

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



MARKOWITZOV MODEL S KVANTITATÍVNOU
POŽIADAVKOU NA PESTROSŤ PORTFÓLIA

DIPLOMOVÁ PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MARKOWITZOV MODEL S KVANTITATÍVNOU
POŽIADAVKOU NA PESTROSŤ PORTFÓLIA

DIPLOMOVÁ PRÁCA

Študijný program: Ekonomicko-finančná matematika a modelovanie
Študijný odbor: 1114 Aplikovaná matematika
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky
Vedúci práce: doc. RNDr. Milan Hamala, CSc.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Michaela Iliťová
Študijný program: ekonomicko-finančná matematika a modelovanie
(Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

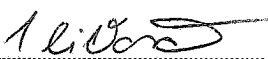
Názov: Markowitzov model s kvantitatívnou požiadavkou na pestrosť portfólia
Markowitz model with quantitative requirement on portfolio diversification

Cieľ: Cieľom diplomovej práce je návrh kvantitatívnej požiadavky na pestrosť portfólia v Markowitzovom modeli, jej formalizácia v tvare konvexnej úlohy, návrh metódy riešenia a numerické experimentovanie s modelom, v rámci ktorého budeme skúmať dopad navrhovanej požiadavky na pestrosť, s ohľadom na zmeny v požadovanom počte akcií v portfóliu, na výnos a riziko.

Vedúci: doc. RNDr. Milan Hamala, CSc.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.
Dátum zadania: 10.02.2015

Dátum schválenia: 11.02.2015

prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu


.....
študent


.....
vedúci práce

Pod'akovanie Touto cestou by som sa chcela pod'akovať môjmu školiteľovi, doc. RNDr. Milanovi Hamalovi, CSc. za jeho pomoc, ochotu, odborné rady, usmernenia a trpezlivosť pri písaní tejto práce, ako aj za poskytnutú voľnosť pri výbere témy. Ďakujem aj mojim rodičom, bratovi, starej mame a priateľom, ktorý pri mne stáli a podporovali ma.

Abstrakt v štátnom jazyku

ILITOVÁ, Michaela: Markowitzov model s kvantitatívnou požiadavkou na pestrosť portfólia [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Milan Hamala, CSc., Bratislava, 2016, 70 s.

Táto práca sa venuje diverzifikačným podmienkam v Markowitzovom modeli. Naším cieľom je navrhnutie kvantitatívnej požiadavky na pestrosť portfólia pre Markowitzov model, formulácia modelu v tvare konvexnej úlohy, návrh metódy jej riešenia a numerické experimentovanie, v ktorom sledujeme dopad zmien požiadavky na pestrosť portfólia na výnos a riziko. V práci popisujeme tri najčastejšie používané formulácie Markowitzovho modelu optimalizácie portfólia. V stručnosti sa venujeme diverzifikácii portfólia v prostredí Markowitzovho modelu a niektorým v literatúre používaným diverzifikačným podmienkam. Formulujeme kvantitatívnu požiadavku na minimálnu pestrosť portfólia v tvare konvexného kvadratického ohraničenia. Uvádzame Markowitzov model rozšírený o lineárne transakčné náklady a požiadavku na minimálnu pestrosť portfólia, pre ktorý testujeme vplyv zmeny požiadavky na pestrosť na strednú hodnotu výnosu a riziko, vyjadrené veľkosťou smerodajnej odchýlky výnosu portfólia. Porovnávame metódy vnútorného a vonkajšieho bodu pri riešení zovšeobecneného Markowitzov modelu pomocou nami vytvoreného generátora úloh. Vplyv zvyšovania hodnoty minimálnej požiadavky na pestrosť portfólia na zmeny výnosu a rizika portfólia prezentujeme na numerických experimentoch, v ktorých na optimalizáciu využívame metódu vnútorného bodu.

Kľúčové slová: Markowitzov model, Diverzifikácia, Metóda vnútorného bodu, Metóda vonkajšieho bodu, Výnos a riziko portfólia

Abstract

ILITOVÁ, Michaela: Markowitz model with quantitative requirement on portfolio diversification [Master thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: doc. RNDr. Milan Hamala, CSc., Bratislava, 2016, 70 p.

This thesis addresses diversification requirements in Markowitz model. Our aim is to propose a quantitative requirement on portfolio diversification in the framework of Markowitz model, to formulate the model as a convex problem, to suggest a method of finding its solution and to numerically experiment, where the impact of changes in the proposed portfolio diversification requirement on return and risk of the portfolio is to be monitored. Three most commonly used formulations of Markowitz portfolio optimisation model are described in this thesis. Portfolio diversification in the framework of Markowitz model, as well as several diversification constraints used in literature are briefly discussed. We formulate quantitative requirement on minimal portfolio diversification as a convex quadratic constraint. Markowitz model extended by linear transactional costs and quantitative constraint on least portfolio diversification is introduced, for which the impact of change in the diversification requirement on the expected value of return and risk, given by volatility of returns of portfolio is tested. Comparison of interior and exterior point methods in solving the extended Markowitz model using a generator of problems, we created, is carried out. The impact of changes in minimal quantitative requirement on portfolio diversification on return and risk is presented in numerical experiments, in which interior point method is used as the optimisation algorithm.

Keywords: Markowitz Model, Diversification, Interior Point Method, Exterior Point Method, Return and Risk of Portfolio

Obsah

Úvod	8
1 Markowitzov model a jeho zovšeobecnenie	10
1.1 Formulácia Markowitzovho modelu	10
1.2 Otázky diverzifikácie portfólia	12
1.2.1 Diverzifikácia portfólia v literatúre	12
1.2.2 Kvantitatívna požiadavka na pestrosť	13
1.3 Zovšeobecnený Markowitzov model	14
1.3.1 Formulácia modelu	14
1.3.2 Teoretické zdôvodnenie	16
2 Programy použité na riešenie optimalizačného modelu	19
2.1 Metóda vnútorného bodu	19
2.2 Metóda vonkajšieho bodu	21
2.3 Evolučné algoritmy	22
3 Numerický experiment	23
3.1 Metodológia	23
3.2 Podmienky optimality	23
3.2.1 Kuhn-Tuckerove podmienky	23
3.2.2 Konzistentnosť úlohy	24
3.3 Algoritmus generátora	26
3.4 Vytvorené programy	27
3.4.1 Generátor úloh	28
3.4.2 Prispôsobenie MATLAB-ovského programu IPM	32
3.4.3 Prispôsobenie MATLAB-ovského programu EPM	35
3.5 Ilustratívny príklad	37
3.6 Porovnanie algoritmov	39
3.6.1 Porovnanie vzhľadom na rozmer úloh n	40
3.6.2 Porovnanie vzhľadom na požiadavku pestrosti portfólia m_1	45
3.7 Vplyv požiadavky na pestrosť portfólia na výnos a riziko	47

Záver	52
Zoznam použitej literatúry	54
Príloha A	56
Príloha B	62
Príloha C	68

Úvod

Odveká túžba ľudí zväčšovať množstvo a hodnotu svojho majetku bola a je hnacím motorom nápadov, myšlienok, postupov a metód zvyšovania zisku. Prelomovým medzi nimi bol model optimalizácie portfólia Harryho Markowitza, ktorý bol publikovaný v roku 1959 v monografii [9], vďaka ktorému investor mohol na jednej strane maximalizovať svoj očakávaný zisk a na druhej strane ohraničiť možné riziko dané veľkosťou smerodajnej odchýlky výnosu portfólia. Od jeho publikovania sa Markowitzovým modelom zaoberalo veľké množstvo autorov, ktorí sa ho pokúšali vylepšiť, alebo aplikovať na konkrétne situácie.

V tejto práci sme sa zamerali na diverzifikačné podmienky v Markowitzovom modeli, ktoré pomáhajú investorovi lepšie regulovať podstupované riziko. Nakoľko v literatúre sme sa v prostredí Markowitzovho modelu nestretli s ohraňčeniami na pestrosť portfólia dané počtom zahrnutých navzájom rôznych akcií, za cieľ tejto práce sme si stanovili návrh takejto kvantitatívnej požiadavky na pestrosť portfólia, jej formuláciu v tvare konvexného kvadratického ohraňčenia vstupujúceho do Markowitzovho modelu, návrh metódy riešenia takto rozšíreného modelu ako aj skúmanie vplyvu zmeny požiadavky na pestrosť na výnos a riziko portfólia.

Práca je členená nasledovne: V prvej kapitole sa venujeme trom používaným formuláciám Markowitzovho modelu a popisujeme premenné, ktoré doňho vstupujú. V skratke sa venujeme problematike diverzifikácie portfólia, ako aj niektorým v literatúre sa vyskytujúcim diverzifikačným podmienkam. Formulujeme kvantitatívne ohraňčenia na pestrosť, avšak nespojito a popisujeme navrhovanú kvantitatívnu požiadavku na minimálnu pestrosť portfólia. Následne formulujeme zovšeobecnený Markowitzov model rozšírený o lineárne transakčné náklady a kvantitatívnu požiadavku na pestrosť, ktorá je formulovaná ako konvexné kvadratické ohraňčenie na minimálny počet navzájom rôznych akcií zahrnutých v portfóliu. Posledná časť prvej kapitoly sa venuje matematickému aparátu a dôkazom potrebným na teoretické zdôvodnenie formulácie požiadavky na pestrosť. Druhá kapitola sa venuje programom použitým na optimalizáciu zovšeobecneného Markowitzovho modelu. Uvádzame v nej formuláciu modelu pomocou štandardnej metódy I. fázy pre potreby metódy vnútorného bodu, popisujeme účelovú funkciu vchádzajúcu do metódy vonkajšieho bodu a v krátkosti sa venujeme aj evolučným algo-

ritmom. Najrozsiahlejšou kapitolou je posledná, tretia kapitola venujúca sa numerickým experimentom. V jej úvode popisujeme použitú metodológiu. Ďalej odvádzame Kuhn-Tuckerove podmienky optimality a podmienky konzistentnosti ohraničení potrebné na vytvorenie generátora úloh zovšeobecneného Markowitzovho modelu. Uvádzame algoritmus generátora úloh, popis jeho kódu, ako aj prispôsobenie MATLAB-ovských programov pre metódu vnútorného a vonkajšieho bodu. Výstupy generátora, spolu s nájdenými riešeniami pomocou metód vnútorného a vonkajšieho bodu demonštrujeme na ilustratívnom príklade malých rozmerov. Následne porovnávame obe metódy pri riešení vygenerovaných úloh pre rôzne počty akcií, z ktorých si volíme portfólio ako aj rôzne podmienky na minimálnu pestrosť. Posledná časť tretej kapitoly sa venuje vplyvu zmien požiadavky na pestrosť na výnos a riziko portfólia, ktorý skúmame pomocou zvyšovania požiadavky na pestrosť vo vygenerovaných úlohách.

1 Markowitzov model a jeho zovšeobecnenie

V tejto kapitole priblížime Markowitzov model výberu portfólia v troch používaných formuláciách, zhrnieme metódy diverzifikácie portfólia v literatúre, predstavíme navrhovanú požiadavku na pestrosť a sformulujeme zovšeobecnený Markowitzov model s požiadavkou na pestrosť a transakčnými nákladmi.

1.1 Formulácia Markowitzovho modelu

Markowitz svoj model prvý krát formuloval už v roku 1952 v článku [8] v zjednodušenej verzii pre 3 cenné papiere. Neskôr model viac rozpracoval v monografii [9] z roku 1959. Od vtedy sa Markowitzovým modelom zaoberalo mnoho autorov a vzniklo veľké množstvo jeho rôznych formulácií. Uvedieme tri z nich, ktoré sa často vyskytujú v literatúre. Tieto a ďalšie formulácie Markowitzovho modelu možno nájsť napríklad v [5], [19] a [18]. Najskôr si však vysvetlíme premenné, ktoré vstupujú do modelu.

Na začiatku si investor zvolí n aktív, do ktorých, resp. do ktorých častí chce investovať. Percento majetku investora, ktoré bude alokované do i -teho aktíva označíme x_i . Tradične sa v modeli nepripúšťajú krátke pozície (shorting) a teda $x_i \geq 0 \quad i = 1, \dots, n$. Vektor $x = (x_1, \dots, x_n)$ potom predstavuje nákup jednotlivých aktív a súčet jeho zložiek je

$$e^T x = 1,$$

kde e je vektor jednotiek $e = (1, \dots, 1)$. Výnos aktíva i , ktorý je náhodnou premennou, označíme r_i . Predpokladáme znalosť prvých dvoch momentov združeného rozdelenia $r = (r_1, \dots, r_n)$

$$E(r) := \mu$$

$$D(r) = E((r - \mu)(r - \mu)^T) := C.$$

Výnos portfólia je rovný $r^T x$. Jeho stredná hodnota je potom

$$E(r^T x) = \mu^T x$$

a smerodajná odchýlka

$$D(r^T x) = x^T C x.$$

Ohraničenia na nákup jednotlivých aktív, či už substitučné, protimonopolné, ohraničenia na minimálne množstvo, intervalové ohraničenia na zastúpenie jednotlivých aktív, alebo iné vstupujú do modelu cez systém nerovností

$$Ax \leq b,$$

kde $A \in R^m \times R^n$ a $b \in R^m$ sú matica a vektor ohraničení.

Ako prvú uvedieme formuláciu Markowitzovho modelu s maximalizáciou strednej hodnoty výnosu pri zhora ohraničenej hodnote smerodajnej odchýlky

$$\begin{aligned} \max \quad & \mu^T x \\ & e^T x = 1 \\ & x \geq \mathbf{0} \\ & Ax \leq b \\ & x^T C x \leq \sigma^2, \end{aligned} \tag{1}$$

kde σ^2 je horné ohraničenie na veľkosť smerodajnej odchýlky výnosu portfólia.

Markowitzov model môžeme formulovať aj ako úlohu minimalizácie smerodajnej odchýlky pri zdola ohraničenej hodnote výnosu

$$\begin{aligned} \min \quad & x^T C x \\ & e^T x = 1 \\ & x \geq \mathbf{0} \\ & Ax \leq b \\ & \mu^T x \geq \mu_{\min}, \end{aligned} \tag{2}$$

kde $\mu_{\min}$ je spodné ohraničenie na veľkosť výnosu portfólia.

Posledná formulácia Markowitzovho modelu, ktorú uvedieme spája maximalizáciu strednej hodnoty výnosu portfólia s minimalizáciou smerodajnej odchýlky

$$\begin{aligned} \max \quad & \mu^T x - \beta x^T C x \\ & e^T x = 1 \\ & x \geq \mathbf{0} \\ & Ax \leq b, \end{aligned} \tag{3}$$

kde $\beta > 0$ je váha, pomocou ktorej investor definuje požadovaný pomer medzi dôrazom kladeným na maximalizáciu výnosu, resp. minimalizáciu smerodajnej odchýlky v procese optimalizácie.

Každý racionálny investor sa snaží maximalizovať svoj výnos $\mu^T x$, avšak nie všetci sú averzný voči riziku a chcú ho minimalizovať. Preto sme sa rozhodli ďalej uvažovať formuláciu Markowitzovho modelu (1), kde sa na jednej strane maximalizuje výnos $\mu^T x$ a na druhej strane si investor môže sám určiť svoj rizikový apetít stanovením adekvátneho ohraničenia σ^2 na jeho smerodajnú odchýlku.

1.2 Otázky diverzifikácie portfólia

1.2.1 Diverzifikácia portfólia v literatúre

Diverzifikáciu autori často prirovnávajú k prísloviu "Nedávajte všetky vajčka do jedného košíčka", nakoľko toto príslovie jasne zachytáva možné dôsledky nediverzifikovanosti portfólia. Avšak správne rozlíšiť, čo znamená "jeden košíček" pri investovaní je náročná úloha. Rozloženie investície do väčšieho množstva aktív samo o sebe nemusí znamenať adekvátne zníženie rizika, ako to poznamenal už Markowitz v článku [8]. Príkladom je zahrnutie cenného papiera, ktorého výnos je silno korelovaný s niektorým z výnosov cenných papierov v portfóliu, alebo investícia do cenných papierov viacerých firiem z rovnakého odvetvia priemyslu. Takáto diverzifikácia zmiernuje dopad na investíciu len pri problémoch, resp. stratovosti cenných papierov niektorých zo zvolených firiem, avšak nie v prípade krízy v danom priemyselnom odvetví. Súvislosťou korelácie výnosov aktív a diverzifikácie sa zaoberal aj Markowitz v už spomínanej monografii [9] z roku 1959. Ani správna diverzifikácia portfólia samozrejme nedokáže eliminovať všetky riziká, avšak dá sa ňou do značnej miery eliminovať nesystémové riziko, čo popisujú autori napríklad v [2] a [14].

V literatúre sa ohraničenia na pestrosť aktív v portfóliu často uvádzajú cez intervalové ohraničenia na zastúpenie jednotlivých aktív nasledovne

$$l_i \leq x_i \leq u_i, \quad i = 1, \dots, n. \quad (4)$$

S formuláciou ohraničení, ktoré sú dané vzťahom (4), resp. jeho špeciálnymi prípadmi sa stretávame napríklad v [9], [14] a [4].

V [7] autori formulujú všeobecnejšie ohraničenie, ktoré definujú cez požiadavku, že nie viac ako zlomok γ celkového majetku bude investované v menej ako r aktívach. Nech $\sum_{i=1}^r x_{[i]}$ je suma r najväčších zložiek vektora x , potom toto ohraničenie vyjadrili ako

$$\sum_{i=1}^r x_{[i]} \leq \gamma,$$

kde ľavá strana predstavuje maximálny obnos držaný v akejkoľvek podmnožine r aktív.

Efektami diverzifikácie na rôzne portfóliá sa bližšie zaoberá [17].

1.2.2 Kvantitatívna požiadavka na pestrosť

Intervalové vymedzenie ohraničení (4) na zastúpenie jednotlivých aktív v portfóliu v Markowitzovom modeli má nepopierateľne svoje opodstatnenie pre investorov, ktorí si vopred určia do ktorých aktív chcú určite investovať isté obnosy a majú predstavu aj o požadovanej výške týchto investícií. Avšak nie je postačujúce v prípade, ak sa investor rozhodne vymedziť len požadovaný počet navzájom rôznych aktív v portfóliu, teda formulovať požiadavku na pestrosť portfólia kvantitatívne. Kvantitatívna požiadavka na pestrosť aktív v portfóliu by bola v praxi využiteľná napríklad v pasívnom investovaní, alebo indexových fondoch. Vo všeobecnosti by poskytovala investorovi nástroj na lepšiu reguláciu "trade-off"-u medzi výnosom a rizikom v portfóliu. Na druhej strane si uvedomujeme, že kvantitatívna požiadavka na pestrosť sama o sebe ešte nemusí byť nástrojom dostatočnej diverzifikácie portfólia. Lepšiu diverzifikáciu pri použití požiadavky na pestrosť by však investor mohol dosiahnuť vhodným výberom aktív vstupujúcich do optimalizácie, napríklad vylúčením aktív s vysokou koreláciou, prípadne spojením kvantitatívnej požiadavky a intervalových ohraničení na pestrosť (4). S kvantitatívnu požiadavku pestrosti portfólia sme sa však v literatúre nestretli, preto sme sa rozhodli ju v tejto práci sformulovať a začleniť do Markowitzovho modelu (1).

Ideálne by bolo, aby počet navzájom rôznych aktív v portfóliu bol z intervalu $\langle m_1, m_2 \rangle$. Taktiež by sme nechceli mať vo veľa aktívach investované príliš malé čiastky, preto by sme uvítali, aby v aspoň m_1 aktívach bolo investované aspoň ρ percent majetku. Inak povedané, chceli by sme, aby počet aktív, do ktorých investujeme nenulové čiastky, t.j. počet kladných zložiek vektora x bol z intervalu $\langle m_1, m_2 \rangle$, kde $m_1, m_2 \in N$,

$m_1 \leq m_2$ a $m_2 \leq n$ a zároveň aby v aspoň m_1 aktívach bolo investované aspoň ρ majetku. Matematicky je tieto požiadavky možné zapísať nasledovne

$$\begin{aligned} 1^T \text{sgn}(\text{sgn}(x - \rho \cdot 1) + 1) &\geq m_1 \\ 1^T \text{sgn}(x) &\leq m_2. \end{aligned} \tag{5}$$

Nevýhodou takejto formulácie je skutočnosť, že ohraničenia (5) nie sú konvexné, ba dokonca ani spojité. Na ich riešenie by sme potrebovali metaheuristické, alebo heuristické algoritmy, napríklad niektorý z evolučných algoritmov. Aby sme však mohli náš model riešiť klasickými optimalizačnými metódami, potrebujeme aby ohraničenia v modeli boli konvexné. Horné ohraničenie, ktoré zabezpečí, aby sme v portfóliu nemali priveľa navzájom rôznych aktív sa nám nepodarilo formulovať konvexne. Požiadavku na investovanie aspoň ρ percent majetku do minimálne m_1 aktív je vo všeobecnosti náročné vyjadriť vo forme konvexného ohraničenia, avšak podarilo sa nám konvexne vyjadriť požiadavku minimálneho množstva navzájom rôznych akcií v portfóliu. Ako si neskôr ukážeme, toto dolné ohraničenie na počet kladných zložiek vektora x pre Markowitzov model (1) sa dá vyjadriť konvexným ohraňčením ako $x^T x \leq \frac{1}{m_1}$.

1.3 Zovšeobecnený Markowitzov model

1.3.1 Formulácia modelu

Pri formulácii zovšeobecneného Markowitzovho modelu pre účely tejto práce vychádzame z Markowitzovho modelu (1), t.j. modelu maximalizácie výnosu. Okrem pridania kvantitatívnej požiadavky na pestrosť portfólia sme sa rozhodli zahrnúť do modelu aj transakčné náklady, kvôli bližšiemu priblíženiu sa realite oproti modelu (1).

Transakčné náklady sme formulovali ako podiel majetku, ktorý je použitý na uhrazenie nákladov spojených s kúpou aktív podľa [7]. Uvažujeme ich lineárne, kde náklady na každý nákup sú proporcionálne k obchodovanému podielu majetku. Mieru nákladov na nákup aktíva i označíme α_i a samozrejme predpokladáme že sú nezáporné pre všetky $i = 1, \dots, n$. Transakčné náklady sú potom rovné $\alpha^T x$. Rozpočtové ohraňčenie môžeme teda formulovať nasledovne

$$e^T x + \alpha^T x = 1,$$

kde e je vektor jednotiek. Rovnicu zjednodušíme zavedením označenia $d = e + \alpha$. Je

zrejme, že $d_i \geq 1$ pre všetky $i = 1, \dots, n$. Rozpočtové ohraničenie vstupujúce do modelu má potom tvar

$$d^T x = 1 \quad (6)$$

a nahradíme ním ohraničenie $e^T x = 1$ z modelu (1). V konečnom dôsledku teda prida-
nie transakčných nákladov má za následok zmenu škálovania premennej $x \geq 0$ oproti
pôvodnému modelu (1). x_i prestávajú byť percentuálnym vyjadrením časti aktuálneho
majetku vlozenej do i -teho aktíva, ale sú podielom i -teho aktíva na majetku pred vy-
konaním transakcie, keďže majetok po transakcii sa zníži o transakčné náklady. Inak
povedané investor vkladá do i -teho aktíva sumu danú násobkom investovaného ma-
jetku a x_i a ako transakčné náklady za nákup i -teho aktíva zaplatí $\alpha_i x_i$ násobok inves-
tovaného majetku. Ak hodnotu investície pre násobíme $e^T x$ získavame veľkosť majetku
po vykonaní transakcii.

Na tomto mieste by sme samozrejme mohli transformovať premenné x_i nasledovne
 $y_i = d_i x_i$, čo by však viedlo k zmene ohraničení, čím by sa stratila ich intuitívna
interpretácia. Pre lepšiu zrozumiteľnosť významu ohraničení preto ponecháme model
(1) v pôvodnom tvare, avšak s rozpočtovým ohraničením (6). Celkový podiel na ma-
jetku, ktorý investor použije na kúpu i -teho aktíva je potom súčtom x_i a transakčných
nákladov $\alpha_i x_i$.

Teraz prejdeme k formulácii požiadavky na pestrosť. Nakoľko horné ohraničenie na
počet kladných zložiek vektora x sa nám nepodarilo vyjadriť konvexne a požiadavku
na investovanie aspoň ρ percent majetku do aspoň m_1 aktív je vo všeobecnosti náročné
vyjadriť v tvare konvexného ohraničenia, zamerali sme sa na preformulovanie spodného
ohraničenia do konvexného tvaru. V ďalšom texte budeme teda pojmom požiadavka
na pestrosť portfólia označovať spodné ohraničenie na počet navzájom rôznych akcií v
portfóliu, resp. kladných zložiek vektora x , ktorú je možné vyjadriť v konvexnom tvare
nasledovne

$$x^T D^2 x \leq \frac{1}{m_1}, \quad kde \quad D^2 = \begin{pmatrix} d_1^2 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & d_n^2 \end{pmatrix}. \quad (7)$$

Matematická formulácia pomocných tvrdení, ako aj dôkazov je uvedená v nasledujúcej
časti.

Markowitzov model (1) po preškáľovaní definovanom vzťahom (6) a zahrnutí konvexnej kvantitatívnej podmienky na pestrosť portfólia (7) nadobudne tvar

$$\begin{aligned}
 \max \quad & \mu^T x \\
 d^T x = \quad & 1 \\
 x \geq \quad & \mathbf{0} \\
 Ax \leq \quad & b \\
 x^T C x \leq \quad & \sigma^2 \\
 x^T D^2 x \leq \quad & \frac{1}{m_1}.
 \end{aligned} \tag{8}$$

Nakoľko presné hodnoty vektora stredných hodnôt výnosov μ a kovariančnej matice výnosov C nie sú v praxi známe, nahradíme ich odhadmi, ktoré označíme c a Q . C je kovariančná matica, teda je symetrická a kladne semi-definitná. Ako jej odhad Q budeme uvažovať symetrickú, kladne definitnú maticu. Kladnú definitnosť budeme neskôr potrebovať kvôli vytvoreniu inverznej matice ku Q pre účely zaručenia konzistentnosti ohraňení. Úlohu prevedieme na minimalizačnú a dostávame finálny tvar modelu

$$\begin{aligned}
 \min \quad & -c^T x \\
 d^T x = \quad & 1 \\
 x \geq \quad & \mathbf{0} \\
 Ax \leq \quad & b \\
 x^T Q x \leq \quad & \sigma^2 \\
 x^T D^2 x \leq \quad & \frac{1}{m_1}.
 \end{aligned} \tag{9}$$

1.3.2 Teoretické zdôvodnenie

Budeme používať nasledovné označenia:

- $e = (1, 1, \dots, 1) \in R^n$
- $y \in R^n \Rightarrow e^T y = \sum_{j=1}^n y_j$
 $\Rightarrow y^T y = \sum_{j=1}^n y_j^2$
- S_n = jednotkový simplex, t.j. $S_n = \{y \in R^n \mid e^T y = 1, y \geq \mathbf{0}_n\}$

Lema 1. $\forall y \in S_n$ platí:

$$\frac{1}{n} \leq y^T y \leq 1. \quad (10)$$

Dôkaz

I [Ľavá nerovnosť v (10)].

Z Cauchyho nerovnosti $(e^T y)^2 \leq (e^T e)(y^T y)$ dostávame:

$$1 \leq n(y^T y),$$

pričom dolná hranica sa dosahuje pre vektor

$$\bar{y} = \frac{1}{n}e = \left[\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right].$$

II [Pravá nerovnosť v (10)].

Pre $y \in S_n$ máme

$$\begin{aligned} 0 &\leq y_j \leq 1 \quad (j = 1, 2, \dots, n) \\ \Rightarrow 0 &\leq y_j^2 \leq y_j \quad (j = 1, 2, \dots, n). \end{aligned}$$

Sčítaním všetkých dielčích nerovníc dostávame

$$\sum_{j=1}^n y_j^2 \leq \sum_{j=1}^n y_j = 1,$$

pričom horná hranica sa dosahuje pre vektory

$$\bar{y} = e_j = (0, 0, \dots, 0, 1, 0, \dots, 0)^T. \blacksquare$$

Dôsledok 1. Ak vektor $y \in S_n$ obsahuje aspoň $(n - m)$ núl, t.j. nanajvýš m kladných zložiek, tak

$$\frac{1}{m} \leq y^T y \leq 1. \quad (11)$$

Dôkaz

Vektor $y \in S_n$, ktorý obsahuje $(n - m)$ núl môžeme zapísať v blokovom tvare nasledovne

$$y = \begin{pmatrix} y^1 \\ y^2 \end{pmatrix} = \begin{pmatrix} y^1 \\ 0_{n-m} \end{pmatrix},$$

kde $y^1 \in R^m$, $y^2 \in R^{n-m}$. Zrejme $y^1 \in S_m$, a preto podľa Lemy 1 platí (11). $\blacksquare$

Obmenou Dôsledku 1 dostávame

Dôsledok 2. Ak pre vektor $y \in S_n$ platí

$$y^T y < \frac{1}{m}, \quad (12)$$

tak y obsahuje aspoň $(m + 1)$ kladných zložiek.

Poznamenávame, že ostrá nerovnosť (12) nie je vhodná pre použitie v modeloch a preto ju musíme upraviť na neostrú nerovnosť

$$y^T y \leq \frac{1}{m}.$$

Tým sme k pôvodnej množine vektorov $y \in S_n$ obsahujúcich aspoň $(m + 1)$ kladných zložiek pridali vektor $\bar{y} = \left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}, 0, 0, \dots, 0 \right)$, obsahujúci $(n - m)$ núl a m kladných zložiek. Teda dostávame nasledovné tvrdenie:

Veta 1. Ak pre vektor $y \in S_n$ platí

$$y^T y \leq \frac{1}{m}, \quad (13)$$

tak y obsahuje aspoň m kladných zložiek.

Nech $d > \mathbf{0}_n$ je kladný n -rozmerný vektor, ktorý určuje d -simplex $S_n^d = \{x \in R^n \mid d^T x = 1, x \geq \mathbf{0}_n\}$ a diagonálnu maticu $D = \text{diag}(d_i)_{1 \leq i \leq n}$.

V nerovnosti (13) aplikujeme substitúciu $y = Dx$. Potom zrejme $y \in S_n$ implikuje $x \in S_n^d$, pretože

$$1 = e^T y = e^T Dx = d^T x.$$

Teda dostávame tvrdenie

Veta 2. Ak pre vektor $x \in S_n^d$ platí

$$x^T D^2 x \leq \frac{1}{m}, \quad (14)$$

tak x obsahuje aspoň m kladných zložiek.

2 Programy použité na riešenie optimalizačného modelu

V tejto časti si predstavíme metódy vnútorného a vonkajšieho bodu, ako sme ich použili na riešenie optimalizačného modelu (9) a v krátkosti sa dotkneme aj evolučných algoritmov.

2.1 Metóda vnútorného bodu

Metóda vnútorného bodu, alebo interior point method (IPM) je v MATLAB-e dostupná cez funkciu `fmincon`. V niektorých prípadoch je pomocou zabudovaného IPM algoritmu v MATLAB-e možné nájsť optimum, aj keď optimalizácia nevychádza z prípustného riešenia, avšak je to skôr výnimka ako pravidlo, hlavne ak sa počiatočný bod nachádza ďaleko od optimálneho riešenia. Preto sme použili štandardnú techniku I. fázy používanú v lineárnom programovaní, aby sme získali prípustné riešenie úlohy (9).

Technika I. fázy spočíva v pridaní novej premennej ξ a predefinovaní ohraničení tak, aby boli splnené pre akýkoľvek štartovací vektor x^0 a zároveň pre $\xi = 0$ boli totožné s pôvodnými ohraničeniami problému. V tejto časti postupne predefinujeme všetky ohraničenia z modelu (9) a následne ho sformulujeme ako problém s vyššou dimenziou.

Ako prvé upravíme ohraničenie $d^T x = 1$. Najskôr pridáme do ohraničenia vektor x^0 tak, aby bola vždy splnená rovnosť, bez ohľadu na zložky vektora x^0 . Dosiahneme to nasledovnou formuláciou

$$d^T x - d^T x^0 + 1 = 1. \quad (15)$$

Keďže potrebujeme, aby vzťah (15) postupne konvergoval k pôvodnému ohraničeniu $d^T x = 1$, pridáme novú premennú $\xi \geq 0$, ktorá bude v prvom kroku $\xi = 1$ a v priebehu optimalizácie ju budeme znižovať smerom k nule. Nové ohraničenie bude mať potom tvar

$$d^T x + (1 - d^T x^0)\xi = 1, \quad (16)$$

v maticovom zápise

$$\begin{pmatrix} d_1 & \cdots & d_n & 1 - d^T x^0 \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix} = 1. \quad (17)$$

Rovnako upravíme aj ohraničenie $Ax \leq b$

$$Ax - Ax^0 + b \leq b$$

$$Ax + (b - Ax^0)\xi \leq b, \quad (18)$$

respektíve

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} & (b - Ax^0)_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & (b - Ax^0)_n \end{pmatrix} \leq b. \quad (19)$$

Pre kvadratické ohraničenia použijeme prenášobenie ξ^2 . Ohraničenie $x^T Qx \leq \sigma^2$ teda upravíme na

$$\begin{aligned} x^T Qx - (x^0)^T Qx^0 + \sigma^2 &\leq \sigma^2 \\ x^T Qx + (\sigma^2 - (x^0)^T Qx^0)\xi^2 &\leq \sigma^2, \end{aligned} \quad (20)$$

potom

$$\begin{pmatrix} x & \xi \end{pmatrix} \begin{pmatrix} q_{11} & \cdots & q_{1n} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ q_{n1} & \cdots & q_{nn} & 0 \\ 0 & \cdots & 0 & \sigma^2 - (x^0)^T Qx^0 \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix} \leq \sigma^2. \quad (21)$$

Rovnakým spôsobom upravíme aj ohraničenie $x^T D^2x \leq \frac{1}{m_1}$

$$\begin{aligned} x^T D^2x - (x^0)^T D^2x^0 + \frac{1}{m_1} &\leq \frac{1}{m_1} \\ x^T D^2x + \left(\frac{1}{m_1} - (x^0)^T D^2x^0 \right) \xi^2 &\leq \frac{1}{m_1}, \end{aligned} \quad (22)$$

a teda

$$\begin{pmatrix} x & \xi \end{pmatrix} \begin{pmatrix} d_1^2 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & d_n^2 & 0 \\ 0 & \cdots & 0 & \frac{1}{m_1} - (x^0)^T D^2x^0 \end{pmatrix} \leq \frac{1}{m_1}. \quad (23)$$

Zápis ohraničení (17), (19), (21) a (23) využijeme pri programovaní, avšak pre lepšiu čitateľnosť model preformulujeme použitím zápisu ohraničení (16),(18), (20) a (22)

takto

$$\begin{aligned}
 \min \quad & -c^T x + 10^{10} \xi \\
 & d^T x + (1 - d^T x^0) \xi = 1 \\
 & \begin{pmatrix} x \\ \xi \end{pmatrix} \geq \begin{pmatrix} 0_n \\ 0 \end{pmatrix} \\
 & Ax + (b - Ax^0) \xi \leq b \\
 & x^T Q x + (\sigma^2 - (x^0)^T Q x^0) \xi^2 \leq \sigma^2 \\
 & x^T D^2 x + \left(\frac{1}{m_1} - (x^0)^T D x^0 \right) \xi^2 \leq \frac{1}{m_1},
 \end{aligned} \tag{24}$$

kde koeficient 10^{10} pri premennej ξ v účelovej funkcii zabezpečí, že ξ pôjde rýchlo k nule.

2.2 Metóda vonkajšieho bodu

Metóda vonkajšieho bodu, alebo exterior point method (EPM) narozdiel od metódy vnútorného bodu nepotrebuje vychádzať z prípustného riešenia. Namiesto toho sa predchádzajúce riešenia používajú ako štartovacie body, čím sa optimalizácia postupne posúva bližšie k lokálnemu extrému. Stručne popíšeme úpravu úlohy $\text{Min}\{f(x) \mid g_i(x) \leq 0\}$ vo všeobecnosti pre metódu EPM, čo neskôr aplikujeme pri programovaní algoritmu EPM na úlohu (9).

Ako sme už spomenuli, algoritmus EPM nepotrebuje vychádzať z prípustného riešenia. Daňou za to je potreba väčšieho množstva behov optimalizačného algoritmu, resp. etáp optimalizácie. Pri ďalšej etape sa použije ako štartovací bod riešenie z predchádzajúcej etapy optimalizácie a týmto spôsobom algoritmus konverguje k lokálnemu extrému. Metóda vnútorného bodu funguje narozdiel od IPM na princípe zvyšovania penalizácie nesplnenia ohraňení. Penalta sa v každej etape navyšuje, čo núti algoritmus rýchlejšie konvergovať. V metóde IPM sa k hodnote účelovej funkcie pripočítavajú nesplnené ohraňenia ako suma logaritmov ich vzdialeností od nuly násobená penaltou, zatiaľ čo v EPM ako suma kvadrátov týchto vzdialeností násobená penaltou.

Vo všeobecnosti ak etapu označíme $l = 1, 2, \dots$ môžeme funkciu vstupujúcu do opti-

malizácie metódou EPM napísať ako

$$\psi_l(x) = f(x) + r_l \sum_i \max\{0, g_i(x)\}^2, \quad (25)$$

kde $f(x)$ je účelová funkcia, $g_i(x) \leq 0$ sú ohraničenia, r_l je hodnota penalty v l -tej etape, platí pre ňu $r_1 < r_2 < r_3 < \dots$ a je daná vzťahom

$$r_{l+1} = \rho r_l, \quad (26)$$

kde r_1 sa volí zväčša rovné jednej a pre ρ platí $\rho > 1$. My sme si zvolili $\rho = 3$.

2.3 Evolučné algoritmy

Evolučné algoritmy (EA) sú podľa [20] optimalizačné a learning algoritmy, založené na náhodnosti, ktoré napodobňujú evolučný princíp prežitia najsilnejších jedincov. V [1] autor zdôrazňuje, že EA sú stavané na úlohy na voľný extrém. Pre ich aplikovanie na úlohy s ohraničeniami je preto potrebný mechanizmus, ktorý zahrnie ohraničenia do ich účelovej funkcie. Najčastejšie sa na zahrnutie ohraničení využívajú penalizačné funkcie. Problémom penalizačných funkcií je závislosť veľkosti penalty od konkrétneho problému, keďže príliš nízka aj príliš vysoká hodnota penalty môže spôsobiť neúspešnosť algoritmu pri hľadaní optima. Nevýhodou EA je ich menšia presnosť, časová náročnosť ako aj problémy s optimalizáciou niektorých problémov vo vyšších dimenziách, ktoré vyplývajú z využitia náhodnosti. Problémom zlyhania metód založených na náhodnosti vo vyšších dimenziách sa zaoberá napríklad [15].

V úlohe (9) sme zo začiatku uvažovali požiadavku na pestrosť danú ohraničeniami (5), ktoré nie sú konvexné. Pokúsili sme sa túto úlohu riešiť jedným z EA, konkrétne evolučnou stratégiou a s ohraničeniami sme sa vysporiadali použitím penalizačnej funkcie stochastického rankingu, ako je to uvedené v [16]. Už pri $n = 10$ sme pozorovali vyššiu časovú náročnosť optimalizácie. Pre $n = 20$ už algoritmus napriek výraznému zvyšovaniu počtu cyklov a vyčíslení optimalizačnej funkcie (čo malo za následok markantné zvyšovanie výpočtového času) veľmi zriedkavo našiel optimálne riešenie. Naším cieľom je však nájsť optimálne riešenia aj pre $n \geq 100$, v primeranom výpočtovom čase, preto sme sa rozhodli upustiť od EA, preformulovať požiadavku na pestrosť do tvaru ako je zahrnutá v modeli (9) a ďalej v práci uvažovať len optimalizáciu pomocou metód IPM a EPM.

3 Numerický experiment

3.1 Metodológia

Na výpočet optimálneho riešenia úloh (9) sme použili, ako sme to už spomenuli v predchádzajúcej kapitole, metódy IPM a EPM. Najskôr sme však museli overiť ich spoľahlivosť a k tomu sme potrebovali sadu úloh (9) s vopred známymi optimálnymi riešeniami. Aby sme však mohli vytvoriť generátor týchto úloh potrebujeme aby spĺňali Kuhn-Tuckerove podmienky optimality a zároveň musíme zabezpečiť existenciu aspoň jedného prípustného riešenia, teda konzistentnosť ohraničení. Odvozeniu Kuhn-Tuckerových podmienok a podmienok konzistentnosti sa venujeme v nasledujúcej sekcii. Ďalej uvádzame algoritmus generátora úloh (9), ktorý spĺňa Kuhn-Tuckerove podmienky ako aj podmienky konzistentnosti, podľa ktorého sme vytvorili v prostredí MATLAB generátor úloh. Podrobne popisujeme kód vytvoreného generátora, ako aj prispôsobenie MATLAB-ovských algoritmov IPM a EPM pre úlohu (9). Výstup generátora s porovnaním riešení získaných metódami IPM a EPM ilustrujeme na príklade malých rozmerov. Následne porovnávame presnosť a dĺžku výpočtového času oboch metód pre rôzne veľkosti n a m_1 . Na záver testujeme vplyv požiadavky na pestrosť na výnos a riziko portfólia, pričom sme využili metódu, ktorá vyšla z porovnania lepšie.

3.2 Podmienky optimality

3.2.1 Kuhn-Tuckerove podmienky

Kuhn-Tuckerove podmienky sú nutnými podmienkami optimality riešenia prvého rádu. Pre zoznámenie sa s nimi odporúčame publikáciu [3].

Pre odvozenie Kuhn-Tuckerových podmienok potrebujeme Lagrangeovu funkciu úlohy (9)

$$L(x, u, v) = -c^T x + u_1^T A x - b + u_2 x^T Q x - \sigma^2 + u_3 x^T D^2 x - \frac{1}{m_1} + v(d^T x - 1),$$
$$x \geq \mathbf{0}_n, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \geq 0_{m+2}, \quad v \in R, \quad (27)$$

kde $u = (u_1, u_2, u_3)^T$ je Lagrangeov multiplikátor pre ohraničenia v tvare nerov-

x	u	v
$\frac{\delta L}{\delta x} \geq \mathbf{0}^T$	$\frac{\delta L}{\delta u} \leq \mathbf{0}^T$	$\frac{\delta L}{\delta v} = \mathbf{0}^T$
$\frac{\delta L}{\delta x} x = 0$	$\frac{\delta L}{\delta u} u = 0$	
$x \geq \mathbf{0}$	$u \geq \mathbf{0}$	

Tabuľka 1: Kuhn-Tuckerove podmienky

ností. Jeho prvých m zložiek, ktoré sme označili ako vektor u_1 prislúcha k ohraničeniam $Ax \leq b$, zložka u_2 ku kvadratickému ohraničeniu $x^T Qx \leq \sigma^2$ a zložka u_3 ku kvadratickému ohraničeniu $x^T D^2 x \leq \frac{1}{m_1}$. Vo všeobecnosti majú Kuhn-Tuckerove podmienky pre úlohu (9) tvar vid' tabuľka (1). Aplikujeme ich na Lagrangeovu funkciu našej úlohy (27). Pre x sú potom rovné

$$\begin{aligned} \frac{\delta L}{\delta x} &= -c + A^T u_1 + 2Qxu_2 + 2D^2 xu_3 + vd := V \geq \mathbf{0} \\ x_i V_i &= 0, \quad i = 1, \dots, n \\ x &\geq \mathbf{0}, \end{aligned} \tag{28}$$

pre u

$$\begin{aligned} \frac{\delta L}{\delta u} &= \left((Ax - b)^T, \quad x^T Qx - \sigma^2, \quad x^T D^2 x - \frac{1}{m_1} \right) := \eta \leq \mathbf{0} \\ u_i \eta_i &= 0, \quad i = 1, \dots, m + 2 \\ u &\geq \mathbf{0} \end{aligned} \tag{29}$$

a nakoniec pre v

$$\begin{aligned} \frac{\delta L}{\delta v} &= d^T x - 1 = 0 \\ d^T x &= 1. \end{aligned} \tag{30}$$

3.2.2 Konzistentnosť úlohy

Aby úloha (9) mala riešenie, musíme zabezpečiť konzistentnosť ohraničení, inak povedané prienik simplexu $S_n = \{x \in R^n \mid d^T x = 1, x \geq \mathbf{0}\}$ s elipsoidmi $x^T Qx \leq \sigma^2$ a

$x^T D^2 x \leq \frac{1}{m_1}$ musí byť neprázdna množina. Budeme teda hľadať minimálnu hodnotu σ^2 , ktorú označíme γ , $\sigma^2 \geq \gamma$ ako bod dotyku simplexu S_n a elipsoidu $x^T Q x$ a rovnako aj minimálnu hodnotu $\frac{1}{m_1}$, v ktorom sa elipsoid $x^T D x$ dotýka simplexu S_n . Postup podrobne demonštrujeme iba pre vyjadrenie γ a pre minimálnu hodnotu $\frac{1}{m_1}$ uvedieme len výsledok, nakoľko pre oba elipsoidy je postup takmer identický.

Hľadanie bodu dotyku začneme tým, že položíme do rovnosti derivácie nadroviny $d^T x$ a elipsoidu $x^T Q x$

$$\begin{aligned}\frac{d}{dx} d^T x &= \frac{d}{dx} x^T Q x \\ d &= 2Q^T x.\end{aligned}$$

x je potom rovné

$$x = \frac{1}{2} Q^{-1} d. \quad (31)$$

Existencia inverznej matice k matici Q vyplýva z jej kladnej definitnosti. Dosadíme takto vyjadrené x do parametrizovanej nadroviny $d^T x = a$

$$d^T \frac{1}{2} Q^{-1} d = a. \quad (32)$$

Predelením parametrom a získame $\hat{x}$ v mieste dotyku

$$\begin{aligned}d^T \frac{1}{2a} Q^{-1} d &= 1 \\ \hat{x} &= \frac{1}{2a} Q^{-1} d.\end{aligned}$$

Získané $\hat{x}$ dosadíme do elipsoidu $x^T Q x$ a využijeme symetrickosť matice $Q = Q^T$

$$\begin{aligned}\hat{x}^T Q \hat{x} &= \frac{1}{2a} d^T Q^{-1} Q \frac{1}{2a} Q^{-1} d \\ &= \frac{1}{4a^2} d^T Q^{-1} d.\end{aligned}$$

Nakoniec dosadíme parameter a zo vzťahu (32)

$$\begin{aligned}\hat{x}^T Q \hat{x} &= \frac{d^T Q^{-1} d}{4 \frac{1}{4} (d^T Q^{-1} d)^2} \\ &= \frac{1}{d^T Q^{-1} d} = \gamma\end{aligned}$$

a tým dostaneme spodné ohraňenie na veľkosť σ^2 vzhľadom na maticu Q a vektor d

$$\sigma^2 \geq \frac{1}{d^T Q^{-1} d}. \quad (33)$$

Rovnakým postupom sme získali aj ohraňenie na minimálnu hodnotu $\frac{1}{m_1}$

$$\begin{aligned}\hat{x}^T D^2 \hat{x} &= \frac{1}{d^T D^{-2} d} \\ &= \frac{1}{n} \leq \frac{1}{m_1},\end{aligned}$$

čo je prakticky vždy splnené.

3.3 Algoritmus generátora

Vstup: počet aktív n , počet ohraňení m , počet kladných zložiek vektora x m_1

Výpočet:

1. x : $n \times 1$ optimálne riešenie

d : $n \times 1$ pomocná vektorová premenná definovaná vzťahom $d = 1 + \alpha$

položme $x = 0_n$, $d = 0_n$

pokiaľ počet kladných zložiek $x < m_1$, tak pre i náhodný index $d_i = 1 + Ro(0, 0.2)$ a $x_i = \frac{1}{m_1 d_i}$

pre $i = 1 : n$, ak $d_i = 0$, tak $d_i = 1 + Ro(0, 0.2)$

2. Q : $n \times n$ diagonálna matica odhadu kovariancie

σ^2 : 1×1 horné ohraňenie na odhadnutú smerodajnú odchýlku výnosu portfólia

položme maticu Q : diagonála Q je z $Ro(0, 1)$, $pom1 : Qpom1 = d$, $\sigma_{min} = \frac{1}{d^T pom1}$

ak $\sigma^2 > \sigma_{min}$, tak $\sigma^2 = x^T Q x$, inak opakuj bod 2.

3. V : $n \times 1$ pomocná vektorová premenná definovaná vzťahom (28)

položme $V = 0_n$

pre $i = 1 : n$, ak $x_i = 0$, tak $V_i = Ro(0, 1)$

4. u : $u = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix}^T$, $(m + 2) \times 1$ vektor Lagrangeových multiplikátorov

ak $m \leq n$, tak u_{kladne} je náhodné celé číslo z $\langle 0, m \rangle$, inak u_{kladne} je náhodné celé číslo z $\langle 0, n \rangle$

$$\text{položme } u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \mathbf{0}_m \\ Ro(0, 1) \\ Ro(0, 1) \end{pmatrix}$$

pokiaľ počet kladných zložiek $u_1 \leq u_{kladne}$, tak pre náhodný index $i \in \{1, 2, \dots, m\}$ $u_i = Ro(0, 1)$

5. A : $m \times n$ matica ohraničení

zložky A sú náhodné celé čísla z $\langle -4, 4 \rangle$, ak $m \leq n$, tak hodnosť A je väčšia rovná m , inak hodnosť A je väčšia rovná n

6. b : $m \times 1$ vektor ohraničení

$$\text{položme } pom2 = Ax, b = 0_m$$

pre $i = 1 : m$, ak $u_i = 0$, tak $b_i = pom2_i + Ro(0, 5)$, inak $b_i = pom2_i$

7. v : 1×1 Lagrangeov multiplikátor

$$v = Ro(0, 1)$$

8. c : $n \times 1$ odhad stredných hodnôt výnosov portfólia

$$\text{položme } D = diag(d^2)$$

$$c = -V + A^T u_1 + 2Qxu_2 + 2Dxu_3 + vd$$

Výstup: $c, d, A, b, Q, \sigma^2, x, u, v$

3.4 Vytvorené programy

Na riešenie modelu (9) formulovaného v prvej kapitole sme použili programovací jazyk MATLAB a zabudované funkcie `fmincon` a `fminunc`. Ich popis je dostupný v dokumentácii MATLAB-u na stránkach [10] pre `fmincon` a [11] pre `fminunc`. Vytvoríme generátor úloh (9) podľa algoritmu popísaného v časti Algoritmus generátora a popíšeme prispôsobenie algoritmov IPM a EPM v prostredí MATLAB pre riešenie úloh (9). Jednotlivé programy budeme uvádzať a vysvetľovať po častiach, neprerušovane sú uvedené v prílohe A.

3.4.1 Generátor úloh

V tejto časti priblížime tvorbu generátora úloh, ktorý sme použili na otestovanie spoľahlivosti prispôsobených algoritmov IPM a EPM ako aj na demonštráciu zmien výnosu a rizika (daného veľkosťou odhadu smerodajnej odchýlky výnosu portfólia) vzhľadom na požiadavku pestrosti.

Pri vytváraní generátora úloh sme využili Kuhn-Tuckerové podmienky dané vzťahmi (28), (29) a (30) ako aj podmienku konzistentnosti (33). Generátor je naprogramovaný ako funkcia

```
function[c,d,A,b,Q,sigma2,x,u,v]=Generator(m,n,m1)
```

kde generované premenné sú pomenované podľa zavedenej symboliky, výnimkou sú len premenné `sigma2`, ktorá zodpovedá σ^2 a `m1`, ktorá zodpovedá m_1 . Počet ohraňčení m , vstupujúci do generátora, je nezáporné celé číslo a počet akcií n je kladné celé číslo. Počet kladných zložiek vektora x `m1`, je celé číslo z intervalu $m_1 \in \langle 1, n \rangle$, nakoľko pre $m_1 = 0$ nie je splnená podmienka (30). Podmienky na vstupujúce premenné m , n a m_1 sme do programu generátora nezahrnuli, nakoľko ich hodnoty budeme priamo zadávať.

Na začiatku sme vygenerovali optimálne riešenie x a pomocnú premennú d . Keďže požadujeme, aby ohraňčenie (7) bolo aktívne a zároveň musí platiť vzťah (6), dostávame nasledovný systém rovníc

$$\begin{aligned}d_1x_1 + d_2x_2 + \dots + d_nx_n &= 1 \\(d_1x_1)^2 + (d_2x_2)^2 + \dots + (d_nx_n)^2 &= \frac{1}{m_1},\end{aligned}$$

kde práve m_1 x_i je kladných. Intuitívne riešenie tohto systému je $d_ix_i = \frac{1}{m_1}$ pre $x_i > 0$. Ďalej chceme, aby miery transakčných nákladov α_i boli náhodné a dosahovali hodnotu nanajvyš 20%, preto zložky pomocnej premennej d budeme generovať z rozdelenia $Ro(1, 1.2)$. Vektor x sme teda generovali nasledovne

```
x=zeros(n,1);
d=zeros(n,1);
while length(x(x>0))<m1
    i=randi([1 n],1,1);
    d(i)=1+0.2*rand(1);
    x(i)=1/(m1*d(i));
```

```
end
```

Vo vyššie uvedenom cykle sa vygenerovali len zložky vektora d pre $x_i > 0$ a ostatné ostali nulové, avšak chceme aby všetky $d_i \in Ro(1, 1.2)$, preto

```
for i=1:n
    if (d(i)==0)
        d(i)=1+0.2*rand(1);
    end
end
```

Maticu Q sme generovali ako diagonálnu, pre zníženie výpočtovej náročnosti úlohy. Vo všeobecnosti je možné vygenerovať symetrickú a kladne definitnú maticu $Q = B^T B + I$ kde B je ľubovoľná matica. Požadujeme, aby ohraňenie $x^T Q x \leq \sigma^2$ bolo v optime aktívne a zároveň musí byť splnená podmienka konzistentnosti (33). Počítanie inverzných matíc v programovacích jazykoch je často menej presné, preto je rozumnejšie použiť systém rovníc. Ohraňenie (33) sme prenášobili pravou aj ľavou stranou, čo je možné, nakoľko obe strany sú kladné

$$\frac{1}{\sigma^2} \leq d^T Q^{-1} d.$$

Zaviedli sme pomocnú premennú $pom_3 = Q^{-1}d$ a riešili sústavu

$$Q pom_3 = d,$$

ktorej výsledok sme dosadili do ohraňenia (33), čím sme získali dolné ohraňenie na premennú σ^2 . Ak $\sigma^2 = x^T Q x$ nespĺňalo dolné ohraňenie, vygenerovali sme maticu Q znovu a postup sme zopakovali

```
rep=1;
while rep==1
    Q=diag(rand(n,1));
    pom1=Q\d;
    sigmamin=1/(d'*pom1);
    sigma2=x'*Q*x;
    if sigma2<sigmamin
        rep=1;
    else
```

```
        rep=0;
    end
end
```

Ďalej sme vygenerovali vektor V tak, aby boli splnené podmienky (28) vzťahujúce sa naň, z rovnomerného rozdelenia na intervale $\langle 0, 1 \rangle$ pre $V_i > 0$

```
V=zeros(n,1);
for i=1:n
    if (x(i)==0)
        V(i)=rand(1);
    end
end
```

Vektor Lagrangeových multiplikátorov u sme generovali tak, aby boli splnené podmienky (29) a zároveň aby počet aktívnych ohraničení v systéme $Ax \leq b$ nebol väčší ako n . Keďže obe kvadratické ohraničenia z úlohy (9) sú aktívne, podľa podmienok (29) sú posledné dve zložky vektora u , t.j. u_2 a u_3 , nenulové. Generovali sme ich z rozdelenia $Ro(0, 1)$. Vektor b sme ešte negenerovali, preto prvých m zložiek vektora u , t.j. vektor u_1 , môžeme generovať náhodne. Vrámcí nich sme si zvolili počet kladných zložiek vektora u_1 , t.j. počet aktívnych ohraničení v $Ax \leq b$ z intervalu $\langle 0, m \rangle$, ak $m \leq n$, inak z $\langle 0, n \rangle$ a generovali sme ich ako náhodné premenné taktiež z rozdelenia $Ro(0, 1)$

```
if m<=n
    u_kladne=randi([0 m],1,1);
else
    u_kladne=randi([0 n],1,1);
end
u=[zeros(m,1);rand(2,1)];
while length(u(u(1:m)>0))<u_kladne
    i=randi([1 m],1,1);
    u(i)=rand(1);
end
```

Maticu A sme generovali tak, aby jej hodnosť bola m ak $m \leq n$, inak n . Za jej zložky

sme stanovili náhodné celé čísla z intervalu $\langle -4, 4 \rangle$

```
A=randi([-4,4],m,n);  
if m<=n  
    while rank(A)<m  
        A=randi([-4,4],m,n);  
    end  
else  
    while rank(A)<n  
        A=randi([-4,4],m,n);  
    end  
end
```

Samozrejme interval $\langle -4, 4 \rangle$ možno zmeniť na všeobecný $\langle A_{min}, A_{max} \rangle$ a zložky generovať ako reálne čísla pomocou príkazu

```
A=Amin+(Amax-Amin)*rand(m,n);
```

Pri generovaní vektora b sme taktiež využili podmienky (29)

```
pom2=A*x;  
b=zeros(m,1);  
for i=1:m  
    if (u(i)==0)  
        b(i)=pom2(i)+5*rand(1);  
    else  
        b(i)=pom2(i);  
    end  
end
```

Lagrangeov multiplikátor v sme generovali z rovnomerného rozdelenia $Ro(0, 1)$

```
v=rand(1);
```

Na záver sme už len dopočítali vektor c , aby boli splnené Kuhn-Tuckerove podmienky a zakončili funkciu

```
D=diag(d.^2);  
c=-V+[A',2*Q*x,2*D*x]*u+v*d;  
end
```

3.4.2 Prispôsobenie MATLAB-ovského programu IPM

V tejto časti popíšeme riešenie modelu (24) pomocou algoritmu IPM zabudovanom vo funkcii `fmincon`. Využijeme maticové zápisy ohraničení (17), (19), (21) a (23) a zápis účelovej funkcie

$$\begin{pmatrix} -c^T & 10^{10} \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix}. \quad (34)$$

Budeme vychádzať z dokumentácie MATLAB-u, ktorá sa venuje kvadratickým ohraničeniam [13] pre funkciu `fmincon`. Keďže model (24) obsahuje nelineárne ohraničenia, museli sme pre ne vytvoriť pomocnú funkciu. Účelovú funkciu sme taktiež definovali samostatne. Poslednou pomocnou funkciou, ktorú sme využili je funkcia na výpočet Hessovej matice, vďaka ktorej sa zníži výpočtový čas. Najskôr si teda popíšeme tieto tri pomocné funkcie a následne sa pozrieme na ich implementáciu do algoritmu IPM.

Účelová funkcia

Použili sme zápis účelovej funkcie (34) a zároveň sme v programe definovali jej gradient pre urýchlenie výpočtov. Výsledná funkcia má tvar

```
function [y,grady] = obj_ipm(z,c_new)
y = -c_new'*z;
grady = -c_new;
end
```

Nelineárne ohraničenia

Nelineárne ohraničenia vstupujú do funkcie `fmincon` cez vstupný argument `nonlcon`. Je potrebné ich definovať ako funkciu pre nelineárne ohraničenia vo forme nerovností aj rovností, bez ohľadu na to či sú v modeli prítomné obe. Vo všeobecnosti definujeme nelineárne ohraničenia nasledovne

$$z^T H_i z + k_i^T z + p_i \leq 0. \quad (35)$$

Rovnako ako pri účelovej funkcii, aj u nelineárnych ohraničení sme definovali gradienty. Výsledná funkcia je potom

```
function [y,yeq,grady,gradyeq] = nonlin_constr(z,H,k,p)
j = length(H);
y = zeros(1,j);
for i = 1:j
```

```
y(i) = z'*Hi*z + ki'*z + pi;  
end  
yeq = [];  
  
grady = zeros(length(z),j);  
for i = 1:j  
    grady(:,i) = 2*Hi*z + ki;  
end  
gradyeq = [];  
end
```

Hessova matica

Poslednou pomocnou funkciou ktorú sme použili je funkcia na výpočet hessiánu. Hessián vstupujúci do funkcie `fmincon` je podľa dokumentácie [12] hessiánom Lagrangeovej funkcie, do ktorej však vstupuje len účelová funkcia f a nelineárne ohraničenia. Ak nelineárne ohraničenia v tvare nerovností označíme $cineq$ a v tvare rovností ceq , Lagrangeova funkcia na výpočet Hessovej matice má tvar

$$L = f + \sum_i \lambda_i cineq_i + \sum_j \lambda_j ceq_j,$$

a teda hessián je

$$\nabla^2 L = \nabla^2 f + \sum_i \lambda_i \nabla^2 cineq_i + \sum_j \lambda_j \nabla^2 ceq_j. \quad (36)$$

Druhá derivácia ľavej strany vzťahu (35) je matica H a Lagrangeove multiplikátory λ_i pre nelineárne ohraničenia z rovnice (36) sa automaticky ukladajú do premennej `lambda.ineqnonlin` (viď dokumentácia [10], časť Output arguments, lambda). Funkcia na výpočet Hessovej matice má potom tvar

```
function hess = hessian(lambda,H)  
n=length(H1);  
hess = zeros(n);  
j = length(H);  
for i = 1:j  
    hess = hess + lambda.ineqnonlin(i)*Hi;
```

```
end  
end
```

IPM

V tejto časti postupne popíšeme kód použitý na výpočet modelu (24) s ohraňeniami v tvare maticového zápisu a účelovou funkciou (34) metódou vnútorného bodu. Najskôr sme definovali štartovací vektor $\begin{pmatrix} x^0 \\ 1 \end{pmatrix}$, kde x^0 je náhodný vektor s n zložkami z $Ro(0, 1)$

```
z0=[rand(n,1);1];
```

Následne sme vymazali obsah premenných H , k a p a definovali ich pre všetky nelineárne ohraňenia vstupujúce do modelu podľa vzťahu (35). Ak je $m1 = 0$ toto ohraňenie je len jedno, inak sú dve a ich konkrétny tvar je daný vzťahmi (21) a (23)

```
clear H  
clear k  
clear p  
H1=([ [Q,zeros(n,1)];zeros(1,n+1)]+[ [zeros(n,n),zeros(n,1)];...  
[zeros(1,n),sigma2-z0'*[ [Q,zeros(n,1)];zeros(1,n+1)]*z0]]);  
k1=zeros(n+1,1);  
p1=-sigma2;  
if m1 ==0  
    H2=([ [D,zeros(n,1)];zeros(1,n+1)]+[ [zeros(n,n),zeros(n,1)];...  
[zeros(1,n),(1/m1)-z0'*[ [D,zeros(n,1)];zeros(1,n+1)]*z0]]);  
    k2=zeros(n+1,1);  
    p2=-(1/m1);  
end
```

Ďalej sme nastavili vlastnosti funkcie `fmincon`. Okrem nastavenia algoritmu na metódu vnútorného bodu sme znížili minimálne hodnoty tolerancii, zvýšili maximálny počet iterácii a vyčíslení funkcie, taktiež sme uviedli, že chceme v procese optimalizácie použiť vlastný hessián a vypočítané gradienty účelovej funkcie aj nelineárnych ohraňení a vyplí sme zobrazovanie výpisov funkcie

```
options = optimoptions(@fmincon,'Algorithm','interior-point',...  
    'GradObj','on','GradConstr','on','Hessian','user-supplied',...  
    'HessFcn',@(z,lambda)hessian(lambda,H),'TolFun',1e-14,'TolCon',...
```

```
1e-14,'MaxIter',100000,'MaxFunEvals',10000,'TolX',1e-14,...  
'Display','off');
```

Už nám ostával len definovať účelovú funkciu a funkciu nelineárnych ohraňení

```
fun = @(z)obj_ipm(z,[c;-10^10]);  
nonlconstr = (z)nonlin_constr(z,H,k,p);
```

a môžeme spustiť samotnú optimalizáciu, v ktorej využijeme lineárne ohraňenia definované vzťahmi (17) a (19)

```
[z,fval,eflag,output,lambda] = fmincon(fun,z0,[A,zeros(m,1)]+...  
[zeros(m,n),b-A*z0(1:(end-1))],b,[d',0]+[zeros(1,n),1-d'*...  
z0(1:(end-1))],1,zeros(n+1,1),[],nonlconstr,options);
```

Keďže výsledkom funkcie `fmincon` je optimum, závislé od zvoleného štartovacieho bodu, algoritmus sme spúšťali viackrát a za riešenie pokladáme to s najlepšou hodnotou účelovej funkcie pre hodnotu `exitflag`, v našom prípade označenú `eflag` rovnú jednej, čo znamená, že riešenie je minimom a zároveň spĺňa ohraňenia (minimálne na úrovni `TolCon`).

3.4.3 Prispôbenie MATLAB-ovského programu EPM

Na rozdiel od IPM, pri metóde vnútorného bodu sme použili funkciu `fminunc` a stačila nám jedna pomocná funkcia, v ktorej sme počítali počítat hodnotu $\psi_l(x)$ podľa vzťahu (25), ktorú sme použili ako účelovú funkciu pri optimalizácii funkciou `fminunc`.

Účelová funkcia $\psi_l(x)$

Funkciu $\psi_l(x)$ sme definovali staticky, t.j. rovnako pre každú etapu l . Hodnota penalty r_1 , ktorá je v programe označená `r`, do nej vstupuje adekvátne pre danú etapu

```
function [y] = obj_epm(z,c,d,A,b,Q,sigma2,m1,D,r)
```

Začali sme načítaním veľkostí premenných m a n

```
m=size(A,1);  
n=size(z,1);
```

Definovali sme jednotlivé ohraňenia $g_i(x) \leq 0$ pre úlohu (9), vrátane ohraňenia $x \geq 0$. Keďže ohraňenie $d^T x = 1$ je rovnosťou, previedli sme ho do tvaru nerovnosti $\text{abs}(d^T x - 1) \leq 0$

```
g=zeros(m+n+3,1);
```

```
g(1:m)=A*z-b;  
g(m+1:m+n)=-z;  
g(m+n+1)=abs(d'*z-1);  
g(m+n+2)=z'*Q*z-sigma2;  
g(m+n+3)=z'*D*z-(1/m1);
```

Vypočítali sme $\sum_i \max\{0, g_i(x)\}^2$ zo vzťahu (25)

```
ohr=0;  
for i=1:m+n+3  
    ohr=ohr+max(0,g(i))^2;  
end
```

Nakoniec sme dosadili účelovú funkciu $-c^T x$ z úlohy (9) a vypočítali hodnotu $\psi_l(x)$ podľa rovnice (25)

```
y = -c'*z+r*ohr;  
end
```

EPM

Samotný algoritmus EPM už potom nebolo ťažké aplikovať. Zvolili sme si náhodný štartovací vektor

```
z0=rand(n,1);
```

Upravili sme nastavenia funkcie `fminunc` na rovnakú toleranciu (kde to bolo možné) ako sme nastavili vo funkcii `fmincon`. Algoritmus sme zvolili Kvázi-Newtonov, keďže na použitie defaultného trust-region algoritmu, by sme potrebovali gradient účelovej funkcie, ktorý je pre funkciu $\psi_l(x)$ náročné vyjadriť. Z rovnakého dôvodu sme do funkcie nevkladali vlastný hessián. Nakoniec sme vypli zobrazovanie výpisov funkcie pre jednotlivé etapy

```
options = optimoptions(@fminunc,'Algorithm','quasi-newton','TolFun',...  
    1e-14,'MaxIter',100000,'MaxFunEvals',10000,'TolX',1e-14,...  
    'Display','off');
```

Nastavili sme počiatočné hodnoty $r_1 = 1$, $\rho = 3$ a počet etáp na 200

```
r=1;  
ro=3;  
maxetapa=200;
```

3.5 Ilustratívny príklad

Definovali sme cyklus optimalizácie pre zvolený počet etáp a v rámci neho funkciu $\psi_l(x)$ pre aktuálnu hodnotu r_l

```
for etapa=1:maxetapa
    fun = @(z)obj_epm(z,c,d,A,b,Q,sigma2,m1,D,r);
```

Spustili sme samotnú optimalizáciu, nastavili riešenie ako nový štartovací vektor a zmenili hodnotu r_l podľa rovnice (26)

```
[z,fval,eflag,output,lambda] = fminunc(fun,z0,options);
z0=z;
r=ro*r;
end
```

Rovnako ako pri IPM, aj EPM sme spúšťali viackrát a za riešenie sme pokladali to s najlepšou hodnotou účelovej funkcie pre hodnotu `eflag>0`, čo znamená, že riešenie spĺňa nastavené tolerancie.

3.5 Ilustratívny príklad

Pre lepšiu ilustráciu uvádzame príklad matíc a vektorov vygenerovaných generátorom úloh pre $n = 4$, $m = 3$ a $m_1 = 2$.

```
c =
    0.8076
   -0.4092
    0.7950
   -0.4313
```

```
d =
    1.1652
    1.0156
    1.1992
    1.0885
```

```
A =
     3     -3     -3     0
    -1     -2     3    -3
     4     -3     1     3
```

3.5 Ilustratívny príklad

```
b =  
    3.1468  
    2.5764  
    4.6997  
Q =  
    0.1067         0         0         0  
         0    0.9619         0         0  
         0         0    0.0046         0  
         0         0         0    0.7749  
sigma2 =  
    0.0204  
u =  
    0  
    0  
    0  
    0.3998  
    0.2599  
v =  
    0.4018
```

Pre túto úlohu je optimálnym riešením

```
x =  
    0.4291  
    0  
    0.4169  
    0
```

Vygenerované vektory a matice sme použili ako vstupy pre metódy IPM a EPM. Aby sme demonštrovali prínos nami zmenených nastavení funkcií **fmincon** a **fminunc**, spustili sme optimalizáciu aj pre pôvodné nastavenia pre zvolené algoritmy a bez vypisovania výstupov. Riešenia tohto príkladu získané pomocou IPM a EPM s nami zmenenými ako aj pôvodnými nastaveniami spolu s optimálnym riešením zachytáva tabuľka (2). Časy výpočtu aj s odchýlkami od optimálneho riešenia $\hat{x}$ a optimálnej hodnoty účelovej

Generátor	IPM	EPM	IPM - pôvodné nast.	EPM - pôvodné nast.
0,42912	0,42912	0,42912	0,42913	0,42922
0	1,85071E-10	8,71260E-11	2,41800E-04	1,75985E-06
0,41694	0,41694	0,41693	0,41672	0,41683
0	1,77480E-10	3,30049E-10	6,43480E-06	1,88320E-06

Tabuľka 2: Porovnanie optimálneho $\hat{x}$ s riešeniami IPM a EPM pre naše aj pôvodné nastavenia

Metóda	Výpočtový čas (v sekundách)	$\ \hat{x} - x^k\ _2$	$ f(\hat{x}) - f(x^k) $
IPM - naše nast.	0,400264	6,76600E-09	5,80629E-10
EPM - naše nast.	9,423279	7,23878E-07	1,63640E-08
IPM - pôv. nast.	0,594605	3,23364E-04	2,69036E-04
EPM - pôv. nast.	7,950172	1,42164E-04	3,22866E-07

Tabuľka 3: Porovnanie IPM a EPM pre naše aj pôvodné nastavenia na ilustratívnom príklade

funkcie $f(\hat{x})$ sú uvedené v tabuľke (3). Obe metódy sme spustili po 10 krát pre každé nastavenia.

Môžeme si všimnúť, že už na príklade malých rozmerov je vidno značný rozdiel výpočtových časov medzi metódami IPM a EPM. Keď porovnáme naše a pôvodné nastavenia pre metódu EPM vidíme, že výpočtový čas sa pre pôvodné nastavenia znížil (čo je dôsledkom menšej požadovanej presnosti riešenia) avšak riešenie je ďalej od optimálneho. Pri metóde IPM je vidno výrazný rozdiel v presnosti nájdeného riešenia pre naše a pôvodné nastavenia v prospech našich nastavení. Výpočtový čas sa pre pôvodné nastavenia paradoxne zvýšil, čo je dôsledok potreby počítať gradienty a Hessianovú maticu. Ďalšie numerické experimenty budeme teda vykonávať s našimi nastaveniami.

3.6 Porovnanie algoritmov

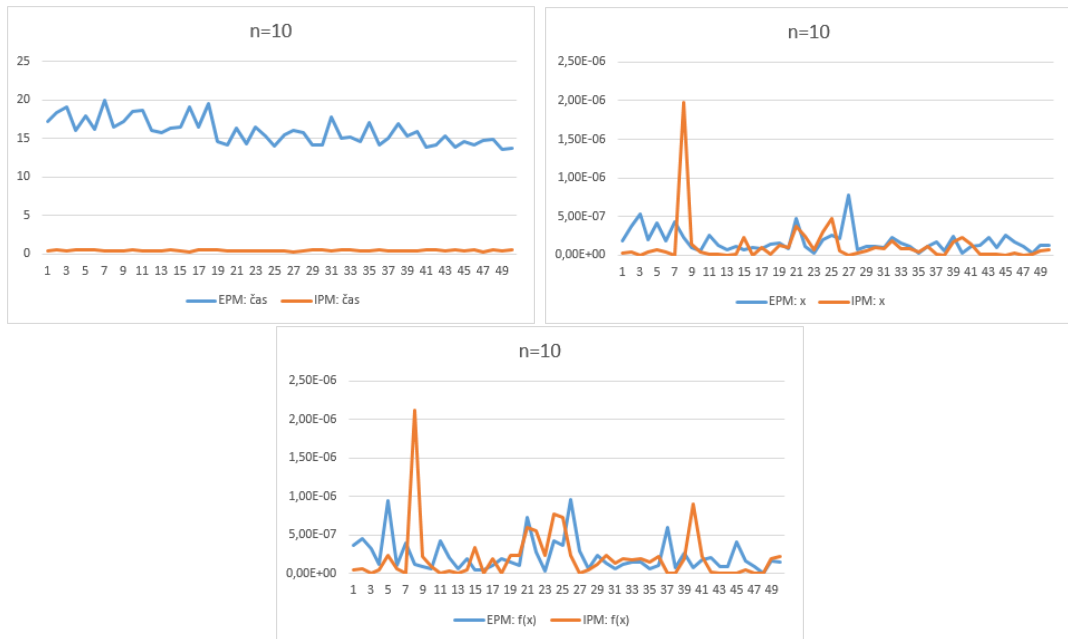
V tejto časti porovnáme algoritmy IPM a EPM pri riešení úloh (9) generovaných pomocou vytvoreného generátora vzhľadom na rôzny počet akcií n a rôzne požiadavky na

minimálnu pestrosť portfólia m_1 . Zameriame sa na čas chodu programu, vzdialenosť od optimálneho riešenia a funkčnej hodnoty v optimálnom riešení.

3.6.1 Porovnanie vzhľadom na rozmer úloh n

Porovnanie algoritmov vzhľadom na rôzny počet akcií n , z ktorých, resp. z ktorých časti vytvárame portfólio sme realizovali pre parametrizované hodnoty $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$. Hodnoty n sme uvažovali rovné 10, 20, 50, 100 a 200. Pre každé z nich sme vygenerovali 50 rôznych úloh (9) a pre každú úlohu sme oba algoritmy spúšťali po 10 krát. Vzdialenosť nájdeného riešenia x^k od optimálneho riešenia $\hat{x}$ úlohy sme vyjadrili cez Euklidovskú normu a vzdialenosti ich funkčných hodnôt cez absolútnu hodnotu.

Obrázok 1 zachytáva porovnanie metód pre $n = 10$ na jednotlivých úlohách. Presnosť nájdených riešení je pre oba algoritmy porovnateľná, čo možno vidieť aj v tabuľke 4. Čo sa výpočtového času týka, algoritmus IPM bol úlohy schopný vypočítať v priemere približne 36 krát rýchlejšie.



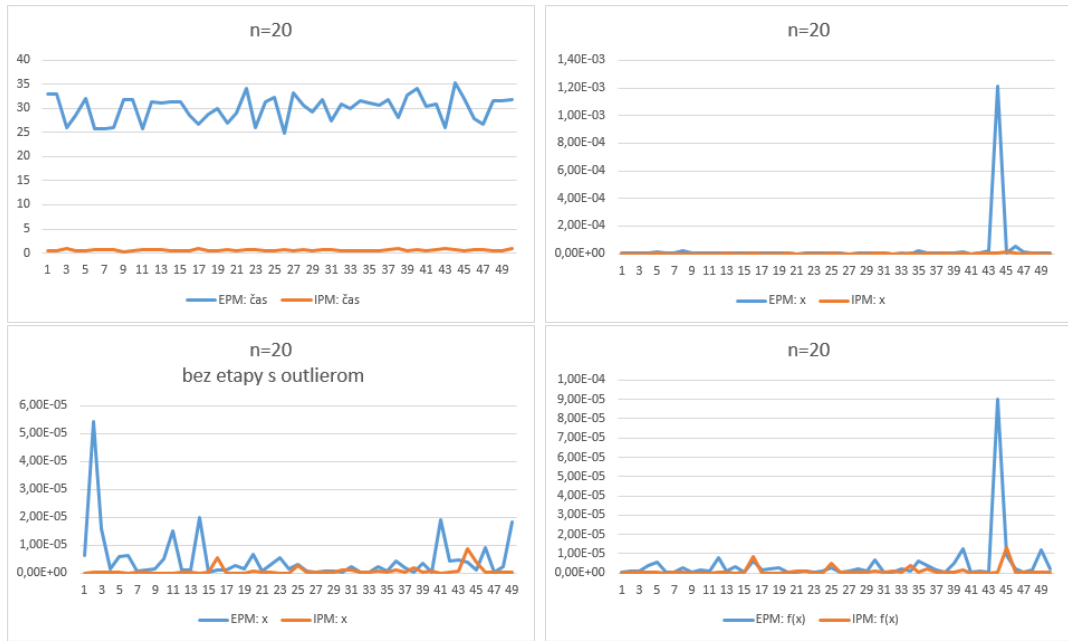
Obr. 1: Porovnanie IPM (oranžová) a EPM (modrá) pre $n = 10$, $m = 5$ a $m_1 = 1$. Hore vpravo: čas, hore vľavo: $\| \hat{x} - x^k \|_2$, dolu: $|f(\hat{x}) - f(x^k)|$.

Pre $n = 20$ už začína byť o čosi presnejší algoritmus IPM, pričom jeho náročnosť na výpočtový čas sa v priemere zvýšila len o necelých 0,2 sekundy. Na druhej strane

Metóda	Výpočtový čas [s]	$\ \hat{x} - x^k\ _2$	$ f(\hat{x}) - f(x^k) $
IPM	0,44148	1,1608E-07	2,0545E-07
EPM	15,91466	1,7390E-07	2,2349E-07

Tabuľka 4: Porovnanie priemerných hodnôt IPM a EPM pre $n = 10$

algoritmus EPM potrebuje takmer dvojnásobok výpočtového času. Na obrázku 2 vidno, že v 44-tej úlohe sa algoritmus EPM výraznejšie vzdialil od optimálneho $\hat{x}$, preto sme uviedli aj porovnanie bez tohto outlieru. Tak isto v tabuľke 5 uvádzame priemerné hodnoty aj bez úlohy 44 pre algoritmus EPM.



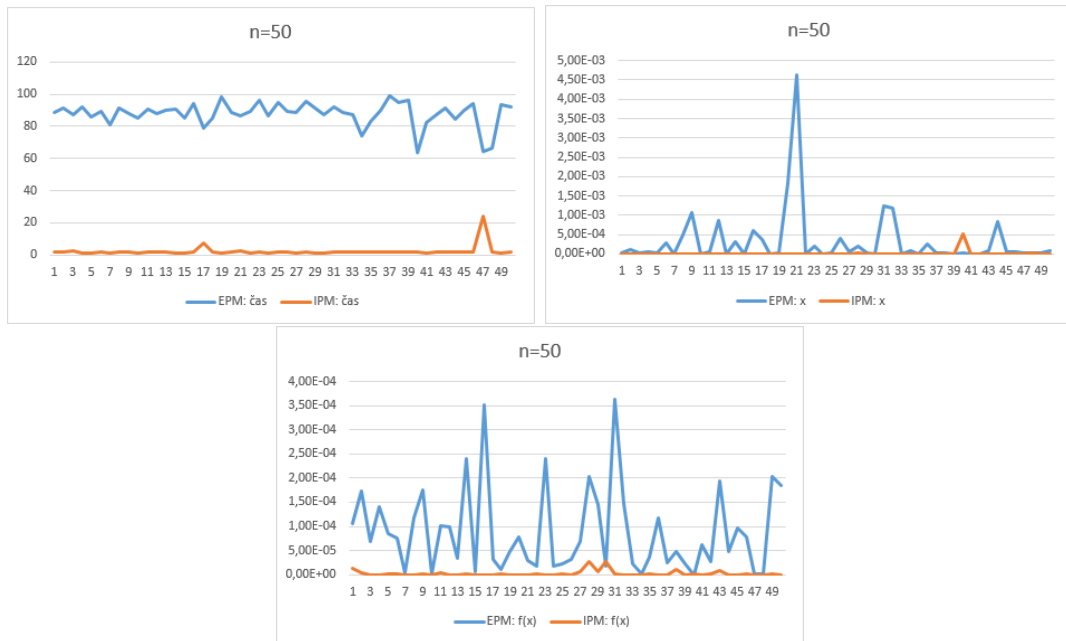
Obr. 2: Porovnanie IPM (oranžová) a EPM (modrá) pre $n = 20$, $m = 10$ a $m_1 = 2$. Hore vpravo: čas, hore vľavo: $\|\hat{x} - x^k\|_2$, dolu vpravo: $\|\hat{x} - x^k\|_2$ bez outlieru, dolu vľavo: $|f(\hat{x}) - f(x^k)|$.

V prípade $n = 50$ nastal výraznejší nárast v hodnote výpočtového času pre metódu IPM, o viac ako trojnásobok, čo dokonca prekonal aj pomerný nárast výpočtového času metódy EPM. Na obrázku 3 môžeme pozorovať stále väčšie zaostávanie metódy EPM čo sa presnosti riešenia aj hodnoty účelovej funkcie týka. V priemere je ale hodnota účelovej funkcie pre riešenie vypočítané metódou EPM stále relatívne blízko optimálnej hodnote účelovej funkcie, ako to zachytáva tabuľka 6. V porovnaní s prie-

Metóda	Výpočtový čas [s]	$\ \hat{x} - x^k \ _2$	$ (f(\hat{x}) - f(x^k)) $
IPM	0,62365	6,4891E-07	8,6569E-07
EPM	29,96634	2,9146E-05	4,1773E-06
EPM bez outliera	29,85678	4,9277E-06	2,4261E-06

Tabuľka 5: Porovnanie priemerných hodnôt IPM, EPM a EPM bez outliera pre $n = 20$

mernými presnosťami pre $n = 20$ došlo k ich poklesu u oboch metód. Výraznejší pokles priemernej presnosti nájdených riešení x^k u metódy IPM je spôsobený jedným outlierom, preto tabuľka 6 zachytáva priemernú presnosť aj po jeho odstránení.



Obr. 3: Porovnanie IPM (oranžová) a EPM (modrá) pre $n = 50$, $m = 25$ a $m_1 = 5$. Hore vpravo: čas, hore vľavo: $\| \hat{x} - x^k \|_2$, dolu: $| (f(\hat{x}) - f(x^k)) |$.

Väčší pomerný nárast výpočtového času u metódy IPM pretrváva aj pre $n = 100$ o viac ako 3,5 násobok, zatiaľ čo u EPM je to o menej ako 2,5 násobok. V absolútnych hodnotách je však v priemere IPM rýchlejšia o takmer 3,5 minúty, ako to vidno v tabuľke 7. Ďalej si môžeme všimnúť, že oproti $n = 50$ sa u metódy IPM zvýšila priemerná presnosť nájdeného x^k , čo však bolo spôsobené outlierom pri $n = 50$. Z obrázku 4 vidno výraznejší rozdiel medzi presnosťou algoritmov v prospech IPM. Môžeme si všimnúť, výrazný pokles presnosti u metódy EPM oproti prípadu $n = 50$. Z porovnania

Metóda	Výpočtový čas [s]	$\ \hat{x} - x^k \ _2$	$ (f(\hat{x}) - f(x^k)) $
IPM	2,09035	1,2077E-05	2,5225E-06
IPM bez outliera	2,10324	1,5815E-06	2,5553E-06
EPM	87,85302	3,1245E-04	8,8403E-05

Tabuľka 6: Porovnanie priemerných hodnôt IPM, IPM bez outliera a EPM pre $n = 50$

je zjavné, že metóda IPM začína byť podstatne lepšia ako metóda EPM pre zvyšujúce sa n .



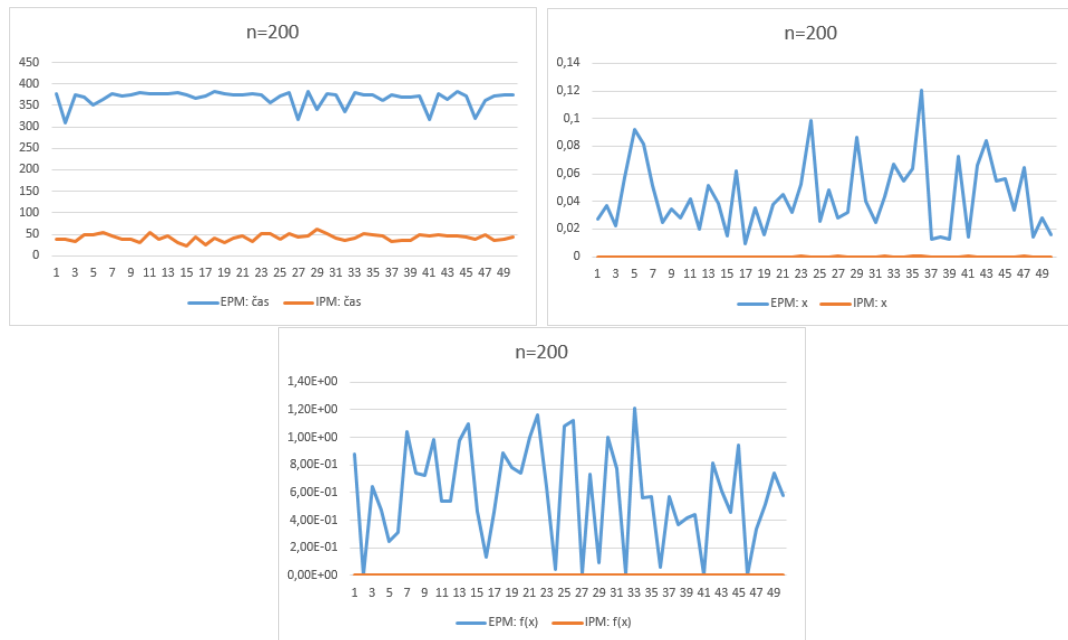
Obr. 4: Porovnanie IPM (oranžová) a EPM (modrá) pre $n = 100$, $m = 50$ a $m_1 = 10$. Hore vpravo: čas, hore vľavo: $\| \hat{x} - x^k \|_2$, dolu: $| (f(\hat{x}) - f(x^k)) |$.

Aj pre $n = 200$ je zjavné, že metóda IPM je výrazne lepšia, napriek tomu, že sa jej výpočtový čas viac ako späťnásobil. Priemerná presnosť oboch metód, ktorú zachytáva tabuľka 8 sa len o málo zhoršila oproti prípadu $n = 100$. Pre metódu IPM, narozdiel od EPM, to znamená stále vysokú presnosť.

Histogramy času, odchýlky nájdeného riešenia od optimálneho $\| \hat{x} - x^k \|_2$, ako aj účelových hodnôt $| (f(\hat{x}) - f(x^k)) |$ pre všetky použité n a obe metódy, na ktorých môžeme lepšie sledovať outlierov, ako aj rozmiestnenie hodnôt a ich zmeny so zvyšujúcim sa n sú dostupné v prílohe B.

Metóda	Výpočtový čas [s]	$\ \hat{x} - x^k\ _2$	$ f(\hat{x}) - f(x^k) $
IPM	2,09035	1,2077E-05	2,5225E-06
EPM	87,85302	3,1245E-04	8,8403E-05

Tabuľka 7: Porovnanie priemerných hodnôt IPM a EPM pre $n = 100$



Obr. 5: Porovnanie IPM (oranžová) a EPM (modrá) pre $n = 200$, $m = 100$ a $m_1 = 20$. Hore vpravo: čas, hore vľavo: $\|\hat{x} - x^k\|_2$, dolu: $|f(\hat{x}) - f(x^k)|$.

Z porovnania pre rôzne hodnoty n z časového hľadiska vychádza lepšie metóda IPM. Metóda EPM mala síce pre nízke hodnoty porovnateľnú, občas dokonca vyššiu presnosť nájdeného riešenia aj účelovej funkcie, avšak so zvyšujúcou sa hodnotou n čoraz viac zaostáva za metódou IPM.

Metóda	Výpočtový čas [s]	$\ \hat{x} - x^k\ _2$	$ f(\hat{x}) - f(x^k) $
IPM	42,26286	1,0731E-05	1,3212E-05
EPM	367,13660	4,3853E-02	5,9152E-01

Tabuľka 8: Porovnanie priemerných hodnôt IPM a EPM pre $n = 200$

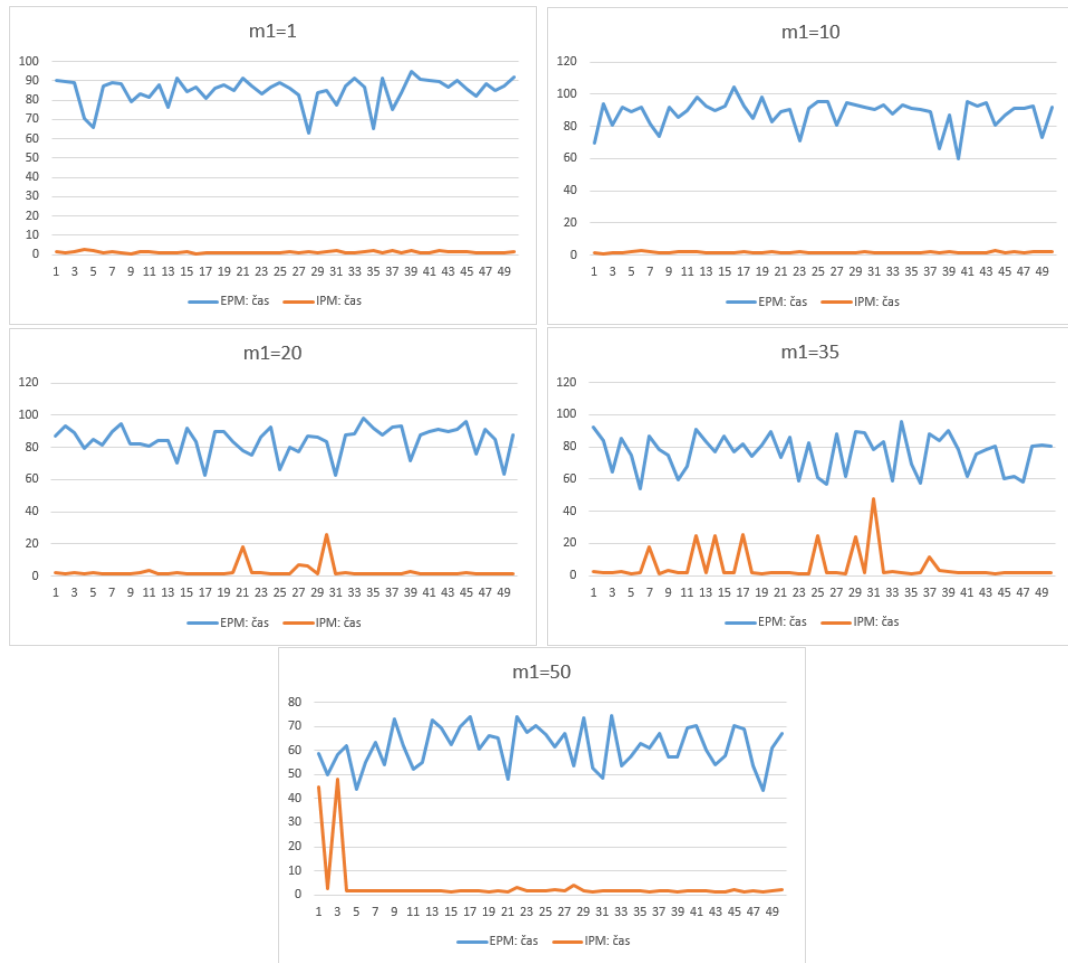
m_1	Metóda	Výpočtový čas [s]	$\ \hat{x} - x^k \ _2$	$ (f(\hat{x}) - f(x^k)) $
1	IPM	1,24766	2,4180E-07	3,8834E-07
	EPM	84,79721	2,6795E-06	3,7302E-06
5	IPM	2,09035	1,2077E-05	2,5225E-06
	EPM	87,85302	3,1245E-04	8,8403E-05
10	IPM	1,74655	6,1443E-06	7,9783E-07
	EPM	88,07031	2,0090E-03	2,5758E-04
20	IPM	2,76529	1,9689E-05	4,1518E-07
	EPM	84,35367	3,1664E-03	2,8242E-04
35	IPM	5,49774	1,5245E-05	5,4093E-08
	EPM	76,25783	1,2587E-03	1,8610E-04
50	IPM	3,49793	3,2322E-08	4,5975E-14
	EPM	61,56393	2,4154E-03	5,0660E-04

Tabuľka 9: Porovnanie priemerných hodnôt IPM a EPM pre rôzne hodnoty m_1

3.6.2 Porovnanie vzhľadom na požiadavku pestrosti portfólia m_1

Porovnanie v predchádzajúcej časti naznačilo, že metóda IPM by mala byť vhodnejšou na riešenie úloh (9). V tejto časti sa preto viac zameriame na ňu. Pre účely porovnania metód vzhľadom na rôzne hodnoty m_1 sme si zvolili $n = 50$, $m = 25$ a $m_1 = 1, 10, 20, 35, 50$, keďže pre $m_1 = 5$ sme porovnanie realizovali v predchádzajúcej časti. Rovnako sme generovali 50 príkladov pre každú hodnotu m_1 a oba algoritmy sme spúšťali po 10 krát každý. Porovnanie časov behu oboch programov, ako aj vzdialenosti nimi nájdeného riešenia x^k od optimálneho $\hat{x}$ a účelových funkcií pre jednotlivé hodnoty m_1 spolu s hodnotami pre $m_1 = 5$ z predchádzajúcej časti zachytáva tabuľka 9. Môžeme si všimnúť zaujímavý vývoj výpočtového času metódy EPM, ktorý sa zo začiatku s vzrastajúcim m_1 zvyšoval, no od hodnoty $m_1 = 10$ sa začal znižovať a pre $m_1 = 35$ a 50 je dokonca nižší ako pre $m_1 = 1$. Výpočtový čas pri metóde IPM pre hodnoty $m_1 = 20, 35$ a 50 je výrazne ovplyvnený prítomnosťou zopár outlierov, ako je to vidno na obrázku 6. Čo sa priemernej presnosti riešenia a odchýlky od optimálnej účelovej funkcie týka, z

3.6 Porovnanie algoritmov

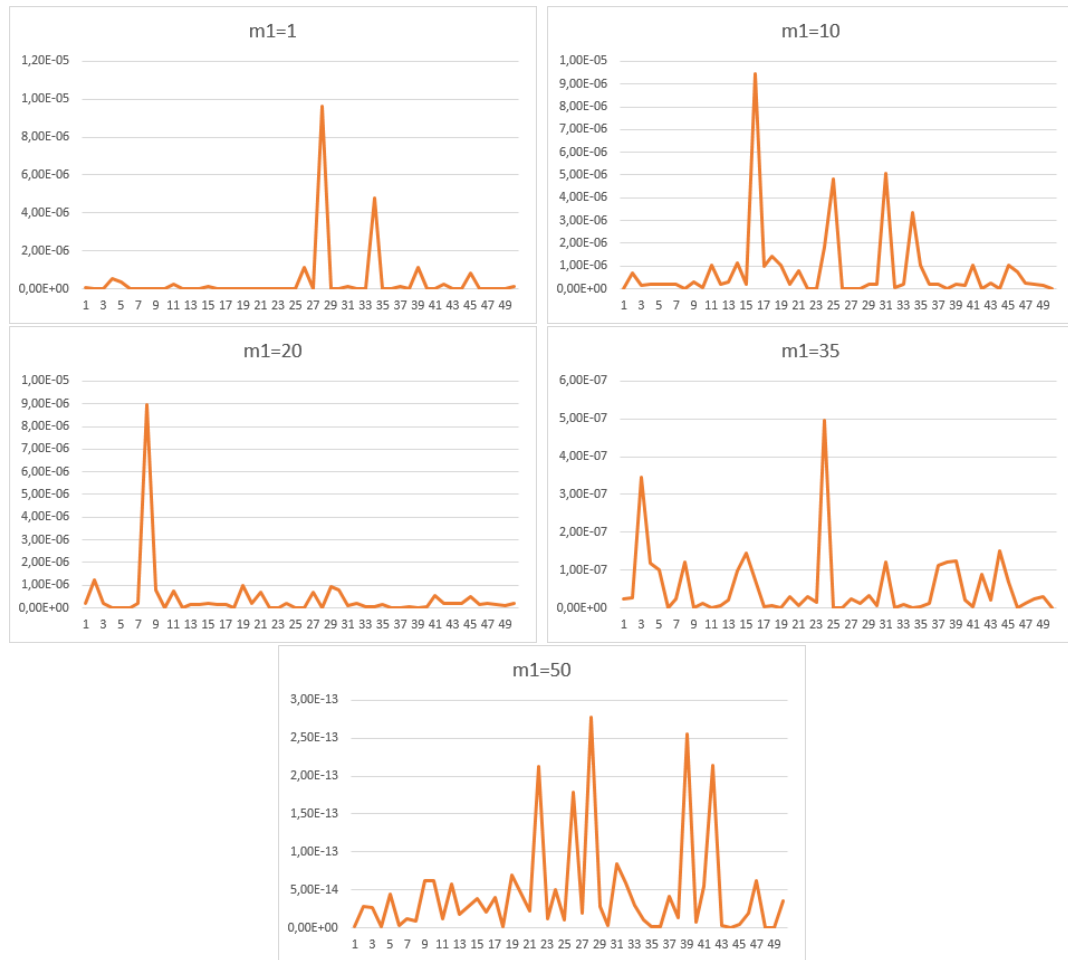


Obr. 6: Porovnanie výpočtového času IPM (oranžová) a EPM (modrá) pre rôzne hodnoty m_1 , $n = 50$ a $m = 25$.

tabuľky 9 vidíme, že metóda IPM je výrazne lepšou ako EPM od $m_1 = 10$. Pre $m_1 = 50$ dosahuje metóda IPM najvyššiu presnosť. Odchýlky od optimálnej hodnoty účelovej funkcie pri použití metódy IPM pre jednotlivé m_1 zachytáva obrázok 7. Vidno na ňom, že odchýlka od účelovej funkcie je vždy menšia ako 10^5 pre všetky hodnoty m_1 .

Z oboch porovnaní metód IPM a EPM, vzhľadom na rôzne n a m_1 vidíme, že pre ich nízke hodnoty sú obe metódy porovnateľné čo sa presnosti týka, metóda EPM je dokonca v zopár prípadoch výrazne presnejšou. So vzrastajúcim n a m_1 však presnosť metódy EPM klesá, zatiaľ čo metóda IPM dosahuje porovnateľnú až väčšiu presnosť. Čo sa veľkosti potrebného výpočtového času týka, metóda IPM je niekoľkonásobne rýchlejšia vo všetkých prípadoch. Z porovnaní je teda zjavné, že IPM je vhodnejšou metódou pre potreby zovšeobecneného Markowitzovho modelu (9). Ďalšie numerické

3.7 Vplyv požiadavky na pestrosť portfólia na výnos a riziko



Obr. 7: Odchýlka od optimálnej hodnoty účelovej funkcie $|f(\hat{x}) - f(x^k)|$ metódy IPM pre rôzne hodnoty m_1 , $n = 50$ a $m = 25$.

experimenty preto budeme uskutočňovať použitím metódy IPM.

3.7 Vplyv požiadavky na pestrosť portfólia na výnos a riziko

V tejto časti sme otestovali vplyv zmien požiadavky na pestrosť, ktorú sme navrhli a začlenili do prostredia Markowitzovho modelu, na strednú hodnotu výnosu portfólia (ďalej len výnos) a smerodajnú odchýlku výnosu portfólia (ďalej len riziko). Na overenie vplyvu požiadavky na pestrosť na výnos a riziko sme pomocou generátora vytvorili úlohy (9) so zadaným m_1 , ktoré sme následne zvyšovali a sledovali sme výnos a riziko novovzniknutých portfólií spĺňajúcich prísnejšiu podmienku na pestrosť. Nakoľko ohraňenie $d^T x = 1$ v spojení s kvadratickým ohraňením $x^T D x \leq \frac{1}{m_1}$ výrazne znižujú množinu prípustných riešení úlohy (9), pre $m_1 = n$ dokonca na jedno-

$m \backslash n$	0	5	10
50	50	28	-
100	50	27	20

Tabuľka 10: Počet úloh vyriešených až do $m_1 = (n - 1)$ pre rôzne n a m

dovú, veľké množstvo ohraničení $Ax \leq b$ môže rýchlejšie viesť k zmenšeniu množiny prípustných riešení na jednobodovú, až prázdnu. Metóda vnútorného bodu potom nie je schopná nájsť riešenie takejto úlohy. Preto sme zvolili v experimentoch nižšie hodnoty m a optimalizáciu sme zastavovali ak metóda IPM nenašla optimum.

Experimenty sme realizovali pre $n = 50$, $m_1 = 5$, $m = 0$ a 5 a $n = 100$, $m_1 = 10$, $m = 0$, 5 a 10 . Pre každú kombináciu sme vygenerovali sadu 50 úloh (9). Hodnotu m_1 sme zvyšovali v každej vygenerovanej úlohe po jednej až do $(n - 1)$. Algoritmus IPM sme spúšťali pokým nenašiel optimum, najviac však 20 krát pre každé m_1 . V prvom behu sa pre urýchlenie výpočtov využívalo riešenie $\hat{x}$ z predchádzajúcej etapy ako štartovací vektor a v ďalších behoch, ak algoritmus IPM nenašiel optimum, náhodný vektor. V prípade, ak pre dané m_1 algoritmus IPM ani po 20 behoch nenašiel optimum, optimalizáciu sme zastavovali. Počet iterácií v nastaveniach algoritmu sme znížili na 200, o približne 150 viac ako priemerný počet iterácií v prípade nájdania optima, čo výrazne urýchlilo výpočty.

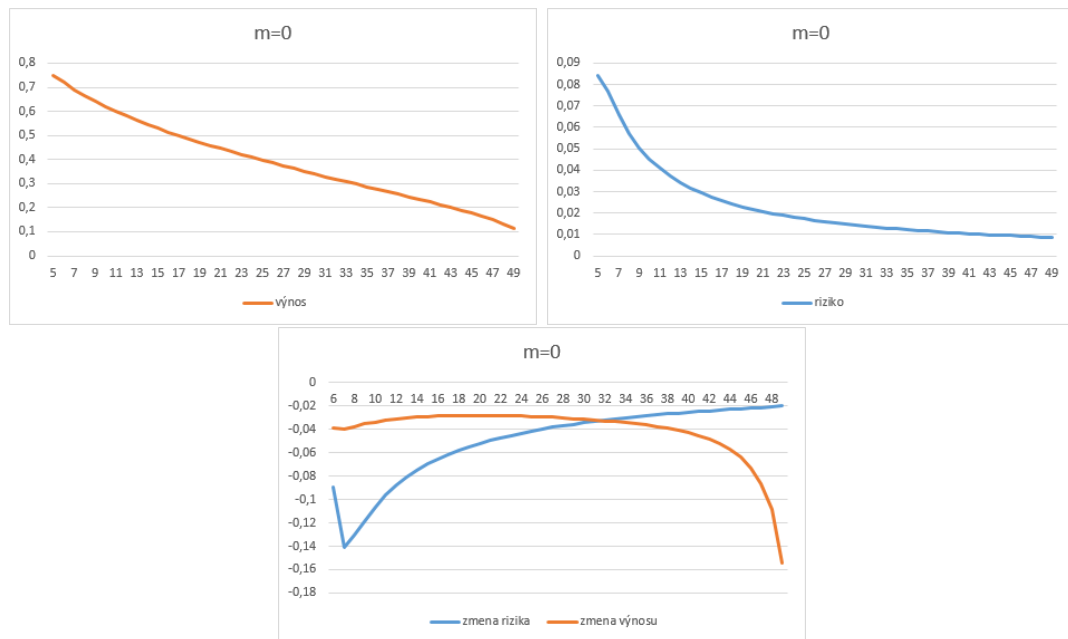
Z 50 úloh vygenerovaných pre každú uvedenú kombináciu n , m_1 a m sme vybrali tie, pre ktoré algoritmus IPM našiel optimálne riešenia až po $m_1 = (n - 1)$. Ich konkrétne počty sú uvedené v tabuľke 14. Môžeme si všimnúť pokles počtu úloh, vyriešených až do konca, so vzrastajúcim m . Ďalej sme pracovali len s týmito.

Pre všetky zvolené kombinácie n , m_1 a m sme vypočítali priemerné riziko a výnos ako aj ich relatívne zmeny z úloh, ktorých počty sú uvedené v tabuľke 14, pre každú hodnotu m_1 od počiatočného až po $(n - 1)$. Tabuľka 11 obsahuje uvedené údaje pre $n = 50$, avšak len pre každé piate m_1 . Obdobná tabuľka pre $n = 100$ je uvedená v prílohe C. Môžeme si všimnúť, že k najvýraznejšiemu poklesu rizika dochádza medzi $m_1 = 5$ a 10 . V prípade výnosu je opačný trend. Vývoj priemerných hodnôt rizika a výnosu, ako aj ich zmien pre všetky hodnoty m_1 prezentujeme na nasledujúcich grafoch.

3.7 Vplyv požiadavky na pestrosť portfólia na výnos a riziko

$m = 0$					$m = 5$			
m_1	Riziko	Δ rizika	Výnos	Δ výnosu	Riziko	Δ rizika	Výnos	Δ výnosu
5	0,0842		0,7479		0,0826		0,9271	
10	0,0451	-0,1066	0,6187	-0,0338	0,0458	-0,1079	0,8003	-0,0265
15	0,0294	-0,0697	0,5295	-0,0292	0,0296	-0,0709	0,7096	-0,0226
25	0,0172	-0,0417	0,3968	-0,0290	0,0173	-0,0418	0,5672	-0,0240
30	0,0143	-0,0344	0,3400	-0,0315	0,0143	-0,0344	0,4972	-0,0267
35	0,0122	-0,0292	0,2865	-0,0354	0,0122	-0,0297	0,4305	-0,0298
40	0,0106	-0,0255	0,2343	-0,0429	0,0106	-0,0258	0,3611	-0,0379
45	0,0094	-0,0226	0,1779	-0,0640	0,0094	-0,0231	0,2841	-0,0557

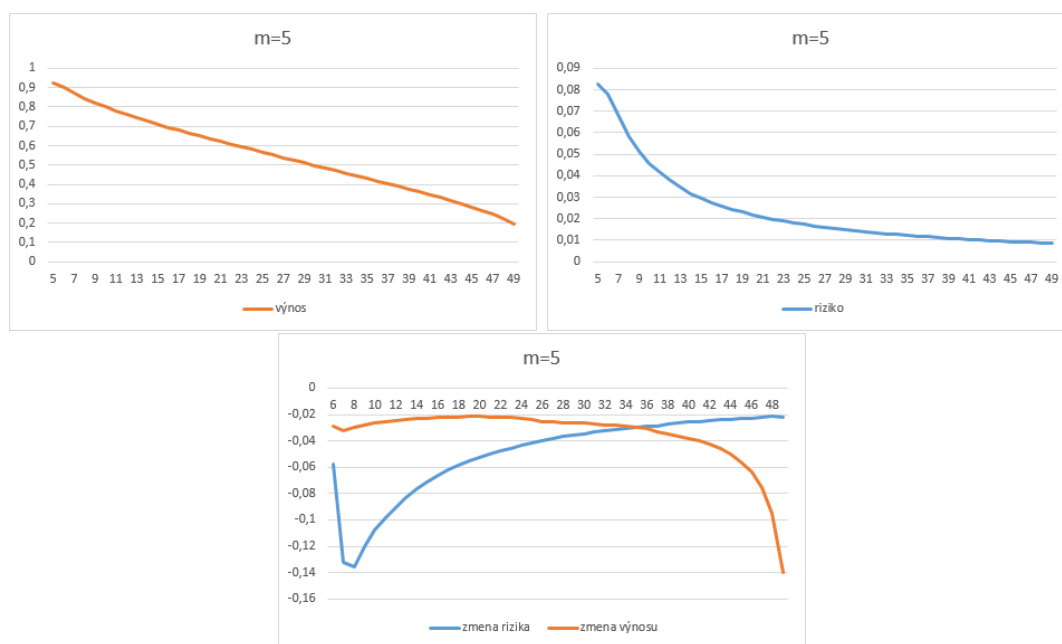
Tabuľka 11: Porovnanie priemerného výnosu, rizika a ich relatívnych zmien pri zmene m_1 pre $n = 50$ $m = 0$ a 5



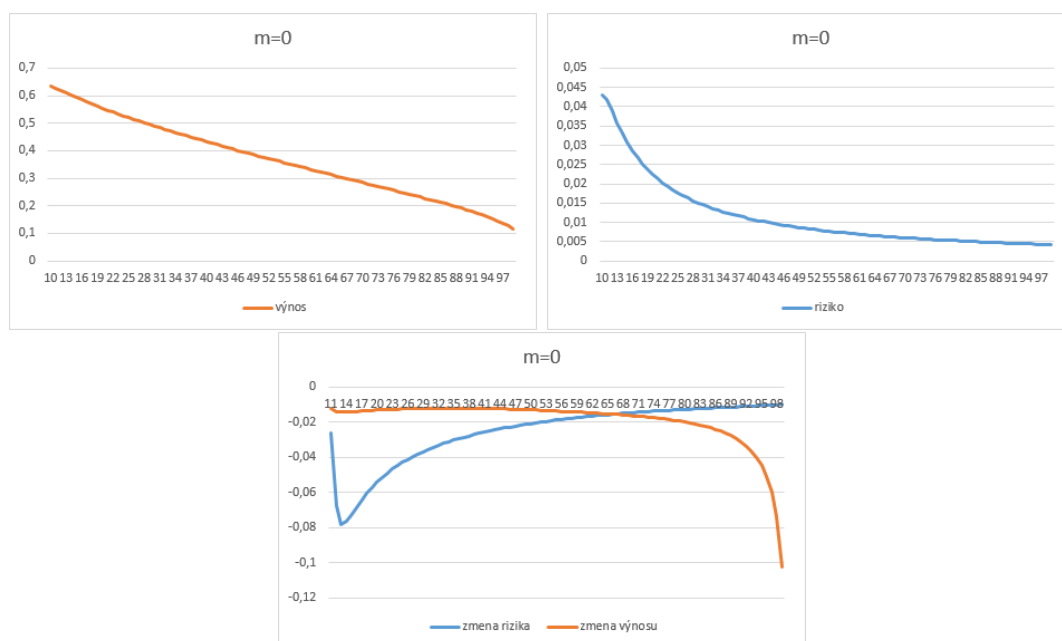
Obr. 8: Priemerný výnos, riziko a ich relatívne zmeny pre $n = 50$ a $m = 0$.

Môžeme si všimnúť, že krivky výnosov a rizík, ako aj ich zmien majú veľmi podobný tvar pre všetky zvolené kombinácie n , m_1 a m . Z obrázkov 8, 9, 10, 11 a 12 vidíme, že pre väčšie hodnoty n a m sa relatívne zmeny výnosov pre menšie m_1 čoraz viac blížia k nule a ich neskorší pokles je čoraz strmší. Ďalej si môžeme na obrázkoch všimnúť, že vo

3.7 Vplyv požiadavky na pestrosť portfólia na výnos a riziko



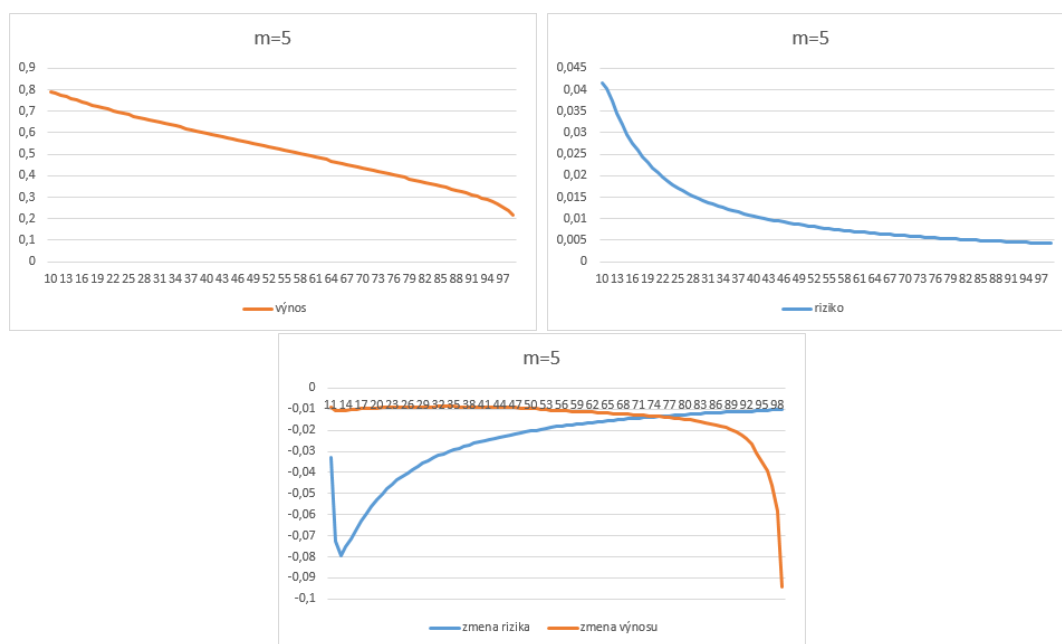
Obr. 9: Priemerný výnos, riziko a ich relatívne zmeny pre $n = 50$ a $m = 5$.



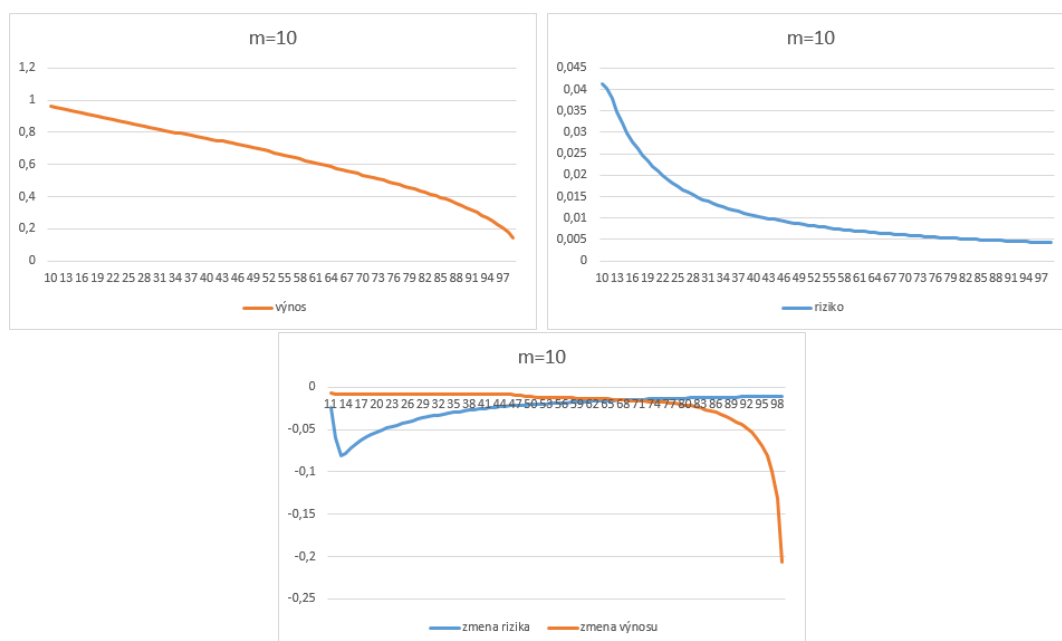
Obr. 10: Priemerný výnos, riziko a ich relatívne zmeny pre $n = 100$ a $m = 0$.

všetkých prípadoch je pokles rizika zo začiatku výrazný, avšak rýchlo sa spomaľuje. V prípade výnosu je zo začiatku miernejší pokles, ktorý sa ďalej relatívne stabilizuje, no na konci je už opäť prudší. Krivky výnosov a rizík sa vo všetkých prípadoch správajú očakávane, teda čím prísnejšia požiadavka na pestrosť portfólia, tým nižšie riziko, ale aj

3.7 Vplyv požiadavky na pestrosť portfólia na výnos a riziko



Obr. 11: Priemerný výnos, riziko a ich relatívne zmeny pre $n = 100$ a $m = 5$.



Obr. 12: Priemerný výnos, riziko a ich relatívne zmeny pre $n = 100$ a $m = 10$.

nižší zisk, pričom pridávanie akcií má pri ich menšom počte väčší vplyv na znižovanie rizika. Ďalej je teda už len na investorovi zvoliť si podľa svojho uváženia prijateľný "trade-off" medzi poklesom rizika a poklesom očakávanej hodnoty výnosu a tým aj finálne portfólio.

Záver

Za cieľ práce sme si zvolili formulovanie kvantitatívnej požiadavky na pestrosť portfólia, ktorú sme vyjadrili konvexným kvadratickým ohraničením (7) v Markowitzovom modeli s lineárnymi transakčnými nákladmi. Ďalej bolo našim cieľom navrhnutie metódy riešenia modelu, kde z porovnávaných metód IPM a EPM vyšla vhodnejšie metóda IPM, ako aj sledovanie vplyvu zmeny ohraničenia na pestrosť portfólia na strednú hodnotu očakávaného výnosu portfólia a riziko, dané veľkosťou smerodajnej odchýlky výnosu portfólia, v numerických experimentoch, ktorých výstupy sa nachádzajú v poslednej časti tretej kapitoly.

V práci sme sa venovali formulácii spomínanej kvantitatívnej požiadavky na pestrosť portfólia v Markowitzovom modeli. Uviedli sme tri používané formulácie Markowitzovho modelu optimalizácie portfólia, z ktorých sme ďalej v práci používali model (1) maximalizácie strednej hodnoty očakávaného výnosu portfólia. Stručne sme sa venovali diverzifikácii portfólia v literatúre, popísali sme kvantitatívnu požiadavku na pestrosť portfólia a navrhli sme ohraničenia na pestrosť (5), ktoré však neboli konvexné. Formulovali sme Markowitzov model (9) zovšeobecnený o lineárne transakčné náklady a kvantitatívnu požiadavku na minimálnu pestrosť portfólia, ktorú sme sformulovali v tvare konvexného ohraničenia (7), čo sme ďalej teoreticky zdôvodnili. Uviedli sme úpravu Markowitzovho modelu (9) štandardnou metódou I. fázy používanou v lineárnom programovaní kvôli ľahšiemu stanoveniu prípustného riešenia pre metódu IPM. Popísali sme optimalizačnú funkciu vstupujúcu do metódy EPM, ako aj určenie hodnoty penalty. V krátkosti sme spomenuli evolučné algoritmy, ktoré sme pre väčšie rozmery n neúspešne aplikovali na model (1) rozšírený o transakčné náklady a ohraničenia na pestrosť (5), čo však v tejto práci konkrétnejšie nepopisujeme. Pomocou Kuhn-Tuckerových podmienok optimality ako aj ohraničení na konzistentnosť sme navrhli a vytvorili generátor úloh (9). Popísali sme prispôsobenie programov IPM a EPM v programovacom jazyku MATLAB na zovšeobecnený Markowitzov model (9). Výstupy generátora, ako aj optimalizáciu pomocou prispôsobených programov IPM a EPM sme ilustrovali na príklade malých rozmerov. Ďalej sme porovnali metódy IPM a EPM pri riešení vygenerovaných úloh (9) pre meniace sa rozmery n a m_1 , z čoho bolo zjavné, že metóda IPM je vhodnejšou na ich riešenie. Na záver sme skúmali zmeny výnosu a rizika pri

zvyšovaní hodnoty požiadavky na pestrosť m_1 vo vygenerovaných úlohách, pre nižší počet ohraňení $Ax \leq b$, t.j. nižšie hodnoty m , keďže pre väčšie m sa zmenšovala množina prípustných riešení, dokonca mohla byť v niektorých prípadoch pre vyššie hodnoty m_1 prázdna.

Prínosom tejto práce je formulácia kvantitatívneho ohraňenia na minimálnu pestrosť portfólia (7) v tvare konvexného ohraňenia v Markowitzovom modeli s lineárnymi transakčnými nákladmi, ktorá poskytuje investorom nástroj na lepšiu reguláciu "trade-off"-u medzi výnosom a rizikom v portfóliu, porovnanie metód IPM a EPM pri riešení zovšeobecneného Markowitzovho modelu (9), ako aj numerické demonštrovanie vplyvu požiadavky na pestrosť na zmenu výnosu a rizika portfólia.

Zoznam použitej literatúry

- [1] Coello, C. A. C.: *Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art*, Computer methods in applied mechanics and engineering 191 (2002), 1245-1287
- [2] Focardi, S. M., Fabozzi, F. J.: *The Mathematics of Financial Modeling and Investment Management*, John Wiley & Sons, Inc., New Jersey, 2004
- [3] Hamala, M., Trnovská, M.: *Nelineárne programovanie, teória a algoritmy*, EPOS, Bratislava, 2013
- [4] Hass, L. H., Schweizer, D., Proelss, J.: *Alternative Investments*, v Baker, K. H., Filbeck, G. ed., Portfolio Theory and Management, Oxford University Press, Oxford, 2013
- [5] Hult, H. et al.: *Risk and Portfolio Analysis: Principles and Methods*, Springer Science & Business Media, New York, 2012
- [6] León, T., Liern., V., Vercher, E.: *Viability of infeasible portfolio selection problems: A fuzzy approach*, European Journal of Operational Research 139 (2002), 178-189
- [7] Lobo, M. S., Fazel, M., Boyd, S.: *Portfolio optimization with linear and fixed transaction costs*, Annals of Operations Research 152 (2007), 341-365
- [8] Markowitz, H.: *Portfolio selection*, The Journal of Finance 7 (1952), 77-91
- [9] Markowitz, H.: *Portfolio selection: Efficient Diversification of Investments*, John Wiley & Sons, Inc., New York, 1959
- [10] MATLAB dokumentácia: fmincon, dostupné na internete (11.4.2016): <http://www.mathworks.com/help/optim/ug/fmincon.html>
- [11] MATLAB dokumentácia: fminunc, dostupné na internete (11.4.2016): <http://www.mathworks.com/help/optim/ug/fminunc.html>
- [12] MATLAB dokumentácia: Hessian, dostupné na internete (11.4.2016): <http://www.mathworks.com/help/optim/ug/hessian.html>

- [13] MATLAB dokumentácia: Linear or Quadratic Objective with Quadratic Constraints, dostupné na internete (11.4.2016): <http://www.mathworks.com/help/optim/ug/linear-or-quadratic-problem-with-quadratic-constraints.html>
- [14] Melicherčík, I., Olšarová, L., Úradníček, V.: *Kapitoly z finančnej matematiky*, EPOS, Bratislava, 2005
- [15] Polyak, B., Shcherbakov, P.: *Why does Monte Carlo fail to work properly in high-dimensional optimization problems?*, Working Paper, The Institute of Control Science, Moskva, 2016, dostupné na internete (11.4.2016): <http://arxiv.org/abs/1603.00311>
- [16] Runarsson, T. P., Yao, X.: *Stochastic Ranking for Constrained Evolutionary Optimisation*, IEEE Transactions on Evolutionary Computation 4 (2000), 284-294
- [17] Ruschendorf, L.: *Mathematical Risk Analysis: Dependence, Risk Bounds, Optimal Allocations and Portfolios*, Springer-Verlag, Berlin, 2013
- [18] Salo, A., Keisler, J., Morton, A.: *Portfolio Decision Analysis: Improved Methods for Resource Allocation*, Springer Science & Business Media, New York, 2011
- [19] Wang, S., Xia, Y.: *Portfolio Selection and Asset Pricing*, Springer-Verlag, Berlin, 2002
- [20] Yu, X., Gen, M.: *Introduction to Evolutionary Algorithms*, Springer-Verlag, London, 2010

Príloha A

Generátor úloh

```
function[c,d,A,b,Q,sigma2,m1,x,u,v]=Generator(m,n,l)

x=zeros(n,1);
d=zeros(n,1);
while length(x(x>0))<m1
    i=randi([1 n],1,1);
    d(i)=1+0.2*rand(1);
    x(i)=1/(m1*d(i));
end
for i=1:n
    if (d(i)==0)
        d(i)=1+0.2*rand(1);
    end
end
rep=1;
while rep==1
    Q=diag(rand(n,1));
    pom1=Q\d;
    sigmamin=1/(d'*pom1);
    sigma2=x'*Q*x;
    if sigma2<sigmamin
        rep=1;
    else
        rep=0;
    end
end
V=zeros(n,1);
for i=1:n
    if (x(i)==0)
        V(i)=rand(1);
    end
end
```

```
        end
    end
    if m<=n
        u_kladne=randi([0 m],1,1);
    else
        u_kladne=randi([0 n],1,1);
    end
    u=[zeros(m,1);rand(2,1)];
    while length(u(u(1:m)>0))<u_kladne
        i=randi([1 m],1,1);
        u(i)=rand(1);
    end
    A=randi([-4,4],m,n);
    if m<=n
        while rank(A)<m
            A=randi([-4,4],m,n);
        end
    else
        while rank(A)<n
            A=randi([-4,4],m,n);
        end
    end
    pom2=A*x;
    b=zeros(m,1);
    for i=1:m
        if (u(i)==0)
            b(i)=pom2(i)+5*rand(1);
        else
            b(i)=pom2(i);
        end
    end
end
```

```
v=rand(1);  
D=diag(d.^2);  
c=-V+[A',2*Q*x,2*D*x]*u+v*d;  
end
```

IPM

Účelová funkcia

```
function [y,grady] = obj_ipm(z,c_new)  
y = -c_new'*z;  
grady = -c_new;  
end
```

Nelineárne ohraničenia

```
function [y,yeq,grady,gradyeq] = nonlin_constr(z,H,k,p)  
j = length(H);  
y = zeros(1,j);  
for i = 1:j  
    y(i) = z'*Hi*z + ki'*z + pi;  
end  
yeq = [];  
grady = zeros(length(z),j);  
for i = 1:j  
    grady(:,i) = 2*Hi*z + ki;  
end  
gradyeq = [];  
end
```

Hessova matica

```
function hess = hessian(lambda,H)  
n=length(H1);  
hess = zeros(n);  
j = length(H);
```

```
for i = 1:j
    hess = hess + lambda.ineqnonlin(i)*Hi;
end
end
```

IPM

```
z0=[rand(n,1);1];
clear H
clear k
clear p
H1=([ [Q,zeros(n,1)];zeros(1,n+1)]+[ [zeros(n,n),zeros(n,1)];...
[zeros(1,n),sigma2-z0'*[ [Q,zeros(n,1)];zeros(1,n+1)]*z0]]);
k1=zeros(n+1,1);
p1=-sigma2;
if m1 ==0
    H2=([ [D,zeros(n,1)];zeros(1,n+1)]+[ [zeros(n,n),zeros(n,1)];...
[zeros(1,n),(1/m1)-z0'*[ [D,zeros(n,1)];zeros(1,n+1)]*z0]]);
    k2=zeros(n+1,1);
    p2=-(1/m1);
end
options = optimoptions(@fmincon,'Algorithm','interior-point',...
    'GradObj','on','GradConstr','on','Hessian','user-supplied',...
    'HessFcn',@(z,lambda)hessian(lambda,H),'TolFun',1e-14,'TolCon',...
    1e-14,'MaxIter',100000,'MaxFunEvals',10000,'TolX',1e-14,...
    'Display','off');
fun = @(z)obj_ipm(z,[c;-10^10]);
nonlconstr = (z)nonlin_constr(z,H,k,p);
[z,fval,eflag,output,lambda] = fmincon(fun,z0,[A,zeros(m,1)]+...
    [zeros(m,n),b-A*z0(1:(end-1))],b,[d',0]+[zeros(1,n),1-d'*...
    z0(1:(end-1))],1,zeros(n+1,1),[],nonlconstr,options);
```

EPMÚčelová funkcia $\psi_l(x)$

```
function [y] = obj_epm(z,c,d,A,b,Q,sigma2,m1,D,r)
m=size(A,1);
n=size(z,1);
g=zeros(m+n+3,1);
g(1:m)=A*z-b;
g(m+1:m+n)=-z;
g(m+n+1)=abs(d'*z-1);
g(m+n+2)=z'*Q*z-sigma2;
g(m+n+3)=z'*D*z-(1/m1);
ohr=0;
for i=1:m+n+3
    ohr=ohr+max(0,g(i))^2;
end
y = -c'*z+r*ohr;
end
```

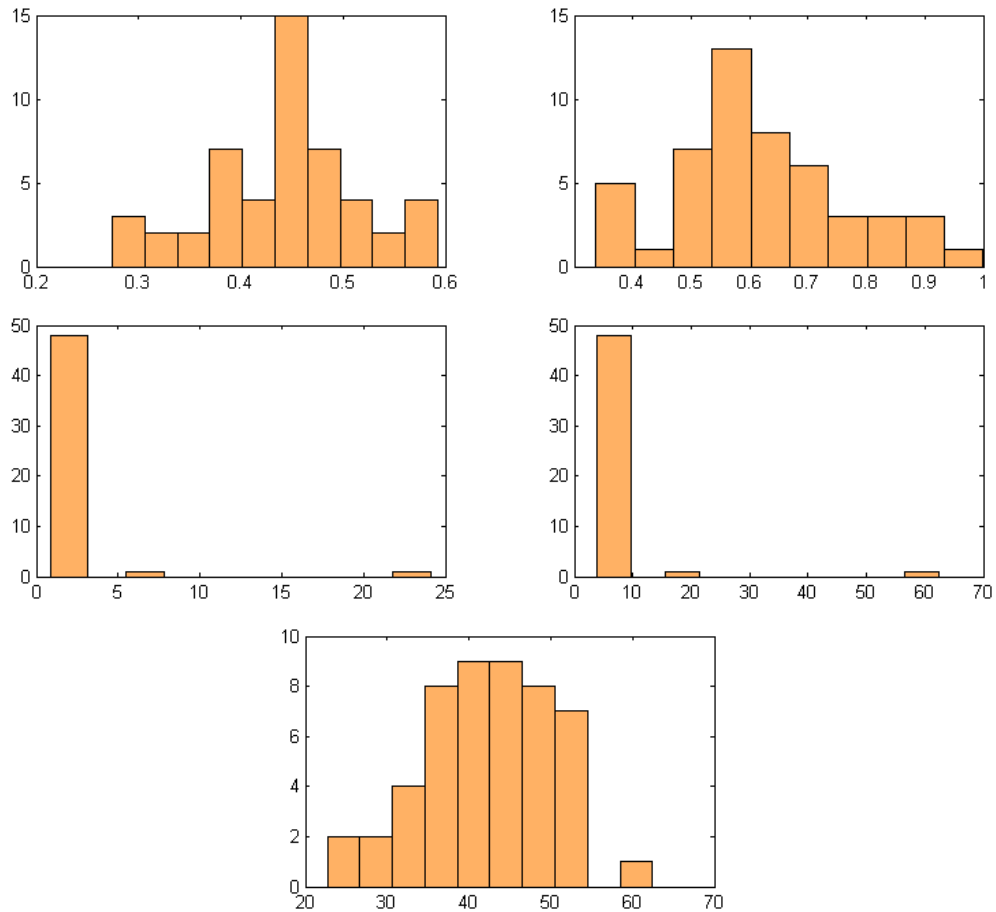
EPM

```
z0=rand(n,1);
options = optimoptions(@fminunc,'Algorithm','quasi-newton','TolFun',...
1e-14,'MaxIter',100000,'MaxFunEvals',10000,'TolX',1e-14,...
'Display','off');
r=1;
ro=3;
maxetapa=200;
for etapa=1:maxetapa
    fun = @(z)obj_epm(z,c,d,A,b,Q,sigma2,m1,D,r);
    [z,fval,eflag,output,lambda] = fminunc(fun,z0,options);
    z0=z;
    r=ro*r;
```

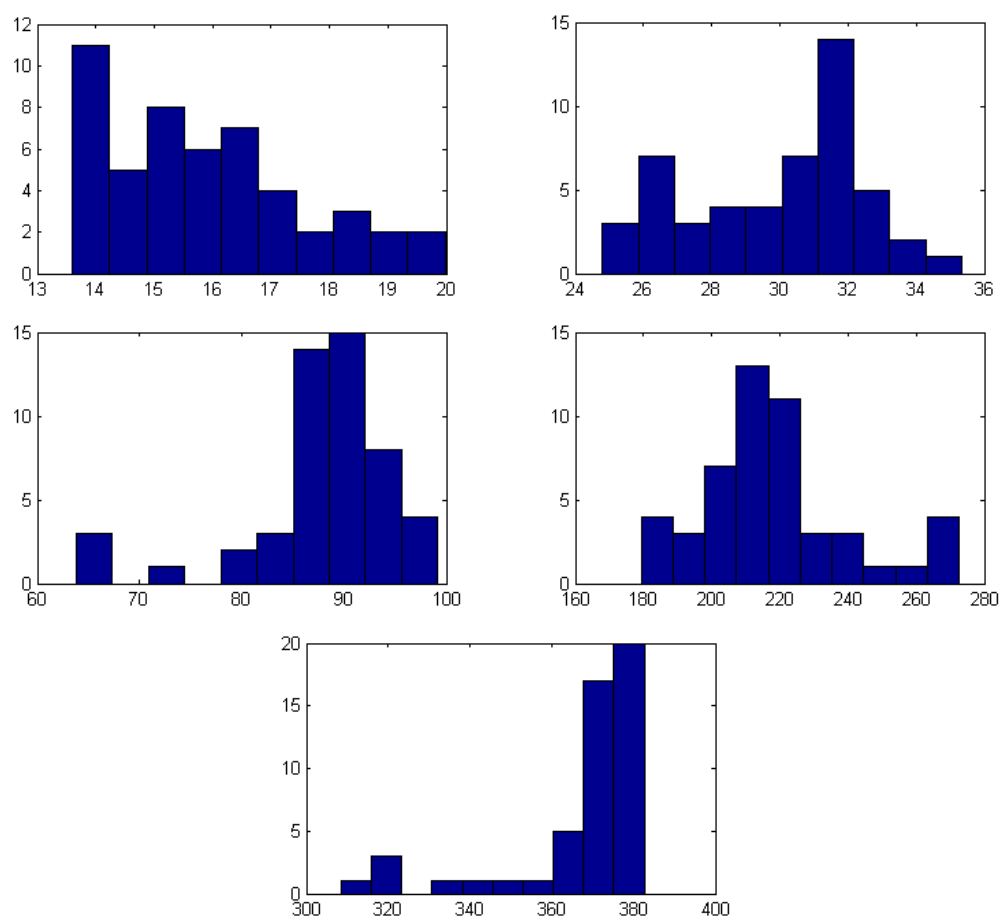
end

Príloha B

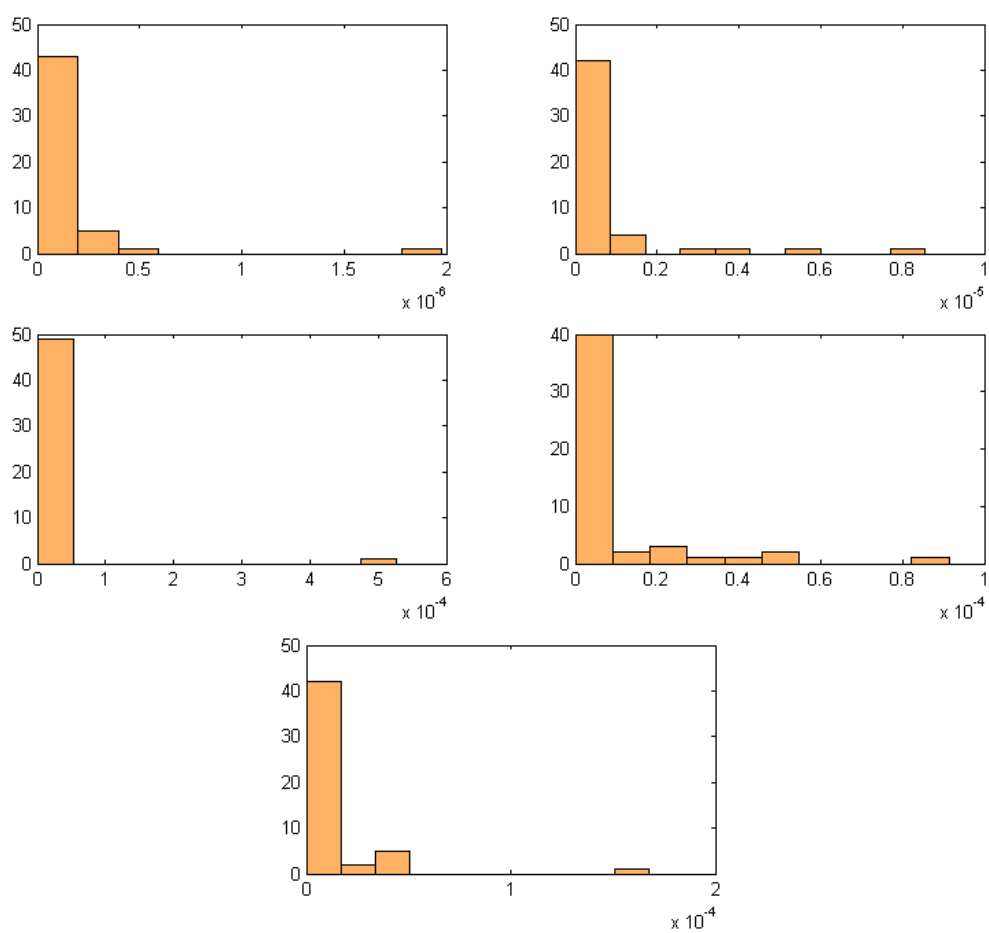
V tejto prílohe uvádzame histogramy času, odchýlky nájdeného riešenia od optimálneho $\| \hat{x} - x^k \|_2$, ako aj účelových hodnôt $| (f(\hat{x}) - f(x^k)) |$ pre všetky použité $n = 5, 10, 20, 50$ a 100 , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$, pre metódy IPM a EPM



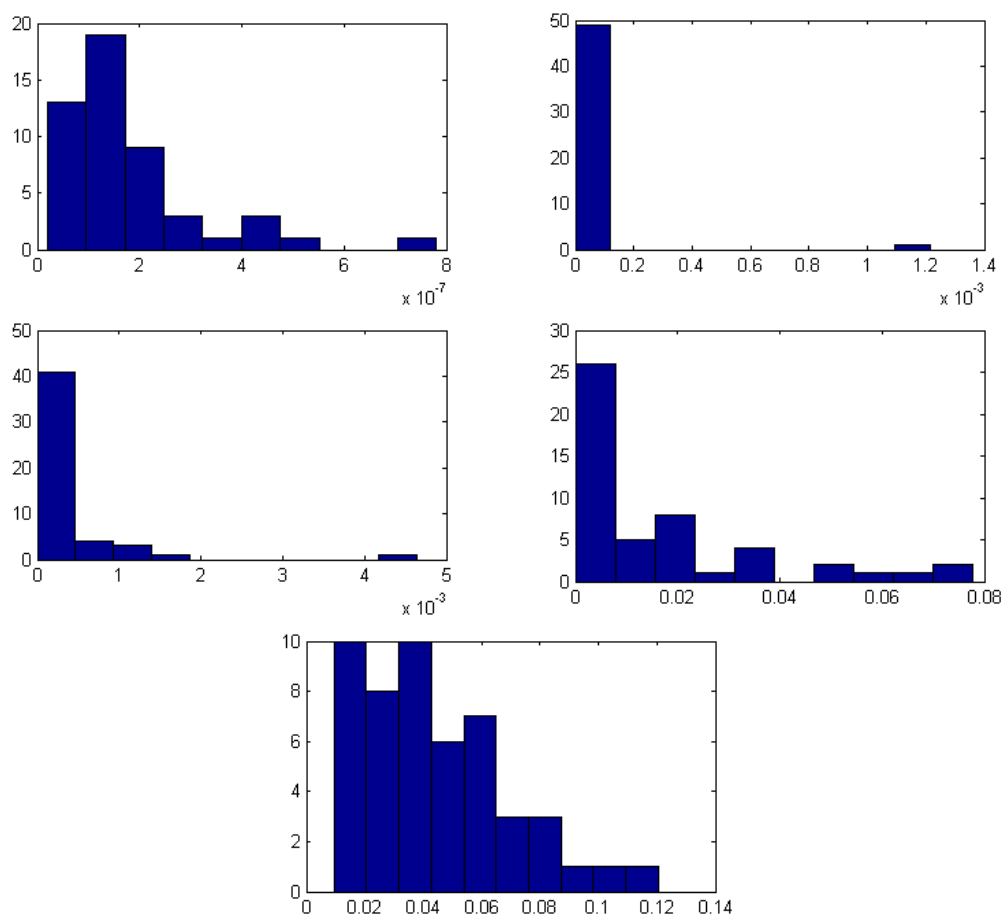
Obr. A.13: Porovnanie výpočtového času metódou IPM pre rôzne veľkosti n , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$.



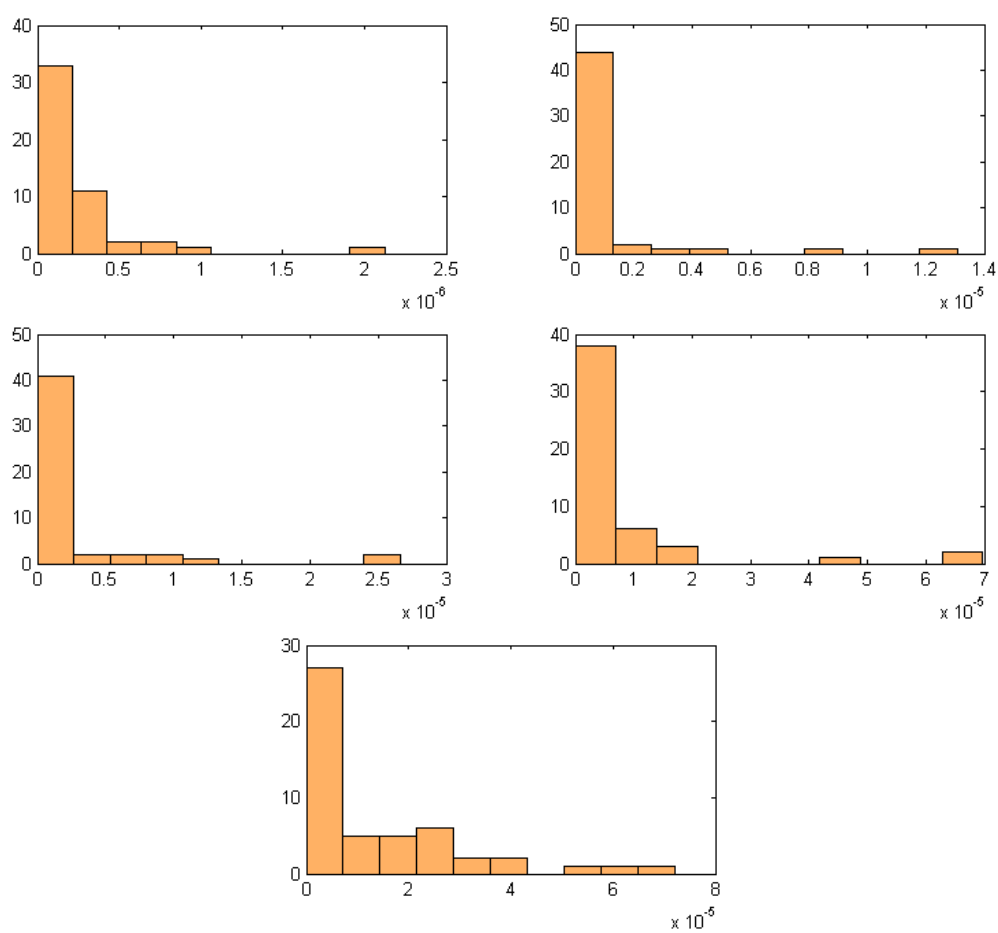
Obr. A.14: Porovnanie výpočtového času metódou EPM pre rôzne veľkosti n , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$.



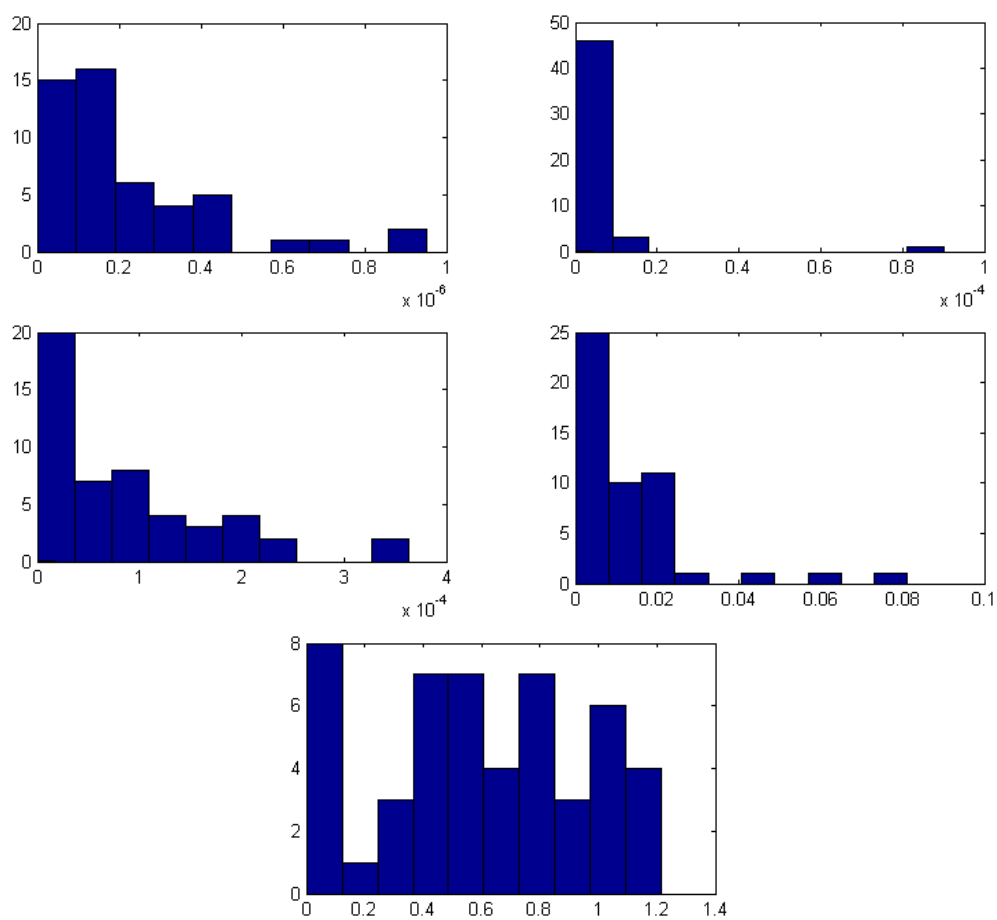
Obr. A.15: Porovnanie odchýlky od optimálneho riešenia $\| \hat{x} - x^k \|_2$ metódou IPM pre rôzne veľkosti n , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$.



Obr. A.16: Porovnanie odchýlky od optimálneho riešenia $\|\hat{x} - x^k\|_2$ metódou EPM pre rôzne veľkosti n , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$.



Obr. A.17: Porovnanie odchýlky od optimálnej hodnoty účelovej funkcie $|f(\hat{x}) - f(x^k)|$ metódou IPM pre rôzne veľkosti n , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$.



Obr. A.18: Porovnanie odchýlky od optimálnej hodnoty účelovej funkcie $|f(\hat{x}) - f(x^k)|$ metódou EPM pre rôzne veľkosti n , $m = \frac{n}{2}$ a $m_1 = \frac{n}{10}$.

Príloha C

Táto príloha obsahuje porovnanie priemerného výnosu, rizika a ich zmien z úloh (9) rozmerov $n = 100$, $m_1 = 10$, $m = 0, 5$ a 10 , pre ktoré sme požadovanú pestrosť portfólia m_1 zvyšovali po jednej až po $(n - 1)$ a riešili ich metódou IPM. Z 50 vygenerovaných úloh (9) pre každú kombináciu rozmerov n , m_1 a m sme do porovnania vybrali len tie, pre ktoré algoritmus IPM našiel riešenia až po $m_1 = (n - 1)$. Pre lepšiu čitateľnosť sú uvedené hodnoty porovnaní len pre každé piate m_1 .

$m = 0$				
m_1	Riziko	Δ rizika	Výnos	Δ výnosu
10	0,0429		0,6362	
15	0,0307	-0,0735	0,5937	-0,0141
20	0,0224	-0,0540	0,5546	-0,0132
25	0,0177	-0,0429	0,5202	-0,0125
30	0,0145	-0,0354	0,4891	-0,0122
35	0,0123	-0,0302	0,4601	-0,0121
40	0,0107	-0,0264	0,4328	-0,0122
45	0,0095	-0,0233	0,4067	-0,0125
50	0,0085	-0,0209	0,3813	-0,0130
55	0,0077	-0,0188	0,3564	-0,0137
60	0,0070	-0,0171	0,3319	-0,0144
65	0,0065	-0,0157	0,3079	-0,0153
70	0,0060	-0,0146	0,2841	-0,0164
75	0,0056	-0,0136	0,2605	-0,0178
80	0,0052	-0,0127	0,2367	-0,0198
85	0,0049	-0,0120	0,2122	-0,0232
90	0,0046	-0,0113	0,1855	-0,0293
95	0,0044	-0,0106	0,1535	-0,0445

Tabuľka 12: Porovnanie priemerného výnosu, rizika a ich relatívnych zmien pri zmene m_1 pre $n = 100$ a $m = 0$

$m = 5$				
m_1	Riziko	Δ rizika	Výnos	Δ výnosu
10	0,0417		0,7913	
15	0,0296	-0,0716	0,7519	-0,0102
20	0,0217	-0,0529	0,7163	-0,0094
25	0,0172	-0,0418	0,6842	-0,0090
30	0,0142	-0,0342	0,6544	-0,0088
35	0,0122	-0,0293	0,6262	-0,0088
40	0,0106	-0,0255	0,5991	-0,0089
45	0,0094	-0,0226	0,5725	-0,0091
50	0,0085	-0,0203	0,5462	-0,0096
55	0,0077	-0,0182	0,5190	-0,0106
60	0,0070	-0,0167	0,4914	-0,0111
65	0,0065	-0,0154	0,4638	-0,0118
70	0,0060	-0,0143	0,4360	-0,0125
75	0,0056	-0,0134	0,4083	-0,0134
80	0,0053	-0,0125	0,3803	-0,0146
85	0,0050	-0,0118	0,3512	-0,0167
90	0,0047	-0,0112	0,3192	-0,0208
95	0,0044	-0,0105	0,2776	-0,0347

Tabuľka 13: Porovnanie priemerného výnosu, rizika a ich relatívnych zmien pri zmene m_1 pre $n = 100$ a $m = 5$

$m = 10$				
m_1	Riziko	Δ rizika	Výnos	Δ výnosu
10	0,0413		0,9648	
15	0,0298	-0,0711	0,9272	-0,0082
20	0,0220	-0,0525	0,8907	-0,0079
25	0,0174	-0,0426	0,8564	-0,0077
30	0,0143	-0,0349	0,8241	-0,0076
35	0,0122	-0,0294	0,7934	-0,0076
40	0,0106	-0,0255	0,7632	-0,0079
45	0,0094	-0,0224	0,7327	-0,0084
50	0,0085	-0,0204	0,6977	-0,0111
55	0,0077	-0,0187	0,6577	-0,0121
60	0,0070	-0,0171	0,6175	-0,0128
65	0,0065	-0,0157	0,5771	-0,0139
70	0,0060	-0,0145	0,5353	-0,0156
75	0,0056	-0,0135	0,4922	-0,0174
80	0,0053	-0,0127	0,4468	-0,0206
85	0,0049	-0,0117	0,3947	-0,0276
90	0,0047	-0,0113	0,3306	-0,0407
95	0,0044	-0,0107	0,2493	-0,0683

Tabuľka 14: Porovnanie priemerného výnosu, rizika a ich relatívnych zmien pri zmene m_1 pre $n = 100$ a $m = 10$