UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

NÁVRH A VYPRACOVANIE WEBOVSKÉHO ROZHRANIA PRE INTERAKTÍVNU MOŽNOSŤ RIEŠENIA ÚLOH NELINEÁRNEHO PROGRAMOVANIA

BAKALÁRSKA PRÁCA

2013

Andrej ŠIŠKA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

NÁVRH A VYPRACOVANIE WEBOVSKÉHO ROZHRANIA PRE INTERAKTÍVNU MOŽNOSŤ RIEŠENIA ÚLOH NELINEÁRNEHO PROGRAMOVANIA

BAKALÁRSKA PRÁCA

Študijný program:	Ekonomická a finančná matematika
Študijný odbor:	1114 Aplikovaná matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	prof. RNDr. Daniel Ševčovič, CSc.

Bratislava 2013

Andrej ŠIŠKA

49010691

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

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

Názov: Návrh a vypracovanie webovského rozhrania pre interaktívnu možnosť riešenia úloh nelineárneho programovania

Cieľ: Cieľom bude vytvoriť webovskú aplikáciu v jazyku PHP na spracovanie a analýzu úloh nelineárneho programovania so zameraním na úlohy o optimálnej skladbe portfólia. Program bude na strane servera využívať otvorený GNU softvér Octave, podobný systému Matlab.

Vedúci: prof. RNDr. Daniel Ševčovič, CSc.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.

Dátum zadania: 25.10.2012

Dátum schválenia: 03.11.2012

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

....
študent

....
vedúci práce

Poďakovanie Na tomto mieste by som rád poďakoval svojmu vedúcemu bakalárskej práce prof. RNDr. Danielovi Ševčovičovi , CSc. za odbornú pomoc, cenné rady a usmernenia, ktoré mi venoval počas vypracovávania bakalárskej práce.

Abstrakt

ŠIŠKA, Andrej: Návrh a vypracovanie webovského rozhrania pre interaktívnu možnosť riešenia úloh nelineárneho programovania [Bakalárska práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: prof. RNDr. Daniel Ševčovič, CSc., Bratislava, 2013, 46 s.

Cieľom práce bolo navrhnúť a naprogramovať webovú aplikáciu na určenie optimálnej skladby portfólia s využitím nelineárneho programovania. Zamerali sme sa najmä na diverzifikáciu portfólia s cieľom minimalizácie rizika pri zachovaní konštantného výnosu. Práca je rozdelená na niekoľko tematických kapitol. Prvá sa zaoberá základnou teóriou nelineárneho programovania. V druhej kapitole píšeme o optimalizácii portfólia, najmä Markowitzovej teórii portfólia, ale pridávame aj pohľad na trhové riziko. Tretia kapitola opisuje technológie použité pri programovaní a predstavuje návrh budúcej aplikácie. Vo štvrtej časti sa nachádza postup programovania aplikácie spolu s vybranými časťami kódu. V poslednej, piatej kapitole predstavujeme výsledný program pomocou niekoľkých ilustračných príkladov.

Kľúčové slová: nelineárne programovanie, diverzifikácia, Markowitzova teória portfólia, GNU Octave, PHP

Abstract

ŠIŠKA, Andrej: Design and implementation of a web interface for option to interactively solve nonlinear programming problems [Bachelor Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: prof. RNDr. Daniel Ševčovič, CSc., Bratislava, 2013, 46 s.

The goal of this bachelor thesis was to design and programm a web application to determine the optimal portfolio composition. We focus mostly on portfolio diversification, aiming to minimize risk while keeping constant rate of return. The thesis is divided into several thematic sections. The first one deals with basic theory of nonlinear programming. In the second, we discuss portfolio optimization, mostly Markowit portfolio theory, but we also offer a view at market risk. Third chapter describes tehnology used when programming and introduces the design of the application. In fourth section we can find the procedure for application programming with selected parts of code. In the last, fifth chapter we introduce the final product with help of several ilustrative examples.

Keywords: nonlinear programming, diversification, Markowitz portfolio theory,
GNU Octave, PHP

Obsah

ZOZNAM OBRÁZKOV

Zoznam obrázkov

ÚVOD

Úvod

Optimalizácia portfólia by mala byť prirodzeným cieľom každého racionálneho investora, ktorý chce maximalizovať svoj zisk. Zisk je však výrazne spojený s rizikom, preto sa optimalizačné metódy spravidla snažia práve minimalizovať riziko. Takou je aj Markowitzova optimalizačná úloha, ktorej cieľom je minimalizovať kvadratickú funkciu rizika pri pevnom zisku, kde sa využívajú metódy nelineárneho programovania.

Táto úloha je však výpočtovo pomerne náročná. Na jej riešenie sa často používajú špecializované programy(napríklad Financial Trading System). Ich nevýhodou je ale ich nedostupnosť pre širokú verejnosť, keďže vo väčšine prípadov ich licencia nie je zadarmo. Okrem toho sú často použiteľné len na zariadení, na ktorom sú nainštalované. Ďalšou možnosťou sú pracné výpočty v programoch ako Microsoft Excel, ktoré sú síce dostatočne rozšírené, no výpočty v nich sú nepohodlné.

Z tohoto dôvodu bude cieľom mojej práce vypracovať webové rozhranie, založené na voľne šíriteľných programovacích jazykoch PHP a Octave, dostupné pre každého s prístupom na internet. Pomocou tejto webovej aplikácie bude možné riešiť Markowitzovu úlohu rôznymi spôsobmi, cez jednoduché rozhranie.

Prvé dve kapitoly práce slúžia ako teoretický úvod do teórie portfólia a nelineárneho programovania potrebného v Markowitzovej teórii.

Tretia kapitola obsahuje popis technológií použitých pri programovaní ako aj návrh aplikácie vrátane jej logickej štruktúry a používateľského rozhrania.

V poslednej kapitole sa dostaneme k samotnému programovaniu, podrobnejšie opíšeme fungovanie programu spolu s niekoľkými ukážkami programovanieco kódu.

1 Úvod do nelineárneho programovania

V tejto časti sa oboznámime zo základnými poznatkami z oblasti nelineárneho programovania. Zameriame sa najmä na definovanie a klasifikáciu úloh nelineárneho programovania, a ich následné riešenie pomocou Lagrangeovej funkcie a multiplikátorov. Budeme pritom vychádzať najmä z Hamalovych a Trnovskej kníh a prednášok [2], [3] a [4]. Tieto vedomosti sú postačujúce pre vyriešenie Markowitzovej úlohy, s ktorou sa stretneme neskôr. Všetky tvrdenia sú uvádzané bez dôkazov, ich znenie je možné dohľadať v [3].

1.1 Definovanie pojmov

Vo svete sa často vyskytujú úlohy, ktoré možno matematicky zapísať ako minimalizáciu (alebo maximalizáciu) nejakej funkcie za určitých obmedzení. Matematicky teda môžeme takúto úlohu definovať v tvare, v akom je uvedená v [4].

Nech máme účelovú fukciu f_0 a $m+1$ funkcií n premenných:

$$f_i : X_i \to \mathbb{R}, i = 0, 1, 2, ..., m,$$

kde $\varnothing \neq X_i \subseteq \mathbb{R}^n$. Ďalej označíme prienik všetkých definičných oborov X_i funkcií f_i nasledovne:

$$X = \bigcap_{i=0}^{m} X_i,$$

a budeme o ňom predpokladať, že je neprázdnou podmnožinou $\mathbb{R}^n$.

Množina prípustných riešení sa definuje takto:

$$K = \{x \in X | f_i(x) \leq 0, i = 1, 2, ..., m\} \neq \varnothing. \tag{1}$$

Z toho je zrejmé, že pre existenciu aspoň jedného prípustného riešenia musí byť množina X neprázdna.

Úloha minimalizovať účelovú funkciu f_0 na množine prípustných riešení K je nazývaná ako úloha matematického programovania. Označuje sa nasledovne:

$$Min \{f_o(x) | x \in X, f_i(x) \leq 0, i = 1, 2, ..., m\},\tag{2}$$

kde nerovnosti $f_i(x) \leq 0$ sa nazývajú $i-$**te ohraničenia úlohy 2**. Môžeme si pritom všimnúť, že prípadnú maximalizáciu účelovej funkcie dosiahneme pomocou vzťahu $max f(x) = -min[-f(x)]$.

Úlohu 2 rozdeľujeme na dva prípady na základe nasledovných kritérií:

- **Úloha lineárneho programovania**, ak $\forall i$ je funkcia f_i lineárna.
- **Úloha nelineárneho programovania**, ak aspoň jedna z funkcií f_i nie je lineárna.

Také prípustné riešenie $\hat{x} \in K$, pre ktoré platí

$$\forall x \in K : f_0(\hat{x}) \leq f_0(x) \tag{3}$$

sa nazýva **optimálnym riešením úlohy 2**. Po dosadení do ohraničení $f_i(x) \leq 0$ nastáva jedna z dvoch možností:

- ohraničenia spĺňajúce $f_i(\hat{x}) = 0$ sa nazývajú **aktívnymi ohraničeniami**.
- ostatné, teda tie pre ktoré platí $f_i(\hat{x}) < 0$, sa nazývajú **neaktívnymi ohraničeniami**.

1.2 Rozdelenie úloh nelineárneho programovania

Vzhľadom k tomu, že niektorí autori uvádzajú ohraničenia aj v tvare $f_i(x) = 0$, je zaužívané rozdelenie úloh nelineárneho programovania v závislosti od počtu a tvaru ohraničení. Rozlišujeme 4 základné typy úloh, ktoré budeme nazývať **U1-U4**. Najjednoduchší prípad nastáva, ak úloha nelineárneho programovania neobsahuje žiadne ohraničenie. Nazýva sa **úloha na voľný extrém** a budeme ju označovať **U1**. Jej matematický zápis vypadá nasledovne:

$$Min\{f(x) | x \in X\}, \tag{4}$$

kde $\varnothing \neq X \subseteq \mathbb{R}^n$ a $f : X \to \mathbb{R}^n$, X je otvorená množina.

Ďalším typom úlohy je **klasická úloha na na viazaný extrém**. Nazýva sa tiež Lagrangeova, pretože tento francúzsky matematik ju sformuloval ako prvý.

Nech máme $m + 1$ funkcií n premenných

$$f_i : X_i \to \mathbb{R}(i = 0, 1, ..., m),$$

$\varnothing \neq X_i \subseteq \mathbb{R}^n$ a $m < n$. Nasledujúca formula sa potom nazýva **klasickou úlohou na viazaný extrém** a označujeme ju (U2):

$$Min \left\{ f_0 \left(x \right) | x \in X, f_i \left(x \right) = 0, i = 1, 2, ..., m \right\}. \tag{5}$$

Klasická úloha teda obsahuje len ohraničenia v tvare rovníc $f_i \left(\hat{x} \right) = 0$. Nasledujúca úloha, nazývaná **úlohou nelineárneho programovania v užšom zmysle**, alebo U3, naopak obsahuje ohraničenia v tvare nerovníc $f_i \left(\hat{x} \right) \leq 0$:

$$Min \left\{ f_0 \left(x \right) | x \in X, f_i \left(x \right) \leq 0, i = 1, 2, ..., m \right\}. \tag{6}$$

Na rozdiel od predchádzajúcej úlohy 5 však počet obmedzení nie je ničím limitovaný. Ostatné predpoklady ostávajú zachované.

Poslednou zmienenou úlohou je **úloha nelineárneho programovania v širšom zmysle** (U4). Táto v sebe spája ohraničenia v tvare rovníc aj nerovníc z úloh 5 a 6:

$$Min \left\{ f_0 \left(x \right) | x \in X, f_i \left(x \right) \leq 0, i = 1, 2, ..., m, f_j \left(x \right) = 0, j = m + 1, m + 2, ..., m + r \right\}. \tag{7}$$

Úlohy na viazaný extrém U2-U4 sú však ekvivalentné a rozdiely medzi nimi iba formálne. Dôkaz možno nájsť v [3, str.16]. V ďalšej časti sa preto budeme zaoberať len úlohou U3 (6).

1.3 Lagrangeova úloha

Nech máme úlohu nelineárneho programovania na viazaný extrém s neprázdnou množinou prípustných riešení a optimálnym riešením. Teda:

$$Min \left\{ f_0 \left(x \right) | x \in X, f_i \left(x \right) \leq 0, i = 1, 2, ..., m, f_j \left(x \right) = 0, j = m + 1, m + 2, ..., m + r \right\},$$

$$K_3 = \{x \in X | f_i(x) \leq 0, i = 1, 2, ..., m\} \neq \emptyset,$$

$$\bar{x} \in K_3.$$

Potom **Lagranegeova funkcia** $L : X \times \mathbb{R}_+^m$ je definovaná takto:

$$L(x, u) = f_o(x) + \sum_{i=1}^{m} u_i f_i(x), \tag{8}$$

kde vektory $u \geq 0_m$ sa nazývajú **vektormi Lagrangeovych premenných**.

Definícia 1.1. *[3, str. 17]*
Nech $\bar{x} \in K_3$ je optimálnym riešením úlohy U36. Vektor $\bar{u} \geq 0_m$ nazveme ***vektorom Lagrangeovych multiplikátorov*** *prináležiacich optimálnemu riešeniu $\bar{x}$ ak pre Langrangeovu funkciu 8 platí:*

$$\forall x \in X : f_0(\bar{x}) \leq L(x, \bar{u}). \tag{9}$$

Je potrebné poznamenať, že existencia vektora Lagrangeovych multiplikátorov závisí od formy, v akej je zapísaná množina prípustných riešení. Teda nemusí nutne existovať ani v prípade, kedy úloha má optimálne riešenie.

Lema 1.2. *[3, str. 18]*
Nech $\hat{x} \in K_3$ je optimálnym riešením úlohy ***U3*** *a $\hat{u} \geq 0_m$ je príslušný vektor Lagrangeovych multiplikátorov. Potom platia nasledujúce vzťahy:*

$$f_o(\hat{x}) = L(\hat{x}, \hat{u}), \tag{10}$$

$$\hat{u}_i f_i(\hat{x}) = 0, \forall i = 1, 2, ..., m. \tag{11}$$

Z predchádzajúcich vzťahov 9 a 10 teda vyplýva, že minimum funkcie $L(x, \hat{u})$ je ekvivalentné s optimálnym riešením $\hat{x} \in K_3$ úlohy (U3). Lagrangeova funckia má teda schopnosť previesť úlohu na viazaný extrém na jednoduchšiu úlohu s voľným extrémom.

Teraz uvedieme definíciu sedlového bodu, pomocou ktorej potom ukážeme, že tento bod je optimálnym riešením úlohy (U3).

Definícia 1.3. *[3, str. 19]*
Majme dve neprázdne množiny $\varnothing \neq X \subseteq \mathbb{R}^n, \varnothing \neq Y \subseteq \mathbb{R}^n$ *a na nich definovanú funckciu* $\Phi : X \times Y \to \mathbb{R}$. *Bod* $(x^0, y^0) \in X \times Y$ *nazývame* ***sedlovým bodom typu min-max*** *funkcie* Φ, *ak platia nasledovné dve nerovnosti:*

$$\forall x \in X, \forall y \in Y : \Phi\left(x^0, y\right) \leq \Phi\left(x^0, y^0\right) \leq \Phi\left(x, y^0\right). \tag{12}$$

Veta 1.4. *[3, str. 20]*
Nech $\hat{x} \in K_3$ *je optimálnym riešením úlohy (U3), t.j. platí 3 a* $\hat{u} \geq 0_m$ *je príslušný vektor Lagrangeovych multiplikátorov, t.j. platí (9). Potom* $(\hat{x}, \hat{u}) \in X \times \mathbb{R}^m_+$ *je sedlovým bodom Lagrangeovej funkcie (8), t.j. platí:*

$$\forall x \in X, \forall u \geq 0_m : L\left(\hat{x}, u\right) \leq L\left(\hat{x}, \hat{u}\right) \leq L\left(x, \hat{u}\right) \tag{13}$$

Veta 1.5. *[3, str. 21]*
Nech $(\hat{x}, \hat{u}) \in X \times \mathbb{R}^m_+$ *je sedlovým bodom Lagrangeovej funkcie (8), t.j. nech platí (13). Potom* $\hat{x} \in X$ *je optimálnym riešením úlohy (U3), t.j. platí (3) a* $\hat{u} \geq 0_m$ *je príslušným vektorom Lagrangeovych multiplikátorov, a teda platí (9).*

Na základe dvoch predošlých viet teda jednoznačne vyplýva, že sedlový bod Lagrangeovej funcie je zároveň optimálnym riešením úlohy (U3), čo využijeme v ďalšej kapitole pri riešení Markowitzovho problému.

2 Teória portfólia

V tejto kapitole sa oboznámime s metódami na výber efektívneho portfólia, ktoré sa používajú v samotnej webovej aplikácii. Hlavne pôjde o znižovanie rizika portfólia pomocou diverzifikácie, ale zameriame sa aj na mieru trhového rizika pri jednotlivých aktívach. Budeme pri tom voľne sledovať výklad príslušných tém z [6].

2.1 Markowitzova teória portfólia

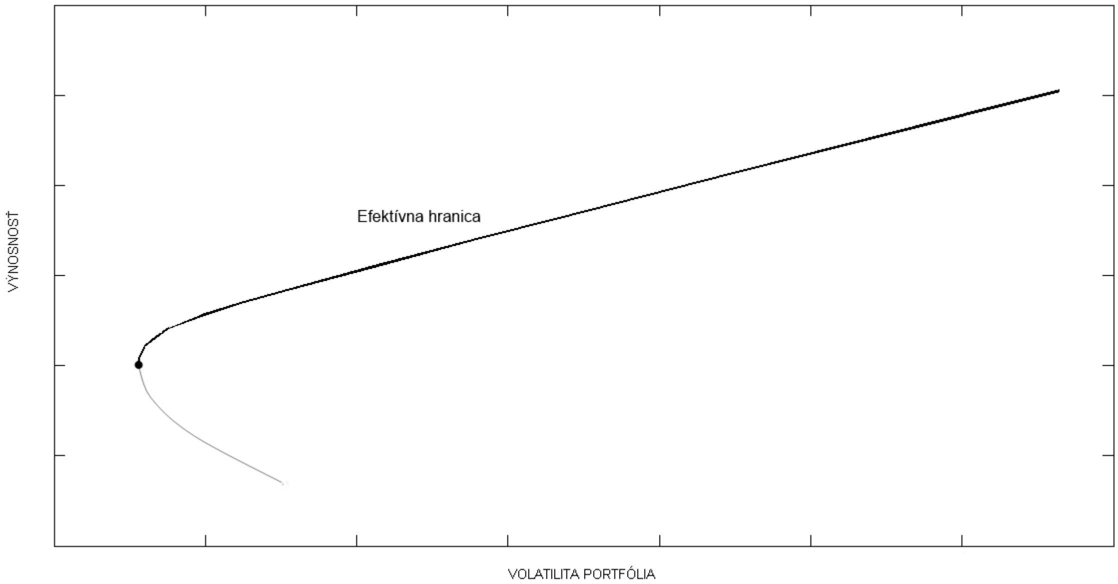
Princípy tejto teórie položil Harry Markowitz v roku 1952. Je známa tiež ako Modern Portfolio Theory(MPT). Markowitz poukázal na to, že nezáleží len na variancii výnosov jednotlivých aktív, ale najmä na tom, aký vplyv majú tieto aktíva na súhrnnú varianciu(rizikovosť) celého portfólia. Pri výbere efektívneho portfólia teda zohľadnil korelácie medzi aktívami. Markowitzova teória portfólia sa preto zaoberá jeho diverzifikáciou tak, aby sa dosiahol čo najvyšší výnos portfólia a zároveň čo najnižšia rizikovosť. Teda:

$$\bar{r}_p \to max \; a \; \sigma_p \to min \tag{14}$$

Takto zostrojená úloha ponúka viac ako jedno riešenie, keďže maximalizácia výnosu a minimalizácia rizika idú proti sebe. Súbor všetkých optimálnych riešení sa dá zobraziť na grafe závislosti výnosu od disperzie portfólia (Obr. 1).

Kombinácie všetkých aktív sa nachádzajú vo vnútri paraboly, pričom jej hranica sa nazýva efektívnou. Portfóliá na tejto hranici sú teda riešeniami Markowitzovej úlohy. Avšak ešte treba vylúčiť portfóliá nachádzajúce sa na spodnej časti hranice paraboly, pretože pre každé z nich exituje iné s rovnakou disperziou, ale vyšším výnosom. Pre nájdenie efektívnych portfólií teda berieme do úvahy len hornú časť efektívnej hranice.

Aby sme dostali úlohu s jedným riešením, môžeme napríklad zafixovať jednu z hodnôt $\bar{r}_p$ a σ_p, pričom budeme iba minimalizovať riziko, resp. maximalizovať výnos. V našom prípade sa budeme zaoberať iba prípadom pevného zisku a minimalizácie rizika. Pre konkrétnu úlohu však potrebujeme zohľadniť investora a jeho postoj k riziku. Rizikovo averzný investor si ako očakávaný výnos $\bar{r}_p$ nastaví nižsiu hodnotu než riziko obľubujúci investor. Fixovaním $\bar{r}_p$ teda dostávame Markowitzovu úlohu v tvare:



Obr. 1: Efektívna hranica

$$\begin{aligned} \min \frac{1}{2} \sum_{i,j=1}^{n} w_i w_j \sigma_{ij}, \\ \sum_{i=1}^{n} w_i \bar{r}_i = \bar{r}_p, \\ \sum_{i=1}^{n} w_i = 1, \\ l_i \leq w_i \leq u_i, \end{aligned} \tag{15}$$

kde náhodné premenné r_i predstavujú výnosy jednotlivých aktív, $\bar{r}_i$ sú ich stredné hodnoty, l_i, u_i sú voliteľné hranice pre mimimálnu a maximálnu hodnotu váhy i-tej akcie, σ_{ij} je kovariancia r_i a r_j a w_i sú váhy aktív v portfóliu, teda výsledok riešenia Markowitzovho problému.

S ohľadom na počet aktív v portfóliu môžu nastať 3 situácie. Pre $n = 1$ je úloha jednoznačná. Pre $n = 2$ je riešenie triviálne, pretože výsledné váhy w_i sú jednoznačne určené poslednými dvoma rovnicami zo sústavy 15. Najzaujímavejší je teda prípad $n \geq 3$, pre ktorý sa Markowitzov problém stáva úlohou kvadratického programovania a je predmetom riešenia našej webovej aplikácie.

Aby úloha 15 mala práve jedno riešenie, musia byť splnené dva predpoklady:

- Výnosy r_i sú lineárne nezávislé, z čoho vyplýva, že kovariančná matica $V = (\sigma_{ij})$ je regulérna a pozitívne definitná, čo potom platí aj pre jej inverznú maticu V^{-1}.

- Existujú aspoň dve rôzne aktíva $1 \leq i, j \leq n$, pre ktoré platí $\bar{r}_i \neq \bar{r}_j$. Pokiaľ by všetky aktíva portfólia mali rovnaký očakávaný výnos $\bar{r}$, potom by úloha 15 nemala riešenie pre $\bar{r} \neq \bar{r}_p$.

Na zákade poznatkov z predošlej časti vieme zapísať Markowitzovu úlohu (15) v tvare Lagrangeovej funkcie:

$$L(w, \lambda, \gamma) = \frac{1}{2} \sum_{i,j=1}^{n} w_i w_j \sigma_{ij} + \lambda \left[\bar{r}_p - \sum_{i=1}^{n} w_i \bar{r}_i \right] + \gamma \left[1 - \sum_{i=1}^{n} w_i \right], \tag{16}$$

pričom v notácii z kapitoly(1) môžeme vyjadriť

$$\begin{aligned} u &= (\lambda, \gamma)^T, \\ f_0 &= \frac{1}{2} \sum_{i,j=1}^{n} w_i w_j \sigma_{ij}, \\ f_1 &= \left[\bar{r}_p - \sum_{i=1}^{n} w_i \bar{r}_i \right], \\ f_2 &= \left[1 - \sum_{i=1}^{n} w_i \right]. \end{aligned}$$

Hľadáme teda minimum funkcie (16), o ktorom vieme, že je optimálnym riešením úlohy (15). Pomocou parciálnych derivácií prvého rádu môžeme vyjadriť úlohu ako sústavu n+2 rovníc o n+2 neznámych. Z dôvodu čitateľnosti uvádzame odvodenie z [6, str. 46].

$$w_p = g + h\bar{r}_p, \tag{17}$$

kde

$$\begin{aligned} g &= \frac{1}{D} \left[B \left(V^{-1} \underline{1} \right) - A \left(V^{-1} \bar{r} \right) \right], \\ h &= \frac{1}{D} \left[C \left(V^{-1} \bar{r} \right) - A \left(V^{-1} \underline{1} \right) \right], \end{aligned}$$

pričom je použité označenie

$$A = \underline{1}^T V^{-1} \bar{r} = \bar{r}^T V^{-1} \underline{1},$$

$$B = \bar{r} V^{-1} \bar{r} > 0$$

$$C = \underline{1} V^{-1} \underline{1} > 0,$$

$$D = DC - A^2$$

a $\underline{1}$ je jednotkový stĺpcový vektor.

2.2 Beta koeficient

Ani dobre diverzifikované portfólio však nie je zbavené rizika. Jeho najvýraznejšou zložkou je tzv. trhové riziko, to znamená citlivosť, s akou reaguje aktívum na pohyb trhu [1, str. 177]. Koeficient beta (β) je parameter aktíva, ktorý určuje práve mieru jeho trhového rizika.

Pre i-te aktívum má beta nasledovný tvar:

$$\beta_i = \frac{cov\,(i, M)}{\sigma_M^2}, \tag{18}$$

kde $cov\,(i, M)$ je kovariancia medzi výnosmi i-teho aktíva a výnosmi trhu a σ_M^2 je variancia výnosov trhu. S použitím historických dát ich vypočítame takto:

$$cov\,(i, M) = \frac{1}{n-1} \sum_{i=1}^{n} (r_i - \bar{r}) (r_{M_i} - \bar{r}_M),$$

$$\sigma_M^2 = \frac{1}{n-1} \sum_{i=1}^{n} (r_{M_i} - \bar{r}_M)^2$$

pričom r je vektor výnosov aktíva a r_M je vektor výnosov trhového portfólia. Toto býva z praktických dôvodov nahrádzané indexami, ktoré pomerne dobre zachytávajú vývoj trhu. V našom prípade budeme používať index $S\&P500$.

Pre výpočet bety celého portfólia sa používa vážený priemer bet jednotlivých aktív:

$$\beta_p = \sum_{i=1}^{n} w_i \beta_i. \tag{19}$$

Betu môžeme chápať ako multiplikátor, ktorý násobí každý pohyb trhu. Napríklad pre aktívum s $\beta = 0.5$ to znamená, že pri raste hodnoty trhového indexu[1] hodnota aktíva stúpne polovičnou rýchlosťou. Môže teda nastať niekoľko základných prípadov, pre ktoré vieme klasifikovať aktíva a ich bety takto:

- $\beta = 1$: Interpretácie je, že ide o typické trhové aktívum, teda získava s stráca hodnotu rovnakým smerom a rýchlosťou ako trh.
- $0 \leq \beta \leq 1$: Aktívum reaguje na fluktuáciu trhu menej, teda nesie menšie trhové riziko.
- $\beta \geq 1$: Akcia je nadpriemerne riziková a prudko reaguje na kolísanie trhu.
- $\beta \leq 0$: V tomto prípade aktívum reaguje na rast trhu poklesom svojej hodnoty a naopak.
- $\beta = 0$: Ide o bezrizikové aktívum, za aké sú vo všeobecnosti považované napríklad štátne dlhopisy. Jeho vývoj s trhom nekoreluje.

[1] spôsob odhadu hodnoty trhu pomocou portfólia zloženého z viacerých reprezentatívnych aktív

3 Návrh webovej aplikácie

V tejto kapitole popíšeme používané programy a technológie a ukážeme si návrh finálnej aplikácie. Stručne popíšeme jej vnútornú štruktúru a fungovanie, ale tiež aj výsledný projekt z pohľadu používateľa, teda funkcionalitu a užívaťeľské rozhranie.

3.1 Použité technológie

Webovská aplikácia pobeží na voľne dostupnom serveri patriacom Fakulte matematiky, fyziky a informatiky Unniverzity Komenského. Využívať pritom bude programovacie jazyky a technológie, ako najmä XHTML, PHP a GNU Octave.

3.1.1 XHTML

XHTML, alebo Extensible HyperText Markup Language, je rozšírenie klasického HTML, teda jazyka, v ktorom sa píšu webové stránky. HTML ako oficiálny jazyk pre WWW(World Wide Web) bol zavedený v roku 1990. Postupne začali vznikať rôzne prehliadače, teda software, ktorý dokáže čítať HTML kód a zobraziť ho v podobe, v akej poznáme WWW dnes. Avšak HTML dokáže zobrazovať len statické stránky, pre dynamické programovanie potrebujeme ďalší programovací jazyk, ktorým je PHP.

3.1.2 PHP

Hypertext Preprocessor, alebo PHP, vznikol v roku 1995 ako voľne dostupný programovací jazyk. Hoci je možné ho použiť aj na programovanie bežných aplikácií, využíva sa hlavne na tvorbu dynamických internetových stránok. Beží na strane servera, to znamená že k užívateľovi sa dostane iba výsledok, ktorý interpretuje prehliadač pri načítaní stránky. Používa sa napríklad na komunikáciu s databázami, prácu so súbormi a aplikáciami, či generovanie obsahu HTML stránky pomocou algoritmov.

3.1.3 GNU Octave

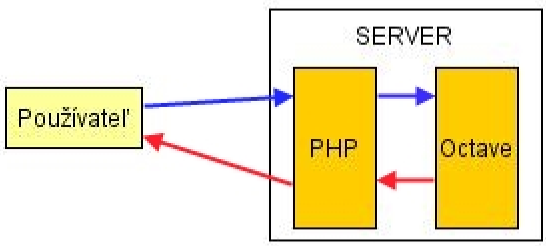
GNU Octave je multifunkčný nástroj schopný zložitých výpočtových operácií, fungujúci vo verziách pre operačné systémy Windows aj Linux. Ako voľne šíriteľný program je často uvádzaný ako alternatíva k Matlabu, na ktorého používanie je potrebné zakúpenie licencie. Toto je vlastne hlavný dôvod prečo uprednostňujeme Octave, keďže naša webovská aplikácia má byť voľne dostupná na internete. S Matlabom má Octave veľmi podobnú syntax, čo znamená, že jednotlivé skripty sú často vzájomne kompatibilné. Nie je to však pravidlo.

Medzi použitie patrí riešenie lineárnych a nelineárnych rovníc, maticové a vektorové operácie, numerické integrovanie a derivovanie, vykresľovanie grafov a iné matematické operácie. Okrem toho sa Octave používa napríklad na analýzu dát, spracovanie obrazu, ekonometriu a štatistiku a ďalšie. Avšak tento programovací jazyk nie je primárne určený na všeobecné účely. Jeho nevýhodou oproti všeobecne zameraným jazykom ako C, C++ je okrem iného najmä rýchlosť.

3.2 Logický model aplikácie

Celé ovládanie aplikácie pre používateľa by malo byť dostupné z prostredia webovej stránky. Všetky ovládaci prvky by mali byť čo najviac jednoduché a intuitívne, no v prípade potreby bude k dispozícii nápoveda. Po načítaní stránky si užívateľ v menu vyberie spôsob, akým chce zadať úlohu. Konkrétny spôsob bude záležať najmä na type dát, ktorými na začiatku disponuje. Môže ísť o už vypočítanú kovariančnú maticu, dáta vo forme hodnôt aktív v daných časoch, alebo len zoznam povolených aktív v portfóliu spolu s časom v ktorom chceme tieto dáta analyzovať.

Samotné webovské rozhranie a jeho súčasti budú fungovať na základe nasledujúcej schémy, ktorá je zjednodušene naznačená aj na Obr. 2 .



Obr. 2: Schéma funkcionality

Vstupné údaje zadané používateľom sa spracujú do textovej podoby pomocou jazyka PHP. V prípade potreby sa získajú ďalšie dáta z internetu [2]. Následne sa preverí, či

[2]http://finance.yahoo.com/

sú dáta kompletné a spĺňajú požadované parametre. Skontrolované dáta sa pomocou textového súboru odošlú na spracovanie pre Octave, ktorý ich vyhodnotí, vypočíta optimálne váhy portfólia a prípadne beta koeficienty pre jednotlivé aktíva.

Výsledok sa opäť pomocou textových súborov prenesie k PHP, ktoré ho zobrazí na webovej stránke v textovej aj grafickej podobe a používateľ dostane možnosť uložiť výsledok do textového súboru, čo môže byť výhodné najmä pri veľkom množstve aktív.

3.3 Kvadratické programovanie v Octave

Podľa [7] má úloha kvadratického programovania na minimalizáciu funkcie

$$min\frac{1}{2}x^T Hx + x^T q \tag{20}$$

pri ohraničeniach

$$Ax = b$$

$$lb \leq x \leq ub$$

$$A_{lb} \leq A_{in} \leq A_{ub}$$

nasledovnú syntax:

$$[x, obj, info, lambda] = qp\left(x_0, H, q, A, b, lb, ub, A_{lb}, A_{in}, A_{ub}\right)$$

Vstupné parametre môžeme opísať takto:

- x_0 je nepovinný parameter pre počiatočný bod algoritmu
- H, q určujú tvar účelovej funkcie podľa 20
- lb, ub sú vektory ohraničení premennej x
- A, b vyjadrujú ohraničenia v tvare rovníc $Ax - b = 0$
- A_{lb}, A_{in}, A_{ub} predstavujú ohraničenia v tvare nerovníc $A_{in} - A_{lb} \geq 0$, resp. $-\left(A_{in} - A_{ub}\right) \geq 0$.

3.4 Používateľské rozhranie

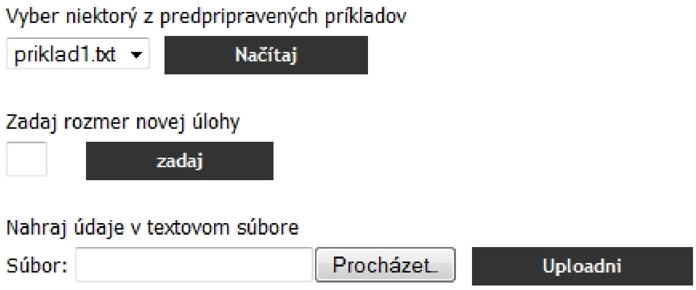
Ako uvádzame v predošlej časti, cieľom je vytvoriť webovskú aplikáciu ktorá bude jednoduchá na obsluhu. Používateľ dostane na výber 3 možnosti pre zadanie úlohy, ku ktorým má prístup cez menu na hlavnej stránke, ako máme možnosť vidieť na obrázku 3.

Obr. 3: Hlavná stránka

3.4.1 Zadávanie pomocou kovariančnej matice

Tento typ zadania bude možné použiť napríklad keď už máme kovariančnú maticu a vektor výnosov vopred vypočítané, alebo sme sa k nim dostali prostredníctvom ďalšej osoby. Používateľ dostane možnosť vybrať novú úlohu, načítať vopred pripravený príklad, alebo nahrať na server dáta v textovom súbore. Tieto možnosti vidíme na obrázku 4.

V prípade novej úlohy vyberie počet aktív v portfóliu. Následne sa zobrazia okná pre vloženie kovariančnej matice a vektora výnosov(Obr. 5), ktoré sú potrebné pre vypočítanie optimálnych váh akcií. Zadané údaje bude možné uložiť do textového súboru pre jednoduchšie použitie v budúcnosti, alebo priamo pokračovať v úlohe.



Obr. 4: Zadávanie kovariančnou maticou

Obr. 5: Kovariančná matica a vektor výnosov

Takto zadané údaje bude potrebné doplniť o celkový očakávaný výnos portfólia a hranice pre percentuálne obmedzenie množstva jedného aktíva v celku.

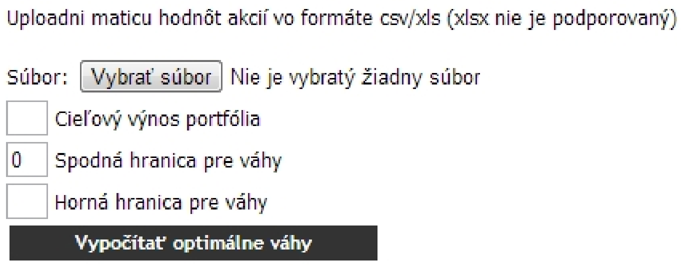
V prípade zvolenia možnosti nahrania textového súboru na server budú údaje v ňom uložené automaticky zobrazené do formulára pre zadávanie úlohy. Cielený výnos a obmedzenia pre aktíva patria medzi preferencie užívateľa, preto sa do súbora ukladať nebudú.

Nakoniec bude možnosť načítať predpripravenú úlohu aj so všetkými parametrami.

3.4.2 Zadávanie pomocou matice hodnôt aktív

V situácii, keď máme k dispozícii historické dáta s hodntotami aktív, nie je potrebné vypočítavať z nich kovariančnú maticu, ani vektor očakávaných výnosov. Pohodlnejšie bude nahrať ich priamo na server, kde tieto výpočty vykoná Octave. Keďže tieto dáta sa najčastejšie objavujú v súboroch typu *.csv a *.xls, teda vo formáte Microsoft Excel, aplikácia ponúka možnosť nahrať ich na server priamo v týchto formátoch(Obr. 6).

Po doplnení preferenčných údajov bude úloha pripravená na odoslanie.

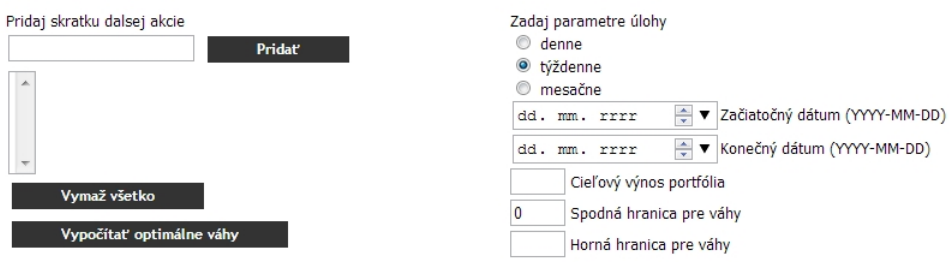


Obr. 6: Zadávanie maticou hodnôt aktív

3.4.3 Zadávanie získaním dát z Yahoo Finance

Posledná možnosť na zadanie vstupných dát bude najjednoduchšia a pravdepodobne najviac používaná. Užívateľ si vo vstupnom formulári vyberie aktíva, medzi ktoré chce rozdeliť svoje portfólio. Tieto aktíva sú reprezentované skratkami, ktoré sú bežne používané na finančných trhoch. Markowitzova teória ďalej vyžaduje časový rozsah a periódu historických dát aktív pre výpočet kovariančnej matice a vektora výnosov. Tieto a ďalšie potrebné údaje, ako cieľový výnos portfólia a hranice pre váhy, budú takisto zadávané v príslušných oknách vo formulári. Vzhľad tohoto formulára máme možnosť vidieť na obrázku 7. Treba ale mať na zreteli, že pokročilé vkladanie dátumu je k dispozícii iba na prehliadačoch podporujúcich najnovšiu verziu HTML. Pre ostatné je k dispozícii vkladanie textovým reťazcom, ktorého požadovaný tvar je uvedený vedľa samotného okna.

V tomto prípade bude taktiež možné spočítať beta koeficienty aktív, keďže na rozdiel od predchádzajúcich spôsobov zadávania poznáme dátumy a periódu. To nám dáva možnosť nahradiť výnosy trhového portfólia vo vzorci $\beta_i = \frac{Cov(r_i,r_M)}{Var(r_M)}$ indexom S&P 500 s rovnakými časovými parametrami ako ostatné aktíva.



Obr. 7: Získavanie dát z Yahoo Finance

4 Implementácia

4.1 Programovanie aplikácie

Ako formálny základ aplikácie sme použili voľne dostupnú šablónu z internetu[3]. Tento základ sme si však po funkčnej aj výzorovej stránke prispôsobili podľa našich potrieb. Ďalším krokom je naprogramovanie aplikácie podľa schémy navrhnutej v predošlej kapitole, ktorá sa vyskytuje v troch mierne odlišných verziách, v závislosti od použitej metódy zadania úlohy. V tejto kapitole uvedieme niekoľko ukážok z programu, zdrojový kód celej aplikácie bude k dispozícii na CD prílohe.

4.1.1 Zadávanie údajov

Vstupné formuláre sú základom zobrazenej webovej stránky a tvoria hlavnú časť komunikácie užívateľa s aplikáciou. Pomocou nich sú zadávané všetky vstupné dáta, pokiaľ nie sú načítavané priamo zo súboru.

Spoločný formulár pre všetky metódy zadávania tvoria údaje pre cieľový výnos a obe hranice pre váhy. Vypadá takto:

```
<input id = "tabulka" type="textfield" name="target_r3" value = <?php echo
    $_SESSION['target_r3']?> /><label for="file">&nbspCieľový výnos
    portfólia</label><br />
<input id = "tabulka" type="textfield" value = "0" name="lower_b3"
    /><label for="lower_b3">&nbspSpodná hranica pre váhy</label><br />
<input id = "tabulka" type="textfield" name="upper_b3" /><label
    for="upper_b3">&nbspHorná hranica pre váhy</label><br /> //$
```

Špecificky, pre zadávanie pomocou kovariančnej matice používateľ vyberie jednu z troch možností. Prvou je načítanie predpripravenej úlohy uloženej v textovom súbore na serveri, ktorá slúži ako ilustračný príklad. Deje sa tak pomocou HTML a PHP kódu:

[3] http://www.phpform.org/

```
<select name="subory">
<?php
    for($i=0; $i < count($arr); $i++) // premenná $arr obsahuje pole názvov
        príkladov
    echo "<option value=". $arr[$i] .">" . $arr[$i] . "</option>";
?>
</select>
<input type="submit" name="nacitaj" value="Nacitaj" />
```

Ďalšia možnosť je vybrať zadanie novej úlohy. Používateľ si v jednoduchom textovom okne vyberie rozmer úlohy (teda počet aktív v portfóliu) a následne sa zobrazí formulár s $n \times n$ oknami pre kovariančnú maticu a $1 \times n$ oknami pre vektor výnosov.

```
<table border="0">
    <?php for($counter = 1;$counter<=$_SESSION['rozmer'];$counter++){ ?>
        <tr>
        <?php for($counter2 =
            1;$counter2<=$_SESSION['rozmer'];$counter2++){ ?>
            <td>
                <input id = "tabulka" type="textfield" name="text_cov<?php
                    echo $counter;?>,<?php echo $counter2;?>" />
            </td>
        <?php }?>
        </tr>
    <?php }?>
 </table>

<table border="0">
    <tr>
    <?php for($counter3 = 1;$counter3<=$_SESSION['rozmer'];$counter3++){ ?>
```

```
        <td>
            <input id = "tabulka" type="textfield" name="text_return<?php
                echo $counter3;?>" />
        </td>
    <?php }?>
    </tr>
</table>
```

Tento formulár je po vyplnení možné odoslať a zobraziť výsledky, alebo ho uložiť do textového súboru pre potreby ďalšieho použitia v budúcnosti. Súbor obsahuje $n+1$ riadkov, pričom prvých n obsahuje prvky kovariančnej matice a v poslednom sa nachádza vektor výnosov. V tomto tvare je možné textový súbor aj nahrať na server, po čom sa automaticky zobrazí vo formulári.

```
<label for="file">Filename:</label>
<input type="file" name="txt" id="file1">
<input type="submit" name="uploadni" value="Uploadni" /><br />
```

Rovnakým spôsobom sa zadáva aj úloha v tvare matice hodnôt akcií. Rozdiel je v tvare, kde celý súbor obsahuje len hodnoty aktív v určitých časoch a v jeho formáte. Namiesto obyčajného textového súboru používame formát kompatibilný s programom Microsoft Excel.

Zadávanie získavaním dát z finance.yahoo.com obsahuje najväčší formulár. Okrem spoločnej časti navyše obsahuje aj textové pole pre zadávanie nových skratiek aktív do portfólia, ktoré sa ukladajú do premennej

```
$_SESSION['skratky']
```

a údaje o čase, v ktorom sledujeme aktíva. Pridaná časť je v takomto tvare:

```
<input type="radio" name="period" value="daily">
```

```
denne<br />
<input type="radio" name="period" value="weekly" checked>
týždenne<br />
<input type="radio" name="period" value="monthly">
mesačne<br />
<input type="date" name="start" value = <?php echo $_SESSION['start']?>
    ><label for="start">Začiatočný dátum (YYYY-MM-DD)</label><br />
<input type="date" name="end" value = <?php echo
    $_SESSION['end']?>><label for="end">Konečný dátum
    (YYYY-MM-DD)</label><br />
```

Podobne ako v prípade kovariančnej matice, aj tu môžeme využiť niekoľko príkadov, po ktorých zvolení sa formulár automaticky vyplní.

4.1.2 Spracovanie dát pomocou PHP

Po stlačení tlačidla "Vypočítať optimálne váhy"sa formulár odošle a jeho obsah sa podrobí kontrole. Pomocou funkcií v PHP skontrolujeme či je formulár úplný a či sú všetky údaje v ňom platné. Pre overenie číselnosti údajov používame zabudovanú PHP funkciu **is_numeric()**.

Nasledovný kód kontroluje správnosť zadaného cieľového výnosu a hraníc pre váhy.

```
function kontrola_hranic($target, $lower, $upper) {
    $kontrola =1;
    if (!( is_numeric($target) && ($target>(-1) ) )) {
        $kontrola = 0;
    }
    if ( (!is_numeric($lower) ) && (!$lower=="") ) {
        $kontrola = 0;
    }
    if ( (!is_numeric($upper) ) && (!$upper=="") ) {
```

```
        $kontrola = 0;
    }
    if (is_numeric($lower) && is_numeric($upper) && ($upper<$lower) ) {
        $kontrola = 0;
    }
    return $kontrola;
}
```

Kvôli prehliadačom nepodporujúcim najnovšiu verziu HTML sme vytvorili funkciu na overenie správnosti formátu zadaného dátumu:

```
function datum($datum) {
    $valid = 1;
    $pole = explode('−', $datum);
    // checkdate je zabudovaná funkcia PHP
    if ((count($pole) != 3) || (!checkdate($pole[1],$pole[2],$pole[0]))){
        $valid = 0;
    }
    return $valid;
}
```

Okrem kontroly zadaných údajov potrebujeme stiahnuť historické dáta všetkých akcií uvedených vo vektore skratiek.

Toto dosiahneme pomocou funkcií **file_get_contents()** a **file_put_contents()**. Dostaneme tým stiahnuté súbory vo formáte csv. Následne pomocou funkcie **fgetcsv** z nich vyberieme práve stĺpec s historickými dátami a uložíme ho do premennej typu pole v PHP.

```
file_put_contents($ptf . " userfiles /Tmp" . $skratka . $id . ".csv",
    file_get_contents($url));
```

```
// prikaz na stiahnutie z yahoo, v premennej $url sa nachádza adresa na priame
    stiahnutie súboru
$myFile = fopen($ptf . "userfiles/Tmp". $skratka . $id . ".csv", "r");
$data = fgetcsv($myFile, 1000, ",");
    unset($pompole);
    while (($data = fgetcsv($myFile, 1000, ",")) != FALSE) {
        $pompole[] = $data[6];
    }
    $pole = array_reverse($pompole);
    // do premennej $pole nacitavam siesty stlpec zo stiahnuteho suboru, pretoze
        prave ten obsahuje zelanu historicke data
    // kvoli opacnemu poradiu datumov pole prevratim
    fclose($myFile);
```

V prípade že všetky údaje boli zadané správne, ich uložíme do textových súborov s jednoznačným identifikačným prvkom pre každého používateľa, ktorý je tvorený tzv. časovou známkou, teda údajom ktorý spája súbor s presným časom, kedy používateľ načítal webovú stránku. Súbory sú zapísané do tvaru čitateľného v Octave. Nasledujúca funckia napríklad s pomocou knižnice **phpexcel** zapisuje do textového súboru maticu s hodnotami akcií a taktiež cieľový výnos s hranicami váh.

```
$myFile = $ptf."userfiles/subor2".$id.".m";
// $id obsahuje časovú známku
$fh = fopen($myFile, 'w') or die("can't open file");
$stringData1 = "P = [";
fwrite($fh, $stringData1);
foreach ($worksheet->getRowIterator() as $row) {
    $cellIterator = $row->getCellIterator();
    $cellIterator->setIterateOnlyExistingCells(false); // Loop all cells, even if
        it is not set
```

```
    foreach ($cellIterator as $cell) {
        if (!is_null($cell)) {
            $p = $cell->getValue();
            fwrite($fh, $p);
            fwrite($fh,",");
        }
    }
    fwrite($fh,";");
}
$stringData2 = "];\n";
fwrite($fh, $stringData2);
$stringData = "rt = [" . $_POST['target_r2'] . "];\n lb = [" .
    $_POST['lower_b2'] . "];\n ub = [" . $_POST['upper_b2'] . "];\n";
fwrite($fh, $stringData);
```

4.1.3 Výpočet v Octave

V prípade zadávania cez kovariančnú maticu je taktiež nutné overiť jej vlastnosti. Kovariančná matica z definície musí byť symetrická a kladne semidefinitná. Na overenie týchto vlastností už ale PHP nestačí, preto použijeme Octave. Príslušný PHP kód, ktorý spúšťa Octave má nasledovný tvar:

```
$command = $pto . " -q --eval \"addpath('".$ptf."/system/');
    over('".$ptf."','".$id."') \" --persist ";
// $pto, $ptf, $id sú parametre
exec($command)
```

Funkcia **over.m** má na starosti overiť kovariančnú maticu uloženú v textovom súbore, ktorého názov je jednoznačne daný parametrami. Octave úlohu vyhodnotí pomocou porovnania matice s jej transpozíciou a overením nezápornosti vlastných hodnôt matice. Výsledok kontroly zapíše do ďalšieho textového súboru, ktorý potom prečíta a vyhodnotí PHP.

```
prikaz = strcat(path_to_files, " userfiles /subor1",id,".m");
run(prikaz);
koniec = 1;
for i = 1:1:length(r)
   if (r(i)<−1)
         koniec = 0;
         endif
         endfor
if ((V == (V')))
    d = eig(V);
    for i=1:1:length(r)
        if (d(i)<=0)
            koniec = 0;
            endif
            endfor
else
    koniec = 0;
    endif

if (rt <−1)
    koniec = 0;
    endif
subor = strcat(path_to_files,' userfiles /overit',id,'.txt');
dlmwrite(subor,koniec);
```

Hlavnou úlohou Octave je samotný výpočet beta koeficientov a váh pre optimálne porfólio. PHP zadá príkaz na výpočet v rovnakej forme ako pri kontrole kovariančnej matice. Octave načíta zadané hodnoty z textového súboru. V prípade, že dáta sú zadané v matici s hodnotami aktív P, sa príslušná kovariančná matica a vektor výnosov vypočíta takto:

```
prikaz = strcat(path_to_files, ' userfiles /subor3', id, '.m');
run(prikaz);
prikaz2 = strcat(path_to_files, ' userfiles /subor3SP', id, '.m');
run(prikaz2);

[n,m] = size(P);
R = zeros(n-1,m);
%vypocitam maticu vynosov
for i = 1:1:(n-1)
  for j = 1:1:m
     R(i,j) = P(i+1,j)/P(i,j) - 1;
  end
end
Er = sum(R,1)/(n-1);
H = cov (R);
```

V prípade, že používame dáta z Yahoo Finance, použijeme jednoduchý výpočet pre beta koeficienty:

```
nSP = length(SP); % hodnoty trhoveho portfolia
SPR = zeros(nSP-1,1);
for j = 1:1:(nSP-1)
     SPR(j) = SP(j+1)/SP(j) - 1;
end

suborB = strcat(path_to_files,' userfiles /vystup3B',id,'.txt');
vystupB = fopen(suborB,'wt');
```

```
for j = 1:1:(length(x))
    pom = (cov(R(:,j), SPR))/(var(SPR));
    fprintf(vystupB,'%f\n',pom);
end
```

Načítané a upravené matice, vektory a premenné upravíme na tvar použiteľný vo funkcii kvadratického programovania, následne ju použijeme a získané výsledky zapíšeme do súboru.

```
f = zeros(m,1);
A = Er;
b = rt;
Aeq = ones(1,m);
beq = 1;
%upravenie hranic na vektory
if(size(lb)>0)
    lb = lb*ones(m,1);
end
if(size(ub)>0)
    ub = ub*ones(m,1);
end
%výpočet a zápis optimálnych váh
[x, obj, info, lambda] = qp ([], H, f, Aeq, beq, lb, ub, b, A, []) ;
%zápis výsledkov do súboru
subor = strcat(path_to_files,' userfiles /vystup3',id, '.txt');
vystup = fopen(subor,'wt');
for i = 1:1:length(x)
    fprintf(vystup,'%f\n',x(i));
end
```

4.1.4 Odoslanie a zobrazenie výsledkov

Textové súbory s vypočítanými váhami a betami načíta opäť PHP do príslušných premenných.

```
$subor = fopen($ptf."userfiles/vystup3".$id.".txt", "rb");
if($subor) {
    while (!feof($subor) ) {
        $r = floatval(fgets($subor));
        $result [] = $r;
    }
    fclose($subor);
}
```

Rovnakým spôsobom v prípade potreby načítavame aj beta koeficienty. PHP tieto výsledky ďalej interpretuje do tabuľkovej a grafickej podoby. Grafy sú vykresľované pomocou knižnice **pchart**[4].

Používateľ má možnosť jedným kliknutím tieto výsledky stiahnuť do súboru typu **csv** a zobraziť ho napríklad programom Microsoft Excel. Toto je užitočné najmä v prípade potreby archivácie výsledkov pre väčšie portfólio kvôli ďalšiemu použitiu.

[4] http://www.pchart.net/

5 Ilustračné príklady

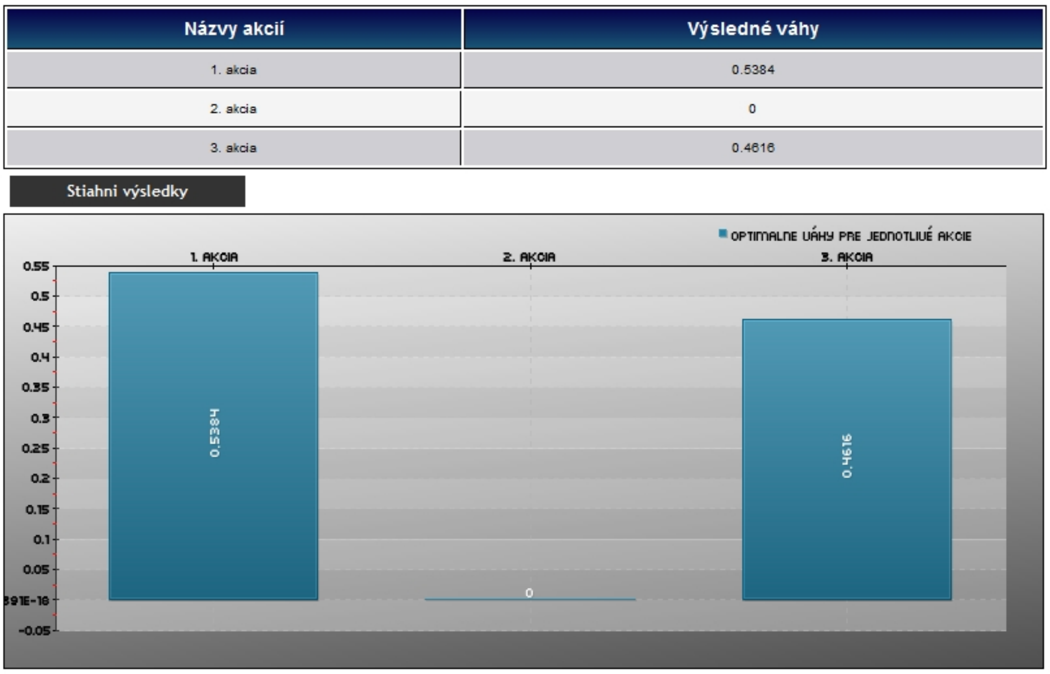
Funkčnosť webovej aplikácie si ilustrujeme na dvoch typických príkladoch. Bude to zadávaním kovariančnou maticou a získavaním dát z Yahoo Finance.

5.1 Príklad 1

V tomto príklade si priblížime zadávanie cez kovariančnú maticu. Keďže chceme načítať maticu, ktorá je už uložená, vyberieme možnosť **načítať príklad**, tak ako to vidno na obrázku č. 8. Matica rozmeru 3×3, čo zodpovedá 3 aktívam v portfóliu, sa aj s príslušným vektorom výnosov načíta do formulára rovnako ako na obrázku č. 9. Pre úplnosť úlohy je potrebné zadať požadovaný výnos portfólia, ktorý si určíme na hodnotu 0.1654. Po stlačení tlačidla **Vypočítať optimálne váhy** aplikácia vykoná výpočet, a výsledné váhy zobrazí do tabuľky a grafu (Obr. 10).

Obr. 8: Príklad 1 - výber úlohy

Obr. 9: Príklad 1 - zadanie



Názvy akcií	Výsledné váhy
1. akcia	0.5384
2. akcia	0
3. akcia	0.4616

Obr. 10: Príklad 1 - výsledok

5.2 Príklad 2

Druhý príklad je zameraný na získavanie dát z Yahoo Finance. Podobne ako v predošlom prípade vyberieme jeden z pripravených príkladov (Obr. 13) a po stlačení tlačidla sa vyplní formulár vybranými údajmi (Obr. 14). Nakoľko dáta získané takýmto spôsobom obsahujú viac informácií, je možné vypočítať aj beta koeficienty pre jednotlivé aktíva.

Akcie pre tento príklad boli vybrané ako súčasť školského projektu. Konkrétne ide o porovnanie 10 spoločností zastupujúcich letecký priemysel. Prehľad vidíme v tabuľke 1.

Ako historické údaje boli vybrané dáta z prvej polovice roku 2012 s týždennou periódou. Za očakávaný týždenný výnos sme si určili jedno percento.

Po odoslaní formulára sa najprv vypočíta kovariančná matica tvaru 10×10 a vektor výnosov veľkosti 1×10. Vektor aj matica sú uvedené pre poradie aktív, v akom sú zadané v tabuľke 1.

Po dokončení výpočtu programom Octave dostávame výsledky vo forme tabuľky a grafu (15). Pri pohľade na výsledné váhy, kovariančnú maticu a vektor výnosov sa zdá, že do portfólia boli vybrané aktíva s nižšou varianciou a kovarianciou s inými aktívami,

UAL	UNITED CONFIDENTAL HLDGS INC
CPA	COPA HOLDINGS SA
RYYAY	RYANAIR
LFL	LAN AIRLINES SA
JBLU	JETBLUE AIRWAYS CORP
HA	HAWAIIAN HOLDINGS INC
GLUX	GREAT LAKES AVIATION LTD
DAL	DELTA AIR LINES INC
ALK	ALASKA AIR GROUP INC
ALGT	ALLEGIANT TRAVEL CO

Tabuľka 1: Zoznam aktív

$$\begin{pmatrix}
0.00523 & 0.00152 & 0.00177 & 0.000601 & 0.0038 & 0.00256 & 0.0021 & 0.00402 & 0.00183 & 0.0016 \\
0.00152 & 0.00134 & 0.000685 & 0.00039 & 0.00125 & 0.00102 & 0.00194 & 0.00155 & 0.00093 & 0.000931 \\
0.00177 & 0.000685 & 0.00158 & 0.000745 & 0.00118 & 0.00112 & 0.00165 & 0.00146 & 0.000832 & 0.000811 \\
0.000601 & 0.00039 & 0.000745 & 0.00187 & 0.000978 & 0.000624 & 0.000316 & 0.000623 & 0.00061 & 0.00029 \\
0.0038 & 0.00125 & 0.00118 & 0.000978 & 0.00434 & 0.00256 & 0.000498 & 0.00306 & 0.00187 & 0.00161 \\
0.00256 & 0.00102 & 0.00112 & 0.000624 & 0.00256 & 0.00381 & 0.00119 & 0.00216 & 0.00115 & 0.00108 \\
0.0021 & 0.00194 & 0.00165 & 0.000316 & 0.000498 & 0.00119 & 0.0212 & 0.00141 & 0.00232 & 0.00201 \\
0.00402 & 0.00155 & 0.00146 & 0.000623 & 0.00306 & 0.00216 & 0.00141 & 0.0039 & 0.00149 & 0.00149 \\
0.00183 & 0.00093 & 0.000832 & 0.00061 & 0.00187 & 0.00115 & 0.00232 & 0.00149 & 0.00133 & 0.00124 \\
0.0016 & 0.000931 & 0.000811 & 0.00029 & 0.00161 & 0.00108 & 0.00201 & 0.00149 & 0.00124 & 0.00177
\end{pmatrix}$$

Obr. 11: Kovariančná matica

napriek často nižšiemu výnosu.

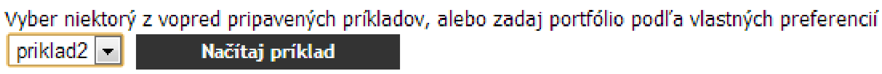
Taktiež by bolo zaujímavé sledovať, ako sa sa správajú výnosy niektorých aktív v porovnaní s indexom S&P 500. Vyberieme si na to aktíva podľa ich kovariancie s trhom, teda vlastne podľa koeficientu beta. Ako vidíme v kovariančnej matici, všetky aktíva s indexom S&P 500 korelujú kladne, preto nemôžeme očakávať nezvykle sa správajúce akíva. Zvolíme si teda aktíva, ktoré majú betu najvyššiu (Lan Airlines), najnižšiu (Allegiant travel) a čo najbližšiu jednotke (Ryanair).

Na obrázkoch 16, 17 a 18 a vidíme porovnanie výnosov týchto aktív s výnosom indexu S&P 500 počas sledovanej polovice roka. Vzhľadom na to, že všetky aktíva po-

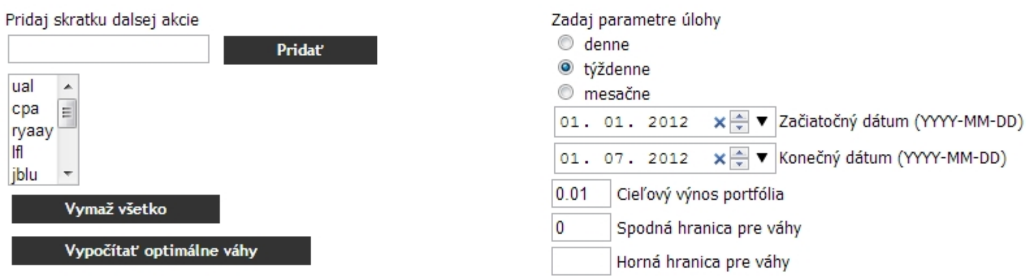
$$\begin{pmatrix} 0.0141 & 0.014 & 0.00435 & 0.00529 & 0.00103 & 0.00886 & 0.0405 & 0.0129 & -0.000476 & 0.0108 \end{pmatrix}$$

Obr. 12: Vektor výnosov

chádzajú z rovnakého odvetvia sa fluktuácie výnosov od seba nelíšia veľmi výrazne. Napriek tomu môžeme vidieť niekoľko pohybov grafu Lan Airlines opačným smerom ako u S&P 500, čo zrejme súvisí s nízkym beta koeficientom. Naopak, Allegian Travel svoj vysoký koeficient potvrdzuje viacerými výraznými pohybmi v smere indexu. Ryanair však najmä v druhom štvťroku pomerne verne kopíroval vývoj na trhu, preto jeho beta nie je prekvapivá.



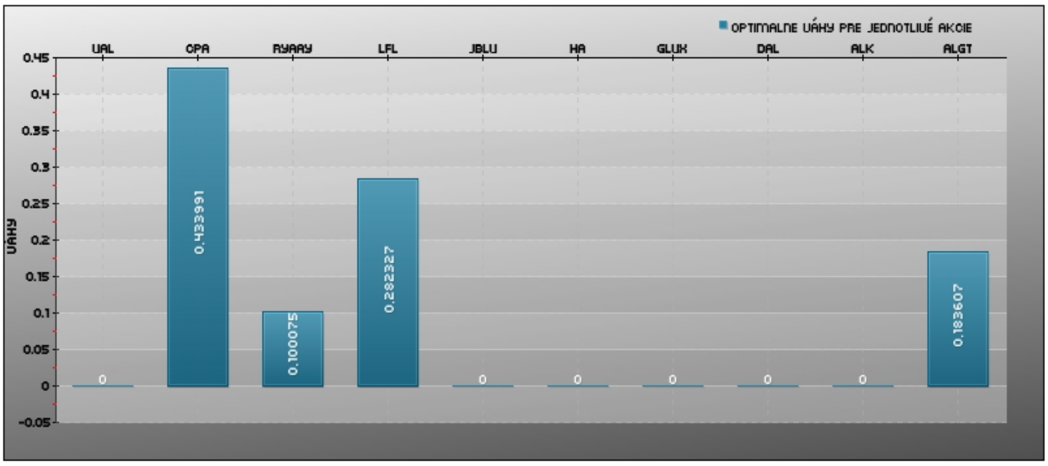
Obr. 13: Príklad 2 - výber úlohy



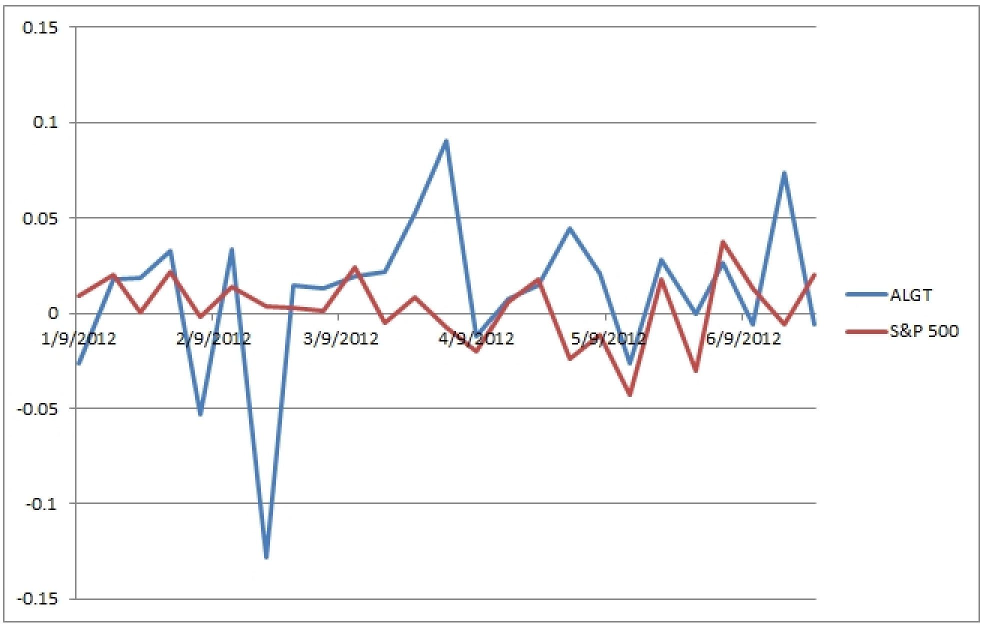
Obr. 14: Príklad 2 - zadanie

Názvy akcií	Výsledné váhy	Výsledné bety
ual	0	0.616895
cpa	0.433991	0.389692
ryaay	0.100075	0.978405
lfl	0.282327	1.351742
jblu	0	0.901746
ha	0	1.240717
glux	0	0.859612
dal	0	0.480887
alk	0	0.760905
algt	0.183607	0.198238

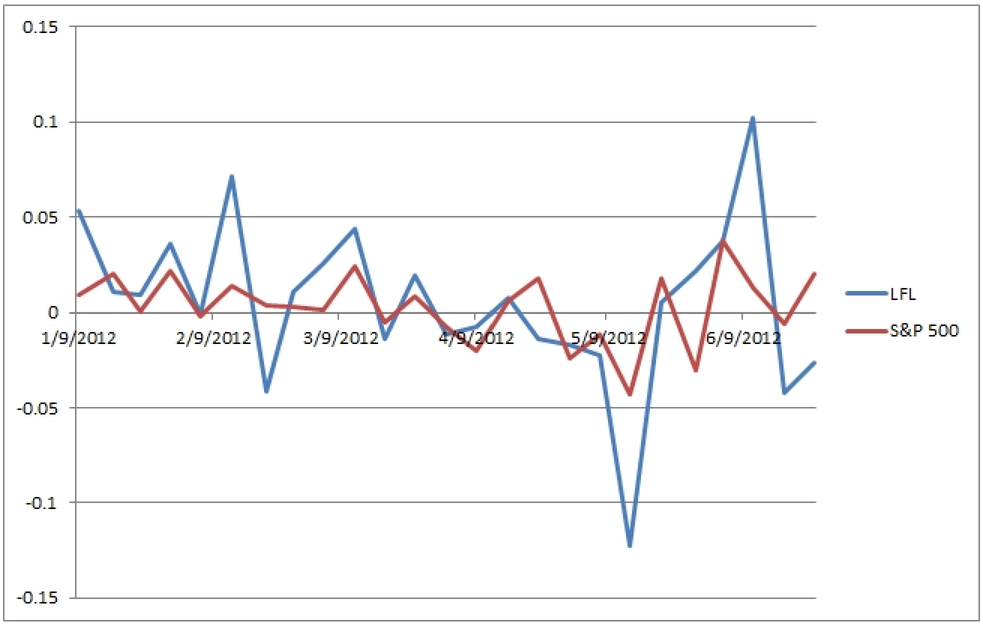
Stiahni výsledky



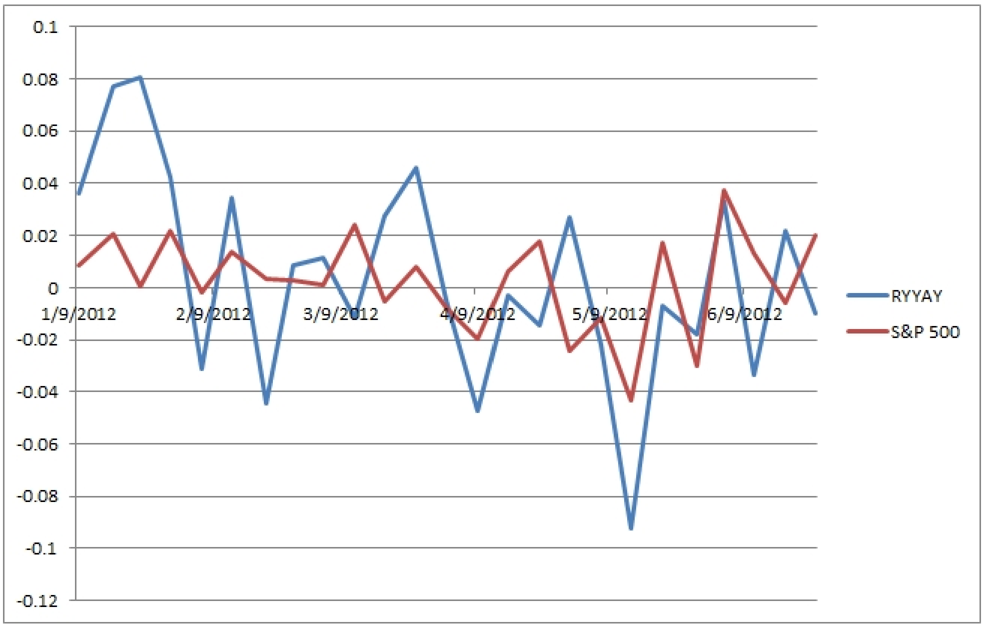
Obr. 15: Príklad 2 - výsledok



Obr. 16: Allegiant Travel



Obr. 17: Lan Airlines



Obr. 18: Ryanair

ZÁVER

Záver

V bakalárskej práci sme sa venovali úlohe o optimálnej skladbe portfólia. Zamerali sme sa najmä na Markowitzovu teóriu portfólia, ktorá hovorí o diverzifikácii portfólia za účelom minimalizácie volatility. Ukázali sme ďalej, ako takúto úlohu riešiť s pomocou kvadratického programovania.

Cieľom práce bolo na základe týchto teoretických vedomostí naprogramovať webovú aplikáciu, ktorej úlohou je práve riešenie Markowitzovej úlohy. Toto sme dosiahli pomocou PHP a Octave, a aplikácia bola umiestnená na voľne dostupný školský server.

Tento projekt by mohol byť prínosom pre každého, kto sa zaujíma o optimalizáciu portfólia pomocou minimalizácie rizika, avšak z finančných, časových, alebo iných dôvodov sú preňho štandardné prostriedky nedostupné alebo nevýhodné. Viem si napríklad predstaviť jeho využitie pri niektorých školských projektoch.

V budúcnosti je možné s aplikáciou ďalej pracovať, napríklad vylepšiť niektoré užívateľské prvky, zjednodušiť výber cieľového výnosu priamym určením bodu na efektívnej hranici, alebo ponúknuť zoznam aktív, z ktorých je možné portfólio zostaviť. Takisto by mohlo byť užitočné rozšíriť Markowitzov model o ďalšie, ako Capital Asset Pricing Model (CAPM) alebo faktorové modely. Do úvahy prichádzajú aj modely, ktoré priamo súvisia s optimalizáciou portfólia, pričom uvažovaná miera rizika nie je jednoduchá variancia ale niektorá zo zložiteších mier rizika ako napr. (Value at Risk) V@R, (conditional Value at Risk) CV@R, (Average Value at Risk) AV@R a ďalšie modely mier rizika obsiahnuté v práci [5].

ZOZNAM POUŽITEJ LITERATÚRY

Zoznam použitej literatúry

[1] Brealey, R. A., Myers, S. C.: *Principles of Corporate Finance, Seventh edition*, The McGraw-Hill Companies, New York, 2003

[2] Hamala, M.: *Nelineárne programovanie*, prednášky, FMFI UK, Bratislava, 2012

[3] Hamala, M.: *Nelineárne programovanie*, Alfa, Bratislava, 1972

[4] Hamala, M., Trnovská, M.: *Nelineárne programovanie*, EPOS, Bratislava, 2013

[5] Kilianová, S.: *Stochastické dynamické optimalizačné modely dôchodkového plánovania*, dizertačná práca, FMFI UK, Bratislava, dostupné na internete (30.5.2013): http://www.iam.fmph.uniba.sk/studium/efm/phd/kilianova/kilianova-thesis.pdf

[6] Melicherčík, I., Olšarová, L., Úradníček, V.: *Kapitoly z finančnej matematiky*, Epos, Bratislava, 2005

[7] GNU Octave: A high-level interactive language for numerical computations, dostupné na internete(30.5.2013): http://math.hawaii.edu/lab/197/octave.pdf

Príloha A

CD príloha CD nosič obsahujúci kompletný zdrojový kód aplikácie.