba sloupcového grafu z importovaných dat 29
- 2.3 Další možnosti při práci s grafy 32
- 2.4 Závěr 38

3. Investice 40

- 3.1 Hodnocení efektivnosti investic 40
- 3.2 Vstupní údaje pro hodnocení efektivnosti investic 41
- 3.3 Statické metody hodnocení investic 43
- 3.4 Dynamické metody hodnocení investic 45

4. Řízení zásob 49

- 4.1 Dodávkový cyklus zásob 49
- 4.2 Modely řízení zásob 50

5. Ekonomická analýza investice do decentralizovaného zdroje energie 55

- 5.1 Bilance zdroje elektrické energie - malé vodní elektrárny 55
- 5.2 Bilance spotřeby elektrické energie - v malé chatě 57
- 5.3 Ekonomická analýza investice do decentralizovaného zdroje energie 58

6. Věžňovo dilema 59

- 6.1 Hra s nulovým a nenulovým součtem 59
- 6.2 Společné rysy her 60
- 6.3 Popis věžňova dilematu 60
- 6.4 Turnaj R. Axelroda ve věžňovu dilematu 61
- 6.5 Popis vybraných strategií věžňova dilematu 62
- 6.6 Demostrace věžňova dilematu 63
- 6.7 Nejlepší strategie věžňova dilematu 67
- 6.8 Věžňovo dilema v praxi 68
- 6.9 Závěr 70

7. Finanční matematika 71

- 7.1 Umořování dluhu 71
- 7.2 Kontokorentní účet 73
- 7.3 Vývoj cen na trhu 75

8. Optimalizační úlohy 77

8.1 Optimalizace výroby 77

8.2 Optimalizace investic 78

8.3 Řízení zásobování 79

9. Použitá literatura 81

1 Řešení rozhodovacích problémů pomocí SW Mathematica

Mnohé rozhodovací problémy z běžného života můžeme s výhodou řešit pomocí matematických metod. Nejčastěji jde o tzv. **vícekritériální rozhodovací procesy**, kdy z množiny **variant** vybíráme tu optimální na základě několika **kritérií**. Tato kritéria přitom mohou mít různé váhy.

V této kapitole ukážeme postup řešení rozhodovacích úloh dvěma metodami, které jsou vhodné pro využití SW *Mathematica*, a to **metodou bazické varianty** a **metodou lineárních dílčích funkcí utility**.

1.1 Teoretický úvod

Vícekritériálním rozhodováním (s konečnou množinou variant) se rozumí následující problém:

Je dána n -prvková množina variant V , které jsou posuzovány dle m stanovených hledisek (kritérií) z množiny K . Úkolem je vybrat z dané množiny variant V variantu, která je nejlepší (optimální) vzhledem ke kritériím z množiny K .

Kritéria hodnocení představují hlediska zvolená rozhodovatelem sloužící k posouzení výhodnosti jednotlivých variant z hlediska dosažení cíle rozhodování. Rozeznáváme kritéria **výnosového typu** (vyšší hodnoty preferovány před nižšími - např. zisk) a kritéria **nákladového typu** (nižší hodnoty preferovány před vyššími - např. náklady). Dále rozlišujeme kritéria kvantitativní (hodnoty vyjádřeny číselně) a kvalitativní (jejich důsledky vyjádřeny slovně).

Při hodnocení variant rozhodovacího problému podle různých hledisek (kritérií) je přirozené vypočítat celkové hodnocení (utilitu, užitek) variant jako průměr hodnocení dílčích, která musí být určitým způsobem standardizována, aby byla sčitatelná. Mají-li jednotlivá hlediska hodnocení rozdílnou důležitost, pak aritmetický průměr je nahrazen průměrem váženým (resp. váženým součtem).

Vícekritériální funkce utility (užitková funkce) u - přiřazuje každé variantě rozhodování utilitu (užitek, ohodnocení) vyjádřenou reálným číslem. Čím je toto číslo větší, tím více rozhodovatel danou variantu rozhodování preferuje. Varianta x , pro kterou bude hodnota užitkové funkce $u(x)$ maximální, se stává variantou optimální a je "vítězkou" našeho rozhodovacího procesu.

Konstrukce vícekritériální funkce utility za jistoty je v obecném případě obtížná, proto se v praxi nejčastěji pracuje s aditivním tvarem této funkce, který lze vyjádřit následující formulí:

$$u(x) = \sum_{j=1}^m v_j \cdot u_j(x) = v_1 \cdot u_1(x) + v_2 \cdot u_2(x) + \dots + v_m \cdot u_m(x)$$

Význam jednotlivých symbolů pro $j = 1, 2, \dots, m$:

$u(x)$ - hodnota vícekritériální funkce utility, neboli celkové hodnocení dané varianty vzhledem k celému souboru kritérií,

v_j - nezáporná normovaná váha kritéria k_j (viz dále),

$u_j(x)$ - hodnota dílčí funkce utility u_j pro variantu x , neboli dílčí hodnocení varianty x vzhledem ke kritériu k_j ,

Jednotlivé metody vícekritériálního hodnocení variant se odlišují způsobem stanovení vah kritérií a způsobem určení dílčích hodnocení $u_j(x)$.

Dílčí funkce utility u_j vyjadřují změnu ohodnocení (přínosu pro rozhodovatele) v závislosti na změnách hodnoty daného kritéria hodnocení (změnách důsledků variant vzhledem k tomuto kritériu). Pro kritéria výnosového typu (kritéria s rostoucí preferencí) je odpovídající dílčí funkce utility vždy rostoucí, přičemž může být konvexní, konkávní nebo lineární.

Konkávni rostoucí dílčí funkce utility odpovídá situaci, kdy rozhodovatel cení stejné přírůstky hodnot daného kritéria stále méně (přírůstky dílčí utility pro stejné velké přírůstky daného kritéria klesají). Konvexní rostoucí dílčí funkce utility zobrazuje naopak situaci, kdy stejné přírůstky hodnot daného kritéria znamenají pro rozhodovatele stále větší přínos (přírůstky dílčí utility pro stejné velké přírůstky daného kritéria rostou).

Pro kritéria nákladového typu (kritéria s klesající preferencí) je odpovídající dílčí funkce utility vždy klesající, a to konkávni (konvexní) v případě, že rozhodovatel cení stejné poklesy hodnot daného kritéria stále více (méně).

V praxi jsou dílčí funkce utility často lineární. V tomto případě znamenají pro rozhodovatele stejné přírůstky (u rostoucí dílčí funkce utility), resp. stejné poklesy (u klesající dílčí funkce utility) hodnot daného kritéria vždy stejný přínos.

Definičním oborem dílčích funkcí utility jsou intervaly hodnot jednotlivých kritérií.

Dílčí funkce utility je zvykem normovat tak, že nabývají hodnot z intervalu mezi 0 a 1. Pro nejhorší hodnotu daného kritéria nabývá dílčí funkce utility hodnotu 0 a pro nejlepší hodnotu kritéria nabývá hodnoty 1.

Důležitou roli ve výpočtu hodnot vícekritériální hodnotící funkce hrají **váhy kritérií**, což jsou nezáporná reálná čísla $v_1, v_2, \dots, v_m$, která vyjadřují rozdílnou významnost jednotlivých kritérií vzhledem k celkovému hodnocení variant. Většina metod vícekritériálního hodnocení pracuje s normovanými vahami, pro které platí $v_1 + v_2 + \dots + v_m = 1$ (lze je vyjádřit v procentech).

Máme-li stanoveny nenormované váhy w_j , pak normované váhy v_j z nich vypočteme podle vzorce

$$v_j = \frac{w_j}{w_1 + w_2 + \dots + w_m}, \quad j = 1, 2, \dots, m$$

Z několika možných metod stanovení vah zvolíme pro naše účely **Saatyho metodu**, která využívá párového srovnávání kritérií a je podrobněji popsána dále.

Pro vlastní vícekritériálního hodnocení variant pak zvolíme dvě metody, které jsou vhodné pro využití SW *Mathematica*, a to **metodu bazické varianty** a **metodu lineárních dílčích funkcí utility** (metodu univerzální standardizace). Princip těchto metod je popsán v rámci komentovaného řešení ukázkového příkladu v následující podkapitole.

1.2 Modelový příklad s komentářem - výběr televizoru

Nyní bude vyřešen jeden modelový příklad řešení rozhodovacího problému (vícekritériálního hodnocení variant) i s komentářem. Algoritmus je univerzální a funguje pro řešení naprosto reálných rozhodovacích úloh z běžného života.

Stačí na vstupu zadat (resp. přepsat stávající) údaje o konkrétní rozhodovací situaci:

- počet kritérií a počet variant
- názvy (slovní popisy) kritérií a názvy variant
- typy kritérií (nákladové nebo výnosové)
- hodnoty jednotlivých kritérií pro všechny varianty
- párové srovnání preference kritérií

Další ukázkové příklady jsou pak vyřešeny v následující podkapitole.

Jako ukázkový příklad užití metod vícekritériálního hodnocení poslouží výběr nejlepšího televizoru ze tří vybraných variant - Samsung, Philips, Thomson.

Jako kritéria rozhodování jsme zvolili cenu (v Kč), úhlopříčku (v cm), jas (v cd/m^2) a barvu. První tři kritéria jsou kvantitativní (příčemž cena je nákladové kritérium, úhlopříčka a jas jsou výnosová kritéria), barva je kvalitativní. V tomto případě přidělíme jednotlivým barvám body z nějaké bodové škály, abychom mohli s hodnotami tohoto kritéria dále počítat.

Nejprve zadáme počet kritérií a počet variant rozhodovacího problému:

```
m = 4; (* zadejte počet kritérií *)
```

```
n = 3; (* zadejte počet variant *)
```

Dále zadáme názvy kritérií $k[1]$ až $k[m]$:

```
k[1] = "cena(Kč)"; (* zadejte název prvního kritéria *)
```

```
k[2] = "úhlopříčka(cm)"; (* zadejte název druhého kritéria *)
```

```
k[3] = "jas(cd/m²)"; (* zadejte název třetího kritéria *)
```

```
k[4] = "barva(body)"; (* zadejte název čtvrtého kritéria *)
```

Dále zadáme názvy variant $v[1]$ až $v[n]$:

```
v[1] = "Samsung"; (* zadejte název první varianty *)
```

```
v[2] = "Philips"; (* zadejte název druhé varianty *)
```

```
v[3] = "Thomson"; (* zadejte název třetí varianty *)
```

Zadáme typy jednotlivých kritérií - vložíme "nak" pro nákladové kritérium resp. "vyn" pro výnosové kritérium.

```
typkrit[1] = "nak"; (*zadejte typ prvního kritéria*)
```

```
typkrit[2] = "vyn"; (*zadejte typ druhého kritéria*)
```

```
typkrit[3] = "vyn"; (*zadejte typ třetího kritéria*)
```

```
typkrit[4] = "vyn"; (*zadejte typ čtvrtého kritéria*)
```

Nyní je třeba zadat hodnoty jednotlivých kritérií pro všechny varianty. Musíme tedy zjistit ceny všech tří televizorů, délky jejich úhlopříček, hodnoty jasu a barvy. Zadáváme ve tvaru $\text{hod}[i, j]$ - např. $\text{hod}[1,2]$ znamená číselnou hodnotu prvního kritéria pro druhou variantu, v našem případě cenu TV Philips. V případě kvalitativního kritéria (barva) přidělujeme jednotlivým hodnotám body z nějaké bodové škály.

```
hod[1, 1] = 14 998; (* zadejte hodnotu prvního kritéria na první variantě *)
```

```
hod[1, 2] = 19 990; (* zadejte hodnotu prvního kritéria na druhé variantě *)
```

```
hod[1, 3] = 13 998; (* zadejte hodnotu prvního kritéria na třetí variantě *)
```

```
hod[2, 1] = 66; (* zadejte hodnotu druhého kritéria na první variantě *)
```

```
hod[2, 2] = 95; (* zadejte hodnotu druhého kritéria na druhé variantě *)
```

```
hod[2, 3] = 82; (* zadejte hodnotu druhého kritéria na třetí variantě *)
```

```
hod[3, 1] = 450; (* zadejte hodnotu třetího kritéria na první variantě *)
```

```
hod[3, 2] = 500; (* zadejte hodnotu třetího kritéria na druhé variantě *)
```

```
hod[3, 3] = 450; (* zadejte hodnotu třetího kritéria na třetí variantě *)
```

```
hod[4, 1] = 1; (* zadejte hodnotu čtvrtého kritéria na první variantě *)
```

```
hod[4, 2] = 0.5; (* zadejte hodnotu čtvrtého kritéria na druhé variantě *)
```

```
hod[4, 3] = 0.5; (* zadejte hodnotu čtvrtého kritéria na třetí variantě *)
```

Sestavíme přehlednou tabulku všech hodnot kritérií pro jednotlivé varianty:

```
tabhodnot = Table[hod[i, j], {i, 1, m}, {j, 1, n}];
```

```
TableForm[tabhodnot, TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	Samsung	Philips	Thomson
cena(Kč)	14 998	19 990	13 998
úhlopříčka(cm)	66	95	82
jas(cd/m^2)	450	500	450
barva(body)	1	0.5	0.5

STANOVENÍ VAH KRITERIÍ

Posledními vstupními údaji, které je třeba k provedení výpočtu optimální varianty zadat, jsou preference vah jednotlivých kritérií. Při použití Saatyho metody se zadávají párově, pro všechny možné dvojice z m prvků (z počtu variant).

Hodnotami této preferenční funkce $s[i,j]$ mohou být pouze čísla 3, 5, 7, 9, a to v případě, že i -té kritérium je (postupně) slabě, dosti, prokazatelně, absolutně významnější než j -té kritérium. Je-li tomu naopak (j -té je významnější než i -té), zadá se příslušná převrácená hodnota. Jsou-li obě kritéria stejně významná, zadá se hodnota 1. Např. zadáme-li $s[1,2] = 5$, znamená to, že první kritérium je pro nás dosti významnější než druhé. Zadáme-li $s[3,4] = 1/3$, znamená to, že čtvrté kritérium je pro nás slabě významnější než třetí.

Zadání preference vah:

```
s[1, 2] = 5;
s[1, 3] = 7;
s[1, 4] = 9;
s[2, 3] = 3;
s[2, 4] = 5;
s[3, 4] = 5;
s[i_, j_] := 1 /; i == j
s[i_, j_] := 1 / s[j, i] /; i > j
```

Nyní bude sestavena Saatyho matice pro stanovení vah kritérií:

```
S = Table[s[i, j], {i, 1, m}, {j, 1, m}];
"S = " MatrixForm[S] (*Saatyho matice stanoveni vah*)
```

$$S = \begin{pmatrix} 1 & 5 & 7 & 9 \\ \frac{1}{5} & 1 & 3 & 5 \\ \frac{1}{7} & \frac{1}{3} & 1 & 5 \\ \frac{1}{9} & \frac{1}{5} & \frac{1}{5} & 1 \end{pmatrix}$$

Lze ukázat (Věta Perron-Frobeniova), že každá takto sestavená matice má maximální reálné vlastní číslo $\lambda > 0$ a odpovídající vlastní vektor má všechny složky kladné.

Normované váhy kritérií stanovíme jako složky normovaného vlastního vektoru matice S odpovídajícího jejímu maximálnímu vlastnímu číslu.

Nejprve tedy zapíšeme matici pro výpočet vlastních čísel matice S :

```
S - λ * IdentityMatrix[4] // MatrixForm (*matice pro vypocet char. polynomu*)
```

$$\begin{pmatrix} 1 - \lambda & 5 & 7 & 9 \\ \frac{1}{5} & 1 - \lambda & 3 & 5 \\ \frac{1}{7} & \frac{1}{3} & 1 - \lambda & 5 \\ \frac{1}{9} & \frac{1}{5} & \frac{1}{5} & 1 - \lambda \end{pmatrix}$$

Její determinant je charakteristickým polynomem matice S :

```
CharacteristicPolynomial[S, λ]
```

$$-\frac{6656}{4725} - \frac{392\lambda}{75} - 4\lambda^3 + \lambda^4$$

Vypočítáme vlastní čísla matice S (kořeny charakteristického polynomu):

```
N[Eigenvalues[S], 6]
```

```
{4.30034, -0.022234 + 1.131242 i, -0.022234 - 1.131242 i, -0.255876}
```

Dále vlastní vektory příslušné vlastním číslům:

N[Eigenvectors[S], 6]

```
{ {15.9452, 4.93924, 2.70405, 1.00000},
  {-1.09259 + 13.77544 i, -4.03137 + 0.67000 i, -0.47280 - 2.66681 i, 1.00000},
  {-1.09259 - 13.77544 i, -4.03137 - 0.67000 i, -0.47280 + 2.66681 i, 1.00000},
  {-20.0671, 8.94069, -4.07165, 1.00000} }
```

Vybereme vlastní vektor příslušný maximálnímu reálnému vlastnímu číslu a normujeme jej (součet normovaných vah musí být roven 1). Dostaneme tak normované váhy kritérií:

```
realvlvektor = N[Eigenvectors[S], 6][[1]];
normarealvlvektoru = Total[realvlvektor];
vahy = (1 / normarealvlvektoru) * realvlvektor
```

(* Normované váhy kritérií *)

```
{0.64848, 0.200876, 0.109972, 0.0406695}
```

Poznámka:

Normované váhy kritérií lze získat ze Saatyho preferenční matice přímo:

```
vahyprimo = (1 / Total[N[Eigenvectors[S], 6][[1]]]) * (N[Eigenvectors[S], 6][[1]])
{0.64848, 0.200876, 0.109972, 0.0406695}
```

V dalším kroku doplníme původní tabulku hodnot kritérií pro jednotlivé varianty o sloupec s vahami kritérií:

```
pridejsloupec[A_, b_] := Transpose[Join[Transpose[A], {b}]]
tabulkasvahami = pridejsloupec[tabhodnot, vahyprimo];
TableForm[tabulkasvahami,
  TableHeadings -> {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}]}]
```

	Samsung	Philips	Thomson	váhy
cena (Kč)	14 998	19 990	13 998	0.64848
úhlopříčka (cm)	66	95	82	0.200876
jas (cd/m ²)	450	500	450	0.109972
barva (body)	1	0.5	0.5	0.0406695

METODA BAZICKÉ VARIANTY

Bazická varianta je fiktivní varianta sestavená jako vektor nejlepších nebo předem zvolených (požadovaných) hodnot kritérií na daném souboru variant. Z každého řádku základní tabulky hodnot tedy vybereme nejlepší hodnotu daného kritéria a ta bude tvořit složku vektoru - bazické varianty.

```
minmax[i_] := If[typkrit[i] == "vyn", Max, Min];
bazickavarianta = Table[minmax[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}]
```

(* Bazická varianta *)

```
{13998, 95, 500, 1}
```

Do základní tabulky hodnot kritérií nyní doplníme bazickou variantu:

```

tabulkasbazvar = pridejsloupec[tabulkasvahami, bazickavarianta];
TableForm[tabulkasbazvar, TableHeadings →
  {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}, {"bazická varianta"}]}]

```

	Samsung	Philips	Thomson	váhy	bazická varianta
cena (Kč)	14 998	19 990	13 998	0.64848	13 998
úhlopříčka (cm)	66	95	82	0.200876	95
jas (cd/m ²)	450	500	450	0.109972	500
barva (body)	1	0.5	0.5	0.0406695	1

Metoda bazické varianty je založena na stanovení dílčích ohodnocení variant vzhledem k jednotlivým kritériím pomocí porovnání hodnot důsledků variant vždy s hodnotami bazické varianty.

Celkové hodnocení varianty x popsané vektorem $(x_1, x_2, \dots, x_m)$ naměřených hodnot kritérií $k_1, \dots, k_m$ je při použití této metody dáno formulí

$$u(x) = \sum_{j=1}^m v_j \cdot u_j(x_j), \quad \sum_{j=1}^m v_j = 1, \quad v_j \geq 0, \quad j = 1, \dots, m$$

Hodnota j -té dílčí funkce utility (j -té dílčí hodnocení) je dáno vztahem

$$u_j(x_j) = \frac{x_j}{x_j^b} \quad \text{v případě kritéria s rostoucí preferencí (výnosového typu), resp.}$$

$$u_j(x_j) = \frac{x_j^b}{x_j} \quad \text{v případě kritéria s klesající preferencí (nákladového typu),}$$

přičemž vektor $(x_1^b, x_2^b, \dots, x_m^b)$ představuje bazickou variantu.

Dílčí funkce utility u pro kritéria výnosového typu jsou tedy lineární, pro kritéria nákladového typu jsou lineární lomené.

Tato metoda standardizace dílčích kritérií odstraňuje vliv rozdílných jednotek měření použitých pro jednotlivá kritéria.

Nyní zpět k příkladu. Provedeme výpočet dílčích hodnocení (s promítnutím normovaných vah kritérií) a výsledky zapíšeme opět do přehledné tabulky:

```

u[p_, q_] := If[typkrit[p] == "vyn", vahyprimo[[p]] * hod[p, q] / bazickavarianta[[p]],
    vahyprimo[[p]] * bazickavarianta[[p]] / hod[p, q]];

```

```

tabulkahodnoceni = Table[u[p, q], {p, 1, m}, {q, 1, n}];
TableForm[tabulkahodnoceni,
  TableHeadings → {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]

```

	Samsung	Philips	Thomson
cena (Kč)	0.60524	0.454100	0.64848
úhlopříčka (cm)	0.139556	0.200876	0.173388
jas (cd/m ²)	0.098975	0.109972	0.098975
barva (body)	0.0406695	0.0203347	0.0203347

Celkové hodnocení každé z variant (hodnotu užité funkce u) dostaneme jako vážený součet dílčích hodnocení:

```

celkhod[j_] := Total[Table[u[p, j], {p, 1, m}]];
celkovehodnoceni = Table[celkhod[j], {j, 1, n}];
TableForm[celkovehodnoceni, TableHeadings → {Table[v[j], {j, 1, n}]}]

```

Samsung	0.88444
Philips	0.785283
Thomson	0.94118

Optimální variantou pak bude ta, jejíž celkové hodnocení je největší:

```
Max[celkovehodnoceni];
indexviteze = Max[Table[If[celkhod[j] == Max[celkovehodnoceni], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze]}], "Output"]]
```

Optimální varianta: Thomson

METODA LINEÁRNÍCH DÍLČÍCH FUNKCÍ UTILITY

Tato metoda vychází z předpokladu, že všechny dílčí funkce utility mají lineární tvar. Tyto funkce se stanoví tak, že nejhorší hodnotě j -tého kritéria (na daném souboru variant) x_j^0 se přiřadí dílčí utilita 0, nejlepší hodnotě x_j^1 dílčí utilita 1 a spojnice těchto bodů jsou pak zobrazením lineárních dílčích funkcí utility. Dílčí hodnotící funkce pro variantu $x = (x_1, x_2, \dots, x_m)$ je definována vztahem

$$u_j(x_j) = \frac{x_j - x_j^0}{x_j^1 - x_j^0}$$

V našem ukázkovém příkladu budeme postupovat dále tak, že do základní tabulky hodnot přidáme místo bazické varianty vektor nejhorších a nejlepších hodnot daného kritéria, a rovněž rozdíl těchto hodnot:

```
minmax2[i_] := If[typkrit[i] == "vyn", Min, Max];
nejhorsihodnoty = Table[minmax2[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];
nejlepsihodnoty = bazickavarianta;
rozdil = nejlepsihodnoty - nejhorsihodnoty;

tabulkasextremy = pridejsloupec[tabulkasvahami, nejhorsihodnoty];
tabulkasextremy2 = pridejsloupec[tabulkasextremy, nejlepsihodnoty];
tabulkasextremy3 = pridejsloupec[tabulkasextremy2, rozdil];
TableForm[tabulkasextremy3, TableHeadings -> {Table[k[i], {i, 1, m}],
Join[Table[v[j], {j, 1, n}], {"váhy"}, {"nejhorší"}, {"nejlepší"}, {"rozdíl"}]}]
```

	Samsung	Philips	Thomson	váhy	nejhorší	nejlepší	rozdíl
cena (Kč)	14 998	19 990	13 998	0.64848	19 990	13 998	-!
úhlopříčka (cm)	66	95	82	0.200876	66	95	29
jas (cd/m ²)	450	500	450	0.109972	450	500	50
barva (body)	1	0.5	0.5	0.0406695	0.5	1	0

V dalším kroku již provedeme výpočet dílčích hodnocení, která vynásobíme vahami kritérií:

```
u2[p_, q_] := vahyprimo[p] * (hod[p, q] - nejhorsihodnoty[p]) / rozdil[p];
tabulkahodnoceni2 = Table[u2[p, q], {p, 1, m}, {q, 1, n}];
TableForm[tabulkahodnoceni2,
TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	Samsung	Philips	Thomson
cena (Kč)	0.54026	0	0.64848
úhlopříčka (cm)	0	0.200876	0.110828
jas (cd/m ²)	0	0.109972	0
barva (body)	0.0406695	0.	0.

Zbývá pouze dílčí hodnocení každé varianty podle všech kritérií sečíst a dostaneme hodnocení celkové:

```
celkhod2[j_] := Total[Table[u2[p, j], {p, 1, m}]];
celkovehodnoceni2 = Table[celkhod2[j], {j, 1, n}];
TableForm[celkovehodnoceni2, TableHeadings -> {Table[v[j], {j, 1, n}]}]
```

Samsung	0.580927
Philips	0.310849
Thomson	0.75931

Optimální variantou je opět ta s nejvyšším celkovým hodnocením:

```
Max[celkovehodnoceni2];
indexviteze2 = Max[Table[If[celkhod2[j] == Max[celkovehodnoceni2], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze2]}], "Output"]]
```

Optimální varianta: Thomson

1.3 Modelový příklad - výběr nejvýhodnějšího dopravního prostředku

Rozhodovací problém - výběr nejvýhodnějšího dopravního prostředku na trase Třebíč - Olomouc mezi variantami vlak, auto, autobus.

Nejprve zadáme všechny potřebné vstupní údaje jako v ukázkovém příkladě:

```

m = 4; (*zadejte počet kritérií*)
n = 3; (*zadejte počet variant*)

k[1] = "cena(Kč)"; (*zadejte název prvního kritéria*)
k[2] = "vzdálenost(km)"; (*zadejte název druhého kritéria*)
k[3] = "čas(min)"; (*zadejte název třetího kritéria*)
k[4] = "dostupnost(body)"; (*zadejte název čtvrtého kritéria*)

v[1] = "vlak"; (*zadejte název první varianty*)
v[2] = "autobus"; (*zadejte název druhé varianty*)
v[3] = "auto"; (*zadejte název třetí varianty*)

typkrit[1] = "nak"; (*zadejte typ prvního kritéria*)
typkrit[2] = "nak"; (*zadejte typ druhého kritéria*)
typkrit[3] = "nak"; (*zadejte typ třetího kritéria*)
typkrit[4] = "vyn"; (*zadejte typ čtvrtého kritéria*)

hod[1, 1] = 123; (*zadejte hodnotu prvního kritéria na první variantě*)
hod[1, 2] = 145;
hod[1, 3] = 405;
hod[2, 1] = 163;
hod[2, 2] = 145;
hod[2, 3] = 135;
hod[3, 1] = 208;
hod[3, 2] = 190;
hod[3, 3] = 87;
hod[4, 1] = 7;
hod[4, 2] = 3;
hod[4, 3] = 5; (*zadejte hodnotu čtvrtého kritéria na třetí variantě*)

(* Zadání preferencí vah kritérií *)
(* 1...stejně významná, 3...slabě významnější, 5...dosti významnější,
7...prokazatelně významnější, 9...absolutně významnější *)

s[1, 2] = 7;
s[1, 3] = 1 / 3;
s[1, 4] = 3;
s[2, 3] = 1 / 9;
s[2, 4] = 1 / 5;
s[3, 4] = 5;
s[i_, j_] := 1 /; i == j
s[i_, j_] := 1 / s[j, i] /; i > j

```

Sestavíme přehlednou tabulku všech hodnot kritérií na jednotlivých variantách:

```

tabhodnot = Table[hod[i, j], {i, 1, m}, {j, 1, n}];
TableForm[tabhodnot, TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]

```

	vlak	autobus	auto
cena(Kč)	123	145	405
vzdálenost(km)	163	145	135
čas(min)	208	190	87
dostupnost(body)	7	3	5

Dále stanovíme váhy kritérií Saatyho metodou:

```
S = Table[s[i, j], {i, 1, m}, {j, 1, m}];
```

```
MatrixForm[S]
```

```
(*Saatyho matice stanoveni vah*)
```

$$\begin{pmatrix} 1 & 7 & \frac{1}{3} & 3 \\ \frac{1}{7} & 1 & \frac{1}{9} & \frac{1}{5} \\ 3 & 9 & 1 & 5 \\ \frac{1}{3} & 5 & \frac{1}{5} & 1 \end{pmatrix}$$

```
vahyprimo = (1 / Total[N[Eigenvalues[S], 6][[1]]]) * (N[Eigenvalues[S], 6][[1]])
```

```
(* váhy kritérií *)
```

```
{0.267418, 0.0398755, 0.56601, 0.126698}
```

Do základní tabulky hodnot přidáme váhy:

```
pridejsloupec[A_, b_] := Transpose[Join[Transpose[A], {b}]]
```

```
tabulkasvahami = pridejsloupec[tabhodnot, vahyprimo];
```

```
TableForm[tabulkasvahami,
```

```
TableHeadings -> {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}]}]
```

	vlak	autobus	auto	váhy
cena (Kč)	123	145	405	0.267418
vzdálenost (km)	163	145	135	0.0398755
čas (min)	208	190	87	0.56601
dostupnost (body)	7	3	5	0.126698

Daný rozhodovací problém řešíme metodou bazické varianty:

```
minmax[i_] := If[typkrit[i] == "vyn", Max, Min];
```

```
bazickavarianta = Table[minmax[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}]
```

```
(* bazická varianta *)
```

```
{123, 135, 87, 7}
```

Bazickou variantu přidáme do tabulky:

```
tabulkasbazvar = pridejsloupec[tabulkasvahami, bazickavarianta];
```

```
TableForm[tabulkasbazvar, TableHeadings ->
```

```
{Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}, {"bazická varianta"}]}]
```

	vlak	autobus	auto	váhy	bazická varianta
cena (Kč)	123	145	405	0.267418	123
vzdálenost (km)	163	145	135	0.0398755	135
čas (min)	208	190	87	0.56601	87
dostupnost (body)	7	3	5	0.126698	7

Vypočítáme dílčí hodnocení všech variant:

```
u[p_, q_] := If[typkrit[p] == "vyn", vahyprimo[[p]] * hod[p, q] / bazickavarianta[[p]],  
vahyprimo[[p]] * bazickavarianta[[p]] / hod[p, q]];
```

```
tabulkahodnoceni = Table[u[p, q], {p, 1, m}, {q, 1, n};
```

```
TableForm[tabulkahodnoceni,
```

```
TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	vlak	autobus	auto
cena (Kč)	0.267418	0.226844	0.081216
vzdálenost (km)	0.0330257	0.0371255	0.0398755
čas (min)	0.236744	0.259172	0.56601
dostupnost (body)	0.126698	0.054299	0.090499

Na závěr vypočteme celkové hodnocení a stanovíme optimální variantu - řešení našeho rozhodovacího problému:

```
celkhod[j_] := Total[Table[u[p, j], {p, 1, m}]];
celkovehodnoceni = Table[celkhod[j], {j, 1, n}];
TableForm[celkovehodnoceni, TableHeadings -> {Table[v[j], {j, 1, n}]}]

vlak      | 0.66389
autobus   | 0.57744
auto      | 0.77760

Max[celkovehodnoceni];
indexviteze = Max[Table[If[celkhod[j] == Max[celkovehodnoceni], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze]}], "Output"]]

Optimální varianta: auto
```

Řešení metodou lineárních dílčích funkcí utility:

Vytvoříme tabulku hodnot s nejhorší a nejlepší hodnotou každého kritéria a s jejich rozdílem:

```
minmax2[i_] := If[typkrit[i] == "vyn", Min, Max];
nejhorsihodnoty = Table[minmax2[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];
nejlepsi hodnoty = bazickavarianta;
rozdil = nejlepsi hodnoty - nejhorsihodnoty;

tabulkasextremy = pridejsloupec[tabulkasvahami, nejhorsihodnoty];
tabulkasextremy2 = pridejsloupec[tabulkasextremy, nejlepsi hodnoty];
tabulkasextremy3 = pridejsloupec[tabulkasextremy2, rozdil];
TableForm[tabulkasextremy3, TableHeadings -> {Table[k[i], {i, 1, m}],
Join[Table[v[j], {j, 1, n}], {"váhy"}, {"nejhorší"}, {"nejlepší"}, {"rozdíl"}]}]
```

	vlak	autobus	auto	váhy	nejhorší	nejlepší	rozdíl
cena (Kč)	123	145	405	0.267418	405	123	-282
vzdálenost (km)	163	145	135	0.0398755	163	135	-28
čas (min)	208	190	87	0.56601	208	87	-121
dostupnost (body)	7	3	5	0.126698	3	7	4

Vypočítáme dílčí hodnocení všech variant:

```
u2[p_, q_] := vahyprimo[p] * (hod[p, q] - nejhorsihodnoty[p]) / rozdil[p];
tabulkahodnoceni2 = Table[u2[p, q], {p, 1, m}, {q, 1, n}];
TableForm[tabulkahodnoceni2,
TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	vlak	autobus	auto
cena (Kč)	0.267418	0.246555	0
vzdálenost (km)	0	0.0256343	0.0398755
čas (min)	0	0.084200	0.56601
dostupnost (body)	0.126698	0	0.063349

Nakonec opět vypočteme celkové hodnocení a stanovíme optimální variantu - řešení našeho rozhodovacího problému druhou metodou:

```

celkhod2[j_] := Total[Table[u2[p, j], {p, 1, m}]];
celkovehodnoceni2 = Table[celkhod2[j], {j, 1, n}];
TableForm[celkovehodnoceni2, TableHeadings -> {Table[v[j], {j, 1, n}]}]

vlak      | 0.394116
autobus    | 0.356389
auto       | 0.66923

Max[celkovehodnoceni2];
indexviteze2 = Max[Table[If[celkhod2[j] == Max[celkovehodnoceni2], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze2]}], "Output"]]

Optimální varianta: auto

```

1.4 Modelový příklad - výběr nejlepšího města k bydlení

Rozhodovací problém - výběr nejlepšího města k bydlení mezi variantami Praha, Ženeva, Štrasburk.

Nejprve zadáme všechny potřebné vstupní údaje jako v ukázkovém příkladě:

```

m = 4; (*zadejte počet kritérií*)
n = 3; (*zadejte počet variant*)

k[1] = "prům. plat (Kč)"; (*zadejte název prvního kritéria*)
k[2] = "vzdálenost od Plzně autem (km)"; (*zadejte název druhého kritéria*)
k[3] = "oblíbenost (body)"; (*zadejte název třetího kritéria*)
k[4] = "cena bydlení-byt (Kč)"; (*zadejte název čtvrtého kritéria*)

v[1] = "Praha"; (*zadejte název první varianty*)
v[2] = "Ženeva"; (*zadejte název druhé varianty*)
v[3] = "Štrasburk"; (*zadejte název třetí varianty*)

typkrit[1] = "vyn"; (*zadejte typ prvního kritéria*)
typkrit[2] = "nak"; (*zadejte typ druhého kritéria*)
typkrit[3] = "vyn"; (*zadejte typ třetího kritéria*)
typkrit[4] = "nak"; (*zadejte typ čtvrtého kritéria*)

hod[1, 1] = 22 000; (*zadejte hodnotu prvního kritéria na první variantě*)
hod[1, 2] = 104 000;
hod[1, 3] = 40 000;
hod[2, 1] = 95;
hod[2, 2] = 870;
hod[2, 3] = 515;
hod[3, 1] = 5;
hod[3, 2] = 9;
hod[3, 3] = 7;
hod[4, 1] = 15 000;
hod[4, 2] = 50 000;
hod[4, 3] = 25 000; (*zadejte hodnotu čtvrtého kritéria na třetí variantě*)

(* Zadání preferencí vah kritérií *)
(* 1...stejně významná, 3...slabě významnější, 5...dosti významnější,
7...prokazatelně významnější, 9...absolutně významnější *)

s[1, 2] = 7;
s[1, 3] = 1 / 3;
s[1, 4] = 5;
s[2, 3] = 1 / 5;
s[2, 4] = 1 / 7;
s[3, 4] = 3;
s[i_, j_] := 1 /; i == j
s[i_, j_] := 1 / s[j, i] /; i > j

```

Sestavíme přehlednou tabulku všech hodnot kritérií na jednotlivých variantách:

```

tabhodnot = Table[hod[i, j], {i, 1, m}, {j, 1, n}];
TableForm[tabhodnot, TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]

```

	Praha	Ženeva	Štrasburk
prům. plat (Kč)	22 000	104 000	40 000
vzdálenost od Plzně autem (km)	95	870	515
oblíbenost (body)	5	9	7
cena bydlení-byt (Kč)	15 000	50 000	25 000

Dále stanovíme váhy kritérií Saatyho metodou:

```
S = Table[s[i, j], {i, 1, m}, {j, 1, m}];
```

```
MatrixForm[S]
```

```
(*Saatyho matice stanoveni vah*)
```

$$\begin{pmatrix} 1 & 7 & \frac{1}{3} & 5 \\ \frac{1}{7} & 1 & \frac{1}{5} & \frac{1}{7} \\ 3 & 5 & 1 & 3 \\ \frac{1}{5} & 7 & \frac{1}{3} & 1 \end{pmatrix}$$

```
vahyprimo = (1 / Total[N[Eigenvectors[S], 6][[1]]]) * (N[Eigenvectors[S], 6][[1]])
```

```
(* váhy kritérií *)
```

```
{0.337812, 0.0451828, 0.467516, 0.149489}
```

Do základní tabulky hodnot přidáme váhy:

```
pridejsloupec[A_, b_] := Transpose[Join[Transpose[A], {b}]]
```

```
tabulkasvahami = pridejsloupec[tabhodnot, vahyprimo];
```

```
TableForm[tabulkasvahami,
```

```
TableHeadings → {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}]}]
```

	Praha	Ženeva	Štrasburk	váhy
prům. plat (Kč)	22 000	104 000	40 000	0.337812
vzdálenost od Plzně autem (km)	95	870	515	0.0451828
oblíbenost (body)	5	9	7	0.467516
cena bydlení-byt (Kč)	15 000	50 000	25 000	0.149489

Daný rozhodovací problém řešíme metodou bazické varianty:

```
minmax[i_] := If[typkrit[i] == "vyn", Max, Min];
```

```
bazickavarianta = Table[minmax[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}]
```

```
(* bazická varianta *)
```

```
{104 000, 95, 9, 15 000}
```

Bazickou variantu přidáme do tabulky:

```
tabulkasbazvar = pridejsloupec[tabulkasvahami, bazickavarianta];
```

```
TableForm[tabulkasbazvar, TableHeadings →
```

```
{Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}, {"bazická varianta"}]}]
```

	Praha	Ženeva	Štrasburk	váhy	bazická va
prům. plat (Kč)	22 000	104 000	40 000	0.337812	104 000
vzdálenost od Plzně autem (km)	95	870	515	0.0451828	95
oblíbenost (body)	5	9	7	0.467516	9
cena bydlení-byt (Kč)	15 000	50 000	25 000	0.149489	15 000

Vypočítáme dílčí hodnocení všech variant:

```
u[p_, q_] := If[typkrit[p] == "vyn", vahyprimo[[p]] * hod[p, q] / bazickavarianta[[p]],
```

```
vahyprimo[[p]] * bazickavarianta[[p]] / hod[p, q]];]
```

```
tabulkahodnoceni = Table[u[p, q], {p, 1, m}, {q, 1, n}];
```

```
TableForm[tabulkahodnoceni,
```

```
TableHeadings → {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	Praha	Ženeva	Štrasburk
prům. plat (Kč)	0.071460	0.337812	0.129928
vzdálenost od Plzně autem (km)	0.0451828	0.00493375	0.0083347
oblíbenost (body)	0.259731	0.467516	0.363624
cena bydlení-byt (Kč)	0.149489	0.0448466	0.089693

Na závěr vypočteme celkové hodnocení a stanovíme optimální variantu - řešení našeho rozhodovacího problému:

```
celkhod[j_] := Total[Table[u[p, j], {p, 1, m}]];
celkovehodnoceni = Table[celkhod[j], {j, 1, n}];
TableForm[celkovehodnoceni, TableHeadings -> {Table[v[j], {j, 1, n}]}]

Praha      | 0.52586
Ženeva     | 0.85511
Štrasburk  | 0.59158

Max[celkovehodnoceni];
indexviteze = Max[Table[If[celkhod[j] == Max[celkovehodnoceni], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze]}], "Output"]]

Optimální varianta: Ženeva
```

Řešení metodou lineárních dílčích funkcí utility:

Vytvoříme tabulku hodnot s nejhorší a nejlepší hodnotou každého kritéria a s jejich rozdílem:

```
minmax2[i_] := If[typkrit[i] == "vyn", Min, Max];
nejhorsihodnoty = Table[minmax2[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];
nejlepsi hodnoty = bazickavarianta;
rozdil = nejlepsi hodnoty - nejhorsihodnoty;

tabulkasextremy = pridejsloupec[tabulkasvahami, nejhorsihodnoty];
tabulkasextremy2 = pridejsloupec[tabulkasextremy, nejlepsi hodnoty];
tabulkasextremy3 = pridejsloupec[tabulkasextremy2, rozdil];
TableForm[tabulkasextremy3, TableHeadings -> {Table[k[i], {i, 1, m}],
Join[Table[v[j], {j, 1, n}], {"váhy"}, {"nejhorší"}, {"nejlepší"}, {"rozdíl"}]}]
```

	Praha	Ženeva	Štrasburk	váhy	nejhorší
prům. plat (Kč)	22 000	104 000	40 000	0.337812	22 000
vzdálenost od Plzně autem (km)	95	870	515	0.0451828	870
oblíbenost (body)	5	9	7	0.467516	5
cena bydlení-byt (Kč)	15 000	50 000	25 000	0.149489	50 000

Vypočítáme dílčí hodnocení všech variant:

```
u2[p_, q_] := vahyprimo[[p]] * (hod[p, q] - nejhorsihodnoty[[p]]) / rozdil[[p]];
tabulkahodnoceni2 = Table[u2[p, q], {p, 1, m}, {q, 1, n}];
TableForm[tabulkahodnoceni2,
TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	Praha	Ženeva	Štrasburk
prům. plat (Kč)	0	0.337812	0.074154
vzdálenost od Plzně autem (km)	0.0451828	0	0.0206966
oblíbenost (body)	0	0.467516	0.233758
cena bydlení-byt (Kč)	0.149489	0	0.106778

Nakonec opět vypočteme celkové hodnocení a stanovíme optimální variantu - řešení našeho rozhodovacího problému druhou metodou:

```

celkhod2[j_] := Total[Table[u2[p, j], {p, 1, m}]];
celkovehodnoceni2 = Table[celkhod2[j], {j, 1, n}];
TableForm[celkovehodnoceni2, TableHeadings -> {Table[v[j], {j, 1, n}]}]

Praha      | 0.194671
Ženeva     | 0.80533
Štrasburk  | 0.435386

Max[celkovehodnoceni2];
indexviteze2 = Max[Table[If[celkhod2[j] == Max[celkovehodnoceni2], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze2]}], "Output"]]

Optimální varianta: Ženeva

```

1.5 Modelový příklad - výběr nejlepšího dárku pro kamarádku

Rozhodovací problém - výběr nejlepšího dárku pro kamarádku ze čtyř variant: tričko, chameleon, čokoláda, plavání se žraloky.

Nejprve zadáme všechny potřebné vstupní údaje jako v ukázkovém příkladě. Tento rozhodovací problém má o 1 variantu více, musíme proto doplnit všechna potřebná data:

```

m = 4; (*zadejte počet kritérií*)
n = 4; (*zadejte počet variant*)

k[1] = "cena (Kč)"; (*zadejte název prvního kritéria*)
k[2] = "originalita (body)"; (*zadejte název druhého kritéria*)
k[3] = "rychlost pořízení (min)"; (*zadejte název třetího kritéria*)
k[4] = "praktičnost (body)"; (*zadejte název čtvrtého kritéria*)

v[1] = "tričko"; (*zadejte název první varianty*)
v[2] = "chameleon"; (*zadejte název druhé varianty*)
v[3] = "čokoláda"; (*zadejte název třetí varianty*)
v[4] = "plavání se žraloky"; (*zadejte název čtvrté varianty*)

typkrit[1] = "nak"; (*zadejte typ prvního kritéria*)
typkrit[2] = "vyn"; (*zadejte typ druhého kritéria*)
typkrit[3] = "nak"; (*zadejte typ třetího kritéria*)
typkrit[4] = "vyn"; (*zadejte typ čtvrtého kritéria*)

hod[1, 1] = 400; (*zadejte hodnotu prvního kritéria na první variantě*)
hod[1, 2] = 2000;
hod[1, 3] = 50;
hod[1, 4] = 4000;
hod[2, 1] = 3;
hod[2, 2] = 7;
hod[2, 3] = 1;
hod[2, 4] = 9;
hod[3, 1] = 15;
hod[3, 2] = 30;
hod[3, 3] = 2;
hod[3, 4] = 60;
hod[4, 1] = 9;
hod[4, 2] = 3;
hod[4, 3] = 1;
hod[4, 4] = 5; (*zadejte hodnotu čtvrtého kritéria na čtvrté variantě*)

(* Zadání preferencí vah kritérií *)
(* 1...stejně významná, 3...slabě významnější, 5...dostí významnější,
7...prokazatelně významnější, 9...absolutně významnější *)

s[1, 2] = 1 / 3;
s[1, 3] = 7;
s[1, 4] = 3;
s[2, 3] = 9;
s[2, 4] = 5;
s[3, 4] = 1 / 5;
s[i_, j_] := 1 /; i == j
s[i_, j_] := 1 / s[j, i] /; i > j

```

Nejprve sestavíme přehlednou tabulku všech hodnot kritérií na jednotlivých variantách:

```
tabhodnot = Table[hod[i, j], {i, 1, m}, {j, 1, n}];
TableForm[tabhodnot, TableHeadings → {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]
```

	tričko	chameleon	čokoláda	plavání se žraloky
cena (Kč)	400	2000	50	4000
originalita (body)	3	7	1	9
rychlost pořízení (min)	15	30	2	60
praktičnost (body)	9	3	1	5

Dále stanovíme váhy kritérií Saatyho metodou:

```
S = Table[s[i, j], {i, 1, m}, {j, 1, m}];
MatrixForm[S]
```

(*Saatyho matice stanovení vah*)

$$\begin{pmatrix} 1 & \frac{1}{3} & 7 & 3 \\ 3 & 1 & 9 & 5 \\ \frac{1}{7} & \frac{1}{9} & 1 & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{5} & 5 & 1 \end{pmatrix}$$

```
vahyprimo = (1 / Total[N[Eigenvalues[S], 6][[1]]]) * (N[Eigenvalues[S], 6][[1]])
(* váhy kritérií *)
```

```
{0.267418, 0.56601, 0.0398755, 0.126698}
```

Do základní tabulky hodnot přidáme váhy:

```
pridejsloupec[A_, b_] := Transpose[Join[Transpose[A], {b}]]
tabulkasvahami = pridejsloupec[tabhodnot, vahyprimo];
TableForm[tabulkasvahami,
  TableHeadings → {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}]}]
```

	tričko	chameleon	čokoláda	plavání se žraloky	váhy
cena (Kč)	400	2000	50	4000	0.267
originalita (body)	3	7	1	9	0.566
rychlost pořízení (min)	15	30	2	60	0.039
praktičnost (body)	9	3	1	5	0.126

Daný rozhodovací problém řešíme metodou bazické varianty:

```
minmax[i_] := If[typkrit[i] == "vyn", Max, Min];
bazickavarianta = Table[minmax[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}]
(* bazická varianta *)
{50, 9, 2, 9}
```

Bazickou variantu přidáme do tabulky:

```
tabulkasbazvar = pridejsloupec[tabulkasvahami, bazickavarianta];
TableForm[tabulkasbazvar, TableHeadings →
  {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}, {"bazická varianta"}]}]
```

	tričko	chameleon	čokoláda	plavání se žraloky	váhy
cena (Kč)	400	2000	50	4000	0.267
originalita (body)	3	7	1	9	0.566
rychlost pořízení (min)	15	30	2	60	0.039
praktičnost (body)	9	3	1	5	0.126

Vypočítáme dílčí hodnocení všech variant:

```

u[p_, q_] := If[typkrit[p] == "vyn", vahyprimo[[p]] * hod[p, q] / bazickavarianta[[p]],
  vahyprimo[[p]] * bazickavarianta[[p]] / hod[p, q]];
tabulkahodnoceni = Table[u[p, q], {p, 1, m}, {q, 1, n}];
TableForm[tabulkahodnoceni,
  TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]

```

	tričko	chameleon	čokoláda	plavání se žraloky
cena (Kč)	0.0334272	0.0066854	0.267418	0.00334272
originalita (body)	0.188669	0.440229	0.062890	0.56601
rychlost pořízení (min)	0.0053167	0.00265837	0.0398755	0.00132918
praktičnost (body)	0.126698	0.0422328	0.0140776	0.070388

Na závěr vypočteme celkové hodnocení a stanovíme optimální variantu - řešení našeho rozhodovacího problému:

```

celkhod[j_] := Total[Table[u[p, j], {p, 1, m}]];
celkovehodnoceni = Table[celkhod[j], {j, 1, n}];
TableForm[celkovehodnoceni, TableHeadings -> {Table[v[j], {j, 1, n}]}]

tričko      | 0.354112
chameleon   | 0.491805
čokoláda    | 0.384261
plavání se žraloky | 0.64107

Max[celkovehodnoceni];
indexviteze = Max[Table[If[celkhod[j] == Max[celkovehodnoceni], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze]}], "Output"]]

Optimální varianta: plavání se žraloky

```

Řešení metodou lineárních dílčích funkcí utility:

Vytvoříme tabulku hodnot s nejhorší a nejlepší hodnotou každého kritéria a s jejich rozdílem:

```

minmax2[i_] := If[typkrit[i] == "vyn", Min, Max];
nejhorsihodnoty = Table[minmax2[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];
nejlepsi hodnoty = bazickavarianta;
rozdil = nejlepsi hodnoty - nejhorsihodnoty;

tabulkasextremy = pridejsloupec[tabulkasvahami, nejhorsihodnoty];
tabulkasextremy2 = pridejsloupec[tabulkasextremy, nejlepsi hodnoty];
tabulkasextremy3 = pridejsloupec[tabulkasextremy2, rozdil];
TableForm[tabulkasextremy3, TableHeadings -> {Table[k[i], {i, 1, m}],
  Join[Table[v[j], {j, 1, n}], {"váhy"}, {"nejhorší"}, {"nejlepší"}, {"rozdíl"}]}]

```

	tričko	chameleon	čokoláda	plavání se žraloky	váhy
cena (Kč)	400	2000	50	4000	0.267
originalita (body)	3	7	1	9	0.566
rychlost pořízení (min)	15	30	2	60	0.039
praktičnost (body)	9	3	1	5	0.126

Vypočítáme dílčí hodnocení všech variant:

```

u2[p_, q_] := vahyprimo[[p]] * (hod[p, q] - nejhorsihodnoty[[p]]) / rozdil[[p]];
tabulkahodnoceni2 = Table[u2[p, q], {p, 1, m}, {q, 1, n}];
TableForm[tabulkahodnoceni2,
  TableHeadings -> {Table[k[i], {i, 1, m}], Table[v[j], {j, 1, n}]}]

```

	tričko	chameleon	čokoláda	plavání se žraloky
cena (Kč)	0.243723	0.135401	0.267418	0
originalita (body)	0.141502	0.424506	0	0.56601
rychlost pořízení (min)	0.0309379	0.0206253	0.0398755	0
praktičnost (body)	0.126698	0.0316746	0	0.063349

Nakonec opět vypočteme celkové hodnocení a stanovíme optimální variantu - řešení našeho rozhodovacího problému druhou metodou:

```

celkhod2[j_] := Total[Table[u2[p, j], {p, 1, m}]];
celkovehodnoceni2 = Table[celkhod2[j], {j, 1, n}];
TableForm[celkovehodnoceni2, TableHeadings -> {Table[v[j], {j, 1, n}]}]

```

tričko	0.54286
chameleon	0.61221
čokoláda	0.307293
plavání se žraloky	0.62936

```

Max[celkovehodnoceni2];
indexviteze2 = Max[Table[If[celkhod2[j] == Max[celkovehodnoceni2], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze2]}], "Output"]

```

Optimální varianta: plavání se žraloky

1.6 Modelový příklad - výběr brokera pro retailovou investiční činnost

Rozhodovací problém - výběr vhodného brokera pro retailovou investiční činnost. Vybírat budeme ze tří amerických brokerů: E*Trade, Fidelity, Just2Trade. Celý postup už zkrátíme.

Nejprve zadáme všechny potřebné vstupní údaje jako v ukázkovém příkladě:

```

m = 4; (* počet kritérií *)
n = 3; (* počet variant *)

k[1] = "cena za transakci ($)"; (* první kritérium *)
k[2] = "minimální hodnota účtu ($)"; (* druhé kritérium *)
k[3] = "dostupnost Level II streamu (body)"; (* třetí kritérium *)
k[4] = "margin rate (%)"; (* čtvrté kritérium *)

v[1] = "E*Trade"; (* první varianta *)
v[2] = "Fidelity"; (* druhá varianta *)
v[3] = "Just2Trade"; (* třetí varianta *)

typkrit[1] = "nak"; (* typ prvního kritéria *)
typkrit[2] = "nak"; (* typ druhého kritéria *)
typkrit[3] = "vyn"; (* typ třetího kritéria *)
typkrit[4] = "nak"; (* typ čtvrtého kritéria *)

hod[1, 1] = 9.99; (* hodnota prvního kritéria na první variantě *)
hod[1, 2] = 7.95;
hod[1, 3] = 2.50;
hod[2, 1] = 500;
hod[2, 2] = 2500;
hod[2, 3] = 2500;
hod[3, 1] = 0.25;
hod[3, 2] = 0.8;
hod[3, 3] = 1;
hod[4, 1] = 8.44;
hod[4, 2] = 8.57;
hod[4, 3] = 5.25;

(* Zadání preferencí vah kritérií *)
(* 1...stejně významná, 3...slabě významnější, 5...dosti významnější,
7...prokazatelně významnější, 9...absolutně významnější *)

s[1, 2] = 3;
s[1, 3] = 5;
s[1, 4] = 7;
s[2, 3] = 3;
s[2, 4] = 5;
s[3, 4] = 3;
s[i_, j_] := 1 /; i == j
s[i_, j_] := 1 / s[j, i] /; i > j

```

Vypočítáme váhy kritérií a sestavíme tabulku hodnot i s vahami:

```
S = Table[s[i, j], {i, 1, m}, {j, 1, m}];
vahyprimo = (1 / Total[N[Eigenvalues[S], 6][[1]]]) * (N[Eigenvalues[S], 6][[1]]);
```

```
tabhodnot = Table[hod[i, j], {i, 1, m}, {j, 1, n}];
pridejsloupec[A_, b_] := Transpose[Join[Transpose[A], {b}]]
tabulkasvahami = pridejsloupec[tabhodnot, vahyprimo];
TableForm[tabulkasvahami,
  TableHeadings → {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}]}]
```

	E*Trade	Fidelity	Just2Trade	váhy
cena za transakci (\$)	9.99	7.95	2.5	0.56501
minimální hodnota účtu (\$)	500	2500	2500	0.262201
dostupnost Level II streamu (body)	0.25	0.8	1	0.117504
margin rate (%)	8.44	8.57	5.25	0.055285

Daný rozhodovací problém řešíme nejprve metodou bazické varianty, přičemž rovnou vypočítáme celková hodnocení všech variant a vybereme variantu optimální:

```
minmax[i_] := If[typkrit[i] == "vyn", Max, Min];
bazickavarianta = Table[minmax[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];

u[p_, q_] := If[typkrit[p] == "vyn", vahyprimo[[p]] * hod[p, q] / bazickavarianta[[p]],
  vahyprimo[[p]] * bazickavarianta[[p]] / hod[p, q]];
```

```
celkhod[j_] := Total[Table[u[p, j], {p, 1, m}]];
celkovehodnoceni = Table[celkhod[j], {j, 1, n}];
TableForm[celkovehodnoceni, TableHeadings → {Table[v[j], {j, 1, n}]}]
Max[celkovehodnoceni];
indexviteze = Max[Table[If[celkhod[j] == Max[celkovehodnoceni], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze]}], "Output"]]
```

```
E*Trade | 0.467361
Fidelity | 0.357987
Just2Trade | 0.790239
```

Optimální varianta: Just2Trade

Řešení metodou lineárních dílčích funkcí utility:

```
minmax2[i_] := If[typkrit[i] == "vyn", Min, Max];
nejhorsihodnoty = Table[minmax2[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];
nejlepsi hodnoty = bazickavarianta;
rozdil = nejlepsi hodnoty - nejhorsihodnoty;
u2[p_, q_] := vahyprimo[[p]] * (hod[p, q] - nejhorsihodnoty[[p]]) / rozdil[[p]];
```

```
celkhod2[j_] := Total[Table[u2[p, j], {p, 1, m}]];
celkovehodnoceni2 = Table[celkhod2[j], {j, 1, n}];
TableForm[celkovehodnoceni2, TableHeadings → {Table[v[j], {j, 1, n}]}]
Max[celkovehodnoceni2];
indexviteze2 = Max[Table[If[celkhod2[j] == Max[celkovehodnoceni2], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze2]}], "Output"]]
```

```
E*Trade | 0.264366
Fidelity | 0.240057
Just2Trade | 0.737799
```

Optimální varianta: Just2Trade

1.7 Modelový příklad - výběr pračky

Rozhodovací problém - výběr pračky z variant Whirlpool, AEG, Bosch (vynechána typová označení), podle následujících kritérií: cena, počet otáček, maximální náplň, energetická třída.

Vstupní údaje:

```

m = 4; (* počet kritérií *)
n = 3; (* počet variant *)

k[1] = "cena (Kč)"; (* první kritérium *)
k[2] = "počet otáček (ot./min)"; (* druhé kritérium *)
k[3] = "maximální náplň (kg)"; (* třetí kritérium *)
k[4] = "energetická třída (A-G)"; (* čtvrté kritérium *)

v[1] = "Whirlpool"; (* první varianta *)
v[2] = "AEG"; (* druhá varianta *)
v[3] = "Bosch"; (* třetí varianta *)

typkrit[1] = "nak"; (* typ prvního kritéria *)
typkrit[2] = "vyn"; (* typ druhého kritéria *)
typkrit[3] = "vyn"; (* typ třetího kritéria *)
typkrit[4] = "vyn"; (* typ čtvrtého kritéria *)

hod[1, 1] = 12990; (* hodnota prvního kritéria na první variantě *)
hod[1, 2] = 13658;
hod[1, 3] = 18077;
hod[2, 1] = 1200;
hod[2, 2] = 1200;
hod[2, 3] = 1400;
hod[3, 1] = 5;
hod[3, 2] = 7;
hod[3, 3] = 7;
hod[4, 1] = 0.5;
hod[4, 2] = 1;
hod[4, 3] = 1;

(* Zadání preferencí vah kritérií *)
(* 1...stejně významná, 3...slabě významnější, 5...dostí významnější,
7...prokazatelně významnější, 9...absolutně významnější *)

s[1, 2] = 3;
s[1, 3] = 3;
s[1, 4] = 3;
s[2, 3] = 3;
s[2, 4] = 3;
s[3, 4] = 1 / 3;
s[i_, j_] := 1 /; i == j
s[i_, j_] := 1 / s[j, i] /; i > j

```

Vypočítáme váhy kritérií a sestavíme tabulku hodnot i s vahami:

```
S = Table[s[i, j], {i, 1, m}, {j, 1, m}];
vahyprimo = (1 / Total[N[Eigenvalues[S], 6][[1]]]) * (N[Eigenvalues[S], 6][[1]]);
```

```
tabhodnot = Table[hod[i, j], {i, 1, m}, {j, 1, n}];
pridejsloupec[A_, b_] := Transpose[Join[Transpose[A], {b}]]
tabulkasvahami = pridejsloupec[tabhodnot, vahyprimo];
TableForm[tabulkasvahami,
  TableHeadings → {Table[k[i], {i, 1, m}], Join[Table[v[j], {j, 1, n}], {"váhy"}]}]
```

	Whirlpool	AEG	Bosch	váhy
cena (Kč)	12 990	13 658	18 077	0.475481
počet otáček (ot./min)	1200	1200	1400	0.274519
maximální náplň (kg)	5	7	7	0.091506
energetická třída (A-G)	0.5	1	1	0.158494

Daný rozhodovací problém řešíme nejprve metodou bazické varianty, přičemž rovnou vypočítáme celková hodnocení všech variant a vybereme variantu optimální:

```
minmax[i_] := If[typkrit[i] == "vyn", Max, Min];
bazickavarianta = Table[minmax[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];

u[p_, q_] := If[typkrit[p] == "vyn", vahyprimo[[p]] * hod[p, q] / bazickavarianta[[p]],
  vahyprimo[[p]] * bazickavarianta[[p]] / hod[p, q]];
```

```
celkhod[j_] := Total[Table[u[p, j], {p, 1, m}]];
celkovehodnoceni = Table[celkhod[j], {j, 1, n}];
TableForm[celkovehodnoceni, TableHeadings → {Table[v[j], {j, 1, n}]}]
Max[celkovehodnoceni];
indexviteze = Max[Table[If[celkhod[j] == Max[celkovehodnoceni], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze]}], "Output"]]
```

```
Whirlpool | 0.855391
AEG       | 0.93753
Bosch     | 0.86620
Optimální varianta: AEG
```

Řešení metodou lineárních dílčích funkcí utility:

```
minmax2[i_] := If[typkrit[i] == "vyn", Min, Max];
nejhorsihodnoty = Table[minmax2[i][Table[hod[i, j], {j, 1, n}]], {i, 1, m}];
nejlepsi hodnoty = bazickavarianta;
rozdil = nejlepsi hodnoty - nejhorsihodnoty;
u2[p_, q_] := vahyprimo[[p]] * (hod[p, q] - nejhorsihodnoty[[p]]) / rozdil[[p]];

celkhod2[j_] := Total[Table[u2[p, j], {p, 1, m}]];
celkovehodnoceni2 = Table[celkhod2[j], {j, 1, n}];
TableForm[celkovehodnoceni2, TableHeadings → {Table[v[j], {j, 1, n}]}]
Max[celkovehodnoceni2];
indexviteze2 = Max[Table[If[celkhod2[j] == Max[celkovehodnoceni2], j, 0], {j, 1, n}]];
CellPrint[TextCell[Row[{"Optimální varianta: ", v[indexviteze2]}], "Output"]]
```

```
Whirlpool | 0.475481
AEG       | 0.663043
Bosch     | 0.524519
Optimální varianta: AEG
```

2 Import dat z MS Excel a tvorba grafů

Mnoho volně dostupných dat je k dispozici například ve Veřejné databázi Českého statistického úřadu (<http://vdb.czso.cz/>). V této ukázce budeme pracovat s jednou z tabulek, které jsou ve Veřejné databázi k dispozici. Vybereme data o počtu obyvatel v krajích, která jsou v kategorii 4 (Obyvatelstvo, volby), podkategorie 40 (Demografie) a tabulka se jmenuje "Počet obyvatel podle pohlaví v krajích". Po kliknutí na téma se objeví tabulka v náhledu, kde je nad ní možné vybrat datum, ke kterému se budou údaje vztahovat. Vybereme například 31.12.2012. Následuje potvrzení našeho výběru pomocí tlačítka "Zobrazit tabulku", následně zvolit "Export" a exportovat tabulku na svůj disk:

ČESKÝ STATISTICKÝ ÚŘAD Veřejná databáze

Napište nám | Web ČSÚ | English

Úvodní stránka | Navigační strom | Statistické ukazatele | O Veřejné databázi | Časté dotazy | Náповěda

Výběr parametrů pro tabulku - Počet obyvatel podle pohlaví v krajích (DEM0030PU_KR)

Provedte výběr hodnot nabízených parametrů, poté potvrďte tlačítkem "Zobrazit tabulku"

Stav k: 31.12.2012

	Počet obyvatel		
	celkem	muži	ženy
Česká republika	10 516 125	5 164 349	5 351 776
Hlavní město Praha	1 246 780	605 484	641 296
Středočeský kraj	1 291 816	637 720	654 096
Jihočeský kraj	636 611	313 613	322 998
Plzeňský kraj	572 687	283 250	289 437
Karlovarský kraj	301 726	148 789	152 937
Ústecký kraj	826 764	408 585	418 179
Liberecký kraj	438 594	215 119	223 475
Královéhradecký kraj	552 946	271 657	281 289
Pardubický kraj	516 440	255 018	261 422
Kraj Vysočina	511 207	253 694	257 513
Jihomoravský kraj	1 168 650	571 982	596 668
Olomoucký kraj	637 609	311 608	326 001
Zlínský kraj	587 693	287 350	300 343
Moravskoslezský kraj	1 226 602	600 480	626 122

Zobrazit tabulku | Aktualizovat

ČESKÝ STATISTICKÝ ÚŘAD Veřejná databáze

Napište nám | Web ČSÚ | English

Úvodní stránka | Navigační strom | Statistické ukazatele | O Veřejné databázi | Časté dotazy | Náповěda

Nacházíte se: 4 - Obyvatelstvo, volby / 40 - Demografie

Tabulka | Graf | Mapa

Počet obyvatel podle pohlaví v krajích

Kód: DEM0030PU_KR
Období: 31.12.2012

Export | Tisk | Nastavení | Dotazy | Změnit parametry

	Počet obyvatel		
	celkem	muži	ženy
Česká republika	10 516 125	5 164 349	5 351 776
Hlavní město Praha	1 246 780	605 484	641 296
Středočeský kraj	1 291 816	637 720	654 096
Jihočeský kraj	636 611	313 613	322 998
Plzeňský kraj	572 687	283 250	289 437
Karlovarský kraj	301 726	148 789	152 937
Ústecký kraj	826 764	408 585	418 179
Liberecký kraj	438 594	215 119	223 475
Královéhradecký kraj	552 946	271 657	281 289
Pardubický kraj	516 440	255 018	261 422
Kraj Vysočina	511 207	253 694	257 513
Jihomoravský kraj	1 168 650	571 982	596 668
Olomoucký kraj	637 609	311 608	326 001
Zlínský kraj	587 693	287 350	300 343
Moravskoslezský kraj	1 226 602	600 480	626 122

Nastavení exportu - Google Chrome

vdb.czso.cz/vdbvo/export.jsp?cislotab=DEM0030PU_KR&kapitc

Nastavení exportu tabulky DEM0030PU_KR

Výběr typu exportu

☒ Export do XLS ☐ Export do XML

Nastavení parametrů exportu výstupního objektu do XLS

Zobrazit poznámky k legendě ☒

Zobrazit poznámky k datům ☒

Exportovat | Zavřít

VDBVO 1.3.24 © Český statistický úřad

Tabulku uložte do libovolného adresáře na Vašem počítači (např. na Plochu/Desktop). Přímo v MS Excel pro zjednodušení upravte tabulku tak, aby vypadala následovně:

	A	B	C	D
1	kraj a počet obyvatel	celkem	muži	ženy
2	Hlavní město Praha	1 246 780	605 484	641 296
3	Středočeský kraj	1 291 816	637 720	654 096
4	Jihočeský kraj	636 611	313 613	322 998
5	Plzeňský kraj	572 687	283 250	289 437
6	Karlovarský kraj	301 726	148 789	152 937
7	Ústecký kraj	826 764	408 585	418 179
8	Liberecký kraj	438 594	215 119	223 475
9	Královéhradecký kraj	552 946	271 657	281 289
10	Pardubický kraj	516 440	255 018	261 422
11	Kraj Vysočina	511 207	253 694	257 513
12	Jihomoravský kraj	1 168 650	571 982	596 668
13	Olomoucký kraj	637 609	311 608	326 001
14	Zlínský kraj	587 693	287 350	300 343
15	Moravskoslezský kraj	1 226 602	600 480	626 122
16				

Takto upravenou tabulku uložte.

2.1 Import tabulky MS Excel

Nyní se podíváme na možnosti importu tabulky MS Excel do prostředí Mathematica. Importovat můžeme oba formáty MS Excel, tedy .xls i .xlsx. Jelikož data stažená z Veřejné databáze jsou s koncovkou .xls, bude demonstrován import tohoto formátu (analogicky lze importovat formát .xlsx, .csv aj.). Stažený soubor byl pro účely tohoto textu přejmenován na "data_mathematica.xls". K samotnému importu slouží příkaz *Import*:

Import["Desktop/data_mathematica.xls"]

```
{{{kraj a počet obyvatel, celkem, muži, ženy},
  {Hlavní město Praha, 1.24678×106, 605 484., 641 296.},
  {Středočeský kraj, 1.29182×106, 637 720., 654 096.},
  {Jihočeský kraj, 636 611., 313 613., 322 998.},
  {Plzeňský kraj, 572 687., 283 250., 289 437.},
  {Karlovarský kraj, 301 726., 148 789., 152 937.},
  {Ústecký kraj, 826 764., 408 585., 418 179.},
  {Liberecký kraj, 438 594., 215 119., 223 475.},
  {Královéhradecký kraj, 552 946., 271 657., 281 289.},
  {Pardubický kraj, 516 440., 255 018., 261 422.},
  {Kraj Vysočina, 511 207., 253 694., 257 513.},
  {Jihomoravský kraj, 1.16865×106, 571 982., 596 668.},
  {Olomoucký kraj, 637 609., 311 608., 326 001.},
  {Zlínský kraj, 587 693., 287 350., 300 343.},
  {Moravskoslezský kraj, 1.2266×106, 600 480., 626 122.}}}
```

K importu dat stačí napsat do argumentu příkazu *Import* relativní cesta k souboru. Lze samozřejmě vepsat i absolutní adresu (ta může být i s mezerami):

```

Import["c:/Users/Vitek/Desktop/data_mathematica.xls"]

{{{kraj a počet obyvatel, celkem, muži, ženy},
 {Hlavní město Praha, 1.24678×106, 605484., 641296.},
 {Středočeský kraj, 1.29182×106, 637720., 654096.},
 {Jihočeský kraj, 636611., 313613., 322998.},
 {Plzeňský kraj, 572687., 283250., 289437.},
 {Karlovarský kraj, 301726., 148789., 152937.},
 {Ústecký kraj, 826764., 408585., 418179.},
 {Liberecký kraj, 438594., 215119., 223475.},
 {Královéhradecký kraj, 552946., 271657., 281289.},
 {Pardubický kraj, 516440., 255018., 261422.},
 {Kraj Vysočina, 511207., 253694., 257513.},
 {Jihomoravský kraj, 1.16865×106, 571982., 596668.},
 {Olomoucký kraj, 637609., 311608., 326001.},
 {Zlínský kraj, 587693., 287350., 300343.},
 {Moravskoslezský kraj, 1.2266×106, 600480., 626122.}}}

```

Tabulka MS Excel se zobrazí jako seznam ve složených závorkách (po řádcích). K tomu, aby vzhled dat byl v přívětivější podobě, slouží například příkaz *Grid* nebo *TableForm*. Ještě předtím však definujeme proměnnou *tabulka*, do které se tabulka uloží. Proměnnou definujeme, abychom si v další práci usnadnili manipulaci s tabulkou.

```

tabulka = Import["Desktop/data_mathematica.xls"]

{{{ "kraj a počet obyvatel", "celkem", "muži", "ženy"},
 { "Hlavní město Praha", 1.24678`*^6, 605484.` , 641296.`},
 { "Středočeský kraj", 1.291816`*^6, 637720.` , 654096.`},
 { "Jihočeský kraj", 636611.` , 313613.` , 322998.`},
 { "Plzeňský kraj", 572687.` , 283250.` , 289437.`},
 { "Karlovarský kraj", 301726.` , 148789.` , 152937.`},
 { "Ústecký kraj", 826764.` , 408585.` , 418179.`},
 { "Liberecký kraj", 438594.` , 215119.` , 223475.`},
 { "Královéhradecký kraj", 552946.` , 271657.` , 281289.`},
 { "Pardubický kraj", 516440.` , 255018.` , 261422.`},
 { "Kraj Vysočina", 511207.` , 253694.` , 257513.`},
 { "Jihomoravský kraj", 1.16865`*^6, 571982.` , 596668.`},
 { "Olomoucký kraj", 637609.` , 311608.` , 326001.`},
 { "Zlínský kraj", 587693.` , 287350.` , 300343.`},
 { "Moravskoslezský kraj", 1.226602`*^6, 600480.` , 626122.`}}}

```

Nyní použijeme příkazy *Grid* a *TableForm* pro zobrazení tabulky. Za samotnou proměnnou ještě specifikujeme, který list s MS Excel souboru chceme zobrazit. V našem případě to bude první list:

```
tabulka[[1]] // Grid
```

kraj a počet obyvatel	celkem	muži	ženy
Hlavní město Praha	1.24678×10^6	605 484.	641 296.
Středočeský kraj	1.29182×10^6	637 720.	654 096.
Jihočeský kraj	636 611.	313 613.	322 998.
Plzeňský kraj	572 687.	283 250.	289 437.
Karlovarský kraj	301 726.	148 789.	152 937.
Ústecký kraj	826 764.	408 585.	418 179.
Liberecký kraj	438 594.	215 119.	223 475.
Královéhradecký kraj	552 946.	271 657.	281 289.
Pardubický kraj	516 440.	255 018.	261 422.
Kraj Vysočina	511 207.	253 694.	257 513.
Jihomoravský kraj	1.16865×10^6	571 982.	596 668.
Olomoucký kraj	637 609.	311 608.	326 001.
Zlínský kraj	587 693.	287 350.	300 343.
Moravskoslezský kraj	1.2266×10^6	600 480.	626 122.

```
tabulka[[1]] // TableForm
```

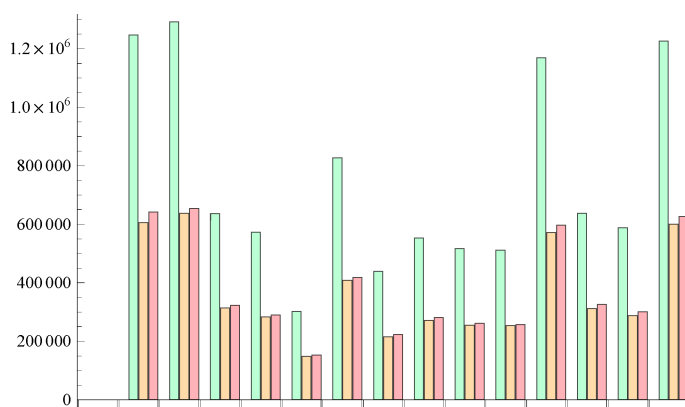
kraj a počet obyvatel	celkem	muži	ženy
Hlavní město Praha	1.24678×10^6	605 484.	641 296.
Středočeský kraj	1.29182×10^6	637 720.	654 096.
Jihočeský kraj	636 611.	313 613.	322 998.
Plzeňský kraj	572 687.	283 250.	289 437.
Karlovarský kraj	301 726.	148 789.	152 937.
Ústecký kraj	826 764.	408 585.	418 179.
Liberecký kraj	438 594.	215 119.	223 475.
Královéhradecký kraj	552 946.	271 657.	281 289.
Pardubický kraj	516 440.	255 018.	261 422.
Kraj Vysočina	511 207.	253 694.	257 513.
Jihomoravský kraj	1.16865×10^6	571 982.	596 668.
Olomoucký kraj	637 609.	311 608.	326 001.
Zlínský kraj	587 693.	287 350.	300 343.
Moravskoslezský kraj	1.2266×10^6	600 480.	626 122.

Snad jediný rozdíl mezi oběma metodami je ten, že *TableForm* zarovná hodnoty vlevo, kdežto *Grid* na střed.

2.2 Tvorba sloupcového grafu z importovaných dat

Tabulku z MS Excel máme tedy importovanou. Pro vytvoření sloupcového grafu z dat využijeme příkaz *BarChart*. Ihned, bez jakékoliv úpravy dat, by sloupcový graf vypadal následovně:

```
BarChart[tabulka]
```



Taková tabulka však není vhodná, jelikož zde chybí legenda, popis grafu, název grafu a navíc se zde zobrazují i buňky tabulky, které slouží jako hlavička (tzn. názvy krajů a nadpisy sloupců v tabulce). Proto je nutné provést

výběr samotných dat, následně popis osy x (kraje ČR) a přidat také legendu (názvy sloupců). Tyto výběry opět zahrneme do proměnných (*data*, *popis*, *legenda*).

Nejprve tedy provedeme výběr dat, kdy pomocí příkazu *Range* (rozsah) vybereme jednotlivé sloupce s daty. Podle tabulky jsou to údaje od druhého sloupce a druhého řádku. Data jsou tedy v rozmezí sloupců 2 až 4 a řádků 2 až 15:

```
data = tabulka[[Range[2, 15], Range[2, 4]]]

{{1.24678 × 106, 605 484., 641 296.}, {1.29182 × 106, 637 720., 654 096.},
 {636 611., 313 613., 322 998.}, {572 687., 283 250., 289 437.},
 {301 726., 148 789., 152 937.}, {826 764., 408 585., 418 179.},
 {438 594., 215 119., 223 475.}, {552 946., 271 657., 281 289.},
 {516 440., 255 018., 261 422.}, {511 207., 253 694., 257 513.},
 {1.16865 × 106, 571 982., 596 668.}, {637 609., 311 608., 326 001.},
 {587 693., 287 350., 300 343.}, {1.2266 × 106, 600 480., 626 122.}}
```

Obdobným způsobem lze vybrat popis osy x:

```
popis = tabulka[[Range[2, 15], 1]]
```

```
{Hlavní město Praha, Středočeský kraj, Jihočeský kraj, Plzeňský kraj, Karlovarský kraj,
 Ústecký kraj, Liberecký kraj, Královéhradecký kraj, Pardubický kraj, Kraj Vysočina,
 Jihomoravský kraj, Olomoucký kraj, Zlínský kraj, Moravskoslezský kraj}
```

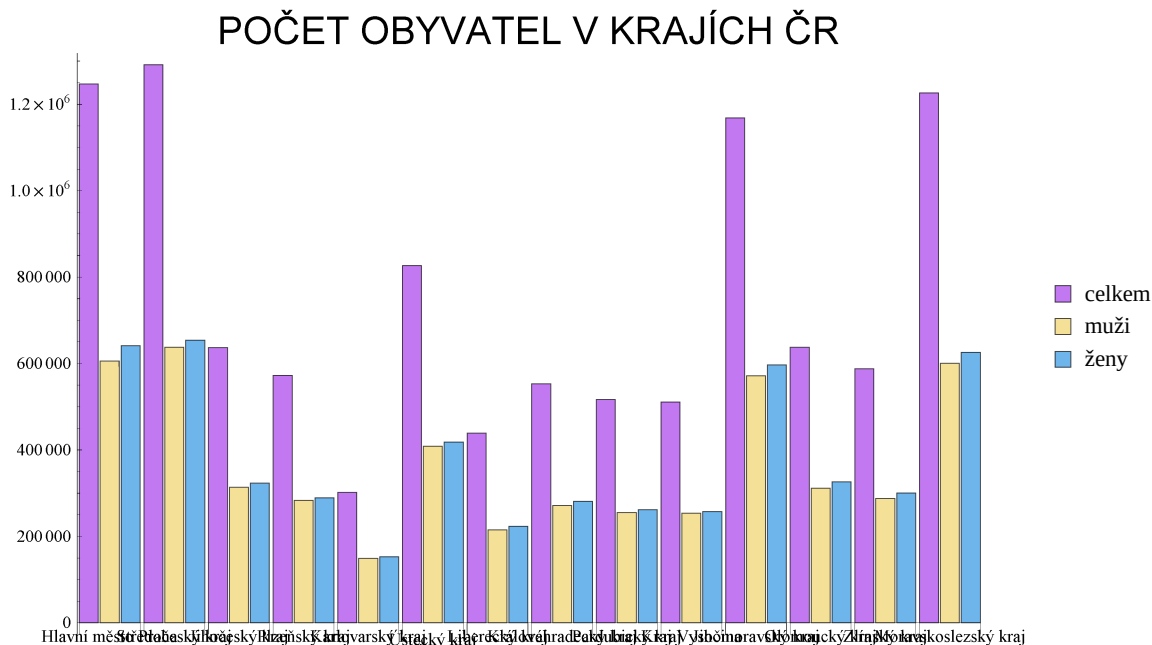
A legenda:

```
legenda = tabulka[[1, Range[2, 4]]]
```

```
{celkem, muži, ženy}
```

Nyní opět pomocí příkazu *BarChart* s uplatněním různých možností vytvoříme sloupcový graf se všemi náležitostmi. Graf opět umístíme do proměnné *graf*:

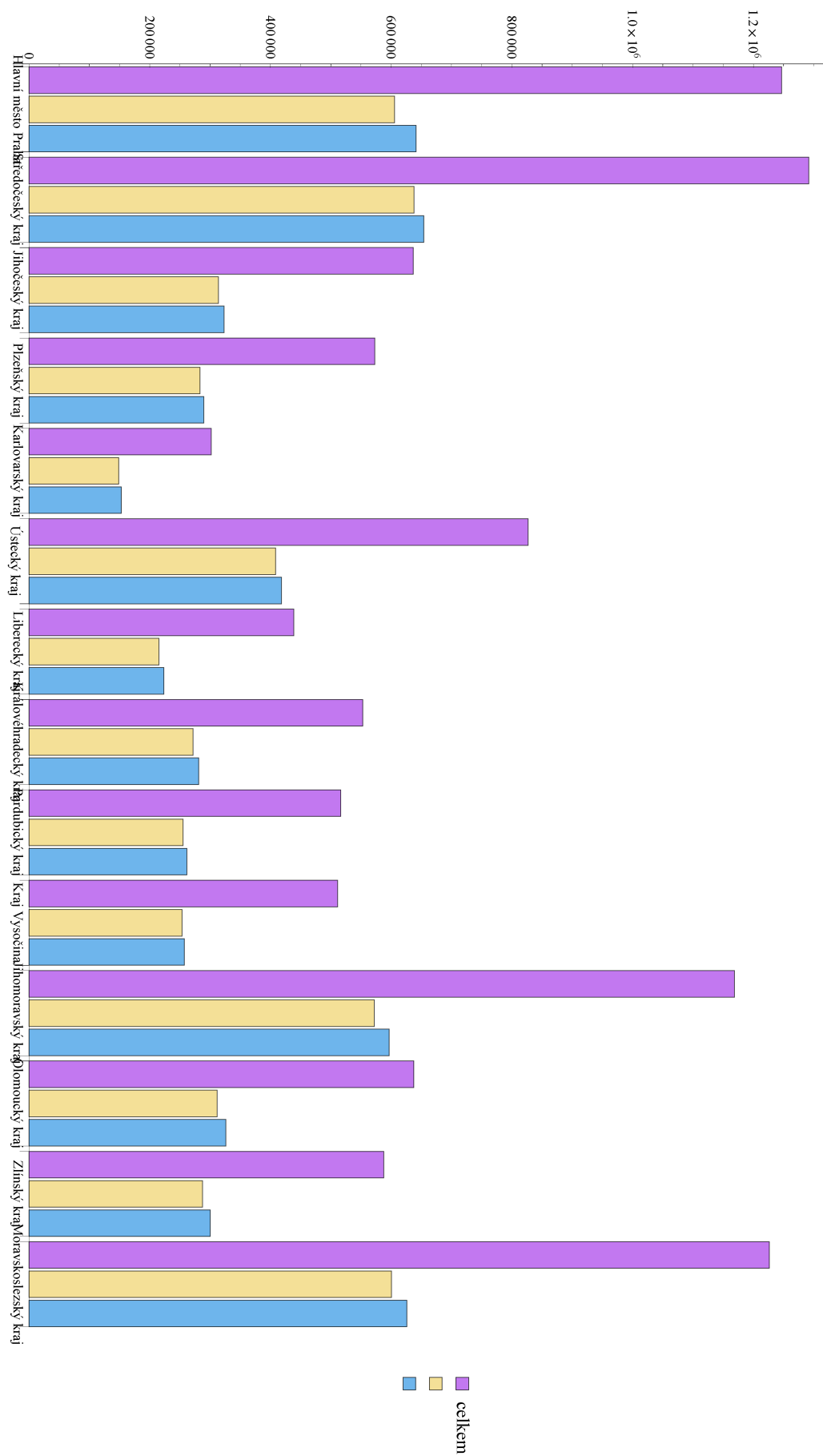
```
graf = BarChart[data, ChartLabels → {popis, None}, PlotLabel →
  StyleForm["POČET OBYVATEL V KRAJÍCH ČR", FontSize → 20, FontFamily → "Arial"],
  ChartStyle → "Pastel", ChartLegends → legenda]
```



Pro tuto publikaci z důvodu čitelnosti grafu jej ještě otočíme pomocí příkazu *Rotate* (to není nutné, pokud nebudete výsledek tisknout):

```
Rotate[graf, -Pi / 2]
```

POČET OBYVATEL V KRAJÍCH ČR



Pro úplnost si ještě vysvětlíme některé možnosti při tvorbě grafů. Příkaz *ChartLabels* umožňuje vytvořit popisky grafu. Obvykle se popisuje každý sloupec zvlášť (ukážeme si později), při volbě parametru v této proměnné na hodnotu *None*, se popis umístí ke každému oddílu grafu (kraj).

Příkaz *PlotLabel* umožní vložit název grafu. Bez nastavení stylů pomocí *StyleForm* se nastaví defaultní hodnoty. V našem případě jsme ale změnili rod písma na Arial a zvětšili velikost písma na 20. Lze upravit i barvu nadpisu pomocí příkazu *FontColor* a tloušťku písma pomocí *FontWeight*.

Příkazem *ChartStyle* můžeme definovat barevnou paletu jednotlivých sloupců (v našem případě pastelové barvy), ale také například ohraničení sloupců apod. *ChartLegend* pak umožňuje zobrazení legendy grafu.

V možnostech sloupcového grafu lze také nastavit mezery mezi jednotlivými sloupci pomocí příkazu *BarSpacing*.

2.3 Další možnosti při práci s grafy

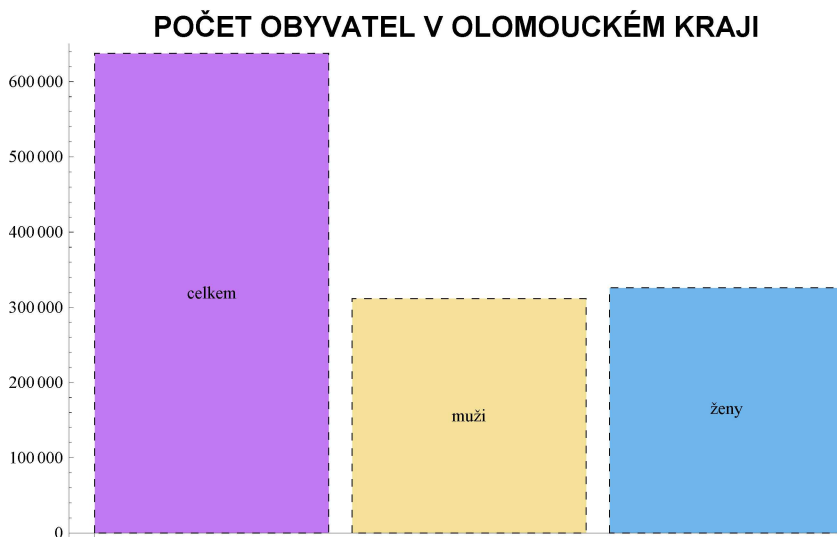
Následující ukázka slouží pro zopakování výběru určitých dat z importované tabulky a také pro ukázkou použití příkazu pro umístění popisků pomocí příkazu *Placed*. Nejprve vybereme data např. za Olomoucký kraj (řádek 13, sloupce 2 až 4):

```
dataOL = tabulka[[13, Range[2, 4]]]
```

```
{637 609., 311 608., 326 001.}
```

Data za Olomoucký kraj máme vybrána a nyní vykreslíme graf. Jelikož se jedná o jeden kraj a navíc jsou popisky sloupců definovány pomocí příkazu *Placed*, není nutné uvádět legendu. Přidáme příkaz *FontWeight* a *ChartBaseStyle* pro ilustraci jejich použití. Jelikož se jedná o jeden graf, není nutné definovat pro popis sloupců proměnnou, ale stačí je vypsát ručně:

```
grafOL = BarChart[dataOL, ChartLabels -> {Placed[{"celkem", "muži", "ženy"}, Center]},  
  PlotLabel -> StyleForm["POČET OBYVATEL V OLOMOUCKÉM KRAJI",  
    FontSize -> 15, FontFamily -> "Arial", FontWeight -> Bold],  
  ChartBaseStyle -> EdgeForm[Dashed], ChartStyle -> "Pastel"]
```



Pokud chceme vykreslit data z více (ale ne všech) údajů importované tabulky v jednom grafu, lze to udělat následovně. Ukážeme si srovnání Olomouckého kraje a Prahy. Nejprve je ale nutné definovat si rozmezí druhého grafu, tedy dat za Hlavní město Praha. Rovněž definujeme ručně popisky grafů. Z dřívějšíka máme již legendu uloženu v proměnné:

```
dataPHA = tabulka[[2, Range[2, 4]]]
```

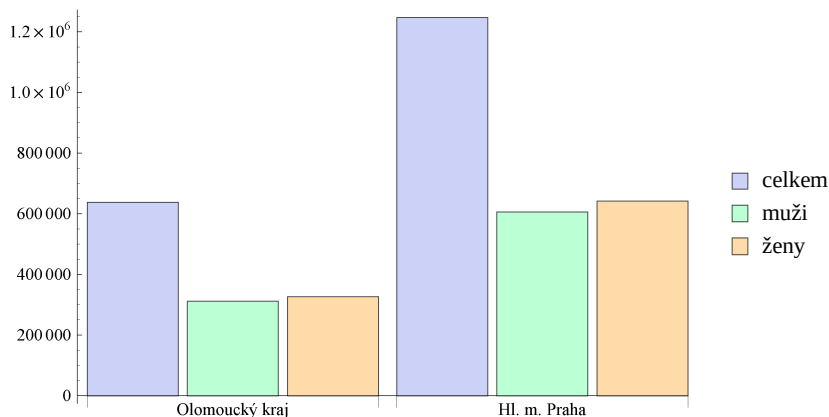
```
{1.24678 × 106, 605 484., 641 296.}
```

```
popisky = {"Olomoucký kraj", "Hl. m. Praha"}
```

```
{Olomoucký kraj, Hl. m. Praha}
```

Nyní je možné vykreslit sloupcový graf:

```
BarChart[{dataOL, dataPHA}, ChartLabels → {popisky, None}, ChartLegends → legenda]
```



Samostatné grafy je možné taky vykreslovat zvlášť s vlastními osami (ne se společnou, jak je uvedenou výše), a to pomocí příkazu *Grid*. Následující příklad vykreslí tímto způsobem grafy pro Olomoucký kraj, Hlavní město Prahu, Karlovarský kraj a Ústecký kraj. Opět bude nutné ale definovat si nejprve rozsah dat pro poslední dva jmenované kraje:

```
dataKV = tabulka[[6, Range[2, 4]]]
```

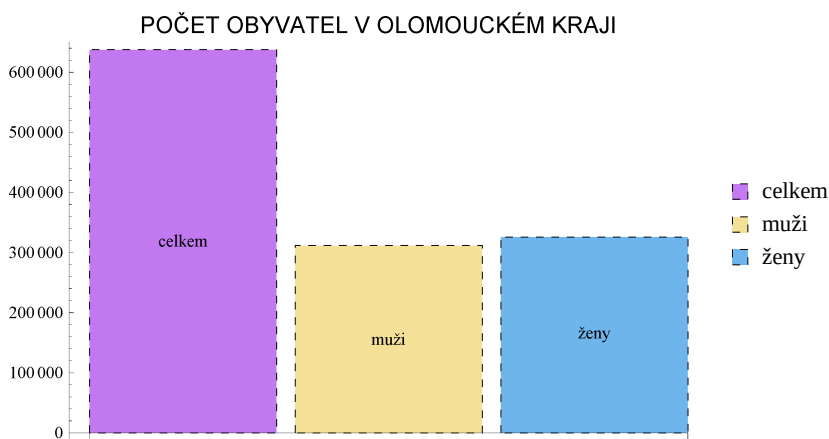
```
{301 726., 148 789., 152 937.}
```

```
dataUST = tabulka[[7, Range[2, 4]]]
```

```
{826 764., 408 585., 418 179.}
```

Nyní vytvoříme samotné grafy pro každý kraj. Použijeme vzor z této kapitoly, kdy pouze zmenšíme názvy grafů:

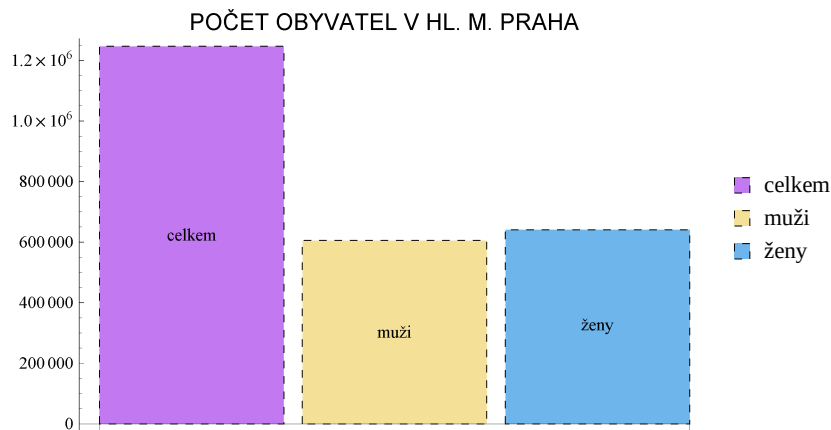
```
grafOL = BarChart[dataOL, ChartLabels → {Placed[{"celkem", "muži", "ženy"}, Center]},  
PlotLabel → StyleForm["POČET OBYVATEL V OLOMOUCKÉM KRAJI",  
FontSize → 12, FontFamily → "Arial"], ChartStyle → "Pastel",  
ChartLegends → {"celkem", "muži", "ženy"}, ChartBaseStyle → EdgeForm[Dashed]]
```



```

grafPha = BarChart[dataPHA,
  ChartLabels → {Placed[{"celkem", "muži", "ženy"}, Center]}, PlotLabel →
  StyleForm["POČET OBYVATEL V HL. M. PRAHA", FontSize → 12, FontFamily → "Arial"],
  ChartStyle → "Pastel", ChartLegends → {"celkem", "muži", "ženy"},
  ChartBaseStyle → EdgeForm[Dashed]]

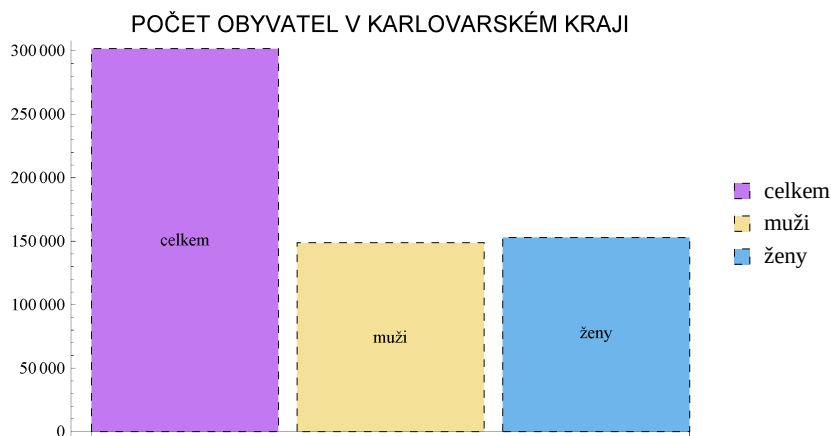
```



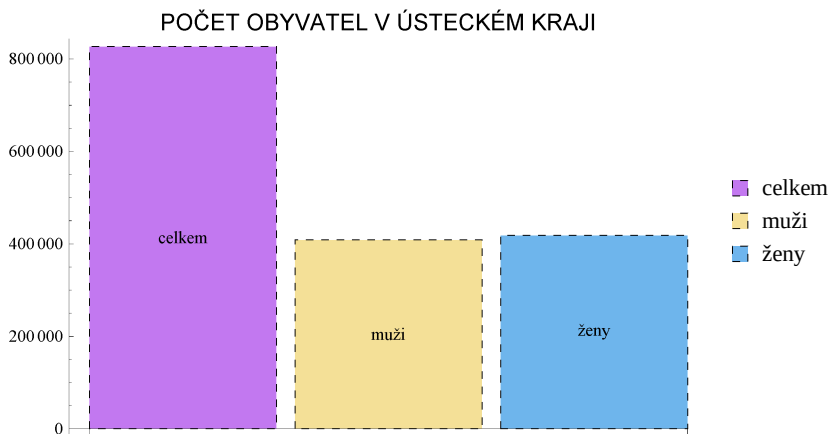
```

grafKV = BarChart[dataKV, ChartLabels → {Placed[{"celkem", "muži", "ženy"}, Center]},
  PlotLabel → StyleForm["POČET OBYVATEL V KARLOVARSKÉM KRAJI",
    FontSize → 12, FontFamily → "Arial"], ChartStyle → "Pastel",
  ChartLegends → {"celkem", "muži", "ženy"}, ChartBaseStyle → EdgeForm[Dashed]]

```

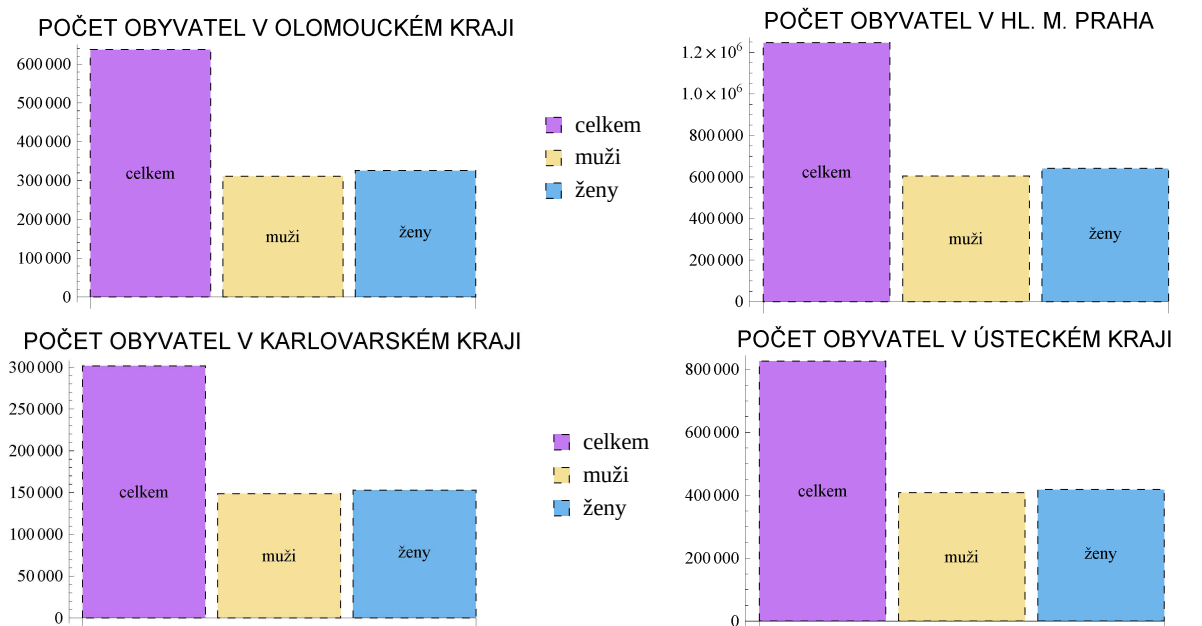


```
grafUST = BarChart[dataUST,
  ChartLabels → {Placed[{"celkem", "muži", "ženy"}, Center]}, PlotLabel →
  StyleForm["POČET OBYVATEL V ÚSTECKÉM KRAJI", FontSize → 12, FontFamily → "Arial"],
  ChartStyle → "Pastel", ChartLegends → {"celkem", "muži", "ženy"},
  ChartBaseStyle → EdgeForm[Dashed]]
```



A dostáváme se k použití příkazu *Grid*, kdy vykreslíme grafy do matice 2x2:

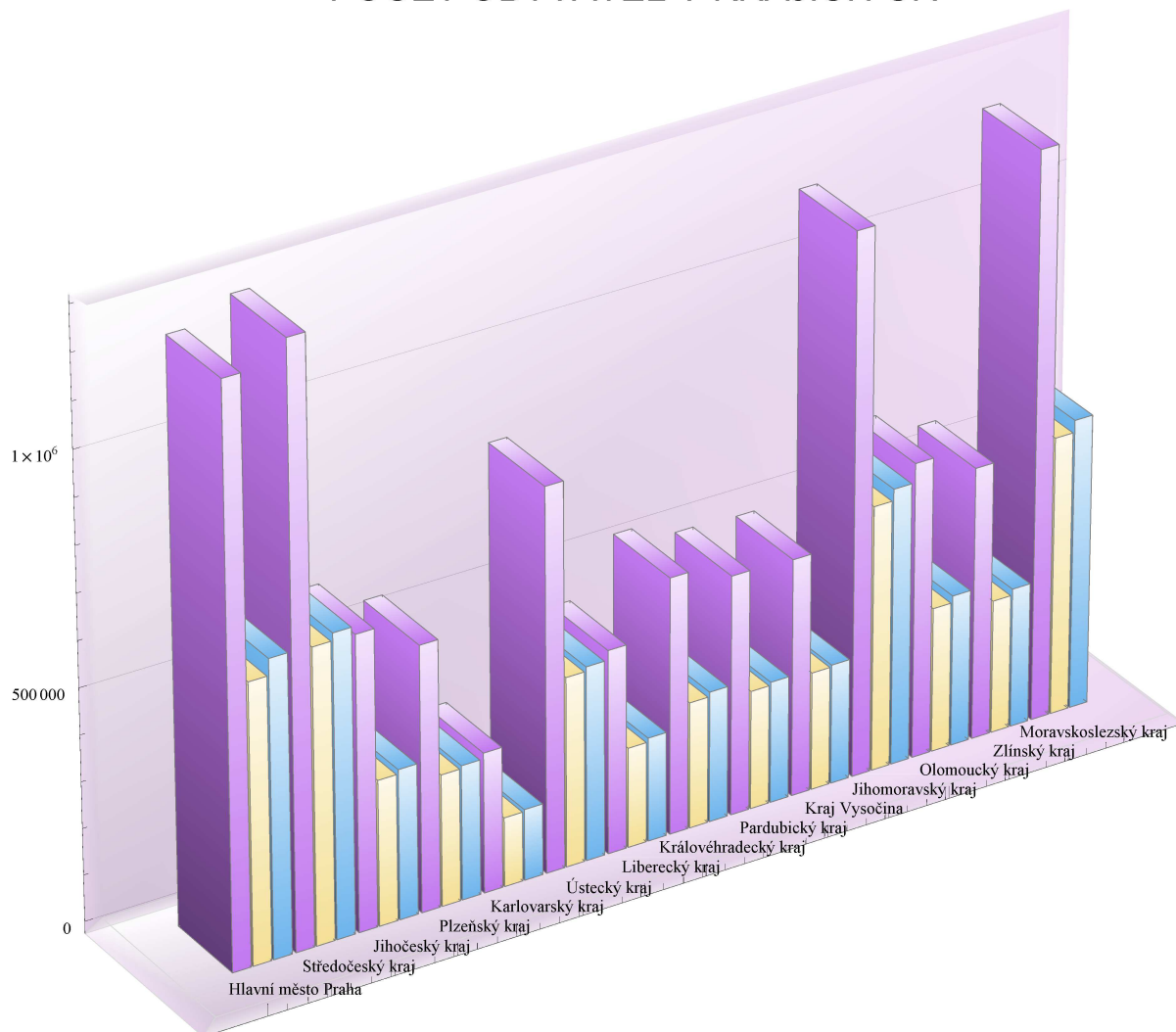
```
Grid[{{grafOL, grafPha}, {grafKV, grafUST}}]
```



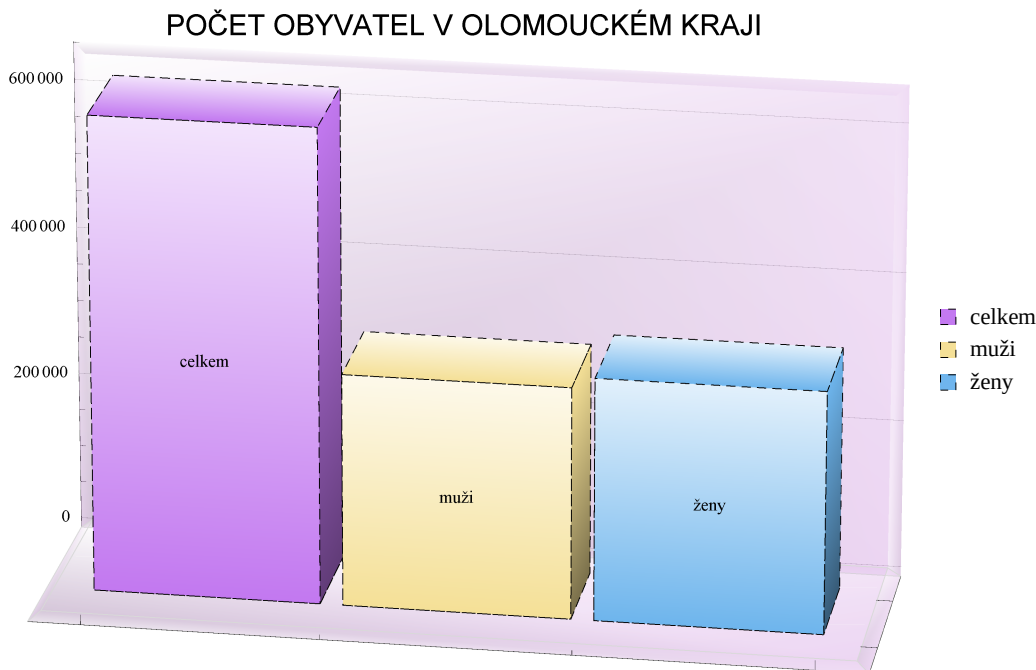
Bez jakýchkoliv úprav dříve definovaných grafů lze vytvořit i jeho 3D interaktivní verzi pomocí příkazu *BarChart3D*:

```
BarChart3D[tabulka[[Range[2, 15], Range[2, 4]]],
ChartLabels -> {popis, None}, PlotLabel ->
StyleForm["POČET OBYVATEL V KRAJÍCH ČR", FontSize -> 20, FontFamily -> "Arial"],
ChartStyle -> "Pastel", ChartLegends -> legenda]
```

POČET OBYVATEL V KRAJÍCH ČR



```
BarChart3D[dataOL,
  ChartLabels -> {Placed[{"celkem", "muži", "ženy"}, Center]}, PlotLabel ->
  StyleForm["POČET OBYVATEL V OLOMOUCKÉM KRAJI", FontSize -> 15, FontFamily -> "Arial"],
  ChartStyle -> "Pastel", ChartLegends -> {"celkem", "muži", "ženy"},
  ChartBaseStyle -> EdgeForm[Dashed]]
```



Analogicky lze vytvářet také kruhové diagramy pomocí příkazu *PieChart* a *PieChart3D*. Nejprve je však potřeba udělat několik úprav, jelikož kruhový diagram znázorňuje relativní hodnoty (poměry) a navíc nelze principiálně do diagramu zahrnout celkový počet obyvatel. Samozřejmě by šlo použít i absolutní hodnoty (diagram stejně zobrazí hodnoty v poměru mezi sebou), ale popis diagramu by pak nebyl správný. Nejprve je nutné vybrat pouze hodnoty pro počet mužů a žen. Vše na příkladu dat z Olomouckého kraje:

```
dataOLVyberM = tabulka[[13, 3]]
```

```
311 608.
```

```
dataOLVyberZ = tabulka[[13, 4]]
```

```
326 001.
```

Máme vybraná data o počtu mužů (*dataOLVyberM*) a žen (*dataOLVyberZ*) v Olomouckém kraji. Nyní je převedeme na procenta. K tomu je nutné v prvním kroku data sečíst, anebo rovnou použít hodnotu z importované tabulky pro celkový počet obyvatel v Olomouckém kraji. Pro ukázkou funkcionality programu *Mathematica* si ukážeme první variantu:

```
suma = dataOLVyberM + dataOLVyberZ
```

```
637 609.
```

```
pomerM = dataOLVyberM / suma * 100
```

```
48.8713
```

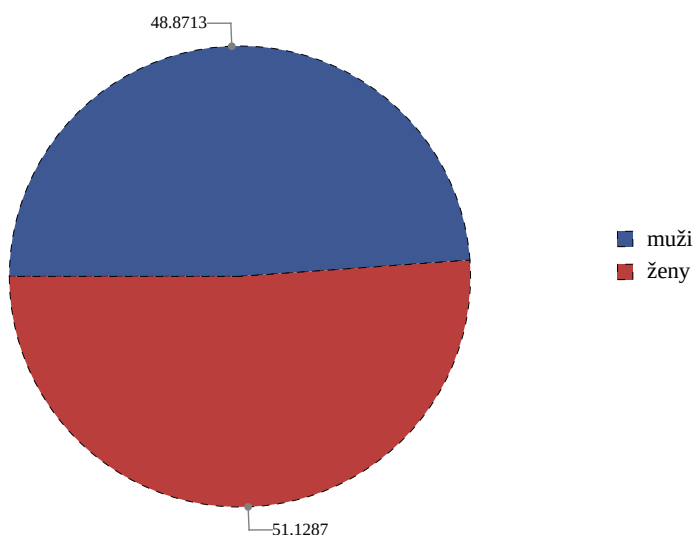
```
pomerZ = dataOLVyberZ / suma * 100
```

```
51.1287
```

Nyní jsme vytvořili poměr mužů (48,8 %) a žen (51,2 %). S malými úpravami pak lze vytvořit kruhový diagram, do jehož definice jsme přidali popisky výsečí a změnili barvy (na paletu *DarkRainbow*; barvy lze určit i pomocí slovního zadání, např. *ChartStyle -> {Red, Green}*). Jinak vše zůstalo stejné jako při vytváření sloupcového grafu:

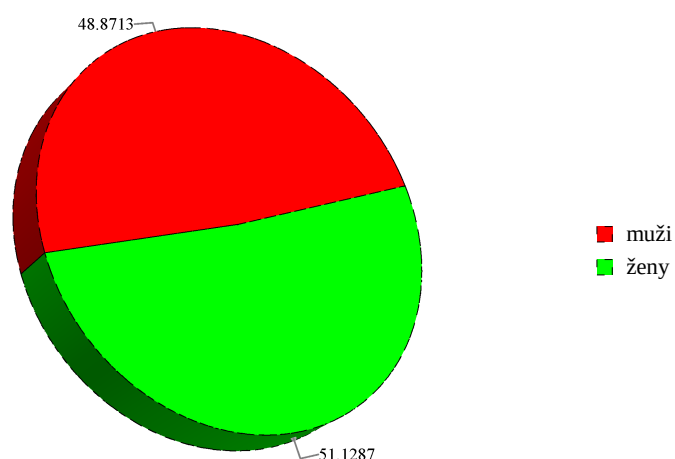
```
PieChart[{pomerM, pomerZ}, ChartLabels -> {Placed[{pomerM, pomerZ}, "RadialCallout"]},
PlotLabel -> StyleForm["POMĚR MŮŽŮ A ŽEN V OLOMOUCKÉM KRAJI",
FontSize -> 15, FontFamily -> "Arial"], ChartStyle -> "DarkRainbow",
ChartLegends -> {"muži", "ženy"}, ChartBaseStyle -> EdgeForm[Dashed]]
```

POMĚR MŮŽŮ A ŽEN V OLOMOUCKÉM KRAJI



```
PieChart3D[{pomerM, pomerZ}, ChartLabels -> {Placed[{pomerM, pomerZ}, "RadialCallout"]},
PlotLabel -> StyleForm["POMĚR MŮŽŮ A ŽEN V OLOMOUCKÉM KRAJI",
FontSize -> 15, FontFamily -> "Arial"], ChartStyle -> {Red, Green},
ChartLegends -> {"muži", "ženy"}, ChartBaseStyle -> EdgeForm[Dashed]]
```

POMĚR MŮŽŮ A ŽEN V OLOMOUCKÉM KRAJI



V této kapitole jsme ukázali, jak lze importovat data do prostředí programu *Mathematica*. Bohatým zdrojem tabelárních dat je Veřejná databáze Českého statistického úřadu, s nimiž jsme zde pracovali. Podle ukázek uvedených v této kapitole lze analogicky zpracovat jakákoliv tabelární data. Byla také demonstrována základní úprava dat, jejich výběr a přiřazení do proměnných, které se využily pro tvorbu grafů. Bylo pracováno zejména se sloupcovým typem grafu, nicméně bylo také ukázáno, jak lze použít i 3D a jiné varianty grafů. Ke konci kapitoly bylo také využito jednoduché funkcionality programu *Mathematica* k vytvoření korektního sloupcového diagramu. *Mathematica* nabízí samozřejmě celou škálu dalších grafů, které je možné aplikovat zejména na větší sady statistických dat.

3 Investice

Podnikové investice jsou jednorázově vynaložené zdroje, které budou v budoucnu přinášet peněžní příjmy během delšího časového období. Investováním odkládáme současnou spotřebu s cílem získání výnosů v budoucnu.

Investování je chápáno jako vynakládání finančních prostředků na obnovu, rozšíření nebo zhodnocení majetku podniku. Investice jsou spojeny s dlouhodobou vázaností vysokého objemu kapitálu a tím i zvýšeným podnikatelským rizikem pro celý podnik. **Realizací investic podnik sleduje zvýšení rentability a maximalizaci hodnoty celého podniku. Investice by měly přinášet peněžní tok v co nejkratší době. Peněžní příjmy by měly splatit kapitálové výdaje a dále přinést jejich zhodnocení.**

Investiční proces členíme do čtyř základních fází, které na sebe navazují.

Fázemi investičního procesu jsou:

- Fáze předinvestiční - zahrnuje identifikaci investičních příležitostí, předběžný výběr z těchto příležitostí a samotné **hodnocení efektivnosti investic** (hodnocení projektů), které by mělo odpovědět na otázku, zda investici realizovat a případně, který z projektů realizovat.
- Investiční fáze - navazuje na schválení projektu, zahrnuje etapu projekční a realizační (zajištění právní, finanční a organizační základny, personální zabezpečení, výběr dodavatelů apod.).
- Provozní fáze - zahrnuje řízení celé etapy realizace projektu, začíná zkušebním provozem a dále obsahuje běžný provoz včetně údržby.
- Fáze ukončení realizace projektu - realizuje se prodejem daného zařízení, nebo jeho likvidací. Případný peněžní příjem z likvidace zařízení zvyšuje ekonomickou efektivnost investičního projektu.

3.1 Hodnocení efektivnosti investic

Hlavním cílem podniku je maximalizace jeho hodnoty, tento cíl musí sledovat i vyhodnocování investic. **Hodnocení investice spočívá v posouzení faktorů výnosnosti, času a rizika.**

Faktory výnosnosti měří přínos investice pro podnik, poměřují diskontované peněžní příjmy SHCF (současná hodnota cash flow, CF; diskontované = převedené na současnou hodnotu) plynoucí z investice s počátečními kapitálovými výdaji.

Faktory času zohledňují skutečnost, že peníze, které máme k dispozici v současnosti, mají vyšší hodnotu, než peníze obdržené v budoucnosti. Žádoucí je co nejkratší doba návratnosti projektu, aby projekt v co nejkratším čase přinesl zpět peníze do něj vložené.

Výše rizikovosti investice se promítá do ceny kapitálu použitého na financování investice (s výší rizika roste cena kapitálu).

Současnou hodnotu příjmu plynoucího z investice v budoucnu (SHCF) stanovíme diskontováním podle následujícího vztahu:

$$SHCF = \frac{CF_i}{(1+k)^i}$$

kde:

SHCF ... současná hodnota cash flow plynoucího z investice v budoucnu (v čase i),

CF_i ... cash flow, hodnota peněžního příjmu plynoucí z investice v čase i,

k ... diskontní sazba,

i ... čas (za jak dlouho daný příjem obdržíme).

Základní rozdělení metod hodnocení efektivnosti investic:

- a) metody statické - zohledňují faktor výnosnosti, neberou v úvahu faktor rizika, čas zohledňují pouze částečně,
- b) metody dynamické - zohledňují všechny posuzované faktory; výnosnost, čas i riziko.

Ekonomické hodnocení investičních projektů můžeme rozložit do následujících kroků:

- a) stanovení doby životnosti investice,
- b) sestavení plánu nákladů a výnosů, resp. příjmů a výdajů pro jednotlivé roky životnosti investice,
- c) určení zdrojů financování investičního projektu, a to včetně poměru mezi užitým vlastním a cizím (zapůjčeným) kapitálem,
- d) určení metody stanovení efektivnosti investice,
- e) odhad diskontní míry pro dynamické metody vyhodnocení,
- f) diskontování peněžního toku - převedení peněžních příjmů z investice na jejich současnou hodnotu,
- g) hodnocení efektivnosti investice zvolenou metodou,
- h) porovnání výsledku výpočtu s jeho akceptační úrovní, předem stanovenou požadovanou hodnotou nebo s hodnotami vypočtenými pro jiné investice.

3.2 Vstupní údaje pro hodnocení efektivnosti investic

Existuje mnoho rozdílných technik hodnocení investic, jako jejich vstupní údaje slouží počáteční kapitálové výdaje (C0), předpokládané cash flow plynoucí z těchto investic v jednotlivých letech realizace projektu (CF_i), doba životnosti investice (n) představující faktor času a vážené náklady na kapitál (WACC) zohledňují faktor rizika pro investory. Je žádoucí vypracovat více variant projektu (pesimistická, reálná nebo optimistická), případně uvažovat minimální hodnoty očekávaných příjmů a maximální hodnoty předpokládaných výdajů.

Stanovení výdajů v souvislosti s realizací investice lze přesně určit na základě nabídek. V souvislosti s realizací investičního projektu mohou nastat výdaje na pořízení dlouhodobého majetku a na změnu čistého provozního kapitálu jako důsledek realizace projektu (změna hodnoty oběžného majetku a krátkodobých závazků). Tyto kapitálové výdaje lze snížit zvýšením závazků, což má pozitivní vliv na cash flow; dále mohou být sníženy o výši příjmů z prodeje zařízení, které je nahrazováno novým (zohledňuje se i daňový efekt z prodeje daného zařízení). Do posouzení výše kapitálových výdajů vstupují také náklady ušlé příležitosti (oportunitní náklady), jako výnosy z činnosti, které by podnik dosáhl, kdyby zvolil realizaci druhé nejlepší možné varianty investice.

$$C0 = PC + \Delta OA - \Delta KZ - PPDM + t \times (PPDM - ZC),$$

kde:

C0 ... kapitálové výdaje,

PC ... pořizovací cena investice (včetně souvisejících výdajů),

OA ... oběžná aktiva (pracovní kapitál),

KZ ... krátkodobé závazky,

PPDM ... peněžní prostředky z prodeje stávajícího (nahrazovaného) zařízení,

t ... sazba daně,

ZC ... zůstatková cena stávající (nahrazované) investice.

Odhad peněžních toků plynoucích z investice je založen na prognózování budoucího odbytu či plánu výroby. Na základě těchto odhadů je sestaven plán výnosů a nákladů (zjednodušený výkaz zisku a ztráty, VZZ) a vypočten předpokládaný zisk. Úpravou zisku lze nepřímou metodou stanovit CF plynoucí z investice (s ohledem na časové období se dále diskontuje).

Jako základ pro stanovení CF pomocí úpravy hospodářského výsledku se používá:

- zisk před zdaněním (EBT, Earnings Before Taxes) - je třeba při stanovování CF připočítat k EBT ná-

ladové úroky a odpisy (nejsou spojené s tokem peněžních prostředků).

- zisk před odpisy, úroky a zdaněním (EBITDA, Earnings before Interest, Taxes, Depreciations and Amortization Charges).

Výše uvedený zisk ve formě EBITDA (případně upravený EBT) je nutné dále upravit o změnu hodnoty čistého pracovního kapitálu (ČPK) vyvolanou provozováním investice (změna v případě snížení ČPK se připočítává k zisku, při zvýšení ČPK se změna odečítá od zisku). Zvýšenou potřebu ČPK je třeba profinancovat, což snižuje přírůstek peněžních prostředků plynoucích z investice. Snížená potřeba ČPK pak příjem peněžních prostředků navyšuje, protože v ČPK je vázáno méně kapitálu. Zisk EBITDA se upravuje také o odhad prodejní ceny pořizované investice (v případě prodeje zařízení po ukončení projektu) a výši příjmů z případného prodeje oběžných aktiv po skončení investice (včetně daňového efektu).

Stanovení diskontní sazby a výše inflace je nedílnou součástí vstupů pro hodnocení investic, neboť jsou spjaty s delším časovým obdobím, a proto musíme respektovat hledisko času a rizika a převést hodnotu peněžních příjmů a výdajů na současnou hodnotu prostřednictvím diskontování. Výdaje diskontujeme pouze v případě, že jsou od zahájení realizace projektu vynakládány po dobu delší než jeden rok. Riziko se s delším obdobím, s ohledem na neurčitost událostí v budoucnu, zvyšuje a promítá se do úrovně diskontní míry. Stanovení diskontní sazby souvisí také se způsobem financování projektu.

Diskontní sazba může být stanovena na úrovni celkových nákladů na podnikový kapitál (Weighted Average Cost of Capital, WACC), odráží cenu cizího kapitálu (včetně daňového efektu) a odměnu vlastníkům za vložený kapitál. Pokud se v průběhu realizace projektu kapitálová struktura mění, WACC podniku kolísá. Pakliže je projekt vysoce rizikový, je nutné hodnotu WACC navýšit o rizikovou přírážku.

Diskontní sazbu lze stanovit:

- na úrovni nákladů na vlastní kapitál,
- jako náklady na kapitál pro cílovou strukturu financování projektu,
- v různé výši, dle předpokládané struktury kapitálu; různé úrovně diskontní míry v jednotlivých letech realizace projektu.

$$WACC = n_c \times (1 - t) \times (CK / K) + n_v \times (VK / K),$$

kde:

n_v ... náklady vlastního kapitálu (%),

VK ... hodnota vlastního kapitálu (K),

K ... hodnota celkového kapitálu (K), $K = VK + CK$,

n_c ... úroková míra cizího kapitálu (%),

t ... sazba daně z příjmů (%),

CK ... hodnota cizího kapitálu (za úplatu, nikoliv závazky z obchodního styku).

Náklady na VK se odvíjejí od rizika podnikání; čím vyšší riziko, tím vyšší náklady na VK. Náklady na VK stanovujeme odhadem, požadovaná výnosnost VK, resp. náklady na VK, se skládají ze dvou složek: základní prémie a rizikové prémie. Základní prémie je závislá na ceně bezrizikových produktů kapitálového trhu (např. na ceně dlouhodobých státních dluhopisů) se stejnou splatností jako je doba životnosti projektu. Základní (časová) prémie je odměnou investora za jeho odloženou spotřebu, riziková prémie je odměnou vlastníka kapitálu za podstupované riziko spojené s podnikatelskou činností.

Náklad na CK představuje úroková sazba spolu s poplatky bankám. Výše úrokové sazby se odvíjí od doby splácení úvěru a rizika jeho nesplacení. Pokud podnik používá CK z více zdrojů s různou úrokovou sazbou, pak je nutné vypočítat vážený aritmetický průměr nákladů na CK. Váhy jsou tvořeny podílem jednotlivých cizích zdrojů (úvěrů) na výši celkového CK.

$$n_v = r_0 + RP,$$

kde:

n_v ... náklady VK, požadovaná výnosnost VK,

r_0 ... výnosnost bezrizikové investice,

RP... riziková prémie.

Diskontní sazba musí respektovat cenovou úroveň, ve které jsou uvedeny peněžní příjmy z investice.

Peněžní příjmy z investice mohou být vyjádřeny:

- v běžných cenách (ceny jednotlivých let respektující vývoj inflace), pak pro posouzení hodnocení investice použijeme nominální diskontní sazbu stanovenou např. na základě WACC,
- ve stálých cenách (v cenách výchozího roku investice), pro posouzení hodnocení investice se do diskontní sazby musí promítnout také míra inflace, použijeme tzv. reálnou hodnotu diskontní sazby (diskontní sazbu očištěnou o míru inflace). Pokud budeme v jednotlivých letech předpokládat výrazně odlišnou míru inflace, je nutné stanovit průměrnou inflaci geometrickým průměrem.

Reálná diskontní sazba

$$rk = [(1 + r) / (1 + m) - 1] \times 100,$$

kde:

rk ... reálná diskontní sazba (v %),

r ... nominální diskontní sazba (v % / 100),

m ... průměrná roční míra inflace (v % / 100).

3.3 Statické metody hodnocení investic

Statické metody hodnocení investic poměřují počáteční výdaje na investici s peněžními příjmy z investice. Nezohledňují však faktor rizika, s faktorem času pracují pouze částečně; nevýhoda jejich užití roste s délkou doby realizace projektu, rizikem a ziskovostí projektu. Statické metody se používají pro hodnocení méně významných projektů či projektů s relativně krátkou dobou životnosti; lze je použít také pro orientační posouzení výhodnosti projektů.

Mezi statické metody hodnocení investic patří:

- průměrný roční výnos,
- průměrná doba návratnosti,
- průměrná procentní výnosnost,
- doba návratnosti.

Statické metody jsou níže představeny a je popsán také postup jejich výpočtu. Prakticky jsou výpočty hodnot ukazatelů hodnocení investic vypočteny pro jeden modelový příklad. Ten je nejprve vypočítán klasicky pomocí kalkulačky a následně je uvedeno řešení prostřednictvím software *Mathematica* (výsledky v tisících Kč). Případné rozdíly ve výsledcích jsou způsobeny zaokrouhlováním.

Příklad:

Podnik plánuje investici ve výši 1 000 000 Kč. Očekávané příjmy plynoucí z projektu v jednotlivých letech jeho životnosti uvádí tabulka č. 1. Doba životnosti projektu je 5 let. Spočítejte hodnoty statických metod hodnocení efektivnosti investice (tyto hodnoty se následně srovnávají s hodnotami výsledků jiných projektů nebo s očekávanou hodnotou).

CF po dobu životnosti projektu					
Rok	1	2	3	4	5
CF (tis. Kč)	200	250	250	450	300

Tabulka č.1

Průměrný roční výnos (ØCF) stanovíme jako součet všech peněžních toků (součet všech CF_i) plynoucích z investice po dobu životnosti projektu, který podělíme počtem let životnosti investice (n).

$$\text{Průměrný roční výnos (ØCF)} = \sum_{i=1}^n \frac{CF_i}{n}$$

kde:

$\sum CF_i$... součet všech CF plynoucích z investice po dobu jejího trvání,

n ... délka životnosti investice.

Průměrný roční výnos výnos ØCF = $\sum CF_i / n = (200 + 250 + 250 + 450 + 300) / 5 = 1\,350 / 5 = 270\,000$ Kč

```
PrumRocVyn[CF_] := Apply[Plus, CF] / Length[CF]
```

```
PrumRocVyn[{200, 250, 250, 450, 300}]
```

```
270
```

Průměrná doba návratnosti (tnávratnost) je podílem kapitálových výdajů na investici (C₀) a průměrného ročního výnosu (ØCF). Průměrná doba návratnosti vypovídá o době, za kterou by měla být splacena počáteční investice při rovnoměrném peněžním příjmu.

tnávratnost = C₀ / ØCF

Průměrná doba návratnosti (tnávratnost) = C₀ / ØCF = 1 000 / 270 = 3,70 rokù

```
PrumDobaNavr[C0_, CF_] := N[C0 / (Sum[CF[[i]], {i, 1, Length[CF]}] / Length[CF]), 3]
```

```
PrumDobaNavr[1000, {200, 250, 250, 450, 300}]
```

```
3.7
```

Průměrná procentní výnosnost (Ø r) podává informaci o tom, kolik procent kapitálu se prostřednictvím peněžních příjmů ročně vrátí. Jedná se o převrácený poměr předchozího ukazatele.

Ø r = ØCF / C₀

Průměrná procentní výnosnost Ø r = ØCF / C₀ = 270 / 1 000 = 0,27 ⇒ 27 %

```
PrumProcVyn[C0_, CF_] := 100 * (Sum[CF[[i]], {i, 1, Length[CF]}] / Length[CF]) / C0
```

```
PrumProcVyn[1000, {200, 250, 250, 450, 300}]
```

```
27
```

Doba návratnosti projektu odráží počet let, za něž se kumulované peněžní příjmy z investice budou rovnat počáteční investici. Postupně načítáme příjmy (CF_i) v jednotlivých letech (CF₁ v roce 1 + CF₂ v roce 2 + ...)

dokud součet nedosáhne (nepřevýší) hodnotu počátečního kapitálového výdaje (zde je uveden výpočet pouze pomocí software *Mathematica*, tabulka konstruovaná ručně by byla shodná s níže uvedeným výpočtem).

```
TabulkaNavratnosti1[CF_, C0_, k_] :=
  Transpose[{Flatten[{"Rok realizace", Range[Length[CF]]}],
    Flatten[{"CF (tis. Kč)", Table[CF[[i]], {i, Length[CF]}]}],
    Flatten[
      {"Kumulované CF (tis. Kč)", Table[Sum[CF[[i]], {i, 1, j}], {j, Length[CF]}]},
      Flatten[{"Zbývá do splacení C0 (tis. Kč)", Table[
        If[C0 - Sum[CF[[i]], {i, 1, j}] ≤ 0, 0, C0 - Sum[CF[[i]] / (1 + k)^i, {i, 1, j}]],
        {j, Length[CF]}]}]}] // TableForm
```

```
TabulkaNavratnosti1[{200, 250, 250, 450, 300}, 1000, 0.12]
```

Rok realizace	CF (tis. Kč)	Kumulované CF (tis. Kč)	Zbývá do splacení C ₀ (tis. Kč)
1	200	200	821.429
2	250	450	622.13
3	250	700	444.185
4	450	1150	0
5	300	1450	0

3.4 Dynamické metody hodnocení investic

Dynamické metody hodnocení efektivnosti investic zohledňují faktor výnosnosti, času i rizika. Jsou vhodným nástrojem k hodnocení efektivnosti strategických investic.

Mezi dynamické metody hodnocení investic patří:

- Čistá současná hodnota,
- Vnitřní výnosové procento,
- Index ziskovosti,
- Diskontovaná doba návratnosti.

Metoda čisté současné hodnoty (ČSH; Net Present Value) je základní a nejvíce používanou dynamickou metodou. Metoda ČSH porovnává kapitálové výdaje spojené s investicí se součtem odhadnutých diskontovaných příjmů. Diskontovat znamená přepočítat hodnotu kapitálových výdajů a příjmů k roku pořízení investice (pomocí diskontní sazby). Jako všechny dynamické metody bere metoda ČSH v úvahu hodnotu peněz v čase a riziko. Ke stanovení ČSH investice potřebujeme znát kapitálové výdaje (C₀), odhad peněžních příjmů plynoucích z investice (CF_i) a diskontní sazbu (k).

$$\text{ČSH} = -C_0 + CF_1 / (1 + k)^1 + CF_2 / (1 + k)^2 + \dots + CF_n / (1 + k)^n$$

$$\text{ČSH} = -C_0 + \sum_{i=1}^n \frac{CF_i}{(1 + k)^i}$$

Výsledkem metody ČSH je číslo (v Kč) udávající, o kolik převyšují diskontované příjmy plynoucí z investice diskontované výdaje; neboli o kolik vzroste hodnota podniku. Investici je možné přijmout, pokud je výsledek metody ČSH větší než nula. V případě, že je hodnota ČSH číslo záporné, příjmy z investice nikdy nevrátí kapitálový vklad. Pokud při porovnávání více projektů dostaneme stejné výsledky ČSH, je třeba použít další metodu. Výhodou metody ČSH je její všeobecná použitelnost, možnost sčítání výsledků ČSH za více investic v portfoliu a možnost stanovení přínosu projektu pro hodnotu podniku. Nevýhodou této metody je vysoká citlivost na hodnotu úrokových měr, které vykazují vysokou proměnlivost. Vyšší úroveň diskontní míry snižuje současnou hodnotu projektu.

Průběžný příklad - stejný příklad jako u statických metod v předchozí kapitole je doplněný o informace, které si navíc žádá výpočet dynamických metod.

Vybrané dynamické metody budou spočítány pro zadání uvedené v Tabulce č.1 z předchozí kapitoly (pro přehlednost je tabulka uvedena i za tímto odstavcem). Příklad modifikujeme pouze o informace nezbytné k výpočtu dynamických metod.

Podnik plánuje jednorázovou investici v roce nula o velikosti 1 000 000 Kč. Očekávané příjmy plynoucí z projektu v jednotlivých letech jedna až pět uvádí Tabulka č. 1. Doba životnosti projektu je 5 let. Hodnota průměrných nákladů na kapitál (WACC) je 12 %. Spočítejte hodnoty vybraných dynamických metod hodnocení efektivnosti investice a posuďte výhodnost této investice.

CF po dobu životnosti projektu					
Rok	1	2	3	4	5
CF (tis. Kč)	200	250	250	450	300

Tabulka č. 1

ČSH = - C₀ + CF₁ / (1 + k)¹ + CF₂ / (1 + k)² + ... + CF_n / (1 + k)ⁿ = - 1 000 + 200 / (1 + 0,12)¹ + 250 / (1 + 0,12)² + 250 / (1 + 0,12)³ + 450 / (1 + 0,12)⁴ + 300 / (1 + 0,12)⁵ = - 1 000 + 200 / 1,12 + 250 / 1,25 + 250 / 1,4 + 450 / 1,57 + 300 / 1,76 = - 1 000 + 178,57 + 200 + 178,57 + 286,62 + 170,45 = - 1 000 + 1 014,21 = 14,21 tis. Kč

```
CSH[CF_, C0_, k_] :=
  -C0 + Sum[CF[[i]] / (1 + k)^i, {i, 1, Length[CF]}]
  (*Apply[Plus, Table[n/(1+k)^i, {i, 1, Length[CF]}]]*)
```

CSH[{200, 250, 250, 450, 300}, 1000, 0.12]

178.57 + 200 + 178.57 + 286.62 + 170.45

1014.21`

ČSH investice je kladná, můžeme investici přijmout. Rozdílné hodnoty mezi výsledky spočtené na kalkulačce a výsledkem přesnějšího výpočtu pomocí software *Mathematica* jsou způsobené průběžným zaokrouhlováním.

Vnitřní výnosové procento (VVP; Internal Rate of Return, IRR) ukazuje relativní výnosnost, kterou investice dosahuje v průběhu své životnosti. Hodnota VVP odráží diskontní sazbu, při které je ČSH rovna nule.

$$-C_0 + \sum_{i=1}^n \frac{CF_i}{(1+k)^i} = 0$$

Hodnotu VVP nelze bez pomoci software určit žádnou matematickou metodou, je třeba ji odhadnout a postupovat cestou pokusů a omylů. S růstem VVP je projekt výnosnější. Projekt je přijatelný, pokud je VVP vyšší, než jsou průměrné náklady na kapitál (WACC).

VVP lze stanovit matematicky jen v případě konvenčního průběhu závislosti ČSH na diskontní sazbě, a to dle níže uvedeného postupu:

- 1) Libovolnou hodnotu diskontní sazby „k“ dosadíme do vzorce pro ČSH a spočteme.
- 2) Pokud je ČSH kladná, pak zvolená hodnota „k“ je nižší než VVP ⇒ k_N, ČSH_N.
- 3) Hodnotu diskontní sazby „k“ zvolíme vyšší než v předešlém případě, spočteme ČSH, bude - li záporná ⇒ k_V; ČSH_V. Pokud bude hodnota ČSH opět kladná, musíme zvyšovat diskontní sazbu „k“, dokud nebude výsledná ČSH záporná.
- 4) Přibližné VVP lze stanovit na základě vzorce:

$$VVP = k_N + \frac{ČSH_N}{(ČSH_N - ČSH_V)} \times (k_V - k_N)$$

Uvedený postup je třeba modifikovat v případě, že zvolená hodnota „k“ v bodě 2) vyjde záporně. Pak tedy z prvního výpočtu stanovíme kV a pro stanovení kN hodnotu „k“ snižujeme a opětovně počítáme ČSH. Výhodou užití metody VVP je to, že nemusíme přesně znát hodnotu diskontní míry. Nicméně pro posuzování investičních záměrů je vhodnější použít ČSH a VVP použít pouze jako metodu doplňkovou.

```
VVR[C0_, CF_] :=  
Solve[-C0 + Sum[CF[[i]] / (1 + k)^i, {i, 1, Length[CF]}] == 0, k]
```

Index ziskovosti (IZ, Profitability Index, PI) měří poměr mezi součtem všech odhadnutých diskontovaných příjmů plynoucích z investice a počátečními kapitálovými výdaji. Investici je možné doporučit k realizaci, pokud je IZ větší než jedna, což odpovídá kladné ČSH. Při porovnávání více projektů je nejvýhodnější ten, který hodnotu jedna přesahuje nejvíce.

$$IZ = \sum_{i=1}^n \frac{CF_i}{(1+k)^i} / C_0$$

$$IZ = [200 / (1+0,12)^1 + 250 / (1+0,12)^2 + 250 / (1+0,12)^3 + 450 / (1+0,12)^4 + 300 / (1+0,12)^5] / 1\,000 = [1\,000 + 178,57 + 200 + 178,57 + 286,62 + 170,45] / 1\,000 = 1\,014,21 / 1\,000 = 1,014$$

```
IZ[CF_, C0_, k_] := Sum[CF[[i]] / (1 + k)^i, {i, 1, Length[CF]}] / C0
```

```
IZ[{200, 250, 250, 450, 300}, 1000, 0.12]
```

```
1.01203
```

Výsledná hodnota IZ je větší než 1, investici je možné doporučit.

Diskontovaná doba návratnosti (DN; Payback Period) odráží počet let, za něž se kumulované diskontované peněžní příjmy z investice vyrovnají počátečním kapitálovým výdajům na investici. Postupně načítáme diskontované příjmy (CF_i) v jednotlivých letech [CF₁ / (1 + k)¹ + CF₂ / (1 + k)² + ... + CF_n / (1 + k)ⁿ], dokud se hodnota diskontovaných příjmů nevyrovná počátečním kapitálovým výdajům. Pro podnik je přijatelný projekt, jehož doba návratnosti splňuje plánovanou dobu navrácení peněžních prostředků. Pokud porovnáváme více projektů, je vhodnější přijmout projekt s nejkratší dobou návratnosti.

Nevýhodou metody doby diskontované návratnosti je, že nebere v úvahu příjmy plynoucí z investice po dobu splacení počátečních výdajů. Touto metodou nelze také objektivně srovnat projekty s rozdílnou dobou životnosti. Užití diskontované doby návratnosti se doporučuje pro projekty s krátkou životností, s vysokým rizikem; tato metoda je vhodná zejména jako metoda doplňková.

```

TabulkaNavratnosti2[CF_, C0_, k_] :=
  Transpose[{Flatten[{"Rok realizace", Range[Length[CF]]}],
    Flatten[{"CF (tis. Kč)", Table[CF[[i]], {i, Length[CF]}]}],
    Flatten[
      {"Diskontovné CF (tis. Kč)", Table[CF[[i]] / (1 + k)^i, {i, Length[CF]}]},
      Flatten[{"Kumulované diskontované CF (tis. Kč)",
        Table[Sum[CF[[i]] / (1 + k)^i, {i, 1, j}], {j, Length[CF]}]}],
      Flatten[{"Zbývá do splacení C0 (tis. Kč)",
        Table[If[C0 - Sum[CF[[i]] / (1 + k)^i, {i, 1, j}] ≤ 0, 0,
          C0 - Sum[CF[[i]] / (1 + k)^i, {i, 1, j}]], {j, Length[CF]}]}]}] // TableForm

```

```
TabulkaNavratnosti2[{200, 250, 250, 450, 300}, 1000, 0.12]
```

Rok realizace	CF (tis. Kč)	Diskontovné CF (tis. Kč)	Kumulované diskontované CF
1	200	178.571	178.571
2	250	199.298	377.87
3	250	177.945	555.815
4	450	285.983	841.798
5	300	170.228	1012.03

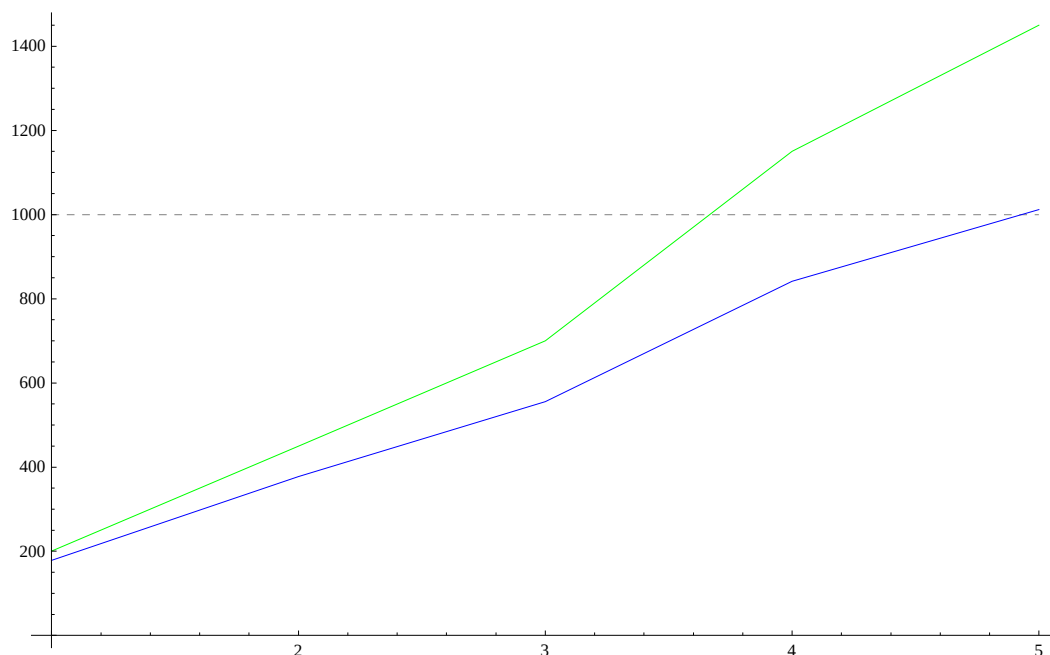
Z výše uvedeného výpočtu je patrné, že investice bude splacena v posledním roce její realizace.

V grafu jsou porovnány dílčí výsledky výpočtu doby návratnosti (zeleně) a diskontované doby návratnosti (modře). Na ose x jsou uvedeny roky trvání investice, na ose y pak kumulovaná výše příjmů z investice (nediskontovaná i diskontovaná). Přerušovanou čárou je značen kapitálový výdaj na investici.

```

Porovnani[CF_, C0_, k_] :=
  ListLinePlot[{Table[C0, {5}], Table[Sum[CF[[i]], {i, 1, j}], {j, Length[CF]}],
    Table[Sum[CF[[i]] / (1 + k)^i, {i, 1, j}], {j, Length[CF]}]},
    AxesOrigin → {1, 0}, ImageSize → 800, PlotStyle → {{Dashed, Gray}, Green, Blue}]

```



4 Řízení zásob

Zásoby patří do oběžného majetku podniku. Jednotlivé složky zásob představují: materiál, nedokončená výroba a polotovary, výrobky, zvířata a zboží. Zásoby jsou tvořeny z důvodu požadavku plynulosti výroby, z důvodu časového a prostorového nesouladu mezi potřebou zásob a jejich dostupností. Podnik držením zásob minimalizuje náklady, které by mohly vzniknout v souvislosti s výpadky v jejich dodávce; může také pružněji reagovat na změny požadavků zákazníků. Vstupy do výroby je třeba zajistit v potřebné kvalitě, množství, ve správném čase a na správném místě. Dalšími rozhodujícími faktory pro zajištění zásob jsou přímé pořizovací náklady výrobních faktorů (ceny zásob), nepřímé pořizovací náklady (přeprava a skladování zásob).

Řízení zásob představuje činnosti zabezpečující jejich optimální výši, s čímž úzce souvisí výše dodávek, jejich frekvence a minimalizace nákladů spojených s dodávkami a skladováním. Významnost řízení zásob se odvíjí od předmětu činnosti podniku. Strategické řízení potřeb zásob zahrnuje řízení potřeb zásob (součást řízení aktiv podniku) a potřebných zdrojů (součást finančního řízení podniku). Operativní řízení zásob zabezpečuje udržování jejich stavů z hlediska množství a struktury.

Řízení zásob stojí před optimalizací dvou pohledů na výši zásob:

- Vyšší úroveň zásob zabezpečuje plynulý chod výroby a snižuje pravděpodobnost jeho možného narušení (může mít za následek ztrátu či případné sankce). Úsporu v souvislosti s vyšší úrovní zásob na skladě je možné dosáhnout také v oblasti nákladů spojených s dodávkou zásob (v případě vyšších dodávek zásob je nižší četnost dodávek a nižší náročnost na čas pracovníků).
- Vysoká úroveň zásob způsobuje vyšší náklady na skladování (prostory, technika, pracovníci) a zároveň je v zásobách vázán kapitál (nejen v samotné zásobě, ale i např. v investici do skladové haly).

Faktory, které ovlivňují řízení zásob, můžeme rozdělit na vnější a vnitřní. Vnějšími faktory jsou zejména nákupní marketing, doprava, umístění podniku a pružnost dodavatelů. Vnitřními faktory jsou technická příprava výroby, charakter výrobního procesu, úroveň logistických procesů, rozsah sortimentu a trend ve spotřebě, úroveň řízení a zainteresovanost managementu. Výrobní faktory pro plynulý proces výroby musí být zajištěny s minimálními náklady. Základní propočty nezbytné pro optimalizaci řízení zásob řeší spotřebu materiálu a jeho nákupní ceny (hladina zásob), bilanci materiálu (bilanční rovnice) a stanovení optimální výše dodávky (skladovací náklady a náklady na dodávku).

4.1 Dodávkový cyklus zásob

Základní bilanční rovnice zásob vyjadřuje základní sladění potřeb zásob s jejich hmotnými zdroji. Výše objednaných zásob ve sledovaném období je závislá na potřebě dané zásoby v podniku, která se odvíjí od spotřeby, výše aktuálního stavu zásoby na skladě a požadavků na výši zůstatku zásoby na skladě.

Zdroje = potřeba
 $Z_p + Q = M_z + Z_k$,

kde:

Q ... velikost dodávky, která musí být v daném období zabezpečena (potřeba),

M_z ... předpokládaná spotřeba zásoby v plánovaném období (spotřeba),

Z_k ... konečná zásoba, která je stanovená jako normovaná zásoba,

Z_p ... počáteční zásoba, skutečný stav zásob v okamžiku sestavení bilance.

Příklad

Zásoba suroviny na počátku období je 100 tun. Podnik plánuje spotřebu této suroviny na 1800 tun za období. Na konci období se požaduje zásoba této suroviny ve výši 200 tun. Vypočítejte, jaká bude celková potřebná výše

dodávky (potřeba) této suroviny za dané období.

$$\begin{aligned} Z_p + Q &= M_z + Z_k, 100 + Q = 1\,800 + 200 \\ Q &= 1\,900 \text{ tun} \end{aligned}$$

Podnik v daném období potřebuje dovést 1 900 tun suroviny.

Ve výpočtu pomocí software *Mathematica* (níže) lze vypočítat jednu vybranou neznámou. Záleží na uvedení vstupních hodnot.

```
BiLRovZas[Zp_, Q_, Mz_, Zk_] :=  
Solve[Zp + Q == Mz + Zk, First@Cases[{Zp, Q, Mz, Zk}, x_ /; ! IntegerQ[x]]]
```

```
BiLRovZas[100, Q, 1800, 200]
```

```
{ {Q -> 1900} }
```

Dodávkový cyklus (t_c) je doba mezi dvěma po sobě jdoucími dodávkami.

$$t_c = t_{obd}/(Q/q),$$

kde :

t_c ... dodávkový cyklus,

t_{obd} ... počet dnů sledovaného období (např. rok),

Q ... předpokládaná potřeba daného druhu zásob,

q ... velikost dodávky.

$$tc[tobd_, Q_, q_] := \frac{tobd}{\frac{Q}{q}}$$

4.2 Modely řízení zásob

Modely řízení zásob lze členit na řízení zásob :

- v situaci závislé potřeby (jednoznačně determinovaná poptávkou po výrobku) - potřeba konkrétní položky zásob je jednoznačně stanovitelná na základě poptávky po jiné položce, známe tedy množství i čas potřeby zásoby. V případě přesných norem pak podnik drží pouze minimální rezervu zásoby.
- v situaci nezávislé potřeby (nezávislá poptávka) - nutnost odhadu potřeby zásoby nebo její stanovení pomocí jiných faktorů, např. časové řady.

Optimalizační model zásob v prostředí jistoty

V situaci nezávislé poptávky existují modely řízení optimalizace průběhu a spotřeby zásob. Lze je dále členit dle jistoty či nejistoty vstupních informací.

Předpoklady modelu průběhu spotřeby, řízení a optimalizace zásob v prostředí jistoty:

- Všechny vstupní informace jsou známy (proměnné jsou konstantami).
- Řízena je jedna položka zásob (známe její cenu, která je stejná pro všechny jednotky a je nezávislá na objemu dodávky).
- Neexistuje časová prodleva mezi objednávkou a dodávkou dané položky.
- Spotřeba položky probíhá rovnoměrně a spojitě.

Optimalizační model je založen na předpokladu, že v souvislosti se zásobami existují **náklady na skladování a náklady na dodávku zásob**.

Přičemž tyto dvě skupiny nákladů působí protichůdně:

- S růstem velikosti jedné dodávky obvykle rostou náklady na skladování a udržování zásob (náklady na držení zásob ND).
- Náklady na doplnění zásob (náklady na dodávku, NDP) naopak rostou s poklesem velikosti dodávek (s vyšším počtem dodávek).

Optimální stav představuje nákladově nejpriznivější objem dodávky, při němž jsou minimální celkové náklady na zabezpečení dodávek, skladování a udržení zásob. Optimalizujeme formou minimalizace funkce celkových nákladů pro nezávisle proměnnou, kterou je velikost dodávky (q).

Náklady na držení zásob

$$ND = q/2 \times N_{si},$$

kde :

ND ... náklady na držení zásob (skladování a udržení),

$q/2$... průměrná zásoba,

N_{si} ... náklady spojené se skladováním jednoho kusu produktu.

$$ND[q_ , N_{si}_] := \frac{q}{2} \times N_{si}$$

Náklady na doplnění zásob

$$NDP = N_{di} \times Q/q,$$

kde :

NDP ... náklady na doplnění zásob (na zabezpečení dodávek),

N_{di} ... náklady vynaložené v souvislosti s jednou dodávkou zásob,

Q ... celková potřeba skladované položky za sledované období (většinou rok),

q ... velikost jedné dodávky v naturálních jednotkách (např.ks).

$$NDP[N_{di}_ , Q_ , q_] := N_{di} \times \frac{Q}{q}$$

Celkové náklady na zajištění zásob jsou součtem výše uvedených dvou skupin nákladů, nákladů na držení zásob (ND; skladování) a nákladů na doplnění zásob (NDP).

$$CN_{opt} = ND + NDP = q/2 \times N_{si} + N_{di} \times Q/q$$

$$CN_{opt}[N_{si}_ , N_{di}_ , q_ , Q_] := \frac{q}{2} \times N_{si} + N_{di} \times \frac{Q}{q}$$

Minimum funkce uvedených nákladů zjistíme prostřednictvím první derivace dle neznámé (q), kterou položíme rovnu nule.

Výsledkem je tzv. **Harrisův - Wilsonův model**:

$$q_{opt} = \sqrt{\frac{2 \times N_{di} \times Q}{N_{si}}}$$

kde:

Ndi ... náklady na zajištění jedné dodávky i-tého materiálu,

Qpi ... velikost roční dodávky (potřeba),

Nsi ... náklady na skladování a udržování zásob, v Kč za jednotku zásoby a sledované období.

$$q[Ndi_ , Qpi_ , Nsi_] := \text{Sqrt} \left[\frac{2 * Ndi * Qpi}{Nsi} \right]$$

Celkové náklady na základě Harrisova - Wilsonova modelu mohou být vyjádřeny také následujícím vztahem:

$$CN_{opt} = \sqrt{2 * Qpi * Nsi * Ndi}$$

Příklad - Harrisův-Wilsonův model

Vypočtete optimální velikost dodávky, dodávkový cyklus, počet dodávek za období a související úroveň celkových nákladů spojených se zásobami. Podnik v daném období spotřebuje 12 000 ks jednoho druhu součástek. Náklady na skladování 1 kusu jsou v daném období 15 Kč. Náklady spojené s objednacím procesem a manipulací, neboli náklady na jednu dodávku zásob jsou 6 000 Kč.

$$q_{opt} = \sqrt{\frac{2 * Ndi * Qpi}{Nsi}}$$

$$q_{opt} = \sqrt{\frac{2 * 12000 * 6000}{15}} = \sqrt{9600000} = 3098 \text{ ks}$$

$$tc = \text{tobd} / (Q/q) = 360 / (12\,000 / 3\,098) = 360 \text{ dní} / 4 \text{ dodávky} = 90 \text{ dní}$$

$$\text{Počet dodávek za sledovanou periodu} = Q / q = 360 / 90 = 4 \text{ dodávky}$$

$$CN_{opt} = ND + NDP = q/2 * Nsi + Ndi * Q/q = 3\,098 / 2 * 15 + 6\,000 * 12\,000 / 3\,098 = 23\,235 + 23\,241 = 46\,476 \text{ Kč}$$

Optimální velikost dodávky je na úrovni 3 098 ks a jsou s ní spojeny celkové náklady ve výši 46 476 Kč za dané období. Dodávkový cyklus je 90 dní, za sledovanou periodu budou realizovány 4 dodávky.

$$q[Ndi_ , Qpi_ , Nsi_] := \text{Round@N} \left[\text{Sqrt} \left[\frac{2 * Ndi * Qpi}{Nsi} \right] \right]$$

$$q[12\,000, 6000, 15]$$

$$3098$$

$$tc[\text{tobd_}, Q_ , q_] := \text{Ceiling} \left[\frac{\text{tobd}}{q} \right]$$

$$tc[360, 12\,000, 3098]$$

$$93$$

$$CN_{opt}[Nsi_ , Ndi_ , q_ , Q_] := N \left[\frac{q}{2} * Nsi + Ndi * \frac{Q}{q} \right]$$

$CN_{opt}[15, 6000, 3098, 12000]$

46 475.8

Optimalizační model zohledňující množstevní rabaty

V praxi je cena zásob často závislá na množství (objemu) nakupovaných zásob a součástí obchodních vztahů jsou množstevní rabaty. Do konstrukce modelu je tedy zahrnuta množstevní sleva (celková finanční úspora, CFU) dosažená při nákupu vyššího objemu dané zásoby.

Další předpoklady konstrukce modelu:

- Jedná se o produkt s nezávislou poptávkou.
- Všechny vstupní informace nejsou známy s jistotou.
- Řízena je jedna položka zásob, cena je závislá na objemu nakupovaných zásob (objemu dodávky).
- Mezi objednávkou a dodávkou existuje časová prodleva.

Celkovou finanční úsporu můžeme vypočítat na základě vztahu:

$$CFU = Q \times r_i \times P,$$

kde:

Q ... celková potřeba položky zásob v daném období,

r_i ... výše množstevního rabatu pro odebíraný objem zásob,

n_p ... počet cenových pásem,

P ... základní cena dané zásoby bez uplatnění množstevního rabatu.

$$CFU[Q_ , r_i_ , P_] := Q * r_i * P$$

Finanční úspora vzniklá uplatněním množstevního rabatu poníží celkové náklady (CN) spojené se zásobami.

Celkový finanční efekt (CFE) vypočteme :

$$CFE = CN - CFU$$

$$CFE[Nsi_ , Ndi_ , Q_ , q_ , r_i_ , P_] := \frac{q}{2} * Nsi + Ndi * \frac{Q}{q} - Q * r_i * P$$

Příklad

Potřeba vybrané zásoby v podniku je 10 000 ks za rok. Náklady na skladování činí 25 Kč/ks ročně. Náklady na dodání zásob představují 5 000 Kč. Množstevní rabaty poskytované dodavatelem zásoby jsou rozvrstveny dle výše odebíraného množství (zobrazeny v následující tabulce). Vypočítejte optimální velikost dodávky při zohlednění celkového finančního efektu, když cena jednoho kusu zásoby je 30 Kč.

Objednávané množství (ks)	Sleva z ceny (%)
0 – 1000	0
1001 – 2500	1
2501 – 3500	2

3501 a více	3
-------------	---

Optimální výše dodávky bez ohledu na výši rabatu je:

$$q_{opt} = \sqrt{\frac{2 * N_{di} * Q_{pi}}{N_{si}}}$$

$$q_{opt} = \sqrt{\frac{2 * 10000 * 5000}{25}}$$

$$q_{opt} = 2\,000 \text{ ks}$$

Optimální objem dodávky spadá do množstevního rabatu 1 % ze základní ceny dané zásoby, tuto skutečnost zahrneme do výpočtu celkového finančního efektu CFU.

$$CFE = q/2 \times N_{si} + N_{di} \times Q / q - Q \times r_i \times P$$

$$CFE = 2\,000/2 \times 25 + 5\,000 \times 10\,000/2\,000 - 10\,000 \times 0,01 \times 30 = 25\,000 + 25\,000 - 3\,000 = 47\,000 \text{ Kč}$$

Při optimální výši dodávky o objemu 2 000 ks bychom dosáhli na množstevní rabat 1 % ze základní ceny výrobku. Je třeba zvážit, zda by případné navýšení objemu dodávky a tudíž navýšení nákladů spojených s dodáním, nepřevážila množstevní sleva v důsledku vyššího odebraného množství. To zjistíme na základě výpočtu CFE pro nejnižší hodnoty v daných cenových pásmech.

$$CFE(2\,501) = q/2 \times N_{si} + N_{di} \times Q / q - Q \times r_i \times P = 2\,501 / 2 \times 25 + 5\,000 \times 10\,000 / 2\,501 - 10\,000 \times 0,02 \times 30 = 31\,262,5 + 19\,992 - 6\,000 = 45\,254,5 \text{ Kč}$$

$$CFE(3\,501) = q/2 \times N_{si} + N_{di} \times Q / q - Q \times r_i \times P = 3\,501 / 2 \times 25 + 5\,000 \times 10\,000 / 3\,501 - 10\,000 \times 0,03 \times 30 = 43\,762,5 + 14\,281,63 - 9\,000 = 49\,044,13 \text{ Kč}$$

Nejnižšího celkového finančního efektu 45 254,4 Kč dosáhneme v případě odběru 2 501 kusů dané zásoby. Množstevní sleva v tomto případě převážila zvýšení nákladů na skladování a dodání.

CFE[25, 5000, 10 000, 2000, 0.01, 30]

47 000.

CFE[25, 5000, 10 000, 2501, 0.02, 30]

45 254.5

CFE[25, 5000, 10 000, 3501, 0.03, 30]

49 044.1

5 Ekonomická analýza investice do decentralizovaného zdroje energie

Naším cílem je zajistit dostatečné množství elektrické energie pro běžný provoz malé obytné jednotky s využitím blízkého vodního zdroje (např. malý chatový domek u říčky).

Pro naše rozhodnutí o zamýšlené instalaci domácí vodní elektrárny je nutno provést energetickou bilanci:

- A) zdroje elektrické energie,
- B) spotřeby elektrické energie.

5.1 Balance zdroje elektrické energie - malé vodní elektrárny

Zásadní veličinou pro kalkulaci energie z vodního zdroje je výška spádu h (m).

$$h = 2;$$

Další důležitou veličinou je průtok Q (m^3 / s).

$$Q = 0.01;$$

Uvažujeme standardní konstanty hustoty vody ρ (kg / m^3) a gravitačního zrychlení g (m / s^2).

$$\rho = 998;$$

$$g = 9.823;$$

Dále musíme definovat účinnost přívodu η_p , která je menší než 1 kvůli sklonu přívodu či tření vody o jeho stěny apod.

$$\eta_p = 0.8;$$

Účinnost turbíny η_T je menší než 1 kvůli tomu, že kinetickou energii vody nedokážeme dokonale převést na mechanickou energii turbíny.

$$\eta_T = 0.7;$$

Účinnost generátoru η_G je menší než 1, neboť nedokážeme dokonale převést mechanickou energii turbíny na elektrickou energii.

$$\eta_G = 0.5;$$

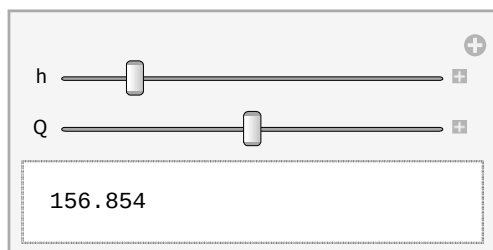
Energie vody E_p (W) z přívodu pro turbínu je dána vztahem:

$$E_p = h \times Q \times \rho \times g \times \eta_p$$

$$156.854$$

Pokud máme představu o tom, jak velikou energii vody E_p (W) z přívodu pro turbínu bychom potřebovali, můžeme si v *SW Mathematica* jednoduše nalézt potřebnou kombinaci hodnot výšky spádu h (m) a průtoku Q (m^3 / s) pomocí funkce *Manipulate*.

Manipulate[$E_p = h \times Q \times \rho \times g \times \eta_p$, { h , 0.5, 10, 0.5}, { Q , 0.005, 0.015, 0.001}]



Energie z turbíny E_T (W) pro generátor je dána vztahem:

$$E_t = E_p \times \eta_t$$

109.798

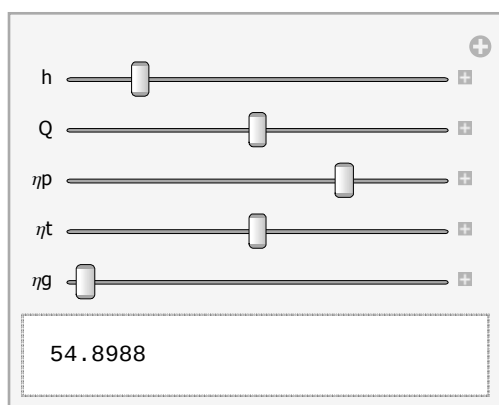
Elektrický výkon P (W) je dán vztahem:

$$P = E_t \times \eta_g$$

54.8988

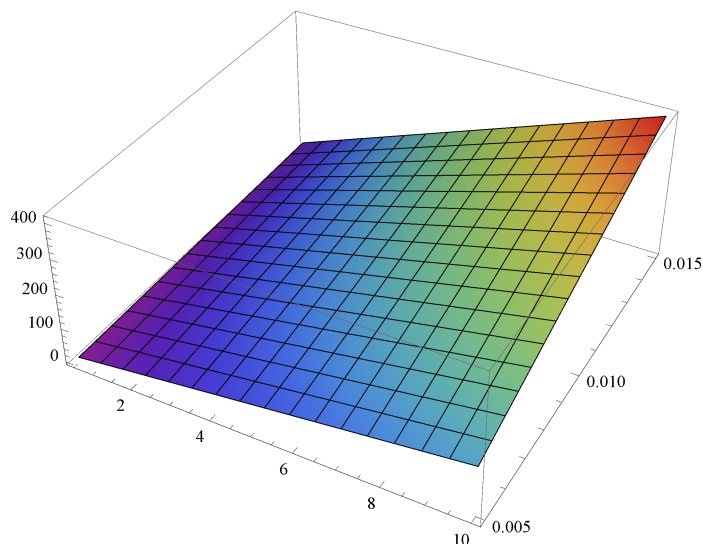
Opět, pokud máme představu o tom, jak veliký elektrický výkon P (W) bychom potřebovali, můžeme si v *SW Mathematica* jednoduše nalézt potřebnou kombinaci hodnot výšky spádu h (m), průtoku Q (m^3/s), účinnosti přívodu η_p , účinnosti turbíny η_t a účinnosti generátoru η_g pomocí funkce *Manipulate*.

```
Manipulate[P = h × Q × ρ × g × ηp × ηt × ηg, {h, 0.5, 10, 0.5}, {Q, 0.005, 0.015, 0.001},
{ηp, 0.5, 0.9, 0.1}, {ηt, 0.5, 0.9, 0.1}, {ηg, 0.5, 0.9, 0.1}]
```



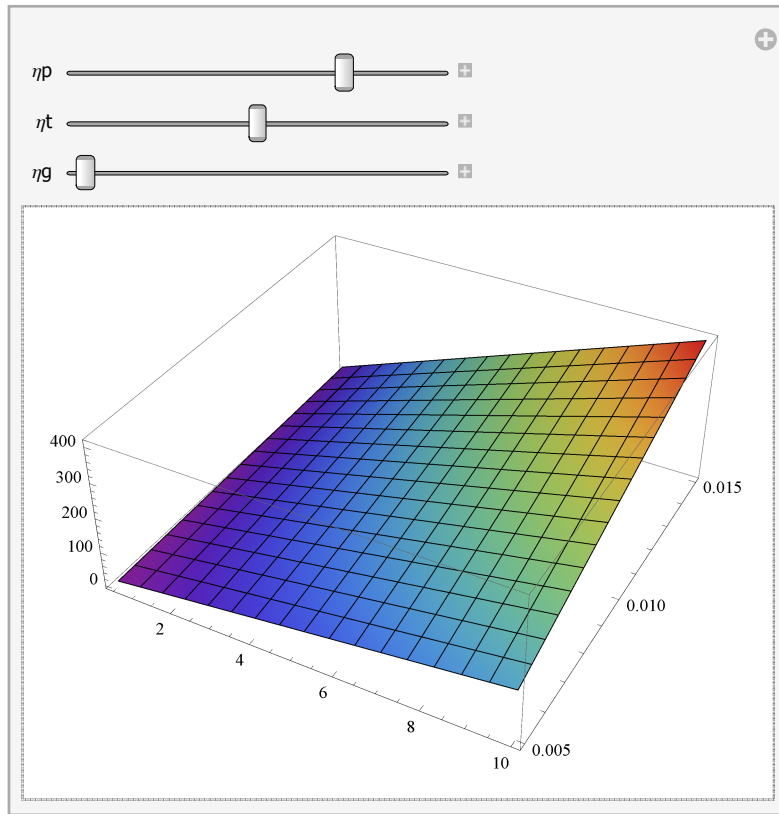
Pro ilustraci je dobré si závislost elektrického výkonu P (W) na výšce spádu h (m) a průtoku Q (m^3/s) znázornit pomocí funkce *Plot3D*.

```
Plot3D[P = h × Q × ρ × g × ηp × ηt × ηg, {h, 0.5, 10},
{Q, 0.005, 0.015}, ColorFunction -> "Rainbow"]
```



Funkci *Plot3D* lze příhodně kombinovat s funkcí *Manipulate*.

```
Manipulate[Plot3D[P = h × Q × ρ × g × ηp × ηt × ηg,
  {h, 0.5, 10}, {Q, 0.005, 0.015}, ColorFunction → "Rainbow"],
{ηp, 0.5, 0.9, 0.1}, {ηt, 0.5, 0.9, 0.1}, {ηg, 0.5, 0.9, 0.1}]
```



Kumulací elektrického výkonu P (W) můžeme získat denně elektrickou energii W (Wh / den) :

$$W = 24 P$$

$$1317.57$$

5.2 Balance spotřeby elektrické energie - v malé chatě

Je nutné spočítat součet denních spotřeb elektrické energie všech spotřebičů $W_D = \sum_{i=1}^n P_{Si} t_{Di}$ (Wh / den). Musíme tedy znát pro každý spotřebič provozní příkon P_{Si} (W) a denní dobu provozu t_{Di} (h / den).

$$W_D = 1000;$$

Dále je nutné počítat s 10% rezervou na spotřebu regulátorů, ztráty ve vedení apod. A_D (Wh / den) je tedy celková očekávaná denní spotřeba.

$$A_D = 1.1 W_D$$

$$1100.$$

A_D (Wh / den) musí být nižší jak W (Wh / den), aby vodní zdroj mohl pokrýt energetické potřeby objektu.

Uvažujeme-li pouze parametr systémového napětí pro nabíjení akumulátorů U_{syst} (V),

$$U_{syst} = 24;$$

potom je požadovaná kapacita akumulátorů C_A (Ah) :

$$C_A = A_D / U_{syst}$$

$$45.8333$$

Pokud chceme, aby baterie pokryly i případný výpadek (např. půl hodiny pro servis turbíny apod.), je třeba vypočítat i koeficient pro nárůst kapacity k_A , který je dán počtem hodin denně 24 a koeficientem hloubky vybití

$h_V (h) :$

$h_V = 0.5;$

$ka = 24 / (24 - h_V)$

1.02128

Potom je minimální kapacita akumulátorů $C_{min} (Ah) :$

$C_{min} = Ca ka / h_V$

93.617

5.3 Ekonomická analýza investice do decentralizovaného zdroje energie

Pro jednoduchou ekonomickou analýzu musíme znát výši investice IN (Kč) (Pozn.: Jde o rozdíl investice do vybudování elektrické přípojky a investice do vybudování vodní elektrárny) a ušetřenou cenu elektrické energie C_{kWh} (Kč / kWh).

$IN = 20\,000;$

$C_{kWh} = 5;$

Potom můžeme určit roční úsporu $C_{úsp}$ (Kč / rok) a prostou návratnost investice R (rok).

$C_{úsp} = 365 Wd C_{kWh} / 1000$

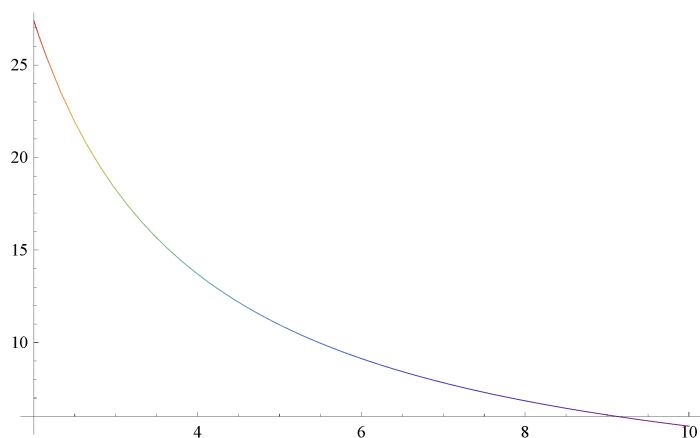
1825

$R = N[IN / C_{úsp}]$

10.9589

Návratnost investice R (rok) si můžeme vyjádřit i graficky v závislosti na ceně elektrické energie C_{kWh} (Kč / kWh).

$\text{Plot}[R = IN / (365 Wd C_{kWh} / 1000), \{C_{kWh}, 2, 10\}, \text{ColorFunction} \rightarrow \text{"Rainbow"}]$



6 Věžňovo dilemma

V roce 1944 byla ustavena nová vědní disciplína s názvem „teorie her“ (Theory of Games). Za zakladatele jsou považováni inmatik John von Neumann a rakouský ekonom Oskar Morgensterne vydáním monografie *Theory of Games and Economic Behavior* (1944). Nemusí být na první pohled zřejmé, že se jedná o ryze matematickou aplikovanou disciplínu, která se zabývá konfliktní situací vycházející z racionálního rozhodování aktérů. Nemějme na mysli pouze agresivní konflikty jako je válka nebo box, ale při jisté míře formalizace lze za konfliktní situaci považovat každou hru, ve které jsou hráči snažící se maximalizovat svou výhru. Teorie her se pak snaží analyzovat tuto situaci a na základě toho vytvořit funkční model, který lze považovat za nejlepší strategii v této „hře“.

Mnohdy si ani nemusíme uvědomovat, že se nacházíme v situaci, která splňuje všechny zmíněné charakteristiky hry. Každodenně se ocitáme v pozici, kdy se rozhodujeme s cílem maximalizovat svůj zisk, například když zápolíme s obchodními řetězci o nejvýhodnější koupi nebo podstupujeme diskuzi s partnerem, kdo bude hlídat děti.

Dalším důvodem, proč se zabývat tímto tématem, může být prostý fakt, že výsledky různých her v rámci teorie her (např. věžňovo dilemma, lov na jelena) se uplatňují v tak odlišných vědeckých disciplínách jako je matematika, ekonomie, sociologie, filosofie, politologie nebo také biologie. Na těchto hrách lze sledovat rozdíl mezi teorií (tj. formalizované zápisy jednotlivých strategií) a praxí (tj. uplatňování těchto strategií při hře samotné). Cílem této kapitoly je tedy představit synergii teorie a praxe u věžňova dilematu vycházejícího z teorie her. Ve výsledku se jedná o neustále se opakující Demingův cyklus postupného zlepšování, který spočívá v plánování, realizaci plánu, kontrole provedení plánu a zahrnutí úprav, které slouží dalšímu plánování, čímž se dostáváme do kruhu onoho zlepšování, či lépe řečeno do spirály. Tento pohyb by měl oscilovat mezi teorií a praxí, jež přináší pochopení dalších významných proměnných ovlivňující formalizaci atd. Tato metoda je v této kapitole uplatňována právě u věžňova dilematu, ale stejně tak může být tento postup využíván i při rozhodování obecně (tj. v managementu).

6.1 Hra s nulovým a nenulovým součtem

Dříve než se dostaneme k věžňovu dilematu, je zapotřebí vysvětlit rozdíl mezi hrou s nulovým a hrou s nenulovým součtem. „Hra s nulovým součtem“ (zero-sum game) se používá pro takové konfliktní situace, kdy hráč vítězí na úkor jiných. Zisk nebo celkový užitek je pak pro každou kombinaci roven nule. Pro snazší představu lze uvést hazardní hru poker, ve které vyhraje ten, kdo obehraje své soupeře o vložené vklady. Výhra po jednotlivých kolech je pak nerovnoměrné přerozdělení vložených částek (většinou jeden vyhrává celý vklad, tzv. bank, a ostatní naopak, vše, co vloží, prohrávají; může však nastat málo pravděpodobná situace v podobě remízy, pak je bank rozdělen na tolik částí, kolik je hráčů ještě ve hře). V takto postavené hře může být vítěz pouze jeden, a to ten, kdo dokáže ukousnout největší díl koláče. Mezi další takové hry patří šachy, tenis, fotbal (všimněme si, že většina sportů).

„Hra s nenulovým součtem“ (non-zero sum game) se odlišuje právě možností mít zisk nebo užitek větší než nula (nebo bohužel i ztrátu větší než původní vklad). To znamená, že mohou nastat kombinace, které nemusí znamenat výhru pouze na úkor dalších hráčů, ale každý z hráčů může obdržet více, než vložil, a to ve vzájemné spolupráci prospěšné oběma stranám (tzv. win-win strategie). Přesto z hlediska jednorázové maximalizace zisku je nejefektivnější podrazit svého protihráče. Jak však budeme argumentovat níže, z dlouhodobější perspektivy se ukazuje, že naopak spolupráce je výhodnější, a to i v případě několika zrad, které mohou od našich spoluhráčů přicházet. S tímto typem her se můžeme setkat daleko častěji než s dříve popisovanou hrou s nulovým součtem. Všude tam, kde se klade důraz na racionální jednání, má smysl hovořit právě o situacích jakožto o hrách s nenulovým součtem (např. při založení obchodní společnosti vkládají zakladatelé podíl, který chtějí společným úsilím v rámci podnikání co nejvíce znásobit. Na druhou stranu může nastat situace, že jejich společný podnik skončí bankrotem a všichni přijdou i o svůj vložený podíl).

	Výhra vs. Vklady	Distribuce výhry
Hra s nulovým součtem	Vždy rovno 0	Vždy jeden hráč
Hra s nenulovým součtem	Může být menší, rovno i větší 0	Může být více hráčů

Tabuľka 1: Typy her

6.2 Společné rysy her

Přesto všechny hry mají některé rysy společné. Prvním z nich je to, že všichni hráči hrají současně, přičemž volí strategie bez znalosti toho, jakou strategii zvolili ostatní. Zároveň platí, že zvolená strategie má vliv na zvyšování či snižování užitku pro jednotlivé hráče. V tomto kontextu se hovoří o strategickém tvaru hry (srov. Fudenberg, Tirole 1994).

Definice - Hra ve strategickém tvaru H je trojice:

$$H = (\{P_i\}_{i \in N}, \{S_i\}_{i \in N}, \{u_i\}_{i \in N})$$

kde

$\{P_i\}_{i \in N}$ představuje konečnou množinu hráčů, kde $N = \{1, 2, \dots, n\}$;

$\{S_i\}_{i \in N}$ je množina strategií těchto hráčů, kde S_i je neprázdná množina strategií i -tého hráče. Prvky těchto množin tvoří čisté strategie.

$\{u_i\}_{i \in N}$ představuje užitky, resp. užitkové funkce, přičemž platí, že

$$u_i : S_1 \times S_2 \times \dots \times S_n \rightarrow \mathbf{R} \text{ pro } i = 1, 2, \dots, n.$$

Hráč tedy usiluje o maximalizaci užitku, na základě volby strategie, aniž by byl obeznámen se strategiemi, které zvolí ostatní.

6.3 Popis věžňova dilematu

Autory věžňova dilematu jsou Merrill Flood a Melvin Dresher, kteří kromě něho popisovali šest dalších experimentů. Nejslavnějším se stalo právě věžňovo dilemma, avšak pod tímto názvem ho poprvé uvádí Albert W. Trucker ve větší snaze přiblížit formální popis Flooda a Dreshera co nejširšímu publiku. Návaznost na tvůrce teorie her je explicitně vyjádřena, když se autoři odvolávají, že právě experimentální hry, mezi které lze bezesporu řadit i dilemma vězně, mají prokázat aplikovatelnost a užitečnost Morgensternovo-Neumanovy teorie (srov. Flood, 1958, s. 6). Obecně se tehdy tyto hry označovaly za problémy při vyjednávání (bargaining problem). Věžňovo dilemma se stalo často citovaným problémem nejenom ve vědeckých publikacích, ale i v každodenním tisku. Donniger (1986, s. 124) zmiňuje, že v šedesátých a sedmdesátých letech bylo napsáno o tomto tématu více než tisíc článků. Proto není divu, že se dilemma vězně objevuje v literatuře v mnoha mutacích (např. Dawkins, 1998, s. 186; Arce, 2010, s. 49; Segal, Hershberger, 1999, s. 36 atd.). Základním předpokladem je myšlenkový experiment týkající se dvou recidivistů, kteří jsou podezřelí ze spáchání velkého zločinu. Ačkoliv vyšetřovatelé mají důkazy pro drobnější delikty, tak aby mohli být podezřelí právoplatně odsouzeni za tento velký zločin, musí učinit doznání. Každý z podezřelých je vyslýchán samostatně a každému je učiněna totožná nabídka. Pakliže se přizná, že skutečně tento zločin spáchali, tak bude propuštěn na svobodu a druhý z obviněných bude potrestán deseti lety žaláře. Každý z nich je tedy přesvědčován k tomu, aby toho druhého zradil. Pokud se ovšem zradí navzájem, tak oba dostanou šest let natvrdo. Jako nejlepší varianta se ukazuje jejich vzájemná spolupráce, kdy sice budou odsouzeni za své drobné prohřešky na jeden rok, nicméně je to nejrychlejší cesta na svobodu pro oba zúčastněné.

	Spolupráce	Zrada
Spolupráce	1:1	5:0
Zrada	0:5	3:3

Tabuľka 2: Věžňovo dilemma

Tento experiment se dá hrát nejenom na roky v žaláři, ale také s penězi, voliči nebo geny. Je však důležité, aby byly výsledky řádně odstupňovány. Pokušení musí nést větší zisk než spolupráce, a ta zase musí být výnosnější než trest za vzájemnou zradu, přičemž oškubání je z pohledu ostatních variant tou nejméně rentabilní (srov. např. Dawkins, 1998, s. 185). Pokušíme si představit, co se asi honí hlavou lidem, než zvolí svůj tah. V podobném duchu se ptá i Dawkins ve své knize Sobecký gen, když za všechny vypočítává. „Vím, že vy můžete zahrát buď spolupráce, nebo zrada. Pokud jste zahráli zrada (takže se řídíte pravým sloupcem), má nejlepší odpověď by byla zrada. Zaplatím sice trest za vzájemnou zradu, ovšem pokud bych zahrál spolupráce, dopadl bych ještě hůře - jako oškubaný. Nyní zvažme druhou z možností, tedy že jste zahráli spolupráce. Z levého sloupce vidíme, že 'zrada' je pro mě opět ten nejlepší tah" (Dawkins, 1998, s. 185).

		Můj tah	
		Spolupráce (A)	Zrada (B)
Tah protihráče	Spolupráce (1)	ODMĚNA (celkem dobré)	OŠKUBÁNÍ (velmi špatné)
	Zrada (2)	POKUŠENÍ (velmi dobré)	ZRADA (dost špatné)

Tabuľka 3: Věžňovo dilemma z pohledu hráče

Podle této úvahy se zdá, že nejlepším řešením tohoto dilematu je vždy zradit. Ke stejnému závěru dochází i další významný pokračovatel tvůrců teorie her, jakým byl John Nash. Tento nositel Nobelovy ceny za ekonomii (2004) je tvůrce tzv. Nashova ekvilibria. Za ekvilibrium (rovnovážný bod) lze považovat stav vzájemné rovnováhy všech částí nějakého celku nebo za vyvážení proti sobě působících sil. Každá konečná hra (finite game) má rovnovážný bod (Nash, 1951, s. 288). K Nashovu ekvilíbriu tedy dochází, když všichni hráči optimalizují svůj zisk, přičemž zároveň jsou známy všechny proměnné ostatních hráčů, podle kterých se rozhodují. Znamená to, že volba protihráčů nijak nemůže ovlivnit mou strategii (tzn. narušit rovnováhu). Od toho je potřeba odlišit optimalitu v Paretově smyslu (např. Marada, 2006, s. 12; Cooper et al., 1996, s. 188), kterou lze vyjádřit jako nejvyšší dosažitelný zisk pro všechny zúčastněné.

Pakliže aplikujeme Nashovo ekvilibrium i optimalitu v Paretově smyslu na věžňovo dilemma, tak získáme odlišná řešení. Zatímco optimalita se týká možností uvedených v předešlé tabulce jako A1, tak Nashovo ekvilibrium splňuje alternativa B2 (srov. Marada, 2006, s. 14). Ačkoliv se může zdát, že varianta A1 je oboustranně výhodná, tak je zde veliké riziko pokušení, a proto je nejstabilnější za všech okolností zrazovat (B2). A právě toto činí z tohoto úkolu dilemma. Je zřejmé, že nejstabilnější strategií je vždy zradit, nicméně nelze si nevšimnout, že při spolupráci lze daleko více získat. Jak tedy ven z tohoto paradoxu?

6.4 Turnaj R. Axelroda ve věžňovo dilematu

Zajímavé řešení tohoto paradoxu mohou přinést výsledky Roberta Axelroda, který uspořádal soutěž různých strategií aplikovaných na toto dilemma. Při této hře na "jedno kolo" (one-shot game) máme pouze dvě možnosti, byť jsme uváděli důvody pro zvolení zejména jedné z nich (zrady). Na druhé straně se při opakování (tzn. dalších kol) nabízí celá řada různých strategií, u nichž nemusí být na první pohled zřejmé, která z nich je nejlepší. Axelrod proto vyzval odborníky na teorii her, aby mu zaslali své strategie do turnaje ve věžňově dilematu. Jednalo se o naprogramované sekvence, kterých nakonec přišlo čtrnáct. Axelrod doplnil ještě strategii „Náhoda“, která spolupracovala či zrazovala zcela náhodně. Je zřejmé, že zaslané strategie by musely být hodně špatné, pokud by dosahovaly obdobných výsledků jako program označený za náhodu. Turnaj probíhal systémem každý s každým, přičemž jedna hra obsahovala 200 kol. „Body byly přiděleny podle následujícího schématu: odměna (za vzájemnou spolupráci) 3 body, pokušení (za jednostrannou zradu) 5 bodů, trest (za vzájemnou zradu) 1 bod a oškubání (spolupráce se zrádcem) 0 bodů (jako ekvivalent těžkého trestu z příkladu se skutečnými vězni)" (Dawkins, 1998,

s. 190). V rámci jednoho kola bylo možné získat maximálně 1 000 bodů (200 tahů po 5 bodech). V případě Nashova ekvilibria by v rámci jednoho utkání bylo možné získat minimálně 200 bodů. Optimalita v Paretově smyslu by přinesla 600 bodů. Nicméně není překvapivé, že většina strategií nedosáhne za jeden zápas ani těch 600 bodů, chcete-li, standardu. Pro lepší představu uvádíme v další podkapitole celou škálu různých strategií uplatňovaných ve věžňovu dilematu včetně několika původních strategií, které se vyskytly ve zmiňovaném turnaji pořádaném Robertem Axelrodem.

6.5 Popis vybraných strategií věžňova dilematu

Půjčka za oplátku (angl. Tit for Tat) - tato strategie začíná spoluprací. Jestliže je zrazena v příštím kole zradí taky. V jiných případech spolupracuje.

Půjčka za dvě oplátky (angl. Tit for Two Tats) - i tato strategie začíná spoluprací. Zrazuje až po dvou za sebou jdoucích zradách spoluhráče. V jiných případech spolupracuje.

Škodolibá (angl. Spiteful) - jedná se o další strategii, která začíná spoluprací, u níž zůstává až do té doby, než je zrazena. Poté však zrazuje až do konce.

Karma (angl. Joss) - začíná spoluprací. Jestliže je však zrazena, tak zrazuje také. Karma spolupracuje v 90 % případech (je zde tedy prostor pro nenadálou zradu ve zbývajících 10 %).

Tester (angl. Tester) - tato strategie začíná zradou. Jestliže je zrada z prvního tahu oplacena, tak dále hraje v souladu se strategií Půjčka za oplátku, v opačném případě hraje strategii Spolupráce-zrada (tzn. střídá spolupráci a zradu bez ohledu na tahy protihráče).

Pavlov (angl. Pavlov) - tato strategie pro změnu začíná spoluprací, pak spolupracuje, až do té doby, kdy oba hráči zvolili stejný tah, v jiných případech zrazuje.

Nedůvěra (angl. Mistrust) - začíná zradou. Jestliže je zrazena, tak zrazuje také. V jiných případech spolupracuje.

Výzkumník (angl. Prober) - Začíná následující trojicí tahů: spolupráce, zrada, zrada. Jestliže spoluhráč spolupracuje současně ve druhém i třetím tahu, tak už hraje pouze zradu. Jestliže oponent spolupracuje ve druhém, anebo ve třetím tahu, tah dále hraje strategii Půjčku za oplátku.

Měkká většina (angl. Soft Majority) - Začíná spoluprací. Tato strategie hraje takovým způsobem, že volí tah, který tvoří většinu z předešlých tahů jejího oponenta se závazkem spolupracovat (tzn. v případě 50 % volí spolupráci).

Tvrdá většina (angl. Hard Majority) - Začíná zradou. Hraje takovým způsobem, že volí tah, který tvoří většinu z předešlých tahů jejího oponenta se závazkem zrazovat (tzn. v případě 50 % volí zradu).

Náhoda (angl. Random) - Náhodně vybírá spolupráci, nebo zradu.

Vždy spolupráce (angl. All C) - tato strategie vždy spolupracuje.

Vždy zrad' (angl. All D) - tato strategie vždy zrazuje.

Spolupráce-zrada (angl. PerCD) - tato strategie střídá spolupráci se zradou bez ohledu na to, jaké tahy volí oponent (tzn. spolupráce-zrada-spolupráce-zrada atd.).

Dvakrát spolupráce a pak zrada (angl. PerCCD) - v této strategii dochází ke stále se opakujícímu cyklu těchto tahů: spolupráce, spolupráce, zrada, a to bez ohledu na to, jaké tahy volí oponent.

Dvakrát zrada a pak spolupráce (angl. PerDDC) - v této strategii dochází ke stále se opakujícímu cyklu těchto tahů: zrada, zrada, spolupráce, a to opět bez ohledu na to, jaké tahy volí oponent.

6.6 Demonstrace věžňova dilematu

Příklad č. 1:

Nyní si můžete vyzkoušet, jakých výsledků byste dosáhli při hře s jednou ze strategií, z nichž některé jsou z původního Axelrodova turnaje.

```
Manipulate[
  If[chosen ≠ strategy, history = {}; chosen = strategy];
  If[turn,

    player2 = PrisonersDilemma[history, strategy, 2];

    history = Append[history, {player1, player2}]; turn = False ];

  Row[{

    Grid[
      Join[{{"Hráč 1", "Hráč 2 (PC)"}, history /. {0 → "Zrada", 1 → "Spolupráce"}], Frame →
        True, Dividers → {{True, True, True}, {True}}, Background → {None, {LightBlue}}]

    ],

    (*Ovládací prvky*)Style["Věžňovo dilema (hráč vs PC)", Bold, 20],
    Spacer[5], {turn, False, ControlType → None},
    {history, {}, ControlType → None}, {player1, ControlType → None},
    {player2, ControlType → None}, {{strategy, "Tit for Tat", "Typ strategie"},
      {"Tit for Tat" → "Tit for Tat", "Tit for Two Tats" → "Tit for Two Tats",
        "Spiteful" → "Spiteful", "Random" → "Random", "AllC" → "AllC", "All" → "All",
        "PerCD" → "PerCD", "PerCCD" → "PerCCD", "PerDDC" → "PerDDC", "Joss" → "Joss",
        "Tester" → "Tester", "Pavlov" → "Pavlov", "Mistrust" → "Mistrust",
        "Soft Majority" → "Soft Majority", "Hard Majority" → "Hard Majority",
        "Prober" → "Prober"}}, {chosen, ControlType → None},
    Row[{Style["Volba hráče"], Spacer[12], Button["Zrada", player1 = 0; turn = True],
      Spacer[8], Button["Spolupráce", player1 = 1; turn = True]}],
    Button["Vynulovat", history = {}],
    (*Tabulka se skóre*)Dynamic[Grid[{
      {Style["Skóre", 18, Bold], SpanFromLeft},
      {Style["Hráč 1"], Style["Hráč 2 (PC)"]},
      If[Length[history] > 1, Map[Apply[Plus, #] &,
        Transpose[history /. {{0, 0} → {3, 3}, {1, 0} → {5, 0}, {0, 1} → {0, 5}}]],
        Flatten[history /. {{0, 0} → {3, 3}, {1, 0} → {5, 0}, {0, 1} → {0, 5}}]]
      }, Frame → True, Background → {None, {LightOrange}}]],
    Alignment → Center, ControlPlacement → Left,
    (*Zde jsou počáteční podmínky*)Initialization ⇒ {chosen = strategy,
      (*Tato funkce obsahuje metodiku daných strategií*)
      PrisonersDilemma[hist_, strategy_, player_] :=
        Switch[strategy,
          "Random", RandomInteger[],
```

```

"All", 0,

"AllC", 1,

"PerCD", Part[{1, 0}, (Length[hist] + 1) - 2 * (Ceiling[(Length[hist] + 1) / 2] - 1)],

"PerCCD",
Part[{1, 1, 0}, (Length[hist] + 1) - 3 * (Ceiling[(Length[hist] + 1) / 3] - 1)],

"PerDDC",
Part[{0, 0, 1}, (Length[hist] + 1) - 3 * (Ceiling[(Length[hist] + 1) / 3] - 1)],

"Tit for Tat",
If[Length[hist] == 0, 1, If[hist[[-1, Switch[player, 1, 2, 2, 1]]] == 0, 0, 1]],

"Tit for Tow Tats",
If[Length[hist] == 0, 1, If[hist[[-1, Switch[player, 1, 2, 2, 1]]] == 0 &&
  hist[[-2, Switch[player, 1, 2, 2, 1]]] == 0, 0, 1]],

"Spiteful", If[Length[hist] == 0, 1,
  If[MemberQ[hist[All, Switch[player, 1, 2, 2, 1]], 0], 0, 1]],

"Pavlov", If[Length[hist] == 0, 1,
  If[hist[[-1, Switch[player, 1, 2, 2, 1]]] == hist[[-1, player]], 1, 0]],

"Joss", If[Length[hist] == 0, 1, If[hist[[-1, Switch[player, 1, 2, 2, 1]]] == 0,
  0, First@RandomChoice[{0.1, 0.9} -> {0, 1}, 1]]],

"Tester", If[Length[hist] == 0, 0,
  If[Length[hist] == 1, 1, If[hist[[2, Switch[player, 1, 2, 2, 1]]] == 0,
    If[hist[[-1, Switch[player, 1, 2, 2, 1]]] == 0, 0, 1],
    Part[{0, 1}, (Length[hist] + 1) - 2 * (Ceiling[(Length[hist] + 1) / 2] - 1)]]],

"Mistrust", If[Length[hist] == 0, 0,
  If[MemberQ[hist[All, Switch[player, 1, 2, 2, 1]], 0], 0, 1]],

"Soft Majority", If[Length[hist] == 0, 1, Which[
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) >
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 1,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) <
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 0,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) ==
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 1,
  True, Print["Error"]]],

"Hard Majority", If[Length[hist] == 0, 0, Which[
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) >
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 1,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) <
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 0,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) ==
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 0,
  True, Print["Error"]]],

"Prober",

```

```

Switch[Length[hist],
  0, 1,
  1, 0,
  2, 0,
  _, If[hist[[2, Switch[player, 1, 2, 2, 1]] == 1 && hist[[3, Switch[player, 1, 2, 2, 1]] ==
    1, 0, If[hist[[1, Switch[player, 1, 2, 2, 1]] == 0, 0, 1]]
]
]]]

```

Příklad č. 2:

Dále máte možnost postavit libovolné strategie proti sobě a zvolit počet kol, ve kterých se ony strategie utkají.

Manipulate[

```

If[chosenP1 ≠ strategyP1 || chosenP2 ≠ strategyP2,
  history = {}; chosenP1 = strategyP1; chosenP2 = strategyP2];
If[Length[history] ≥ turns, trigg = False];
If[trigg,
  Pause[.2];
  player1 = PrisonersDilemma[history, strategyP1, 1];
  player2 = PrisonersDilemma[history, strategyP2, 2];
  history = Append[history, {player1, player2}];
];
Grid[
  Join[{{"Hráč 1", "Hráč 2(PC)"}, history /. {0 → "Zrada", 1 → "Spolupráce"}],
  Frame → True, Dividers → {{True, True, True}, {True}},
  Background → {None, {LightBlue}}],

(*Ovládací prvky*)Style["Věžňovo dilema (simulace)", Bold, 20], Spacer[5],
{turn, False, ControlType → None}, {history, {}, ControlType → None},
{player1, ControlType → None}, {player2, ControlType → None},
{{strategyP1, "Tit for Tat", "Hráč 1"}, {"Tit for Tat" → "Tit for Tat",
  "Tit for Two Tats" → "Tit for Two Tats", "Spiteful" → "Spiteful",
  "Random" → "Random", "AllC" → "AllC", "All" → "All", "PerCD" → "PerCD",
  "PerCCD" → "PerCCD", "PerDDC" → "PerDDC", "Joss" → "Joss", "Tester" → "Tester",
  "Pavlov" → "Pavlov", "Mistrust" → "Mistrust", "Soft Majority" → "Soft Majority",
  "Hard Majority" → "Hard Majority", "Prober" → "Prober"}},
{{strategyP2, "Tit for Tat", "Hráč 2"}, {"Tit for Tat" → "Tit for Tat",
  "Tit for Two Tats" → "Tit for Two Tats", "Spiteful" → "Spiteful",
  "Random" → "Random", "AllC" → "AllC", "All" → "All", "PerCD" → "PerCD",
  "PerCCD" → "PerCCD", "PerDDC" → "PerDDC", "Joss" → "Joss", "Tester" → "Tester",
  "Pavlov" → "Pavlov", "Mistrust" → "Mistrust", "Soft Majority" → "Soft Majority",

```

```

    "Hard Majority" → "Hard Majority", "Prober" → "Prober"}},
{{turns, 10, "Počet tahů"}, InputField[#, Number] &},
Button[Dynamic@If[trigg, "Stop", "Start"],
  If[trigg, trigg = False, trigg = True; history = {}]],
Button["Vynulovat", history = {}],
(*Tabulka se skóre*)Dynamic[Grid[{
  {Style["Skóre", 18, Bold], SpanFromLeft},
  {Style["Hráč 1"], Style["Hráč 2"]},
  If[Length[history] > 1, Map[Apply[Plus, #] &,
    Transpose[history /. {{0, 0} → {3, 3}, {1, 0} → {5, 0}, {0, 1} → {0, 5}}]],
  Flatten[history /. {{0, 0} → {3, 3}, {1, 0} → {5, 0}, {0, 1} → {0, 5}}]]
}, Frame → True, Background → {None, {LightOrange}}]],

{chosenP1, ControlType → None}, {chosenP2, ControlType → None},
{trigg, ControlType → None}, Alignment → Center, ControlPlacement → Left,
(*Zde jsou počáteční podmínky*)
Initialization ⇒ {chosenP1 = strategyP1, chosenP2 = strategyP2, trigg = False,
  (*Tato funkce obsahuje metodiku daných strategií*)
  PrisonersDilemma[hist_, strategy_, player_] :=
    Switch[strategy,
      "Random", RandomInteger[],

      "All", 0,

      "AllC", 1,

      "PerCD", Part[{1, 0}, (Length[hist] + 1) - 2 * (Ceiling[(Length[hist] + 1) / 2] - 1)],

      "PerCCD",
      Part[{1, 1, 0}, (Length[hist] + 1) - 3 * (Ceiling[(Length[hist] + 1) / 3] - 1)],

      "PerDDC",
      Part[{0, 0, 1}, (Length[hist] + 1) - 3 * (Ceiling[(Length[hist] + 1) / 3] - 1)],

      "Tit for Tat",
      If[Length[hist] == 0, 1, If[hist[[1, Switch[player, 1, 2, 2, 1]]] == 0, 0, 1]],

      "Tit for Tow Tats",
      If[Length[hist] == 0, 1, If[hist[[1, Switch[player, 1, 2, 2, 1]]] == 0 &&
        hist[[2, Switch[player, 1, 2, 2, 1]]] == 0, 0, 1]],

      "Spiteful", If[Length[hist] == 0, 1,
        If[MemberQ[hist[[All, Switch[player, 1, 2, 2, 1]]], 0], 0, 1]],

      "Pavlov", If[Length[hist] == 0, 1,
        If[hist[[1, Switch[player, 1, 2, 2, 1]]] == hist[[1, player]], 1, 0]],

      "Joss", If[Length[hist] == 0, 1, If[hist[[1, Switch[player, 1, 2, 2, 1]]] == 0,
        0, First@RandomChoice[{0.1, 0.9} → {0, 1}, 1]]],

      "Tester", If[Length[hist] == 0, 0,
        If[Length[hist] == 1, 1, If[hist[[2, Switch[player, 1, 2, 2, 1]]] == 0,
          If[hist[[1, Switch[player, 1, 2, 2, 1]]] == 0, 0, 1],
          Part[{0, 1}, (Length[hist] + 1) - 2 * (Ceiling[(Length[hist] + 1) / 2] - 1)]]],

      "Mistrust", If[Length[hist] == 0, 0,

```

```

If[MemberQ[hist[All, Switch[player, 1, 2, 2, 1]], 0], 0, 1]],

"Soft Majority", If[Length[hist] == 0, 1, Which[
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) >
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 1,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) <
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 0,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) ==
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 1,
  True, Print["Error"]]],

"Hard Majority", If[Length[hist] == 0, 0, Which[
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) >
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 1,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) <
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 0,
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 1] / Length[hist]) ==
  (Count[hist[All, Switch[player, 1, 2, 2, 1]], 0] / Length[hist]), 0,
  True, Print["Error"]]],

"Prober",
Switch[Length[hist],
  0, 1,
  1, 0,
  2, 0,
  _, If[hist[2, Switch[player, 1, 2, 2, 1]] == 1 && hist[3, Switch[player, 1, 2, 2, 1]] ==
    1, 0, If[hist[-1, Switch[player, 1, 2, 2, 1]] == 0, 0, 1]]
]
}]
]

```

Vězňovo dilema (simulace)

Hráč 1: Tit for Tat [V]

Hráč 2: Tit for Tat [V]

Počet tahů: 10

[Start]

[Vynulovat]

Skóre

	Hráč 1	Hráč 2
Hráč 1		
Hráč 2		

Hráč 1 | Hráč 2 (PC)

6.7 Nejlepší strategie vězňova dilematu

Nakonec zvítězila ta nejjednodušší strategie, která se jmenovala “Půjčka za oplátku” (Tit for Tat). Naprogramoval ji Anatol Rapoport a byla založena na tom, že v prvním tahu tato strategie spolupracuje a pak už jenom kopíruje tahy svého spoluhráče. Když spoluhráč spolupracuje, tak tato Půjčka za oplátku spolupracuje také. Jestliže zradí, zradí rovněž. Protihráč tedy rozhoduje o výsledném zisku, který se zvyšuje s počtem tahů, kdy spolupracuje.

Téměř každá strategie měla zabudovaný mechanismus odvety za zradu. Axelrod oddělil “milé” (nice) od “podlých” (nasty) strategií. Zatímco milé strategie nikdy nezradí jako první (viz Půjčka za oplátku), tak podle strategie to dělávají. Mezi milé strategie patří kromě již zmíněné Půjčky za oplátku také Půjčka za dvě oplátky, zatímco mezi podlé strategie patří Nedůvěra, Tester či Výzkumník. Je pozoruhodné, že z 15 strategií, které se zúčastnily Axelrodova turnaje, se na prvních místech umístily právě milé strategie. “Půjčka za oplátku získala v průměru 504, 5 za jednu hru, tedy velmi dobrý výtěžek – 84 % standardu. Ostatní milé strategie se umístily jen o něco za ní s výtěžkem 83,4 % do 78,6 % standardu. Pak následovala veliká mezera mezi nimi a 66,3 %, které získala strategie zaslaná pod jménem Graaskamp, nejúspěšnější z podlých strategií” (Dawkins, 1998, s.192). Tato nejúspěšnější z podlých strategií se však v prvních padesáti tazích chovala stejně jako Půjčka za oplátku a až v následujícím tahu zradila. Další členění může být na “odpouštějící” (forgiving) a “neodpouštějící” (non-forgiving) strategie. Základním rysem odpouštějící strategie je možnost vymanit se z cyklu vzájemných zrad a po případné zradě zvolit opět možnost spolupráce.

Kategorie strategie	Kritérium
"milé" / "podlé"	nezradí první / může zradit první
"odpouštějící" / "neodpouštějící"	i po zradě zvolí spolupráci / po zradě už vždy jednom zrazuje

Tabulka 4: Věžňovo dilema

Podle těchto informací se tedy zdá, že nejlepší strategií při řešení věžňova dilematu je milá a odpouštějící strategie, kterou Rapoport nazval Půjčkou za oplátku. Pokud se však jedná o jednokolovou hru, tak je za všech okolností výhodnější zradit. Tento závěr lze uplatnit i při hře, u které dopředu víme, na kolik kol se hraje. Poslední kolo se totiž stává obdobou jednokolové hry, kdy racionální aktér by měl podle všeho zradit.

6.8 Věžňovo dilema v praxi

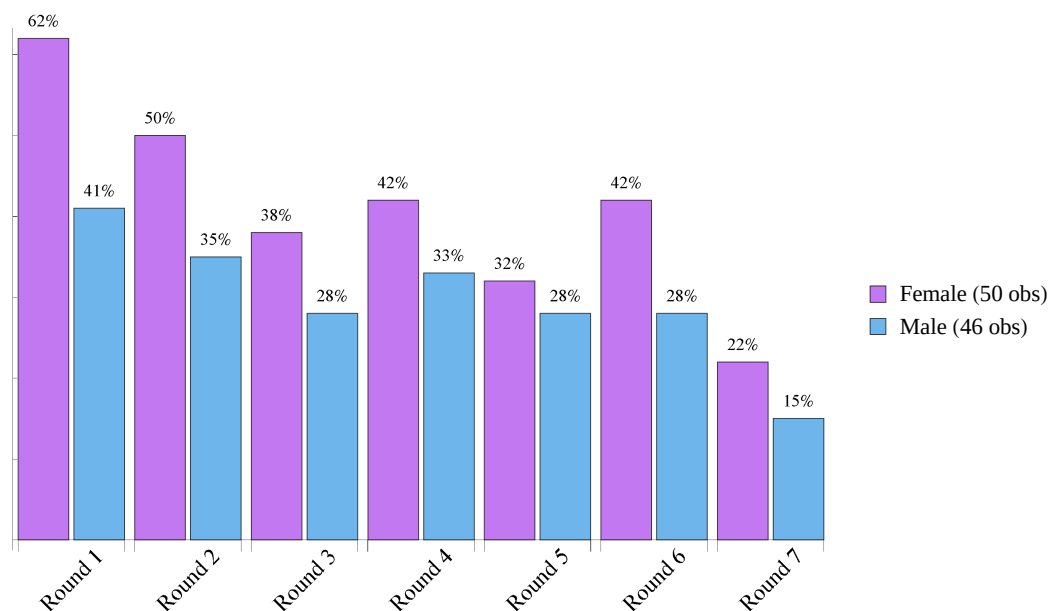
V předešlé kapitole jsme formulovali teoretické závěry věžňova dilematu. Dalo by se tedy očekávat, že se podle nich budou řídit i lidé v rámci experimentů aplikujících toto dilema v praxi. Používám přitom Demingův cyklus (Plan - Do - Check - Act) jako metaforu. Tato spirála postupného zlepšování na základě zpětné vazby začíná plánováním zlepšení (Plan), které je následně zrealizováno (Do), aby mohlo být ověřeno (Check) a následně plošně implementováno (Act). V našem případě se jedná o zlepšování herní strategie věžňova dilematu. Tímto by se podle předešlých informací měla stát strategie zrady v jednokolové hře a v případě více kolové hry Půjčka za oplátku. Dříve než se dostaneme k samotnému popisu některých experimentů, během nichž bylo aplikováno věžňovo dilema, tak je potřeba uvést významné proměnné, které zvyšují ochotu aktérů spolupracovat. Patří mezi ně příbuzenství (Hamilton, 1964 a, 1964 b), věk, gender, příslušnost k sociální třídě, rasa (např. Sampson, Kardush, 1965), postavení ve skupině (Brown, Abrams, 1986), vliv genetiky, obdobného IQ, sociální blízkosti (Segal, Hersberger, 1999) atd. Samotné zvýšení ochoty ke spolupráci lze chápat jako odchylku od nejlepší strategie ve věžňově dilematu alespoň v rámci jednokolové hry. V případě více kolové hry není jednoznačné, zda vede zvýšená ochota ke spolupráci k lepším výsledkům. Segal a Hersberger (1999) zkoumali rozdíly při hraní věžňova dilematu u 59 jednovaječných (monozygotních - MZ) a 37 dvojvaječných (dizygotních - DZ) dvojčat. Zatímco jednovaječná dvojčata mají 100 % genetickou shodu, tak dvojvaječná stejně jako i ostatní sourozenci mají shodu pouze 50 %. Ve výzkumu Segala a Hersbergera se potvrdila hypotéza, že jednovaječná dvojčata mezi sebou spolupracují čteněji než dvojvaječná (Segal, Hersberger, 1999, s. 34). Zdá se tedy, že existují genetické předpoklady pro spolupráci.

Zygotnost	Počet	Body - průměr / (rozmezí)
MZ	59	339,48
Rozmezí		(200–531)
Muži	21	328,62
Ženy	38	345,47
DZ	37	304,27
Rozmezí		(200–451)
Muži	13	271,31
Ženy	24	322,13
Pohlaví celkem		
Muži	34	306,71
Ženy	62	336,44

Tabulka 5: Věžňovo dilema u monozygotních a dizygotních dvojčat (Segal, Hershberger, 1999, p. 39)

Další autoři poukazují na rozdílné osobnostní typy, z nichž některé vždy spolupracují (např. Andreoni, Miller, 1993; Cooper et al., 1996; McKelvey, Palfrey, 1992). V tomto případě ponecháme stranou, zda se jedná o ryzí altruisty nebo o postupné budování důvěry (např. Cooper et al., 1996), byť se jedná přibližně o 12 % - 13 % (Cooper et al., 1996, s. 189), nebo pouze o 5 % (McKelvey, Palfrey, 1992, s. 820) z celkové populace. Nic to nemění na tom, že se experimentálně potvrdilo, že někteří lidé stále spolupracují nejenom ve více kolové hře, ale i v jednokolových hrách (srov. Cooper et al., 1996, s. 188). Spolupráci dokonce mohou zvyšovat signály, které prozrazují, jaký osobnostní typ je váš protihráč. Jedná se především o signály doprovázené silnými emocemi, mezi které patří sympatie, soucit, vina a stud (srov. Brosig, 2002, s. 277). Je důležité, aby tyto signály nepodléhaly volní kontrole, čímž by bylo přinejmenším velmi obtížné je napodobit. Zároveň je však musí postřehnout druhá strana a jak jinak než na základě emocionální citlivosti, chcete-li, empatie. Emoce se tak stávají obtížně kontrolovatelnými komunikačními kanály, které zvyšují pravděpodobnost spolupráce mezi těmi, kteří tyto emoce vzájemně pozorují, a tedy i zakoušejí. “Schopnost rozpoznat, zdali jsou druzí ochotni spolupracovat, by proto měla být připsána emočnímu nastavení jedince, stejně jako sama dispozice ke spolupráci” (Brosig, 2002, s. 277).

Míra spolupráce při experimentování v rámci věžňova dilematu závisí také na pohlaví. Ženy daleko častěji volí v této hře spolupráci než muži (srov. Ortmann, Tichy, 1999, s. 329).



Graf 1: Míra spolupráce dle pohlaví po jednotlivých kolech (Ortmann, Tichy, 1999, p. 335)

Tento závěr potvrzují i výše uvedené výsledky ochoty spolupráce na základě zygotnosti. Je velmi důležité, aby disciplína, která byla vynalezena a dále rozpracována muži, byla také více zpřístupněna ženám. Toto přehlížení

druhého pohlaví, jak explicitně upozorňuje titul nejznámější knihy Simon de Beauvoirové, vede ke zkreslování teoretických znalostí. Toto tvrdošíjné prosazování ryze mužské optiky kritizovala Gilliganová v rámci teoretických předpokladů Kohlbergova popisu morálního vývoje. Kohlberg je tvůrce metodiky, kdy na základě řešení Heinzova dilematu (zda ukrást předražený lék pro svou těžce nemocnou ženu) rozlišuje několik etap morálního vývoje. Podle Gilliganové škála vytvořená Kohlbergem zvýhodňuje muže (srov. Hunt, 2000, s. 361 nn.; Gilliganová, 2001). Podobné námitky lze vznést i na dřívější teoretické formulace nejlepší strategie věžňova dilematu. Dopadl by stejně Axelrodem pořádaný turnaj o nejlepší strategii ve věžňově dilematu, jestliže by se ho zúčastnily ženy? Nejsou sice známy identity všech účastníků turnajů, které pořádal Axelrod, přesto se lze oprávněně domnívat, že pokud to nebyli pouze samí muži, tak alespoň drtivá většina.

6.9 Závěr

V této kapitole je představeno věžňovo dilema a jeho aplikace velice povrchně. Nelze reflektovat celou diskusi o tomto příspěvku teorie her nejenom z důvodu velkého počtu publikací, ale zároveň i pestrosti vědeckých disciplín, v nichž jsou realizovány experimenty využívající věžňova dilematu. Klíčovou otázkou zůstává, proč rozšiřovat popsané stránky o tomto oblíbeném tématu. Mezi nesporné výhody teorie her patří detailní popis pravidel, dynamiky procesů, aktérů a všech dalších okolností, které lze vyjádřit souhrnným termínem hra. Přesto jsme od sebe rozlišili hry s nulovým součtem a hry s nenulovým součtem. Pouze u některých her s nenulovým součtem může na základě spolupráce docházet ke vzájemnému obohacování zúčastněných aktérů. Právě věžňovo dilema je takovou hrou. Přesto lze uvést společné rysy všech her vyjádřené v následující formulaci:

$$H = (\{P_i\}_{i \in N}, \{S_i\}_{i \in N}, \{u_i\}_{i \in N})$$

Kdy se tedy konečný počet hráčů (P) snaží maximalizovat své užitky (u) prostřednictvím volby strategií (S), aniž by byli vzájemně obeznámeni se strategiemi svých spoluhráčů. Věžňovo dilema lze v tomto smyslu pokládat za příklad takové hry, která je však zároveň velmi jednoduchou hrou. A ačkoliv každý hráč má pouze dvě možnosti (spolupracovat nebo zradit), tak v případě více kolových experimentů dochází k rozdílným strategiím, které vedou i k rozdílnému výsledku (např. počtu bodů). Na druhou stranu tato jednoduchost umožňuje zkoumat celou řadu faktorů, které mohou zvyšovat míru spolupráce nebo zrady. V teoretické části jsme uvedli důvody, proč je nejlepší strategií v jednokolovém věžňově dilematu zradit a ve více kolové verzi uplatňovat strategii Půjčka za oplátku (v prvním kole spolupracovat a v dalších zrcadlit to, co zvolil v předešlém tahu protihráč). V praktické části jsme uváděli celou řadu faktorů (např. genetické predispozice, osobnostní typ, pohlaví atd.), které zvyšují ochotu ke spolupráci. Přesto by bez teoretického modelu byly praktické aplikace věžňova dilematu těžko ukotvitelné. V této chvíli celá řada experimentů přináší fakta o důležitosti jednotlivých faktorů. V následující fázi bude potřeba tato fakta utřídit a znovu formalizovat na vyšší úrovni. Tento proces může připomínat známý Demingův cyklus (P-D-C-A), na kterém se však uplatňují i na první pohled nesourodé vědecké disciplíny jako informatika, sociologie, ekonomie, psychologie, politologie, matematika nebo biologie. Ve výsledku získáme velmi komplexní pohled na mnoho každodenních situací. Všude tam, kde dochází ke střetu zájmu jednotlivců či celých skupin, lze hovořit o hře, která může mít všechny rysy věžňova dilematu.

7 Finanční matematika

7.1 Umořování dluhu

Zadání

Půjčka 96 500 Kč bude umořována ročními splátkami ve výši 9 650 Kč.

Roční úroková míra je 4,5%.

a) Stanovte umořovací plán.

b) Graficky zachyťte průběh umořování dluhu.

Řešení

Především je třeba stanovit počet splátkových období n . Když označíme

u_0 výši půjčky,

q úrokovou míru,

d výši konstantních periodických splátek,

pak počet splátkových období n se rovná nejbližší vyššímu celému číslu, pro které

$$\frac{\ln d - \ln(d - u_0)}{\ln(1 + q)}.$$

Když s_k je výše splátky v k -tém splátkovém období, spočteme

ur_k část splátky s_k , která soužijí k zaplacení úroků v k -tém splátkovém období,

pl_k část splátky s_k , která slouží k umořování zbytkového dluhu v k -tém splátkovém období,

u_k výši zbývajcího dluhu v k -tém splátkovém období.

Výpočet provedeme pomocí vztahů

$$ur_k = \begin{cases} 0 & \text{když } k = 0 \\ qu_{k-1} & \text{když } k = 1, \dots, n \end{cases}$$

$$pl_k = \begin{cases} 0 & \text{když } k = 0 \\ s_k - qu_{k-1} & \text{když } k = 1, \dots, n \end{cases}$$

$$u_k = \begin{cases} 0 & \text{když } k = 0 \\ (1 + q)u_{k-1} - s_k & \text{když } k = 1, \dots, n \end{cases}$$

```

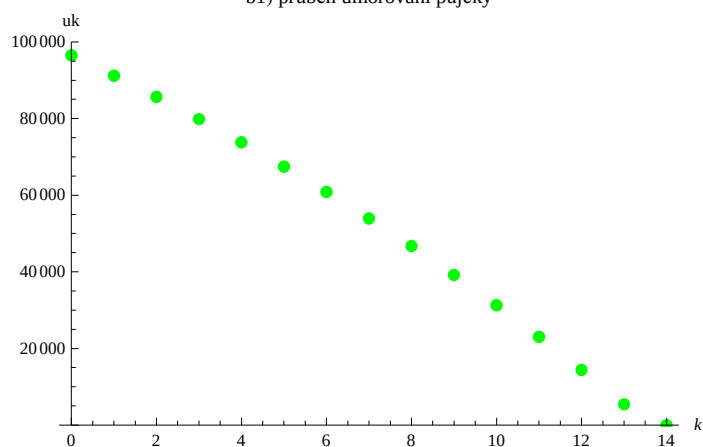
q = 0.045; d = 9650; u[0] = 96500;
n = Round[(Log[d] - Log[d - q * u[0]]) / Log[1 + q]];
" je počet platebních období " n
Clear[s, ur, pl, u, k];
s[0] = 0; ur[0] = 0; pl[0] = 0; u[0] = 96500;
s[k_] := d;
ur[k_] := q * u[k - 1];
u[k_] := (1 + q) * u[k - 1] - s[k];
pl[k_] := s[k] - q * u[k - 1];
s[n] = (1 + q) * u[n - 1];
tuu = Table[{k, s[k], ur[k], pl[k], u[k]}, {k, 0, n, 1}];
Panel[TableForm[tuu, TableHeadings -> {None, {"k", "sk", "urk", "plk", "uk"}},
"a) Umořovací plán"]
ListPlot[Table[{k, u[k]}, {k, 0, n, 1}], PlotRange -> {0, 100000},
PlotStyle -> {Green, PointSize[0.02]}, AxesLabel -> {k, uk},
PlotLabel -> "b1) průběh umořování půjčky"]
ListPlot[{Table[{k, ur[k]}, {k, 0, n, 1}], Table[{k, pl[k]}, {k, 0, n, 1}]},
PlotRange -> {0, 10000}, PlotStyle -> PointSize[0.02], AxesLabel -> {k, {urk, plk}},
PlotLabel -> "b2) průběh splácení úroků a zbytku dluhu v k-tém splátkovém období "]
14 je počet platebních období

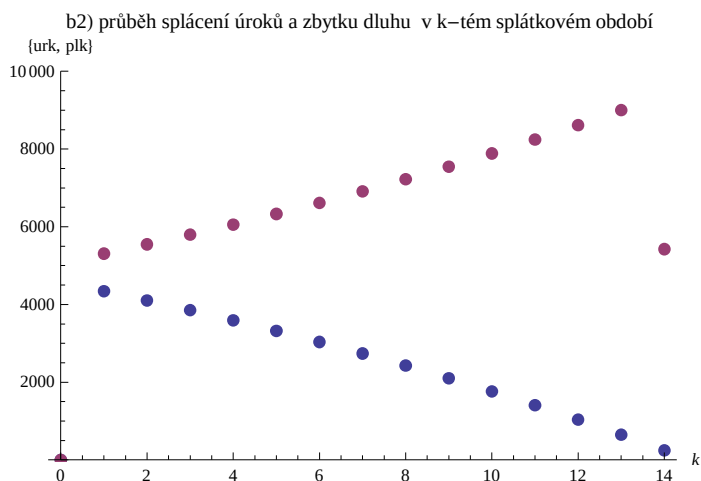
```

a) Umořovací plán

k	sk	urk	plk	uk
0	0	0	0	96500
1	9650	4342.5	5307.5	91192.5
2	9650	4103.66	5546.34	85646.2
3	9650	3854.08	5795.92	79850.2
4	9650	3593.26	6056.74	73793.5
5	9650	3320.71	6329.29	67464.2
6	9650	3035.89	6614.11	60850.1
7	9650	2738.25	6911.75	53938.4
8	9650	2427.23	7222.77	46715.6
9	9650	2102.2	7547.8	39167.8
10	9650	1762.55	7887.45	31280.3
11	9650	1407.61	8242.39	23037.9
12	9650	1036.71	8613.29	14424.7
13	9650	649.109	9000.89	5423.76
14	5667.83	244.069	5423.76	0.

b1) průběh umořování půjčky





V každém z řádků tabulky a) je pro k-té splátkové období zachycena výše splátky, dále její část, která slouží k zaplacení úroku, část, která slouží k umořování zbytkového dluhu a výše zbývajícího dluhu.

Na grafu b1) je možné sledovat, jak rychle dluh klesá.

Na grafu b2) je vidět, že zatímco splátky úroků klesají, fialově označené hodnoty umořování dluhu rostou.

7.2 Kontokorentní účet

Zadání

V následující tabulce jsou zachyceny pohyby na kontokorentním účtu s úvěrovým rámcem 50 000 Kč.

Počet dní	datum	příjmy	výdaje	zůstatek	čerpání debetu
81	2.2.	10 000	---	10 000	50 000
67	24.4.	---	25 000	-15 000	35 000
79	30.6.	5000	---	-10 000	40 000
34	17.9.	---	50 000	-60 000	---
46	21.10.	40 000	---	-20 000	---
25	6.12.	30 000	---	10 000	---
□	31.12.	---	---	10 000	---

Při krátkodobém překročení debetu si banka účtuje přírůzek 5% k debetním úrokům a dále pohotovostní provizi z nevyužitého rámce ve výši 0,5% p.a. Úroková míra z kladných zůstatků je 2% p.a. a z debetních zůstatků 12% p.a. Proveďte uzávěrku tohoto účtu. Pro výpočet použijte standard ACT/360 (angl. zůstatkový způsob).

Řešení

Když označíme

N počet období, kdy nedochází k žádnému pohybu na účtu,

d_i , $i = 1, \dots, N$ počet dní v i -tém období,

p roční úrokovou míru,

K příslušný zůstatek,

UC_i , $i = 1, \dots, N$ úroková čísla,

UD úrokový dělitel,

pak pro celkový úrok U platí, že

$$U = \frac{\sum_{i=1}^N UC_i}{UD}, \text{ kde } UC_i = \frac{Kd_i}{100}, UD = \frac{360}{p}.$$

Celkový úrok spočteme pro K_1 kreditní zůstatek, K_2 debetní zůstatek, K_3 pohotovostní provizi a K_4 překročení úvěrového rámce. Dílčí výsledky zachytíme v tabulce. Nakonec stanovíme konečný zůstatek. Ten obdržíme tak, že k zůstatku z posledního období připočteme kreditní úroky a odečteme debetní úroky, pohotovostní provize z nevyužitého rámce a úroky za překročení debetu.

```

R = 50 000;
o = 6;
d = {81, 67, 79, 34, 46, 25};
K1 = {10 000, 0, 5000, 0, 40 000, 30 000};
K2 = {0, 25 000, 0, 50 000, 0, 0};
z[0] = 0;
z[i_] := z[i - 1] + K1[[i]] - K2[[i]];
tab1 = Table[{i, d[[i]], K1[[i]], K2[[i]], z[i]}, {i, 1, o}];
TableForm[tab1,
  TableHeadings → {None, {"poř.č.", "dní", "příjmy.", "výdaje", "zůstatek"}}]

KZ = Table[If[z[i] ≥ 0, z[i], 0], {i, 1, o}];
uc1[i_] := KZ[[i]] * d[[i]] / 100;
u[1] = N[(2 / 360) * Sum[uc1[i], {i, 1, o}]];

DZ = Table[If[z[i] < 0, z[i], 0], {i, 1, o}];
uc2[i_] := DZ[[i]] * d[[i]] / 100;
u[2] = N[(12 / 360) * Sum[uc2[i], {i, 1, o}]];

K3 = Table[If[z[i] ≥ 0, R, R + z[i]], {i, 1, o}];
NR = Table[If[K3[[i]] ≥ 0, K3[[i]], 0], {i, 1, o}];
uc3[i_] := NR[[i]] * d[[i]] / 100;
u[3] = -N[(0.5 / 360) * Sum[uc3[i], {i, 1, o}]];

PR = Table[If[K3[[i]] < 0, K3[[i]], 0], {i, 1, o}];
uc4[i_] := PR[[i]] * d[[i]] / 100;
u[4] = N[(5 / 360) * Sum[uc4[i], {i, 1, o}]];

tab2 = Table[{i, d[[i]], KZ[[i]], uc1[i], -DZ[[i]], -uc2[i]}, {i, 1, o}];
TableForm[tab2,
  TableHeadings → {None, {"poř.č.", "dní", "kred.zůst.", "UC1", "deb.zůst.", "UC2"}}]

tab2 = Table[{i, d[[i]], NR[[i]], uc3[i], -PR[[i]], -uc4[i]}, {i, 1, o}];
TableForm[tab2,
  TableHeadings → {None, {"poř.č.", "dní", "pohot.prov.", "UC3", "překr.rámce", "UC4"}}]

"Konečný zůstatek:"
z[6] + u[1] - u[2] - u[3] - u[4]

```

poř.č.	dní	příjmy.	výdaje	zůstatek	
1	81	10 000	0	10 000	
2	67	0	25 000	-15 000	
3	79	5000	0	-10 000	
4	34	0	50 000	-60 000	
5	46	40 000	0	-20 000	
6	25	30 000	0	10 000	
poř.č.	dní	kred.zůst.	UC ₁	deb.zůst.	UC ₂
1	81	10 000	8100	0	0
2	67	0	0	15 000	10 050
3	79	0	0	10 000	7900
4	34	0	0	60 000	20 400
5	46	0	0	20 000	9200
6	25	10 000	2500	0	0

poř.č.	dní	pohot. prov.	UC ₃	překr. rámce	UC ₄
1	81	50 000	40 500	0	0
2	67	35 000	23 450	0	0
3	79	40 000	31 600	0	0
4	34	0	0	10 000	3400
5	46	30 000	13 800	0	0
6	25	50 000	12 500	0	0

Konečný zůstatek:

11 860.3

V první tabulce je ze zadaných údajů spočten zůstatek v jednotlivých obdobích. Druhá tabulka obsahuje v jednotlivých sloupcích velikost kreditního a debetního zůstatku spolu s příslušnými úrokovými čísly. Ve třetí tabulce je pohotovostní provize a velikost překročení úvěrového rámce spolu s příslušnými úrokovými čísly.

7.3 Vývoj cen na trhu

Zadání

Sestavte matematický model trhu pro případ, kdy poptávka převyšuje nabídku. Graficky zachyťte vývoj cen pro parametry modelu $x_0 = 5$, $a = 2$, $b = 10$, $k_p = 4$, $k_n = 1, \dots, 8$. Analyzujte, jak změna koeficientu k_n ovlivňuje kvalitu řešení.

Řešení

Označme

$y_{k,p}$ - objem zboží na straně poptávky v k -tém časovém intervalu,

$y_{k,n}$ - objem zboží na straně nabídky v k -tém časovém intervalu.

Na trhu dojde k rovnovážnému stavu, pokud bude nabídka odpovídat poptávce, tj. když

$$(1) \quad y_{k,n} = y_{k,p}.$$

Taková situace je však pouze ideálním stavem. Jedním z důvodů vzniku nerovnováhy může být to, že nabízeného zboží je na trhu nedostatek. Model, který takové situaci odpovídá, budeme nazývat modelem se zpožděním na straně nabídky a budeme ho konstruovat za předpokladu, že funkce, které zachycují závislost nabídky a poptávky na ceně, jsou lineární.

Protože výrobci reagují na nedostatečnou nabídku zvýšením výroby až po uplynutí jednoho časového období v závislosti na předchozí ceně zboží a poptávka reaguje ihned bez zpoždění, lze závislost nabídky a poptávky na ceně zboží vyjádřit ve tvaru

$$(2) \quad y_{k,n} = k_n x_{k-1} - b$$

$$(3) \quad y_{k,p} = -k_p x_k + a, \text{ kde } a, b, k_n, k_p > 0 \text{ jsou parametry modelu. Přitom}$$

k_n je konstanta, která odráží citlivost výrobců na cenu zboží,

k_p je konstanta představující citlivost spotřebitelů na cenu zboží a

x_k je cena zboží v k -tém časovém intervalu.

Dosažením vztahů (2) a (3) do rovnice (1) dostaneme následující matematický model

$$(3) \quad x_k = A x_{k-1} + B, \text{ kde } A = \frac{-k_n}{k_p}, B = \frac{a+b}{k_p}.$$

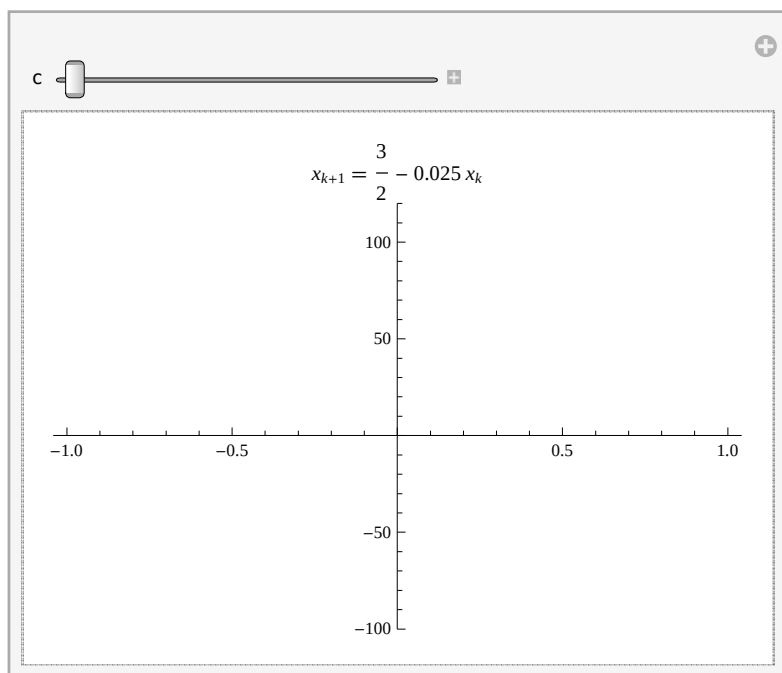
Iterace (3) pro zadané parametry zachytíme graficky a zjistíme, jak se mění jejich vlastnosti v závislosti na změně parametru k_n .

```

Clear[x, kn, kp, a, b, A, c];
kn[c_] = c; a = 2; kp = 4; b = 4;
A[c_] := -kn[c] / kp; B = (b + a) / kp;
y = A[c] * x_k + B;
"x_{k+1}=" y
x[0, c_] = 5;
x[k_, c_] := A[c] * x[k - 1, c] + B;
tab[c_] := Table[{k, x[k, c] // N}, {k, 1, 9, 1}];
Manipulate[ListLinePlot[tab[c], PlotRange -> {-100, 120},
  PlotStyle -> Dashed, PlotMarkers -> Automatic, PlotStyle -> PointSize[0.05],
  PlotLabel -> "x_{k+1} = -c / kp * "x_k" + (b + a) / kp], {c, .1, 5.8, .1}]

```

$$x_{k+1} = \left(\frac{3}{2} - \frac{c x_k}{4} \right)$$



Změna koeficientu k_n se odráží v hodnotě směrnice A , která ovlivňuje kvalitu řešení. Z provedené animace je možné učinit následující závěry: Pokud $-1 < A < 0$, ceny konvergují. Pro $A = -1$ ceny oscilují mezi dvěma hodnotami. Pro $A < -1$ ceny divergují.

8 Optimalizační úlohy

8.1 Optimalizace výroby

Zadání

Balírný a pražírny kávy plánují na následující období výrobu dvou směsí kávy Mocca a Standard. Pro výrobu obou směsí mají při tom na toto období smluvně od dodavatelů k dispozici tři druhy kávových bobů k1, k2, k3 postupně o kapacitě 40, 60 a 25 tun, které se navzájem liší kvalitou a nákupní cenou. Při výrobě směsí je třeba dodržovat technologické postupy, které mimo jiné určují, jaké procento jednotlivých komponent bude použito při této výrobě. Následující tabulka ukazuje skladbu obou směsí (v tunách komponenty na 1 tunu směsi):

komponenta	Mocca	Standard	kapacita (t)
k1	0, 5	0, 25	40
k2	0, 5	0, 5	60
k3	0	0, 25	25

Na základě přímých a nepřímých nákladů souvisejících s výrobou a vzhledem k předpokládané ceně obou směsí byl vykalkulován zisk, který činí 20 000 Kč, resp 14 000 Kč na jednu tunu směsi Moca resp Standard. Management podniku chce naplánovat produkci firmy tak, aby celkový zisk byl maximální.

Řešení

V tomto příkladu se jedná o úlohu z oblasti plánování výroby. Označme

x vyprodukované množství kávy Mocca

y vyprodukované množství kávy Standard

Na základě zadání sestavíme matematický model úlohy

$$z = 2000x + 14000y \rightarrow \max$$

$$0,5x + 0,25y \leq 40$$

$$0,5x + 0,5y \leq 60$$

$$0,25y \leq 25$$

$$x, y \geq 0$$

Ten nejprve vyřešíme numericky a pak graficky.

"Numerické řešení:"

Clear[x, y] *

```
Maximize[{20000 * x + 14000 * y, 0.5 * x + 0.25 * y <= 40, 0.5 * x + 0.5 * y <= 60,
0.25 * y <= 25, x >= 0, y >= 0}, {x, y}]
```

"Grafické řešení:"

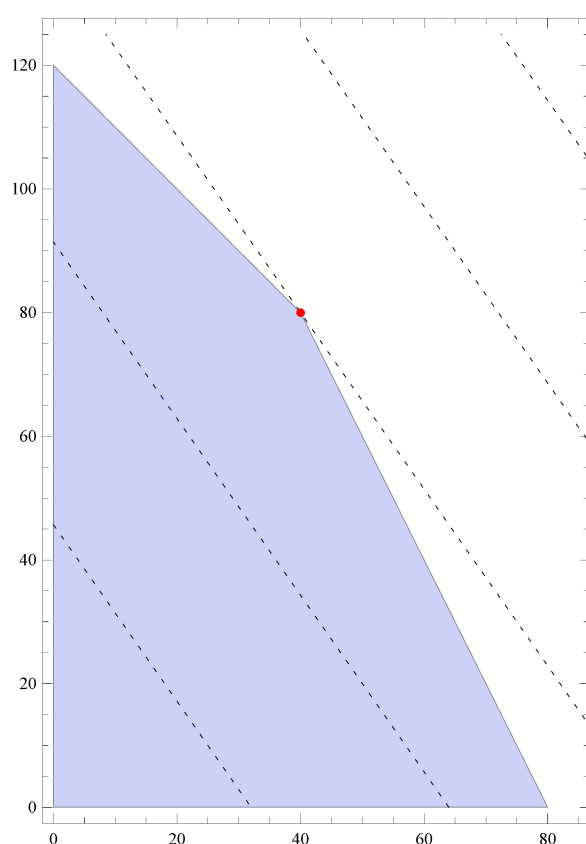
Show[

```
RegionPlot[0.5 * x + 0.25 * y <= 40 && 0.5 * x + 0.5 * y <= 60, {x, 0, 85}, {y, 0, 125},
AspectRatio -> Automatic],
ContourPlot[20000 * x + 14000 * y, {x, 0, 105}, {y, 0, 125},
ContourShading -> False, Contours -> 5, ContourStyle -> Dashing[Small]],
Graphics[{Red, PointSize[Medium], Point[{40, 80}]}]]
```

Numerické řešení:

```
{1.92 × 106 Null, {Null (x → 40.), Null (y → 80.)}}
```

Grafické řešení:



V první výstupní buňce se nachází hodnoty obou proměnných $x=40$, $y=80$ (kolik tun směsi Mocca a kolik tun směsi Standard se má vyrobit) a hodnota účelové funkce 1 920 000 (odpovídající v tomto případě zisku podniku v Kč).

Ve druhé výstupní buňce je grafické řešení úlohy. Přípustná oblast, která je určena výše uvedenými nerovnostmi, je vyznačena modře. Čárkované přímky znázorňují účelové funkce pro různé hodnoty účelové funkce z . Červený bod představuje optimální řešení úlohy.

8.2 Optimalizace investic

Zadání

Vedení investiční společnosti zvažuje investici do akcií 4 firem produkujících nápoje. Aby firma předešla ztrátám plynoucím z rizika spojeného s investováním do soukromého sektoru, rozhodlo se vedení společnosti část peněz investovat do vládních obligací. Celková investovaná částka činí 2 mil. Kč. Z dlouhodobého sledování finančního trhu vyplývají roční procenta výnosů a indexy rizika u sledovaných cenných papírů, které uvedené v následující

tabulce

□	Výnos	Index rizika
České pivovary a.s.	12 %	0, 07
Víno Morava a.s.	9 %	0, 09
Moravská švestka a.s.	15 %	0, 05
České mlékárny a.s.	7 %	0, 03
Vládní obligace	6 %	0, 01

Na poradě managementu společnosti bylo rozhodnuto, že

- Do akcií Českých mlékáren se nesmí investovat více než 200 tis. Kč.
- Investice do vládních obligací musí činit alespoň 20 % všech investic.
- Z hlediska diversifikace portfolia se do akcií žádné z firem vyrábějících alkoholické nápoje nesmí investovat více než 800 tis. Kč.
- Celkový index rizika portfolia nesmí přesáhnout hodnotu 0,05.

Cílem společnosti je maximalizovat roční výnos portfolia při dodržení všech uvedených podmínek.

Řešení

Když označíme

x_i , $i = 1, \dots, 8$, výši investice do i -tého podniku,
má matematický model úlohy tvar

$$z = 0,12x_1 + 0,09x_2 + 0,15x_3 + 0,07x_4 + 0,06x_5 \rightarrow \max$$

$$x_1 + x_2 + x_3 + x_4 + x_5 \leq 2000$$

$$x_4 \leq 200$$

$$x_5 \leq 400$$

$$x_1 \leq 800$$

$$x_2 \leq 800$$

$$x_3 \leq 800$$

$$(0,07x_1 + 0,09x_2 + 0,05x_3 + 0,03x_4 + 0,01x_5) / 200 \leq 0,05$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Najdeme numerické řešení úlohy.

```
Maximize [{0.12 * x1 + 0.09 * x2 + 0.15 * x3 + 0.07 * x4 + 0.06 * x5,
  x1 + x2 + x3 + x4 + x5 <= 2000, x4 <= 200, x5 >= 400, x1 <= 800, x2 <= 800,
  x3 <= 800,
  (0.07 * x1 + 0.09 * x2 + 0.05 * x3 + 0.03 * x4 + 0.01 * x5) / 2000 <= 0.05, x1 >= 0,
  x2 >= 0, x3 >= 0, x4 >= 0}, {x1, x2, x3, x4, x5}]
{240., {x1 -> 800., x2 -> 0., x3 -> 800., x4 -> 0., x5 -> 400.}}
```

Ve výstupní buňce je uvedena výše investic do jednotlivých podniků a vládních obligací, pro kterou se dosáhne maximálního ročního výnosu portfolia. Údaje jsou uvedeny v tis. Kč.

8.3 Řízení zásobování

Zadání

Společnost je schopná zajistit plynulou výrobu při dodávkách materiálu ve výši 20 000 t. Cena materiálu je 9 000 Kč/t. Cena spojená s dodávkou je 7 500 Kč. V minulém roce cena skladování se pohybovala kolem 15% průměrné roční skladovací ceny. Určete optimální výši dodávky, délku dodacího cyklu a cenu jedné dodávky.

Řešení

Označme

S spotřebu nakupované položky za období délky T ,

c cenu skladované položky,

n_i náklady na jednotku objednávky,

n_j procento, které tvoří náklady na udržování zásob,

$\bar{x}$ průměrnou spotřebu zboží,

Q velikost dodávky,

$N(Q)$ nákladovou funkci.

Pokud je spotřeba nakoupeného zboží v čase lineární, pak

$$\bar{x} = \frac{Q}{2}.$$

Nákladová funkce je součtem nákladů spojených s pořízením zásob a nákladů vynaložených na udržování zásob:

$$N(Q) = \frac{Q}{2} T n_s c + \frac{S}{Q} n_j.$$

Nejprve stanovíme velikost dodávky tak, aby hodnota nákladové funkce byla co nejmenší. V podstatě se jedná o stanovení stacionárního bodu nákladové funkce $N(Q)$.

```
Clear[Q, T, ns, c, nj, S];
NC[Q_] = Q * T * ns * c / 2 + S * nj / Q;
v = Solve[D[NC[Q], Q] == 0, Q];
"optimální výše dodávky"
StandardForm[v[[2]]]
optimální výše dodávky
```

$$\left\{ Q \rightarrow \frac{\sqrt{2} \sqrt{n_j} \sqrt{S}}{\sqrt{c} \sqrt{n_s} \sqrt{T}} \right\}$$

Následující část odpovídá řešení naší úlohy pro zadané parametry.

"Pro T=1 rok, S=20000 t, c=9000 Kč, ns=0.15, nj=7500 Kč:"

T = 1; S = 20000; c = 9000; ns = 0.15; nj = 7500;

Q = Sqrt[2] Sqrt[nj] Sqrt[S] / (Sqrt[c] Sqrt[ns] Sqrt[T]);

NC[Q_] = Q * T * ns * c / 2 + S * nj / Q;

t[Q_] = T Q / S;

TableForm[{Q, NC[Q], Round[365 * t[Q]]}, TableHeadings -> {"Velikost opt.dodávky[t]",
"Cena opt. dodávky [Kč/t]", "Délka dodacího cyklu [dny]"}, None]

Pro T=1 rok, S=20000 t, c=9000 Kč, ns=0.15, nj=7500 Kč:

Velikost opt.dodávky[t]	471.405
Cena opt. dodávky [Kč/t]	636 396.
Délka dodacího cyklu [dny]	9

9 Použitá literatura

- Abell, M. L., Braselton J. P.: *Mathematica by example*. Elsevier, London. 2009.
- Andreoni, J., Miller, J. H.: Rational Cooperation in the Finitely Repeated Prisoner's Dilemma: Experimental Evidence. *Economic Journal*, 1993, Vol. 103, No. 418, p. 570-585.
- Arce, D. G.: Economics, Ethics and the Dilemma in the Prisoner's Dilemmas. *American Economist*, 2010, Vol. 55, No. 1, p. 49-57.
- Bohanesová, E., *Finanční matematika*, UP Olomouc. 2006.
- Brosig, J.: Identifying Cooperative Behavior: Some Experimental Results in a Prisoner's Dilemma Game. *Journal of Economic Behavior & Organization*, 2002, Vol. 47, No. 3, p. 275-290.
- Brown, R. J., Abrams, D.: The Effects of Intergroup Similarity and Goal Interdependence on Intergroup Attitudes and Task Performance. *Journal of Experimental Social Psychology*, 1986, Vol. 22, No. 1, p. 78-98.
- Cooper, R., Dejong, D. V., Forsythe, R., Ross, T. W.: Cooperation without Reputation: Experimental Evidence from Prisoner's Dilemma Games. *Games and Economic Behavior*, 1996, Vol. 12, No. 2, p. 187-218.
- Dawkins, R.: *Sobecký gen*. 1. vyd. Mladá fronta, Praha. 1998. 319 s.
- Dluhošová, D.: *Finanční řízení a rozhodování podniku*. Druhé upravené vydání. Ekopress, Praha. 2008. 192 s.
- Donniger, C.: Is It Always Efficient to be Nice? In Dickman and Mitter (eds) *Paradoxical Effects of Social Behavior* Heidelberg: Physica Verlag, 1986, p. 123-134.
- Food, M.: Some Experimental Games. *Management Science*, 1958, Vol. 5., No. 1, p. 5-26.
- Fotr, J., Souček, I.: *Investiční rozhodování a řízení projektů*. První vydání. Grada, Praha. 2011. 416 s.
- Fotr, J., Píšek, M.: *Exaktní metody ekonomického rozhodování*. Academia, Praha. 1986.
- Fotr, J., Dědina, J.: *Manažerské rozhodování*. VŠE, Praha. 1993.
- Fudenberg, D., Tirole. J.: *Game theory*. Massachusetts, MIT Press. 1991.
- Gillingan, C.: *Jiným hlasem: o rozdílné psychologii žen a mužů*. 1. vyd. Portál, Praha. 2001.
- Gross, I.: *Kvantitativní metody v manažerském rozhodování*. Grada Publishing, Praha. 2003.
- Hamilton, W. D.: The genetical evolution of social behavior (I). *Journal of Theoretical Biology*, 1964a, Vol. 7, p. 1-16.
- Hamilton, W. D.: The genetical evolution of social behavior (II). *Journal of Theoretical Biology*, 1964b, Vol. 7, p. 17-52.
- Hrdý, M., Horová, M.: *Finance podniku*. 1. vyd. Praha: Wolters Kluwer ČR, 2009. 180 s.
- Hušek, R., Mañas, M.: *Matematické modely v ekonomii*. SNTL, Praha. 1989.
- Hunt, M.: *Dějiny psychologie*. 1. vyd. Portál, Praha. 2000.
- Chmela, L.: *Energetický management [studijní podklady]*. MVŠO, Olomouc. 2013. (nepublikováno)
- Jablonský, J.: *Operační výzkum*, Professional Publishing, Praha. 2002.
- Kislingerová, E.: *Manažerské finance*. 2. přepracované a rozšířené vydání. C. H. Beck, Praha. 2007. 745 s.
- Marada, T.: *Nashovo ekvilibrium: bakalářská práce*. Matematicko-fyzikální fakulta Univerzita Karlova v Praze, katedra pravděpodobnosti a matematické statistiky, Praha. 2006. s. 32.
- McKelvey, R. D., Palfrey, T. R.: An Experimental Study of the Centipede Game. *Econometrica*, 1992, Vol. 60, No. 4, p. 803-836.
- MECHANIKA DVŮR KRÁLOVÉ. Turbína SETUR. URL: <http://www.mechanikakd.cz/chata.html>
- Nash, J.: Non-Cooperative Games. *Annals of Mathematics*, 1951, Vol. 54, No. 2, p. 286-295.
- Ortmann, A., Tichy, L. K.: Gender differences in the laboratory: evidence from prisoner's dilemma games. *Journal of Economic Behavior & Organization*, 1999, Vol. 39, No. 3, p. 327-339.
- Sampson, E. E., Kardush, M.: Age, Sex, Class, and Race Differences in Response to a Two-Person Non-Zero-Sum Game. *Journal of Conflict Resolution*, 1965, Vol. 9, No. 2, p. 212-220.
- Segal, N. L., Hersherberger, S. L.: Cooperation and Competition Between Twins: Findings from a Prisoner's Dilemma Game. *Evolution and Human Behavior*, 1999, Vol. 20, No. 1, p. 29-52.
- Scholleová, H.: *Ekonomické a finanční řízení pro neekonomy*. První vydání. Grada, Praha. 2008. 256 s.
- Strouhal, J., Židlická, R.: *Účetnictví - Velká kniha příkladů*. 1. vydání. Computer Press, Praha. 2007. 452 s.
- Strouhal, J.: *Finanční řízení firmy v příkladech*. Vydání první. Computer Press, Brno. 2006. 178 s.

- Strouhal, J., Židlická, R., Knapová, B.: Účetnictví - Velká kniha příkladů. 1. vydání. Computer Press, Praha. 2009. 659 s.
- Synek, M. a kol.: Podniková ekonomika. 4. přeprac. a dopl. vyd. C. H. Beck, Praha. 2006. 473 s.
- Synek, M. a kol.: Manažerská ekonomika. 5. aktualizované a doplněné vydání. Grada, Praha. 2011. 480 s.
- Synek, M., Kopkáně, H., Kubálková, M.: Manažerské výpočty a ekonomická analýza. 1. vydání. C. H. Beck, Praha. 2009. 301 s.
- Šiman, J., Petera, P.: Financování podnikatelských subjektů. Teorie pro praxi. 1. vydání. C. H. Beck, Praha. 2010. 192 s.
- Šotová, J.: Diskrétní modelování pomocí iterací funkcí. VVŠ PV Vyškov. 2000
- Říha, J., Láta, F., Kainzová, V., Mošová, V., Vyšín, I., Švrček, F., Richterek, L.: Software Mathematica v přírodních vědách a ekonomii. UPOL, Olomouc. 2012.
- Talašová, J.: Fuzzy metody vícekritériálního hodnocení a rozhodování. VUP, Olomouc. 2003.
- Wellin, P., Calkins, H.: M101: A First Course in Mathematica. Illinois, USA: Wolfram Research. 2011.
- Archiv studentských prací autora (V. Slezák).
- Homepage of Wolfram Research. URL: <http://www.wolfram.com>.