

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ALGORITMY PRE ZÁKLADNÚ ÚLOHU DYNAMICKEJ
OPTIMALIZÁCIE PORTFÓLIA
DIPLOMOVÁ PRÁCA

Bratislava 2012

Bc. Jozef Kalman

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



ALGORITMY PRE ZÁKLADNÚ ÚLOHU DYNAMICKEJ
OPTIMALIZÁCIE PORTFÓLIA
DIPLOMOVÁ PRÁCA

Študijný program: Ekonomická a finančná matematika
Študijný odbor: 9.1.9 Aplikovaná matematika, 1114
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky,
FMFI UK, Mlynská dolina, 842 48 Bratislava
Školiteľ: Mgr. Soňa Kilianová, PhD.

Bratislava 2012

Bc. Jozef Kalman



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Jozef Kalman
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Algoritmy pre základnú úlohu dynamickej optimalizácie portfólia

Cieľ: Základou úlohou dynamickej optimalizácie portfólia budeme rozumieť úlohu prerozdelenia prostriedkov za účelom investovania do množiny aktív, pričom uvažujeme pravidelné prehodnocovanie investícií. Predmetom optimalizácie budú rôzne miery rizika spojeného s investovaním, ako napr. conditional value at risk. Cieľom diplomovej práce bude porovnať výhodnosť použitia rôznych existujúcich algoritmov stochastického programovania na riešenie týchto úloh - pre rôzne miery rizika aj rôzne veľkosti úloh - a možno zistiť, či je niektorý algoritmus pre tento typ úloh výhodnejší než ostatné.

Vedúci: Mgr. Soňa Kilianová, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Dátum zadania: 13.01.2011

Dátum schválenia: 14.01.2011
prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné prehlásenie

Čestne prehlasujem, že som diplomovú prácu vypracoval samostatne, s pomocou literatúry uvedenej v zozname, konzultácií s vedúcou diplomovej práce a s využitím teoretických vedomostí.

V Bratislave, apríl 2012

.....
Jozef Kalman

Podakovanie

Touto cestou by som sa rád poďakoval vedúcej diplomovej práce

Mgr. Soni Kilianovaj, PhD.

*za cenné rady a pripomienky, za poskytnutie literatúry ale hlavne za podporu
a motiváciu pri písaní diplomovej práce.*

KALMAN, Jozef: Algoritmy pre základnú úlohu dynamickej optimalizácie portfólia. [Diplomová práca]-Univerzita Komenského v Bratislava. Fakulta matematiky, fyziky a informatiky. Katedra aplikovanej matematiky a štatistiky. Vedúci diplomovej práce: Mgr. Soňa Kilianová, PhD. Bratislava 2012

Abstrakt

Uvažovanie náhodných údajov v optimalizačných procesoch často vedie na optimalizačné úlohy obrovských rozmerov. V týchto prípadoch predstavujú dekompozičné algoritmy výrazné zefektívnenie optimalizačného procesu. Táto diplomová práca sa zaoberá aplikovaním dekompozičných metód stochastického programovania na jedno a viac-periódovú základnú úlohu optimalizácie portfólia s Conditional Value-at-Risk (CVaR) ako rizikovou mierou, s cieľom porovnať výhodnosť použitia jednotlivých metód pre danú úlohu. Predstavíme L-shaped metódu a jej viaczovú a regularizovanú modifikáciu. Popíšeme CVaRmin metódu, ktorá je špeciálne navrhnutá na riešenie jedno-periódovej úlohy optimalizácie portfólia s CVaR v účelovej funkcii. Navrhujeme viaczovú verziu CVaRmin metódy. Využitím princípov použitých pri odvodení jednozovovej CVaRmin metódy zefektívňujeme vnorenú verziu L-shaped metódy na riešenie viac-periódovej úlohy optimalizácie portfólia. Rozsiahlym testovaním vyhodnotíme efektívnosť aplikovaných metód.

Kľúčové slová:

Úloha optimalizácie portfólia, Stochastické programovanie, L-shaped metóda, CVaRmin metóda, Conditional Value-at-Risk

KALMAN, Jozef: Algorithms for basic dynamic portfolio optimization problem.
[Master's thesis]-Comenius University in Bratislava. Faculty of mathematics, physics
and informatics. Department of applied mathematics and statistics.

Thesis supervisor: Mgr. Soňa Kilianová, PhD.

Bratislava 2012

Abstract

Considering stochastic data in optimization processes often leads to overwhelming optimization problems. In these cases usage of decomposition algorithms makes an optimization process markedly more effective. This master's thesis deals with application of decomposition algorithms of stochastic programming to the one and multi-period basic portfolio optimization problem with Conditional Value-at-Risk in objective function. Our aim is to compare an expedience of particular methods for given problem. We introduce the singlecut L-shaped method. The multicut and regularized modification of the L-shaped method is given as well. We describe the CVaRmin method, which has been specially proposed for optimization processes with CVaR in objective function. We derive the multicut version of the CVaRmin method and also improve efficiency of the nested versions of the L-shaped method for solving multi-period optimization problem. This is done by involving the same principles that have been used to derive CVaRmin method. The efficiency of all considered methods is analyzed in numerical experiments.

Key words:

Portfolio optimization problem, Stochastic programming, L-shaped method, CVaRmin method, Conditional Value-at-Risk

Obsah

Úvod	1
1 Formulácia základnej úlohy optimalizácie portfólia	2
1.1 Modelovanie a meranie rizika	2
1.1.1 Jedno-periódová základná úloha optimalizácie portfólia	3
1.1.2 Conditional Value-at-Risk	5
1.1.3 Viac-periódová základná úloha optimalizácie portfólia	7
1.2 Motivácia použitia dekompozičných algoritmov	13
2 Stochastické lineárne úlohy	16
2.1 Dvoj-stupňové stochastické lineárne úlohy	16
2.1.1 Odvodenie Dvojstupňových SLÚ	16
2.1.2 Dvojstupňové stochastické lineárne úlohy s pevnou kompenzáciou	18
2.1.3 Vlastnosti dvojstupňových SLÚ s pevnou kompenzáciou	19
2.2 Viac-stupňové stochastické lineárne úlohy s pevnou kompenzáciou	23
2.2.1 Formulácia viac-periódovej úlohy optimalizácie portfólia v tvare viac-stupňovej SLÚ s pevnou kompenzáciou	25
3 Algoritmy na riešenie jedno-periódovej a viac-periódovej základnej úlohy optimalizácie portfólia	30
3.1 Jednerezová L-shaped metóda	30
3.2 Princíp fungovania L-shaped metódy	32
3.3 Viacrezová L-shaped metóda	34
3.4 Regularizovaná L-shaped metóda	36
3.5 CVaRmin metóda	38
3.6 Viacrezová CVaRmin metóda	40
3.7 Priame metódy	41
3.7.1 Simplexová metóda	41
3.7.2 Metóda vnútorného bodu	42
3.8 Vnorená jednerezová L-shaped metóda	43
3.9 Vnorená viacrezová L-shaped metóda	46
3.10 Zefektívnenie jedno a viacrezovej vnorenej L-shaped metódy	47

4	Numerické výsledky a implementácia	49
4.1	Jedno-periódová úloha optimalizácie portfólia	49
4.1.1	Závislosť výpočtového času od parametra M	51
4.1.2	Závislosť výpočtového času od parametra K	56
4.1.3	Závislosť výpočtového času od parametra α	61
4.1.4	Regularizovaná L-shaped metóda	63
4.1.5	Zhrnutie	64
4.2	Viac-periódová úloha optimalizácie portfólia	66
4.2.1	Závislosť výpočtového času od parametra K	67
4.2.2	Závislosť výpočtového času od parametra M	69
4.2.3	Závislosť výpočtového času od parametra α	71
4.2.4	Zhrnutie	72
	Záver	74
	Zoznam použitej literatúry	75
	Príloha A	78
	Príloha B	79
	Príloha C	85

Úvod

Tento rok uplynulo 60 rokov od publikovania Markowitzovho slávneho článku *Portfolio selection*, ktorý sa prvýkrát objavil v marcovom vydaní *Journal of Finance* v roku 1952. Úlohy optimalizácie portfólia aj po takto dlhej dobe patria medzi najčastejšie riešené optimalizačné úlohy. Výpočtová náročnosť jednotlivých úloh do veľkej miery závisí od uvažovanej rizikovej miery. V Markowitzovej teórii portfólia sa riziko meria štandardnou odchýlkou, ktorú ale v súčasnosti nahrádza riziková miera Value-at-Risk (VaR). Jednou z najviac kritizovaných nevýhod miery VaR je práve jej vysoká výpočtová náročnosť. Vhodnou alternatívou k miere VaR môže byť riziková miera Conditional Value-at-Risk (CVaR), ktorá má z výpočtového hľadiska, v porovnaní s VaR, oveľa užitočnejšie vlastnosti. V roku 2000 publikovali autori *Rockefeller* a *Uryasev* ďalší významný článok *Optimization of Conditional Value at Risk*, v ktorom vyjadrili mieru CVaR v tvare maximalizačnej formuly. S využitím tejto skutočnosti a predpokladu o diskretnom rozdelení náhodných údajov vstupujúcich do úlohy optimalizácie portfólia s CVaR ako rizikovou mierou, je možné túto úlohu vyjadriť v tvare lineárneho programovania, čo predstavuje značné výpočtové zjednodušenie.

Potreba uvažovať náhodné vstupné údaje má za následok obrovské rozmery týchto optimalizačných úloh. Matica ohraničení týchto úloh má však špeciálnu blokovú štruktúru, ktorú efektívne využívajú dekompozičné metódy lineárneho stochastického programovania.

Motiváciou tejto práce bolo porovnať rôzne algoritmy, ktorými sa úloha optimalizácie portfólia dá riešiť, a zistiť, či niektorý z nich poskytuje výrazné výhody oproti ostatným algoritmom. Práca sa skladá zo štyroch kapitol a štruktúra práce je nasledovná.

V prvej kapitole naformulujeme jedno a viac-periódovú základnú úlohu optimalizácie portfólia a predstavíme rizikovú mieru Conditional Value-at-Risk. Druhá kapitola je venovaná teórii dvoj a viac-stupňových stochastických lineárnych úloh s pevnou kompenzáciou. V tretej kapitole predstavíme a vysvetlíme princíp fungovania uvažovaných dekompozičných metód. Budú to jednerezová, viacrezová a regularizovaná verzia L-shaped metódy, CVaRmin metóda a jedno a viacrezová vnorená L-shaped metóda. Vlastným prínosom tejto práce bude odvodenie viacrezovej CVaRmin metódy v podkapitole 3.6, zefektívnenie vnorených verzií L-shaped metódy pre typ úloh uvedených v prvej kapitole a rozsiahly numerický experiment zameraný na testovanie efektívnosti jednotlivých metód. Náplňou štvrtej kapitoly sú výsledky numerického experimentu.

Kapitola 1

Formulácia základnej úlohy optimalizácie portfólia

Cieľom kapitoly je predstavenie základnej úlohy optimalizácie portfólia. Postupne zdefinujeme jedno-periódovú a viac-periódovú úlohu optimalizácie portfólia. Jedno-periódová úloha optimalizácie portfólia pozostáva len z jednej časovej periódy, na začiatku ktorej je potrebné určiť investičnú stratégiu (určiť váhy aktív v portfóliu). Následne, na konci časovej periódy, realizácia výnosov aktív určí dopad zvolenej investičnej stratégie na konečnú hodnotu portfólia. Vo viac-periódovej úlohe uvažujeme $H > 1$ časových periód. Na začiatku každej periódy máme, na základe realizácií výnosov aktív z predchádzajúcich časových periód, možnosť prehodnotiť investičnú stratégiu (v každej časovej perióde môžeme zmeniť váhy aktív v portfóliu). Úlohou je minimalizovať riziko investície alebo maximalizovať výnos portfólia za podmienky splnenia určitých ohraničení. Na meranie rizika investície budeme používať rizikovú mieru *Conditional Value-at-Risk*. Predstavenie a uvedenie základných vlastností tejto miery je tiež náplňou tejto kapitoly. Napokon sa budeme venovať motivácii pre riešenie úloh optimalizácie portfólia dekompozičnými metódami, pretože ako sa ukáže neskôr, aplikovanie priamych metód (napr. simplexová metóda, metóda vnútorného bodu) na tieto úlohy prináša v mnohých prípadoch značné komplikácie. Dekompozičným metódam sa budeme bližšie venovať v Kapitole 3.

1.1 Modelovanie a meranie rizika

V tejto časti zdefinujeme jedno a viac-periódovú základnú úlohu optimalizácie portfólia. Predstavíme rizikovú mieru *Conditional Value-at-Risk*. V súčasnosti sa používajú rôzne rizikové miery. Medzi najznámejšie rizikové miery patrí Value-at-Risk (VaR) a Conditional Value-at-Risk (CVaR). Tieto dve miery sú v súčasnosti aj predmetom rozsiahleho výskumu. Dôvodom prečo sme si vybrali mieru Conditional Value-at-Risk je zvyšujúci sa záujem o túto rizikovú mieru. Použitie CVaR má aj veľký potenciál a v krátkej dobe môže nahradiť mieru VaR, ktorá je v súčasnosti najpoužívanejšou

rizikovou mierou. Ďalším dôvodom je, že značná časť nevýhod miery VaR je použitím miery CVaR odstránená. Tieto nevýhody ako aj základné vlastnosti mier VaR a CVaR si stručne popíšeme v podkapitole 1.1.2. Ďalšie známe miery rizika sú:

- Štandardná odchýlka (Standard deviation),
- Priemerná najvyššia odchýlka (Mean absolute deviation),
- Dolná štandardná odchýlka (Lower standard deviation),
- Minimálna strata (Minimal loss efficiency).

Definície týchto rizikových mier ako aj ich vlastnosti sa dajú nájsť napr. v [20].

1.1.1 Jedno-periódová základná úloha optimalizácie portfólia

Pod jedno-periódovou základnou úlohou optimalizácie portfólia budeme rozumieť úlohu s jedným časovým horizontom. Na začiatku časového horizontu zvolíme investičnú stratégiu a následne už len počkáme na realizáciu výnosov aktív a výsledný efekt tejto realizácie na konečnú hodnotu portfólia. Výnosy aktív sú reprezentované náhodným vektorom ξ . O náhodnom vektore ξ predpokladáme, že pochádza z konečného diskrétného rozdelenia, čiže výnosy aktív môžu nadobúdať konečný počet realizácií. Jednu realizáciu náhodného vektora ξ budeme nazývať scenár.

Pred zadefinovaním úlohy je užitočné uviesť základné predpoklady úlohy a použité označenia. Predpoklady úlohy:

- neuvažujeme transakčné náklady,
- váhy v portfóliu môžu byť aj záporné (shortselling je dovolený),
- cieľom je minimalizovať riziko vyjadrené pomocou funkcionálu \mathcal{D} ,
- v úlohe uvažujeme iba dva druhy ohraničení - ohraničenie na minimálny požadovaný výnos a rozpočtové ohraničenie, pričom bez ujmy na všeobecnosti predpokladáme normalizáciu rozpočtu na 1.

Základné označenia:

- $\xi = (\xi^1, \xi^2, \dots, \xi^M)^T$ - náhodný vektor výnosov aktív v portfóliu, kde ξ^m , $m = 1, \dots, M$ reprezentuje výnos m -tého aktíva a M označuje celkový počet aktív v portfóliu;
- vektor $\xi_k = (\xi_k^1, \xi_k^2, \dots, \xi_k^M)^T$ reprezentuje jeden scenár, ktorý nastáva s pravdepodobnosťou p_k ($\sum_{k=1}^K p_k = 1$, $p_k \geq 0$, $k = 1, \dots, K$), $k = 1, \dots, K$ a K označuje počet scenárov;
- $\Xi \in \mathbf{R}^K \times \mathbf{R}^M$ - nosič vektora ξ , kde $\Xi_{k,m} = \xi_k^m$ a $k = 1, \dots, K$; $m = 1, \dots, M$, t.j. nosič $\Xi \in \mathbf{R}^K \times \mathbf{R}^M$ obsahuje všetky možné realizácie náhodného vektora ξ ;

- $x \in \mathbf{R}^M$ - vektor váh v portfóliu. Váhy zodpovedajú hodnote investovanej do príslušného aktíva a nie percentuálne vyjadrenej časti z celkového rozpočtu;
- $r \in \mathbf{R}^M$ - vektor očakávaných výnosov, $r = \mathbb{E}(\xi)$;
- $\mu \in \mathbf{R}$ - minimálny požadovaný výnos, $r^T x \geq \mu$;
- $\mathbf{1}_M$ jednotkový vektor s rozmermi $(M \times 1)$.

Jedno-periódovú základnú úlohu optimalizácie portfólia budeme podľa [20] uvádzať v tvare:

$$\begin{aligned} \min_x \mathcal{D}[x^T \xi] \\ \mathbb{E}[x^T \xi] &\geq \mu, \\ \mathbf{1}_M^T x &\leq 1, \\ (x &\geq 0). \end{aligned} \tag{1.1}$$

Funkcionál \mathcal{D} , ktorý reprezentuje riziko portfólia, definujeme podľa [20] ako:

Definícia ([20], str. 36) *Deviation risk functional*

Zobrazenie \mathcal{D} z lineárneho priestoru \mathcal{Y} náhodných premenných definovaných na pravdepodobnostnom priestore $(\Omega, \mathcal{F}, \mathcal{P})$ do reálnych čísel sa nazýva *Deviation risk functional*, ak pre všetky $Y \in \mathcal{Y}$ splňa nasledovné vlastnosti:

- *Invariancia vzhľadom na posun* : $\mathcal{D}(Y + c) = \mathcal{D}(Y)$, $\forall c \in \mathbf{R}$,
- *Konvexnosť* : $\mathcal{D}(\lambda Y^1 + (1 - \lambda)Y^2) \leq \lambda \mathcal{D}(Y^1) + (1 - \lambda)\mathcal{D}(Y^2)$, $\lambda \in [0, 1]$,
- *Monotónnosť* : $Y^1 \leq Y^2 \Rightarrow \mathbb{E}(Y^1) - \mathcal{D}(Y^1) \leq \mathbb{E}(Y^2) - \mathcal{D}(Y^2)$.

S meraním rizika sa spája aj iný typ funkcionálu, tzv. Acceptability functional \mathcal{A} . Je to funkcionál popisujúci preferencie v zmysle, že vyššia hodnota funkcionálu znamená vyššie preferencie. Definujeme ho podľa [20] nasledovne:

Definícia ([20], str. 37) *Acceptability functional*

Zobrazenie \mathcal{A} z lineárneho priestoru \mathcal{Y} náhodných premenných definovaných na pravdepodobnostnom priestore $(\Omega, \mathcal{F}, \mathcal{P})$ do reálnych čísel sa nazýva *acceptability functional*, ak pre všetky $Y \in \mathcal{Y}$ splňa nasledovné vlastnosti:

- *Ekvivariancia vzhľadom na posun* : $\mathcal{A}(Y + c) = \mathcal{A}(Y) + c$, $\forall c \in \mathbf{R}$,
- *Konkávnosť* : $\mathcal{A}(\lambda Y^1 + (1 - \lambda)Y^2) \geq \lambda \mathcal{A}(Y^1) + (1 - \lambda)\mathcal{A}(Y^2)$, $\lambda \in [0, 1]$,
- *Monotónnosť* : $Y^1 \leq Y^2 \Rightarrow \mathcal{A}(Y^1) \leq \mathcal{A}(Y^2)$.

Vzťah medzi funkcionálmi \mathcal{A} a \mathcal{D} je daný vzťahom:

$$\mathcal{D}(Y) = \mathbb{E}(Y) - \mathcal{A}(Y).$$

Typickým predstaviteľom funkcionálu \mathcal{D} je štandardná odchýlka $\mathcal{D}(Y) = \sigma(Y)$. Ako príklady funkcionálu \mathcal{A} si môžeme uviesť napr. miery Value-at-Risk $\mathcal{A}(Y) = VaR_\alpha(Y)$ alebo Conditional Value-at-Risk $\mathcal{A}(Y) = CVaR_\alpha(Y)$, z ktorých obe zdefinujeme v nasledujúcej podkapitole.

1.1.2 Conditional Value-at-Risk

Riziková miera CVaR (Podmienená hodnota v riziku) je v posledných rokoch predmetom intenzívneho výskumu v oblasti risk manažmentu. Bola vyvinutá ako náhrada za rizikovú mieru VaR (Hodnota v riziku), ktorá je v súčasnosti asi najpoužívanjšou mierou v sektore bankovníctva a financií. Pre lepšie porozumenie CVaR si stručne zdefinujeme aj mieru VaR, keďže CVaR je úzko spojená s práve VaR.

Value-at-Risk bola vyvinutá Leavensom v [16] v roku 1945. Celý vývoj miery VaR je pekne zhrnutý v [13]. Vo väčšine publikácií je miera VaR definovaná pre funkciu strát. Nakoľko, pre potreby tejto práce je výhodnejšia definícia pre funkciu výnosov, uvádzame definíciu miery VaR podľa [20].

Definícia ([20], str. 57) *Value-at-Risk*

α kvantil distribučnej funkcie G

$$VaR_\alpha(Y) = VaR_\alpha(G) = G^{-1}(\alpha), \quad 0 < \alpha < 1,$$

sa nazýva *Value at risk*.

O funkcii G predpokladáme, že je distribučnou funkciou výnosovej funkcie Y . Vtedy nám VaR_α hovorí akú najväčšiu stratu by sme nemali prekročiť s danou pravdepodobnosťou α . Výhodou VaR je jeho veľmi ľahká interpretovateľnosť. Medzi nevýhody miery VaR patrí napr. fakt, že VaR neposkytuje žiadnu informáciu o veľkosti strát, ktoré nastávajú s pravdepodobnosťou menšou ako α , VaR nie je subaditívna miera, výskyt viacnásobných extrémov a náročný numerický výpočet. Tieto nevýhody a mnoho ďalších boli podnetom k vzniku CVaR. Hlbšia analýza výhod a nevýhod miery VaR ako aj vzájomné porovnanie mier VaR a CVaR je uvedené v [27] a [28].

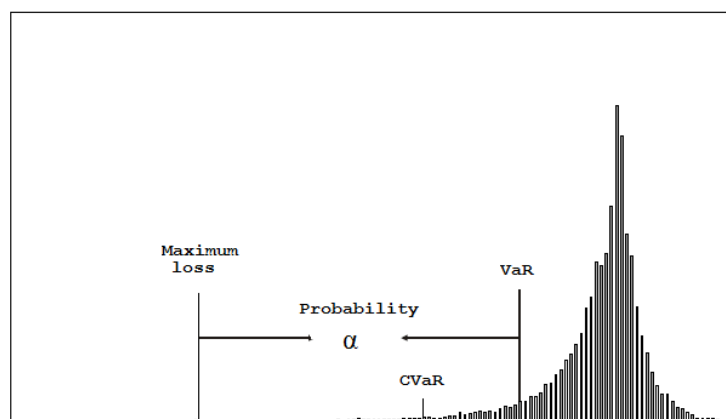
Definícia ([22]) *Conditional Value-at-Risk*

Conditional Value-at-Risk na hladine významnosti α je definovaná ako:

$$CVaR_\alpha(Y) = CVaR_\alpha(G) = \frac{1}{\alpha} \int_0^\alpha G^{-1}(\alpha), \quad 0 < \alpha < 1$$

kde funkcia G je distribučnou funkciou náhodnej premennej Y .

O funkcii G opäť predpokladáme, že je distribučnou funkciou výnosovej funkcie Y . Ako vidieť z definície, $CVaR_\alpha$ hovorí, akú stratu máme očakávať, ak strata prekročí hodnotu VaR_α . $CVaR_\alpha$ spĺňa ekvivarianciu na posun, konkávnosť, monotónnosť, pozitívnu homogénnosť, subaditivitu a poskytuje informáciu o veľkosti strát za VaR_α . Značná časť nedostatkov miery VaR_α je zavedením miery $CVaR_\alpha$ odstránená. $CVaR_\alpha$ sa zvykne nazývať aj Average Value at Risk ([20]), Expected shortfall ([1]) a Tail Value at Risk ([7]). Pre $CVaR_\alpha$ a VaR_α platí vzťah uvádzaný v [20]: $CVaR_\alpha < VaR_\alpha$, ktorý je zobrazený na Obr. 1.1. Hlavnou výhodou miery $CVaR$ je, že môže byť vyjadrená ako optimálna hodnota maximalizačnej úlohy, čo hovorí aj Veta 1.



Obr. 1.1: Vzťah VaR a CVaR

Veta 1 ([22])

Conditional Value-at-Risk môže byť vyjadrená ako optimálna hodnota nasledujúcej optimalizačnej úlohy:

$$CVaR_\alpha(Y) = \max\left\{a - \frac{1}{\alpha}\mathbb{E}(Y - a)^- : a \in \mathbf{R}\right\}, \text{ kde } f^- = -\min(f, 0).$$

Podotýkame, že optimálna hodnota parametera a zodpovedá hodnote VaR_α . Takýto prepis prináša z hľadiska optimalizácie portfólia výrazné zjednodušenie, keďže v prípade spojitého rozdelenia ide o úlohu konvexného programovania a dokonca v prípade diskrétného rozdelenia vektora výnosov sa jedná o úlohu lineárneho programovania. Ako už bolo uvedené, riziko portfólia bude vyjadrené pomocou funkcionálu \mathcal{D} , ktorý bude v našom prípade definovaný ako:

$$\begin{aligned}\mathcal{D}(x^T \xi) &= \mathbb{E}(x^T \xi) - CVaR_\alpha(x^T \xi) \\ &\Downarrow \\ \mathcal{D}(x^T \xi) &= \mathbb{E}(x^T \xi) - \max\{a - \frac{1}{\alpha} \mathbb{E}(x^T \xi - a)^- : a \in \mathbf{R}\}\end{aligned}$$

a nazýva sa *Conditional Value-at-Risk Deviation*. Pre účelovú funkciu dostávame:

$$\begin{aligned}\min_x \{ \mathbb{E}(x^T \xi) - \max_a \{ a - \frac{1}{\alpha} \mathbb{E}[(x^T \xi - a)^-] \} \} \\ \Downarrow \\ \min_{x,a} \{ \mathbb{E}(x^T \xi) - a + \frac{1}{\alpha} \mathbb{E}[(x^T \xi - a)^-] \}\end{aligned}$$

potom úloha (1.1) nadobúda tvar:

$$\begin{aligned}\min_{x,a} \quad & \mathbb{E}(x^T \xi) - a + \frac{1}{\alpha} \mathbb{E}[(x^T \xi - a)^-] \\ & \mathbb{E}(x^T \xi) \geq \mu, \\ & \mathbf{1}_M^T x \leq 1, \\ & (x \geq 0).\end{aligned}\tag{1.2}$$

Týmto sme ukončili formuláciu jedno-periódovej základnej úlohy optimalizácie portfólia. Úloha pozostáva z minimalizácie rizika vyjadreného pomocou miery Conditional Value-at-Risk deviation za podmienky dosiahnutia určitej úrovne strednej hodnoty portfólia, dodržania rozpočtu a prípadných ohraničení na váhy portfólia v prípade neuvažovania shortsellingu. V ďalšej časti naformulujeme viac-periódovú úlohu optimalizácie portfólia.

1.1.3 Viac-periódová základná úloha optimalizácie portfólia

Doposiaľ sme sa zaoberali úlohami optimalizácie portfólia, v ktorých sme uvažovali len jednu realizáciu náhodného vektora reprezentujúceho výnosy aktív v portfóliu, t.j. jednu časovú periódu a prislúchajúce rozhodnutie (x). V praxi však často vzniká potreba riešiť úlohy, v ktorých je potrebné urobiť rozhodnutie viackrát v priebehu sledovaného obdobia. Úlohami tohto typu sa budeme zaoberať v tejto kapitole, t.j. cieľom tejto kapitoly bude zovšeobecnenie jedno-periódovej úlohy optimalizácie portfólia na viac-periódovú úlohu optimalizácie portfólia.

Viac-periódová úloha optimalizácie portfólia pozostáva z $H > 1$ časových periód, ktorým prislúcha $H > 1$ realizácií náhodného vektora reprezentujúceho výnosy aktív. Na začiatku každej časovej periódy je potrebné urobiť rozhodnutie (zvoliť nové váhy v portfóliu) zodpovedajúce prehodnoteniu investičnej stratégie, na základe výnosov aktív v predchádzajúcich časových periódach. Na začiatok opäť uvedieme základné oz-

načenia a predpoklady, potom predstavíme viac-periódovú rizikovú mieru Conditional Value-at-Risk Deviation a jej vlastnosti a napokon naformulujeme viac-periódovú úlohu optimalizácie portfólia.

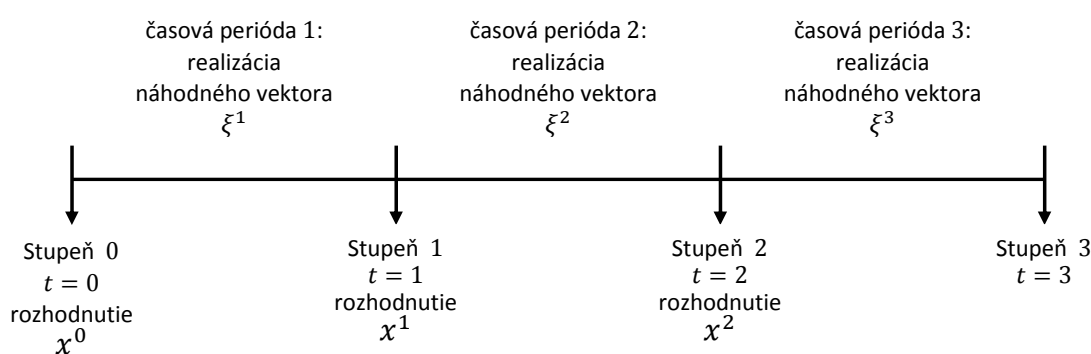
Predpoklady tejto úlohy optimalizácie portfólia sa v podstate nelíšia od predpokladov uvedených v podkapitole 1.1.1. Neuvažujeme transakčné náklady a váhy v portfóliu môžu byť záporné. Cieľom úlohy je v každej časovej perióde minimalizovať riziko pri podmienke dosiahnutia stanovenej úrovne očakávanej hodnoty portfólia na konci poslednej časovej periódy (na konci celého sledovaného obdobia). Počiatočný rozpočet opäť normalizujeme na 1 a v každej ďalšej perióde je rozpočet daný hodnotou portfólia na začiatku danej časovej periódy.

Základné označenia úlohy:

- celkové sledované obdobie rozdelíme na $H > 1$ časových periód. Bez ujmy na všeobecnosti predpokladáme rovnakú dĺžku pre všetky časové periody a budeme ich označovať, $t = 1, \dots, H$. Začiatky, respektíve konce časových periód sú reprezentované stupňami (časovými bodmi). Počet stupňov je $H + 1$, t.j. stupeň 0 zodpovedá začiatku sledovaného obdobia a stupeň H koncu. Časová perióda t je daná časovým intervalom medzi stupňami $t - 1$ a t ;
- $\xi^t = (\xi^{t,1}, \xi^{t,2}, \dots, \xi^{t,M})^T$ označuje náhodný vektor výnosov aktív v portfóliu v časovej perióde t , kde $\xi^{t,m}$, $m = 1, \dots, M$ reprezentuje výnos m -tého aktíva a M označuje celkový počet aktív v portfóliu;
- vektor $\xi_k^t = (\xi_k^{t,1}, \xi_k^{t,2}, \dots, \xi_k^{t,M})^T$ reprezentuje jednu realizáciu náhodného vektora ξ^t , ktorá nastáva s pravdepodobnosťou p_k^t ($\sum_{k=1}^{K^t} p_k^t = 1$, $p_k^t \geq 0$, $k = 1, \dots, K^t$), $k = 1, \dots, K^t$ a K^t označuje počet realizácií náhodného vektora ξ^t ;
- $\Xi^t \in \mathbf{R}^{K^t} \times \mathbf{R}^M$ - nosič náhodného vektora ξ^t , kde $\Xi_{k,m}^t = \xi_k^{t,m}$ a $k = 1, \dots, K^t$; $m = 1, \dots, M$; $t = 1, \dots, H$;
- $x^{t-1} \in \mathbf{R}^M$ - vektor váh v portfóliu, t.j. rozhodnutie na začiatku časovej periódy t ;
- $Y^t \in \mathbf{R}$ - výnos investície na konci časovej periódy t , $Y^t = (\xi^t)^T x^{t-1}$.

Všimnime si, že náhodný proces $\xi = (\xi^1, \dots, \xi^H)$, reprezentujúci vývoj výnosov aktív počas celého sledovaného obdobia, a rozhodovací proces $x = (x^0, \dots, x^{H-1})$ majú indexovanie posunuté o jedna. Je to spôsobené tým, že na začiatku prvej časovej periódy je potrebné urobiť rozhodnutie bez informácie o realizácii výnosov aktív. Na druhej strane, na konci poslednej časovej periódy už nie je potrebné žiadne rozhodnutie, ale ešte nastane posledná realizácia výnosov aktív, ktorá určí konečnú hodnotu portfólia. Celkový proces optimalizácie pre $H = 3$ je znázornený na Obr. 1.2.

Ako vidieť na Obr. 1.2, postupnosť jednotlivých krokov (udalostí) optimalizácie sa dá popísať nasledovne. Rozhodnutie x^0 , robí sa na začiatku prvej časovej periódy,

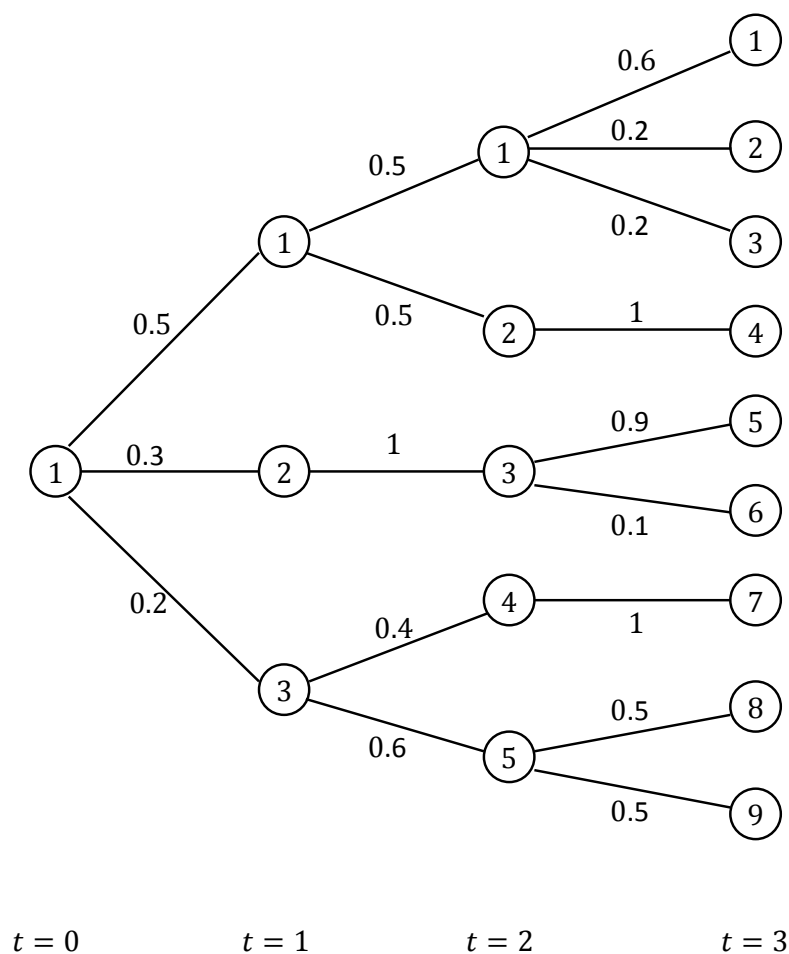
Obr. 1.2: Proces optimalizácie pre $H = 3$

sa robí bez prítomnosti akejkoľvek informácie o realizácii výnosov aktív. Nasleduje realizácia náhodného vektora ξ^1 , ktorá určí výnos zvolenej investície Y^1 a taktiež aj obnos prostriedkov dostupných pre investovanie v druhej časovej perióde $(1 + Y^1)$. Následné rozhodnutie x^1 sa už robí za prítomnosti informácie o realizácii náhodného vektora ξ^1 , po ktorom opäť nastáva realizácia náhodného vektora ξ^2 určujúca výnos investície v perióde 2 aj rozpočet pre periódu 3 atď. Dôležité je si uvedomiť, že s postupujúcim časom rastie aj informačná hodnota, daná všetkými predchádzajúcimi realizáciami náhodných vektorov, dostupná pri určovaní investičnej stratégie.

Na zahrnutie informácie do úlohy optimalizácie portfólia je potrebné zdefinovať filtráciu \mathcal{F} . Filtrácia $\mathcal{F} = (\mathcal{F}^0, \mathcal{F}^1, \dots, \mathcal{F}^H)$ je rastúca postupnosť σ -algebier, pre ktoré platí $\mathcal{F}^t \subseteq \mathcal{F}^{t+1}$, t.j. σ - algebra \mathcal{F}^t zodpovedá dostupnej informácii v stupni t a \mathcal{F}^H zodpovedá informácii na konci poslednej časovej periódy (v stupni H). Úroveň informácie v časovej perióde $t + 1$ je aspoň taká ako v časovej perióde t .

V prípade konečného rozdelenia jednotlivých náhodných vektorov ξ^t sa dá náhodný proces ξ reprezentovať pomocou stromu scenárov. Strom scenárov pozostáva z $H + 1$ stupňov. Stupeň $t = 0$ označuje koreň stromu a ostatné stupne $t = 1, \dots, H$ reprezentujú konce jednotlivých časových periód. Uzly stromu v jednotlivých stupňoch t zodpovedajú realizáciám náhodného vektora ξ^t . Počet uzlov v stupni t sa rovná počtu realizácií náhodného vektora ξ^t , čiže K^t ($K^0 = 1$) a označujeme ich rovnako ako jednotlivé scenáre, t.j. $k \in K^t$. Každý uzol má práve jedného predchodcu, t.j. každý uzol v stupni t (okrem stupňa $t = 0$) je prepojený s jedným uzlom v stupni $t - 1$. Každý uzol v stupni $t = 0, \dots, H - 1$ má aspoň jedného nasledovníka, t.j. je prepojený s aspoň jedným uzlom v stupni $t + 1$. Na Obr. 1.3 je zobrazený príklad stromu scenárov pre $H = 3$.

Čísla na čiarach medzi uzlami zodpovedajú podmieneným pravdepodobnostiam jednotlivých realizácií. Každá cesta od koreňa k uzlom v stupni H , ktoré sa nazývajú aj listy stromu, reprezentuje jeden konkrétny priebeh vývoja výnosov aktív v portfóliu. Jednotlivé priebehy sa nazývajú scenáre. Všimnime si rozdiel medzi scenármi v jedno-periódovej úlohe, kde scenárom nazývame jednu realizáciu náhodného vektora



Obr. 1.3: Strom scenárov

a scenármi vo viac-periódovej úlohe, kde scenárom nazývame jeden priebeh náhodného procesu. Krúžky s číslami označujú jednotlivé uzly (realizácie) v príslušnom stupni. Na príklade stromu z Obr. 1.3 si pre lepšie porozumenie popíšeme jednotlivé σ -algebry zodpovedajúce úrovniám informácie dostupnej v prislúchajúcich časových periódach.

- \mathcal{F}^0 je tzv. triviálna σ -algebra $\mathcal{F}^0 = \{0, \Omega\}$, čo znamená, že k dispozícii nemáme žiadnu informáciu,
- \mathcal{F}^1 je generovaná množinami: $\{1, 2, 3, 4\}$, $\{5, 6\}$, $\{7, 8, 9\}$, t.j. k dispozícii máme informáciu o realizácii výnosov v prvej časovej perióde,
- \mathcal{F}^2 je generovaná množinami: $\{1, 2, 3\}$, $\{4\}$, $\{5, 6\}$, $\{7\}$, $\{8, 9\}$, t.j. popri informácii z \mathcal{F}^1 máme k dispozícii aj informáciu o realizácii výnosov v druhej časovej perióde,
- \mathcal{F}^3 je generovaná množinami: $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{6\}$, $\{7\}$, $\{8\}$, $\{9\}$, t.j. máme k dispozícii kompletnú informáciu.

V ďalšom texte zdefinujeme rizikové miery viac-periódovú Conditional Value-at-Risk a viac-periódovú Conditional Value-at-Risk Deviation a ich vlastnosti.

Definícia ([20]) *Multi-period acceptability functional*

Viac-periódový funkcionál $\mathcal{A}(Y; \mathcal{F})$ sa nazýva *multi-period acceptability functional*, ak platí $\mathcal{A}(Y) < \infty$ pre všetky Y a $\mathcal{A}(Y) > -\infty$ pre nejaké Y a spĺňa nasledovné vlastnosti:

- *Informačná monotónnosť:* Ak $\mathcal{F}^t \subseteq \mathcal{F}^t$ pre všetky t , potom

$$\mathcal{A}(Y^1, \dots, Y^H; \mathcal{F}^0, \dots, \mathcal{F}^{H-1}) \leq \mathcal{A}(Y^1, \dots, Y^H; \mathcal{F}_t^0, \dots, \mathcal{F}_t^{H-1}).$$

- *Predvídateľná ekvivariancia vzhľadom na posun:*

$$\mathcal{A}(Y^1, \dots, Y^t + C^t, \dots, Y^H; \mathcal{F}) \leq \mathbb{E}(C^t) + \mathcal{A}(Y^1, \dots, Y^H; \mathcal{F})$$

pre všetky \mathcal{F}^{t-1} merateľné funkcie C^t .

- *Konkávnosť:* Zobrazenie $(Y^1, \dots, Y^H) \mapsto \mathcal{A}(Y^1, \dots, Y^H; \mathcal{F})$ je konkávne.
- *Monotónnosť:*

$$\forall t \ Y^t \leq \tilde{Y}^t \Rightarrow \mathcal{A}(Y^1, \dots, Y^H; \mathcal{F}) \leq \mathcal{A}(\tilde{Y}^1, \dots, \tilde{Y}^H; \mathcal{F}).$$

Definícia ([20]) *Viac-periódová Conditional Value-at-Risk*

Nech $Y = (Y^1, \dots, Y^H)$ je náhodný proces. Potom pre dané postupnosti konštánt $c = (c^1, \dots, c^H)$, pravdepodobností $\alpha = (\alpha^1, \dots, \alpha^H)$ a filtráciu $\mathcal{F} = (\mathcal{F}^0, \dots, \mathcal{F}^{H-1})$ je Viac-periódová Conditional-Value-at-Risk definovaná ako:

$$CVaR_{c,\alpha}(Y, \mathcal{F}) = \sum_{t=1}^H c^t \mathbb{E} \left(CVaR_{\alpha^t}(Y^t, \mathcal{F}^{t-1}) \right).$$

Miera $CVaR_{c,\alpha}(Y, \mathcal{F})$ spĺňa vlastnosti uvedené v predchádzajúcej definícii a navyše spĺňa vlastnosť pozitívnej homogenity, t.j.

$$\mathcal{A}(\lambda Y^1, \dots, \lambda Y^H; \mathcal{F}) = \lambda \mathcal{A}(Y^1, \dots, Y^H; \mathcal{F})$$

pre všetky $\lambda > 0$.

Definícia ([20]) *Multi-period deviation risk functional*

Viac-periódový funkcionál $\mathcal{D}(Y; \mathcal{F})$ sa nazýva *multi-period deviation risk functional*, ak platí $\mathcal{D}(Y) > -\infty$ pre všetky Y a $\mathcal{D}(Y) < \infty$ pre nejaké Y a spĺňa nasledovné vlastnosti:

- *Informačná antimonotónnosť:* Ak $\mathcal{F}^t \subseteq \mathcal{F}_t^t$ pre všetky t , potom

$$\mathcal{D}(Y^1, \dots, Y^H; \mathcal{F}^0, \dots, \mathcal{F}^{H-1}) \geq \mathcal{D}(Y^1, \dots, Y^H; \mathcal{F}_t^0, \dots, \mathcal{F}_t^{H-1}).$$

- *Predvídateľná invariancia vzhľadom na posun:*

$$\mathcal{D}(Y^1, \dots, Y^t + C^t, \dots, Y^H; \mathcal{F}) \leq \mathcal{D}(Y^1, \dots, Y^H; \mathcal{F})$$

pre všetky \mathcal{F}^{t-1} merateľné funkcie C^t .

- *Konkávnosť:* Zobrazenie $(Y^1, \dots, Y^H) \mapsto \mathcal{D}(Y^1, \dots, Y^H; \mathcal{F})$ je konverzné.

Definícia ([20]) *Viac-periódová Conditional Value-at-Risk Deviation*

Nech $Y = (Y^1, \dots, Y^H)$ je náhodný proces. Potom pre dané postupnosti konštánt $c = (c^1, \dots, c^H)$, pravdepodobností $\alpha = (\alpha^1, \dots, \alpha^H)$ a filtráciu $\mathcal{F} = (\mathcal{F}^0, \dots, \mathcal{F}^{H-1})$ je *Viac-periódová Conditional-Value-at-Risk Deviation* definovaná ako:

$$\begin{aligned} CVaRD_{c,\alpha}(Y, \mathcal{F}) &= \sum_{t=1}^H c^t \mathbb{E} \left(CVaRD_{\alpha^t}(Y^t, \mathcal{F}^{t-1}) \right) \\ &= \sum_{t=1}^H c^t \mathbb{E} \left(\mathbb{E}(Y^t, \mathcal{F}^{t-1}) - CVaR_{\alpha^t}(Y^t, \mathcal{F}^{t-1}) \right). \end{aligned}$$

Riziková miera $CVaRD_{c,\alpha}(Y, \mathcal{F})$ spĺňa vlastnosti uvedené v predchádzajúcej definícii a navyše spĺňa vlastnosť pozitívnej homogenity, t.j.

$$\mathcal{D}(\lambda Y^1, \dots, \lambda Y^H; \mathcal{F}) = \lambda \mathcal{D}(Y^1, \dots, Y^H; \mathcal{F})$$

pre všetky $\lambda > 0$.

Ako vidieť z definície, minimalizáciou $CVaRD_{c,\alpha}(Y, \mathcal{F})$ v každej časovej perióde minimalizujeme očakávanú hodnotu rizika investície, pričom máme k dispozícii informáciu z predchádzajúcich časových období. Pod viac-periódovou úlohou optimalizácie portfólia budeme podľa [15] rozumieť úlohu v tvare:

$$\begin{aligned} \min_{x^0, \dots, x^{H-1}} \sum_{t=1}^H c^t \mathbb{E} \left[CVaRD_{\alpha^t} \left((\mathbf{1} + \xi^t)^T x^{t-1}, \mathcal{F}^{t-1} \right) \right] \\ \mathbb{E} \left[(\mathbf{1} + \xi^H)^T x^{H-1}, \mathcal{F}^{H-1} \right] \geq \mu, \\ \mathbf{1}^T x^0 = 1, \\ \mathbf{1}^T x^t = (\mathbf{1} + \xi^t)^T x^{t-1}, \quad t = 1, \dots, H-1, \\ (x^t \geq 0, \quad t = 0, \dots, H-1), \end{aligned} \tag{1.3}$$

kde c^t je diskontný faktor v časovej perióde t .

Týmto sme naformulovali viac-periódovú úlohu optimalizácie portfólia, ktorej cieľom je minimalizovať riziko investície v každej časovej perióde za podmienky dosiahnutia stanovenej úrovne strednej hodnoty portfólia na konci sledovaného obdobia a dodržania rozpočtových ohraničení, ktoré sú dané realizáciou náhodného vektora výnosov. V prípade nedovolenia shortsellingu je potrebné dodržať aj nezápornosť váh aktív v portfóliu. V ďalšej podkapitole ozrejníme motiváciu tejto práce a uvedieme prvotné očakávania ohľadom efektívnosti použitia jednotlivých algoritmov, ktorých popis je náplňou Kapitoly 3.

1.2 Motivácia použitia dekompozičných algoritmov

V tejto časti by sme chceli ozrejmiť dôvody, prečo sme sa rozhodli testovať dekompozičné metódy na úlohách z predchádzajúcich častí. Následne uvedieme, aké výsledky očakávame pri použití jednotlivých metód, so zameraním hlavne na efektívnosť použitia (výpočtový čas a pamäťové nároky).

Za predpokladu o diskrétnom rozdelení vektora výnosov, pridaním pomocnej premennej $z_k := \max(a - x^T \xi_k, 0)$ a s použitím $f^- = -\min(f, 0) = \max(-f, 0)$ môžeme písať:

$$\mathbb{E}[(x^T \xi - a)^-] = \sum_{k=1}^K p_k \max(a - x^T \xi_k, 0), \quad (1.4)$$

pričom

$$\max(a - x^T \xi_k, 0) = \begin{cases} a - x^T \xi_k & \text{ak } a > x^T \xi_k, \text{ t.j. } z_k \geq a - x^T \xi_k, \\ 0 & \text{ak } a \leq x^T \xi_k, \text{ t.j. } z_k \geq 0. \end{cases} \quad (1.5)$$

Úloha (1.2) sa teda dá prepísať do tvaru úlohy lineárneho programovania:

$$\begin{aligned} \min_{x, a, z} \quad & x^T r - a + \frac{1}{\alpha} p^T z \\ & -a + x^T \xi_k + z_k \geq 0, \quad k = 1, \dots, K, \\ & x^T r \geq \mu, \\ & \mathbf{1}_M^T x \leq 1, \\ & z \geq 0, (x \geq 0). \end{aligned} \quad (1.6)$$

Matica ohraničení úlohy (1.6) má rozmery $(K+2) \times (M+K+1)$ a tak pre veľký počet scenárov nadobúda úloha v tvare (1.6) obrovské rozmery. Riešenie takto veľkých úloh priamou metódou (napr. simplexová metóda alebo metóda vnútorného bodu) sa stáva neefektívnym a pamäťovo veľmi náročným. Napríklad, v prípade $M = 10$ a $K = 10000$ má matica ohraničení úlohy (1.6) rozmery $(10002) \times (10011)$. Na uloženie takejto matice je potrebná pamäť 800.72 MB.

V prípade viac-periódovej úlohy optimalizácie portfólia sa problém s veľkosťou matice ohraničení stáva ešte výraznejším. S využitím predpokladu o diskretnom rozdelení náhodných vektorov ξ^t , použitím reprezentácie CVaR z Vety 1 a použitím podobného prepisu s premennou z ako v prípade úlohy (1.6) sa dá úloha (1.3) vyjadriť v tvare úlohy deterministického lineárneho programovania:

$$\begin{aligned} \min_{x,a,z} \quad & \sum_{t=0}^{H-1} \sum_{k=1}^{K^t} c^t p_k^t \left(\sum_{j \in \{k\}^+} \left(pc(j) (\mathbf{1} + \xi_j^{t+1})^T x_k^t \right) - a_k^t + \frac{1}{\alpha^t} \sum_{j \in \{k\}^+} pc(j) z_j^{t+1} \right) \\ & -a_k^t + (\mathbf{1} + \xi_j^{t+1})^T x_k^t + z_j^{t+1} \geq 0, \quad k = 1, \dots, K^t, \quad t = 0, \dots, H-1, \quad j \in \{k\}^+, \\ & z_j^{t+1} \geq 0, \quad k = 1, \dots, K^t, \quad t = 0, \dots, H-1, \quad j \in \{k\}^+, \\ & \sum_{k=1}^{K^{H-1}} \sum_{j \in \{k\}^+} p_j^H (\mathbf{1} + \xi_j^H)^T x_k^{H-1} \geq \mu, \\ & \mathbf{1}_M^T x^0 = 1, \\ & (\mathbf{1}_M + \xi_j^t)^T x_k^{t-1} = \mathbf{1}_M^T x_j^t, \quad k = 1, \dots, K^t, \quad t = 1, \dots, H-1, \quad j \in \{k\}^+, \\ & (x^t \geq 0, \quad t = 1, \dots, H-1). \end{aligned} \tag{1.7}$$

kde $pc(j)$ sú definované ako:

$$pc(j) = \frac{p(j)}{\sum_{l \in \{k\}^+} p(l)}, \quad k = 1, \dots, K^t, \quad t = 0, \dots, H-1, \quad j \in \{k\}^+,$$

a nazývajú sa podmienené pravdepodobnosti, t.j. sú to pravdepodobnosti nastatia realizácie j ak v predchádzajúcej časovej perióde nastala realizácia k . Symbolom $\{k\}^+$ sme označili množinu nasledovníkov uzla k v danom stupni t . Vektor x_k^t zodpovedá rozhodnutiu v uzli k a stupni t , vektor ξ_j^{t+1} reprezentuje j -tu realizáciu náhodného vektora ξ^{t+1} a premenné a_k^t a $z_j^{t+1} = \max(a_k^t - (\mathbf{1} + \xi_j^{t+1})^T x_k^t, 0)$, $j \in \{k\}^+$ zodpovedajú prepisu $CVaRD_{\alpha^t}((\mathbf{1} + \xi^{t+1})^T x_k^t)$ v uzli k a stupni t podľa Vety 1. Matica ohraničení v tomto prípade nadobúda rozmery:

$$\left(1 + K^H + 2 \sum_{t=1}^{H-1} K^t \right) \times \left(M + 1 + K^H + (M + 2) \sum_{t=1}^{H-1} K^t \right).$$

V prípade stromu scenárov zobrazenom na Obr. 1.3 a pri uvažovaní $M = 5$ sú rozmery matice ohraničení 26×71 . Napríklad, pri uvažovaní $T = 10$, $M = 5$ a len troch možných realizácií v každom uzle, úloha nadobúda rozmery 118096×265716 a na uloženie takto veľkej matice je potrebná pamäť 250 TB.

Práve z tohto dôvodu sme sa rozhodli implementovať a následne testovať dekom-

pozičné metódy, ktoré, ako sa dozvieme neskôr, využívajú špeciálnu štruktúru matice ohraničení úloh typu (1.6). Použitie metód tohto typu by mohlo znamenať zefektívnenie optimalizácie úlohy (1.6). Konkrétne budeme uvažovať metódy založené na Bendersovej dekompozícii, a to jednorezovú, viacrezovú a regularizovanú verziu L-shaped metódy pre dvoj-stupňové stochastické lineárne úlohy s pevnou kompenzáciou uvádzané v [6]. Ďalej to bude metóda vyvinutá špeciálne na riešenie optimalizačných úloh s Conditional Value-at-Risk v účelovej funkcii. Metóda sa nazýva CVaRmin metóda a prvýkrát bola uvedená v [2]. Metóda CVaRmin je odvodená z jednorezovej L-shaped metódy pre dvoj-stupňové stochastické lineárne úlohy. V literatúre sme nenašli žiadnu inú verziu CVaRmin metódy, a preto sme rozhodli odvodiť a implementovať aj viacrezovú verziu CVaRmin metódy, čo považujeme za jeden z vlastných prínosov tejto práce. V prípade viac-stupňových úloh optimalizácie portfólia budeme môcť použiť len dekompozičné metódy, nakoľko sú pamäťové nároky pri použití priamych metód (simplexová metóda, metóda vnútorného bodu) príliš vysoké. Konkrétne použijeme jednorezovú a viacrezovú vnorenú L-shaped metódu pre viac-stupňové stochastické lineárne úlohy s pevnou kompenzáciou uvádzané taktiež v [6]. Tieto metódy modifikujeme, použijúc podobné princípy, aké boli použité pri odvodení CVaRmin metódy, čím by sme mohli dosiahnuť zefektívnenie vnorených verzií L-shaped metódy pre riešenie úloh z prvej kapitoly.

Z numerických výsledkov očakávame vyššiu efektívnosť dekompozičných metód v porovnaní s priamymi metódami. Taktiež očakávame dominanciu jednotlivých verzií CVaRmin metódy nad verziami L-shaped metódy, pretože ako sme uviedli, CVaRmin metóda je určená špeciálne na riešenie úloh typu (1.6) a L-Shaped metóda je všeobecnejšou metódou vhodnou na použitie pre širšie spektrum úloh. V prípade porovnania jednotlivých verzií (CVaRmin aj L-Shaped) je veľmi ťažko povedať, ktorá verzia bude najefektívnejšia, pretože efektívnosť jednotlivých verzií môže závisieť na konkrétnom charaktere úlohy. V prípade viac-periódových úloh optimalizácie portfólia očakávame vyššiu efektívnosť modifikovaných metód.

Kapitola 2

Stochastické lineárne úlohy

Jedno-periódová základná úloha optimalizácie portfólia zodpovedá dvoj-stupňovej stochastickej lineárnej úlohe s pevnou kompenzáciou a úloha v tvare viac-periódovej základnej úlohy optimalizácie portfólia bude zodpovedať tvaru viac-stupňovej stochastickej lineárnej úlohy s pevnou kompenzáciou. Náplňou tejto kapitoly bude definícia týchto typov úloh vo všeobecnosti a uvedením základných vlastností, ktoré budú neskôr potrebné ku konštrukcii dekompozičných algoritmov. Kapitola je rozdelená na dve časti. V prvej sa budeme venovať odvodu dvoj-stupňových stochastických lineárnych úloh. Náplňou druhej časti budú viac-stupňové stochastické lineárne úlohy. Na konci každej časti uvedieme prepis príslušnej úlohy optimalizácie portfólia z Kapitoly 1.

2.1 Dvoj-stupňové stochastické lineárne úlohy

Stochastické lineárne úlohy (SLÚ) sú optimalizačné úlohy vedúce na úlohy lineárneho programovania, v ktorých je potreba uvažovať dáta (vstupné údaje) stochastického charakteru. Takéto typy úloh majú použitie takmer v každej oblasti vedy. Ich škála použitia siaha od oblasti telekomunikácie a medicíny až po oblasť financií. V tejto časti zdefinujeme tzv. dvojstupňové SLÚ. Už z názvu sa dá vytušiť, že sa bude jednať o úlohy, ktorých proces optimalizácie pozostáva z dvoch častí (stupňov). V prvom rade si uvedieme odvodenie týchto úloh, ktoré bolo podnietené zvyšujúcou sa potrebou uvažovať dáta stochastického charakteru. Neskôr uvedieme aj základné vlastnosti týchto úloh, ktoré budú potrebné v Kapitole 3 pri konštrukcii algoritmov.

2.1.1 Odvodenie Dvojstupňových SLÚ

Deterministickú lineárnu úlohu môžeme podľa [6] zdefinovať v tvare:

$$\begin{aligned} \min_x z &= c^T x \\ Ax &= b, \\ x &\geq 0, \end{aligned} \tag{2.1}$$

kde $x \in \mathbf{R}^n$ označuje vektor rozhodovacích premenných, $c \in \mathbf{R}^n$ účelový gradient, $A \in \mathbf{R}^{m \times n}$ maticu koeficientov ohraničení a $b \in \mathbf{R}^m$ vektor pravých strán ohraničení. Hodnota $z = c^T x$ zodpovedá účelovej funkcii a množina prípustných riešení je definovaná ako $\{x | Ax = b, x \geq 0\}$. Optimálne riešenie x^* patrí do množiny prípustných riešení a splňa $c^T x \geq c^T x^*$ pre všetky prípustné x . Poznamenávame, že každú úlohu lineárneho programovania vieme prepísať na úlohu v tvare (2.1). Využívame poznatok, že maximalizačná úloha sa dá previesť na minimalizačnú zavedením substitúcie, $\max_x c^T x = \min_x -c^T x$. Ohraničenia typu nerovnosti vieme prepísať na ohraničenia typu rovnosti pridaním doplnkovej nezápornej premennej. Odčítaním tejto premennej vieme urobiť opačný prepis. Prípadné ohraničenia na premenné x vieme, pomocou substitúcie $x = x^+ - x^-$ kde $x^+, x^- \in \mathbf{R}_+^n$ a doplnkových nezáporných premenných, opäť previesť na ohraničenia typu rovnosti. Teda úlohu (2.1) môžeme považovať za všeobecnú formuláciu úlohy lineárneho programovania. Viac o tomto prepise a lineárnych úlohách sa dá nájsť napr. v [21].

Takto definované úlohy predstavujú vhodný model pre širokú škálu optimalizačných problémov. Avšak predpokladom úloh definovaných v tvare (2.1) je nestochastickosť alebo fixnosť vstupných údajov. V reálnych optimalizačných problémoch sa však veľmi často nachádzajú náhodné, stochastické vstupné údaje. Náhodnosť bude v práci reprezentovaná náhodným vektorom ξ . Predpokladá sa, že pravdepodobnostné rozdelenie vektora ξ poznáme a jeho parametre možno odhadnúť z historických dát, ktoré máme k dispozícii. Ako príklad takýchto úloh môžeme uviesť príklad o farmárovi, uvádzaný v [6], kde je potrebné určiť, koľko a akej obilniny zasiať, aby sme maximalizovali svoj zisk, pričom veľkosť úrody je náhodná, daná počasím. Ďalším príkladom môže byť úloha, ktorú rieši každý obchodník, a to aké množstvá tovaru nakúpiť tak, aby maximalizoval zisk, pričom dopyt zákazníkov je náhodný. Posledným uvedeným príkladom môže byť určenie optimálnej investičnej stratégie, pričom výnosy jednotlivých finančných derivátov sú náhodné. Lineárne úlohy, v ktorých sú prítomné náhodné údaje sa nazývajú stochastické lineárne úlohy. SLÚ, v ktorých je po realizácii náhodných udalostí možné urobiť nejaké dodatočné rozhodnutia, sa nazývajú SLÚ s kompenzáciou. V takto definovanej úlohe môžeme rozhodovacie premenné rozdeliť do dvoch skupín:

- *Prvostupňové premenné* - sú to premenné, ktoré je potrebné určiť ešte pred realizáciou náhodného vektora;
- *Druhostupňové premenné* - reprezentujú rozhodnutia, ktoré je možné urobiť po realizácii náhodného vektora a tak dodatočne ovplyvniť výstup.

Prvostupňové premenné budú reprezentované vektorom x , zatiaľ čo druhostupňové premenné budú reprezentované vektorom y ($y(\omega)$ alebo $y(\omega, x)$), kde ω označuje náhodné udalosti určujúce vektor ξ . Postupnosť rozhodnutí a udalostí môžeme ilustrovať nasledovne:

$$x \rightarrow \xi(\omega) \rightarrow y(\omega, x).$$

2.1.2 Dvojstupňové stochastické lineárne úlohy s pevnou kompenzáciou

Stochastickú lineárnu úlohu s pevnou kompenzáciou môžeme podľa [6] zdefinovať v tvare:

$$\begin{aligned} \min_x z &= c^T x + \mathbb{E}_\xi[\min_{y(\omega)} q(\omega)^T y(\omega)] \\ Ax &= b, \\ Wy(\omega) &= h(\omega) - T(\omega)x, \\ x &\geq 0, \quad y(\omega) \geq 0, \end{aligned} \tag{2.2}$$

pričom vektor $x \in \mathbf{R}^{n_1}$ ako aj $c \in \mathbf{R}^{n_1}$, $A \in \mathbf{R}^{m_1 \times n_1}$ a $b \in \mathbf{R}^{m_1}$ boli popísané v predchádzajúcej časti. Ďalej matica $W \in \mathbf{R}^{m_2 \times n_2}$ sa nazýva kompenzačná matica a budeme o nej predpokladať, že je nestochastická. Práve tento predpoklad má za následok to, že tento typ úloh nazývame úlohami s pevnou kompenzáciou. Stochastickosť úlohy je daná vektorom $\xi(\omega) = (q(\omega), h(\omega), T_1(\omega), \dots, T_{m_2}(\omega))$ s $N = n_2 + m_2 + (m_2 \times n_1)$ prvkami, kde $\forall \omega : q(\omega) \in \mathbf{R}^{n_2}$, $h(\omega) \in \mathbf{R}^{m_2}$, $T(\omega) \in \mathbf{R}^{m_2 \times n_2}$ a $T_i(\omega)$ predstavuje i -ty riadok matice $T(\omega)$. Matica $T(\omega)$ sa nazýva technologická matica a funkcia $q(\omega)^T y(\omega)$ predstavuje kompenzačné (penalizačné) náklady dané realizáciou vektora ξ . Ako je vidieť z úlohy (2.2), veľkosť penalizačných nákladov závisí na veľkosti rozdielu $h(\omega) - T(\omega)x$. Zavedenie penalizačných nákladov je úzko spojené s potrebou pridať do lineárnych úloh stochastické dáta. Ak uvažujeme stochastické dáta v úlohe lineárneho programovania, môžeme úlohu v tvare (2.1) podľa [14] prepísať do tvaru:

$$\begin{aligned} \min_x z &= c^T x \\ Ax &= b, \\ Tx &= h, \\ x &\geq 0, \end{aligned} \tag{2.3}$$

pričom $Ax = b$ zodpovedá nestochastickým ohraničeniam a $Tx = h$ stochastickým ohraničeniam. Keďže realizácia náhodných udalostí v čase rozhodovania o hodnote vektora x nie je známa, tak po realizácii náhodných udalostí takmer vždy platí $Tx \neq h$, a teda stochastické ohraničenia sú porušené. Práve z tohto dôvodu boli zavedené penalizácie za porušenie stochastických ohraničení, ktoré sú dané veľkosťou porušenia $h - Tx$. Veľkosť penalizačných nákladov pre konkrétnu realizáciu vektora ξ je optimálnou hodnotou úlohy

$$Q(x, \xi(\omega)) = \min_y \{q(\omega)^T y \mid Wy = h(\omega) - T(\omega)x, y \geq 0\}.$$

Úlohu (2.2) možno zdefinovať aj v odlišnom tvare, ktorý sa zvykne nazývať deterministický ekvivalent. Uvádzame ho podľa [6]:

$$\begin{aligned} \min_x z &= c^T x + Q(x) \\ Ax &= b, \\ x &\geq 0, \end{aligned} \tag{2.4}$$

kde

$$Q(x) = \mathbb{E}_\xi Q(x, \xi(\omega)), \tag{2.5}$$

pričom

$$Q(x, \xi(\omega)) = \min_y \{q(\omega)^T y \mid Wy = h(\omega) - T(\omega)x, y \geq 0\}. \tag{2.6}$$

Ak pre nejaké ω a x je úloha $Q(x, \xi(\omega))$ neprípustná, definujeme $Q(x, \xi(\omega)) = \infty$. Tiež môže nastať situácia, že pre nejaké ω a x je úloha $Q(x, \xi(\omega))$ zdola neohraničená. V takom prípade definujeme $Q(x, \xi(\omega)) = -\infty$, avšak poznamenávame, že sa jedná o zle definovanú úlohu, ktorej účelová funkcia sa môže ľubovoľne znižovať. Takéto úlohy nebudeme v tejto práci uvažovať. Na deterministickom ekvivalente je dobre viditeľné rozdelenie úlohy (2.2) na dva stupne a taktiež aj postupnosť udalostí uvedená v predchádzajúcej časti. Úloha (2.4) reprezentuje prvý stupeň optimalizácie, ktorá sa robí za prítomnosti neistoty ohľadom realizácie náhodného vektora ξ , čiže v prvom stupni robíme rozhodnutie o premenných x ešte pred realizáciou náhodného vektora. Túto neistotu reprezentuje funkcia $Q(x)$, ktorá je vo všeobecnosti nelineárna a teda prvý stupeň optimalizácie je úlohou nelineárneho programovania. Zo vzťahu (2.5) je zrejmé, že funkcia $Q(x)$ reprezentuje očakávanú hodnotu penalizačných nákladov vzhľadom na vektor ξ . Následne, po realizácii náhodného vektora, máme možnosť urobiť korekčné rozhodnutia za účelom znížiť penalizačné náklady na minimálnu možnú mieru. Za týmto účelom sa rieši úloha (2.6), ktorá je už v tomto prípade nestochastická, daná konkrétnou realizáciou náhodných udalostí ω .

2.1.3 Vlastnosti dvojstupňových SLÚ s pevnou kompenzáciou

V tejto časti uvedieme základné vlastnosti dvojstupňových SLÚ s pevnou kompenzáciou, ktoré sú kľúčové z hľadiska použitia algoritmov uvedených v Kapitole 3. Taktiež uvedieme vety, z ktorých plynú dôležité výsledky ohľadom existencie a vlastností riešenia deterministického ekvivalentu (2.4) – (2.6). Odteraz budeme predpokladať diskkrétne rozdelenie náhodného vektora ξ , pretože v úlohe optimalizácie portfólia budeme uvažovať náhodný vektor s diskrétnym rozdelením. Predpoklad diskrétného náhodného vektora je používaný vo veľkom rozsahu v rôznych reálnych aplikáciách buď priamo, predpokladaním diskrétného rozdelenia, alebo diskretizáciou spojitých rozdelení. Z tohto dôvodu budú aj všetky závery uvedené v práci platné len pre diskkrétne náhodné vek-

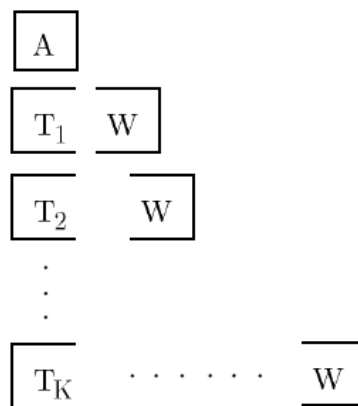
tory.

Nech nosič vektora ξ je definovaný ako $\Xi = \{\xi_k^T, k = 1, \dots, K\}$ kde číslom K budeme označovať počet realizácií vektora ξ . ξ_k predstavuje k -tu realizáciu náhodného vektora ξ , ktorá bude nastávať s pravdepodobnosťou p_k ($p_k > 0$ a $\sum_{k=1}^K p_k = 1$). V prípade diskrétného rozdelenia vektora ξ je možné úlohu (2.4) – (2.6) vyjadriť v tvare, ktorý sa nazýva *Extenzívna forma*:

$$\begin{aligned} \min_{x, y_k} z &= c^T x + \sum_{k=1}^K p_k q_k^T y_k \\ Ax &= b, \\ W y_k &= h_k - T_k x, \quad k = 1, \dots, K, \\ x &\geq 0, \quad y_k \geq 0, \quad k = 1, \dots, K. \end{aligned} \tag{2.7}$$

Celá náhodnosť úlohy (2.2) je v úlohe (2.7) explicitne rozpísaná, čím dostávame úlohu deterministického lineárneho programovania. Úlohy tohto typu sú pomerne ľahko riešiteľné, keďže sa jedná o úlohu lineárneho programovania avšak pri veľkom počte realizácií dosahuje úloha obrovských rozmerov. Práve táto skutočnosť bola podnetom pre odvodenie tzv. dekompozičných metód, ktoré využívajú špeciálnu blokovú štruktúru matice ohraničení úlohy (2.7) zobrazenú na Obr. 2.1. Na riešenie takýchto úloh môžeme použiť:

- pivotné metódy - simplexová a duálna simplexová metóda,
- metódy vnútorného bodu,
- dekompozičné metódy - L-shaped metóda, dekompozičné metódy založené na metóde vnútorného bodu.



Obr. 2.1: Bloková štruktúra úlohy (2.7)

Ďalej si uvedieme základné vlastnosti SLÚ s pevnou kompenzáciou. Konkrétne sa zameriame na množiny prípustnosti, účelovú funkciu a podmienky optimality uvádzané

v [6]. Nech $K_1 = \{x | Ax = b, x \geq 0\}$ označuje množinu definovanú nestochastickými ohraničeniami úlohy (2.2). Túto množinu môžeme charakterizovať ako prvostupňovú množinu prípustnosti. Nech $K_2 = \bigcap_{\xi \in \Xi} K_2(\xi)$ reprezentuje druhostupňovú množinu prípustnosti, pričom $K_2(\xi) = \{x | \exists y \geq 0 : Wy = h(\omega) - T(\omega)x\}$ je tzv. elementárna množina prípustnosti pre ľubovoľné $\xi \in \Xi$. Všimnime si, že túto množinu tvoria všetky vektory $x \in K_1$ také, pre ktoré vieme nájsť prípustné y pre všetky realizácie ξ , a je definovaná ako:

$$K_2 = \{x | \forall \xi, \exists y \geq 0 : Wy = h(\omega) - T(\omega)x\}.$$

Veta 2 ([6], str. 86)

- (i) Pre všetky realizácie ξ je elementárna množina prípustnosti $K_2(\xi)$ uzavretý konvexný polyéder.
- (ii) Ak má vektor ξ konečné diskkrétne rozdelenie, potom množina K_2 je tiež uzavretý konvexný polyéder.

Dôkaz je uvedený v [6], str. 86.

Z uvedenej vety plynie, že množina druhostupňových prípustných riešení je uzavretý konvexný polyéder, čo je z hľadiska existencie optimálneho riešenia užitočný poznatok. Ďalej nás budú zaujímať vlastnosti účelovej funkcie úlohy (2.4). O vlastnostiach účelovej funkcie pre konečné diskkrétne rozdelenie vektora ξ hovorí nasledovná veta.

Veta 3 ([6], str. 89)

Pre všetky ξ je funkcia $Q(x, \xi(\omega))$

- (i) po častiach lineárna konvexná funkcia v T, h ;
- (ii) po častiach lineárna konkávna funkcia v q ;
- (iii) po častiach lineárna konvexná funkcia v x pre všetky $x \in K_1 \cap K_2$.

Ak má náhodný vektor ξ konečné diskkrétne rozdelenie, potom je funkcia $Q(x)$ po častiach lineárna a konvexná v x pre všetky $x \in K_1 \cap K_2$.

Dôkaz je uvedený v [6], str. 89.

Časť (iii) je obzvlášť dôležitá v praxi, a to nielen pre tvorbu efektívnych algoritmov, ale poskytuje aj dostatočné podmienky pre existenciu optimálneho riešenia, keďže sa jedná o úlohu s po častiach lineárnou a konvexnou funkciou v x definovanou na uzavretom konvexnom polyédri $K_1 \cap K_2$. Na záver ešte uvedieme základné typy pevných kompenzačných matíc. Vo všeobecnosti rozlišujeme tri typy úloh:

- úlohy s relatívne úplnou kompenzáciou - sú to úlohy, kde platí $K_1 \subset K_2$;

- úlohy s úplnou kompenzáciou - pri tomto type úloh musí platiť :

$$\forall t \in \mathbf{R}^{m_2} \exists y \geq 0 : Wy = t.$$

Poznamenávame, že pre takéto úlohy je množina K_2 totožná z množinou K_1 , pretože pre ľubovoľný vektor $x \in K_1$ vieme nájsť také y , pre ktoré platí horeuvedená podmienka. Táto vlastnosť je z výpočtového hľadiska veľmi užitočná, ako sa neskôr dozvieme v Kapitole 3;

- úlohy s jednoduchou kompenzáciou - tvar kompenzačnej matice $W = (I, -I)$, kde I je identická matica. Všimnime si, že aj úlohy tohto typu spĺňajú podmienku pre úlohy s úplnou kompenzáciou a teda tu tiež nastáva zhodnosť množín K_1 a K_2 .

Na určenie úloh s úplnou kompenzáciou slúži nasledovná lema uvedená v [14]. Túto lemu budeme potrebovať na určenie typu kompenzačnej matice v prípade viac-periódovej úlohy optimalizácie portfólia.

Lema ([14], str. 201)

Matica $W \in \mathbf{R}^{m_2 \times n_2}$ spĺňa podmienku úplnej kompenzácie práve vtedy, keď má hodnotu m_2 a pre ľubovoľnú množinu $\{W_{i_1}, W_{i_2}, \dots, W_{i_{m_2}}\}$ lineárne nezávislých stĺpcov W , sú lineárne ohraničenia

$$\begin{aligned} Wy &= 0, \\ y_k &\geq 1, \quad k = 1, \dots, m_2, \\ y &\geq 0. \end{aligned}$$

prípustné.

Na záver tejto časti naformulujeme jedno-periódovú základnú úlohu optimalizácie portfólia v tvare dvoj-stupňovej SLŮ s pevnou kompenzáciou. Premenné x a a z úlohy (1.6) sú prvostupňové rozhodovacie premenné a ohraničenia na minimálny výnos a rozpočet zodpovedajú prvostupňovým ohraničeniam. Prvostupňová úloha má tvar:

$$\begin{aligned} \min_{x,a} z &= r^T x - a + \frac{1}{\alpha} Q(x, a) \\ Ax &= b, \\ (x &\geq 0), \end{aligned} \tag{2.8}$$

kde $Q(x, a)$ je očakávaná hodnota rozdielu $(a - x^T \xi)^-$. Druhostupňová úloha nadobúda tvar:

$$\min_z \{z \mid -z + d = -a + x^T \xi, z \geq 0, d \geq 0\}, \tag{2.9}$$

kde d je pomocná premenná pri prepise ohraničení do tvaru rovností. Z (2.9) je vidieť,

že kompenzačná matica nadobúda tvar $W = (-1, 1)$, takže sa jedná o úlohu s jednoduchou kompenzáciou a z vlastností úloh tohto typu vieme, že každé prípustné riešenie prvostupňovej úlohy bude prípustné aj v úlohe druhostupňovej. Týmto sme ukončili teóriu o dvoj-stupňových SLÚ s pevnou kompenzáciou a v nasledujúcej podkapitole sa budeme venovať všeobecnejšiemu prístupu, a to viac-stupňovým SLÚ s pevnou kompenzáciou.

2.2 Viac-stupňové stochastické lineárne úlohy s pevnou kompenzáciou

Náplňou predchádzajúcej časti tejto kapitoly bol koncept stochastických lineárnych úloh, v ktorých proces rozhodovania pozostával z dvoch rozhodnutí (dva rozhodovacie stupne) a jednej realizácie náhodného vektora. V praxi sa však často stretávame s potrebou riešiť úlohy, v ktorých je potrebné uvažovať rozhodovací proces, ktorý sa vyvíja v čase a jednotlivé rozhodnutia môžu byť ovplyvnené predchádzajúcou realizáciou náhodného procesu. Na riešenie problémov tohto typu slúži koncept viac-stupňových stochastických lineárnych úloh s pevnou kompenzáciou, ktorými sa budeme zaoberať v tejto podkapitole.

Viac-stupňovú SLÚ s pevnou kompenzáciou budeme uvádzať podľa [6] v tvare:

$$\begin{aligned} \min_{y^0, \dots, y^H} (c^0)^T y^0 + \mathbb{E}_{\xi^1} \left[\min \left(c^1(\omega^1) \right)^T y^1 + \dots + \mathbb{E}_{\xi^H} \left[\min \left(c^H(\omega^H) \right)^T y^H \right] \dots \right] \\ W^0 y^0 &= h^0, \\ T^0(\omega^1) y^0 + W^1 y^1(\omega^1) &= h^1(\omega^1), \\ &\vdots \\ T^{H-1}(\omega^H) y^{H-1}(\omega^{H-1}) + W^H y^H(\omega^H) &= h^H(\omega^H), \\ y^0 \geq 0, \quad y^t(\omega^t) &\geq 0, \quad t = 1, \dots, H-1, \end{aligned} \tag{2.10}$$

kde vektor $c^0 \in \mathbf{R}^{n_0}$, vektor $h^0 \in \mathbf{R}^{m_0}$ a matice $W^t \in \mathbf{R}^{m_t \times n_t}$, $t = 0, \dots, H$ sú dané a nenáhodné. Všetky ostatné dáta úlohy (2.10) môžu byť náhodné a dané realizáciou náhodného vektora $\xi^t(\omega^t)^T = \left(c^t(\omega^t)^T, h^t(\omega^t)^T, T_1^{t-1}(\omega^t)^T, \dots, T_{m_t}^{t-1}(\omega^t)^T \right)$, ktorý má rozmery $N^t = n_t + m_t + (m_t \times n_{t-1})$ a je definovaný na pravdepodobnostnom priestore s filtráciou (Ω, \mathcal{F}, P) pre $t = 1, \dots, H$. Rovnako ako v prípade úlohy (1.3), $\xi = (\xi^1, \dots, \xi^H)$ označuje náhodný proces a $y = (y^0, \dots, y^H)$ reprezentuje rozhodovací proces, ktorého členy y^t môžu vo všeobecnosti závisieť od vývoja náhodného procesu do času t . Označme $\Xi^t \subset \mathbf{R}^{N^t}$ nosič náhodného vektora ξ^t .

Rovnako ako v prípade dvoj-stupňových SLÚ s pevnou kompenzáciou, zdefinujeme tvar deterministického ekvivalentu aj pre úlohu (2.10). Deterministický ekvivalent

definujeme podľa [6] v tvare rekurzcie. Majme terminálnu podmienku v tvare:

$$\begin{aligned} Q^H(y^{H-1}, \xi^H(\omega^H)) &= \min_{y^H} (c^H(\omega^H))^T y^H(\omega^H) \\ W^H y^H(\omega^H) &= h^H(\omega^H) - T^{H-1}(\omega^H) y^{H-1}, \\ y^H(\omega^H) &\geq 0. \end{aligned} \quad (2.11)$$

Následne môžeme pre $t = 1, \dots, H - 1$ definovať rekurziu:

$$\begin{aligned} Q^t(y^{t-1}, \xi^t(\omega^t)) &= \min_{y^t} (c^t(\omega^t))^T y^t(\omega^t) + Q^{t+1}(y^t) \\ W^t y^t(\omega^t) &= h^t(\omega^t) - T^{t-1}(\omega^t) y^{t-1}, \\ y^t(\omega^t) &\geq 0, \end{aligned} \quad (2.12)$$

kde

$$Q^{t+1}(y^t) = \mathbb{E}_{\xi^{t+1}} \left[Q^{t+1}(y^t, \xi^{t+1}(\omega)) \right].$$

Riešenie úlohy v tvare (2.10) je ekvivalentné s riešením úlohy:

$$\begin{aligned} \min_{y^0} (c^0)^T y^0 + Q^1(y^0) \\ W^0 y^0 &= h^0, \\ y^0 &\geq 0. \end{aligned} \quad (2.13)$$

Poznamenávame, že rekurzia (2.11)–(2.13) v podstate predstavuje dekompozíciu úlohy (2.10), ktorá je základom dekompozičných algoritmov. V prípade $H = 1$ dostávame úlohu analogickú s dvoj-stupňovou SLÚ s pevnou kompenzáciou. Pri predpoklade konečného diskretného rozdelenia náhodných vektorov, tvoriacich náhodný proces ξ , je možné deterministický ekvivalent (rekurziu) (2.11)–(2.13) prepísať v tvare extenzívnej formy. Prepis je analogický s prepisom úlohy (1.7) a tak ho tu nebudeme uvádzať, ale uvedieme špeciálnu blokovú štruktúru matice ohraničení extenzívnej formy.

Na Obr. 2.2 je zobrazená špeciálna bloková štruktúra úloh typu (2.11)–(2.13) na príklade stromu scenárov zobrazenom na Obr. 1.3. Túto blokovú štruktúru ako aj vlastnosti definované Vetou 4 sú veľmi dôležité pre použitie efektívnych dekompozičných algoritmov.

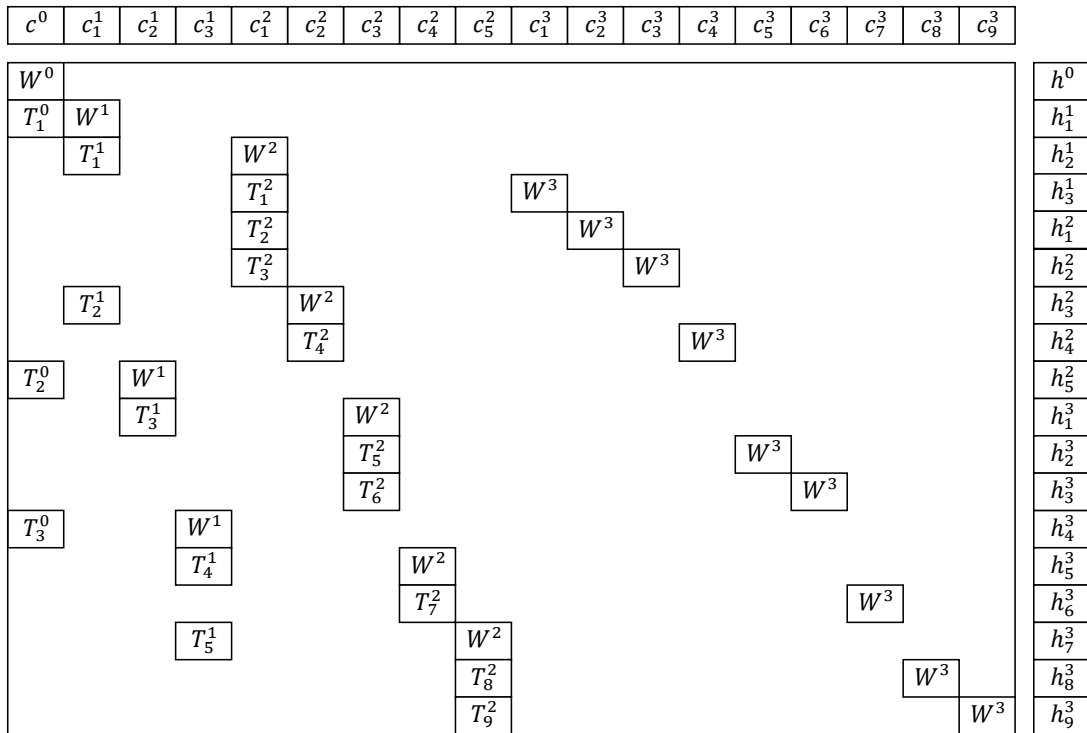
Definujme množiny prípustnosti podľa [6] ako:

$$K_t = \{y^t | Q^{t+1}(y^t) < \infty\}.$$

Veta 4 ([6], str. 129)

Množiny K_t a funkcie $Q^{t+1}(y^t)$ sú konvexné pre $t = 0, \dots, H - 1$ a ak Ξ^t je konečná pre $t = 1, \dots, H$, potom K_t a $Q^{t+1}(y^t)$ sú polyedrické.

Dôkaz je uvedený v [6].



Obr. 2.2: Bloková štruktúra úlohy (2.11)–(2.13)

Týmto sme ukončili teóriu viac-stupňových SLÚ s pevnou kompenzáciou a v ďalšej podkapitole uvedieme prepis viac-periódovej úlohy optimalizácie portfólia (1.3) uvedenej v podkapitole 1.1.3 na viac-stupňovú SLÚ s pevnou kompenzáciou.

2.2.1 Formulácia viac-periódovej úlohy optimalizácie portfólia v tvare viac-stupňovej SLÚ s pevnou kompenzáciou

Na to, aby sme mohli úlohu v tvare (1.3) preformulovať do tvaru rekurzcie (2.11)–(2.13), potrebujeme vedieť rozseparovať všetky ohraničenia úlohy (1.3) do tvaru $W^t y_k^t = h_k^t - T_k^{t-1} y_{\{k\}^-}^{t-1}$, $k = 1, \dots, K^t$, $t = 1, \dots, H$, kde $\{k\}^-$ označuje predchodcu uzla k a vektor y_k^t obsahuje všetky rozhodovacie premenné vystupujúce v príslušnom uzle, t.j. x_k^t, z_k^t, a_k^t . Ohraničenie na minimálnu úroveň očakávanej hodnoty portfólia zjavne takéto rozseparovanie znemožňuje a teda úlohu v tvare (1.3) nie je možné preformulovať do tvaru rekurzcie (2.11)–(2.13). Tomuto problému sa dá vyhnúť prepísaním úlohy (1.3) do mierne odlišného tvaru. Pri prepise využijeme vzťah uvádzaný v [20], ktorým

sa nám podarí zbaviť problémového ohraničenia. Jedná sa o vzťah medzi úlohami:

$$\begin{aligned} \min_x \quad & \mathcal{D}(x^T \xi) \\ & \mathbb{E}(x^T \xi) \geq \mu, \\ & \mathbf{1}_M^T x \leq 1, \\ & (x \geq 0) \end{aligned} \tag{2.14}$$

a

$$\begin{aligned} \max_x \quad & \mathbb{E}(x^T \xi) - \delta \mathcal{D}(x^T \xi) \\ & \mathbf{1}_M^T x \leq 1, \\ & (x \geq 0). \end{aligned} \tag{2.15}$$

Kde parameter δ reprezentuje averziu k riziku. Hodnota $\delta > 0$ zodpovedá riziko averznému investorovi, $\delta = 0$ riziko neutrálnemu investorovi a pri $\delta < 0$ sa jedná o riziko obľubujúceho investora. Vzťah medzi úlohami (2.14) a (2.15) je nasledovný:

- Nech $\delta > 0$. Ak x_δ je riešenie úlohy (2.15) a $\mu = \mathbb{E}(x_\delta^T \xi)$, potom x_δ je riešením aj úlohy (2.14).

Dôkaz tohto vzťahu je uvedený v [20]. Z tohto dostávame, že pri vhodne zvolenej hodnote parametra δ , bude riešenie úloh (2.16) a (2.17) totožné. V úlohách (2.16) a (2.17) sme pre zjednodušenie zápisu použili $\mathcal{D}(\cdot) = CVaRD_\alpha(\cdot)$ a $Y^t = (\mathbf{1} + \xi^t)^T x^{t-1}$.

$$\min_{x^0, \dots, x^{H-1}} \sum_{t=1}^H c^t \mathbb{E}[\mathcal{D}(Y^t, \mathcal{F}^{t-1})]$$

$$\begin{aligned} \mathbb{E}(Y^H, \mathcal{F}^{H-1}) & \geq \mu, \\ \mathbf{1}^T x^0 & = 1, \\ \mathbf{1}^T x^t & = (\mathbf{1} + \xi^t)^T x^{t-1}, \quad t = 1, \dots, H-1, \\ (x^t & \geq 0, \quad t = 0, \dots, H-1) \end{aligned} \tag{2.16}$$

$$\min_{x^0, \dots, x^{H-1}} \sum_{t=1}^{H-1} c^t \mathbb{E}[\mathcal{D}(Y^t, \mathcal{F}^{t-1})] + c^H \mathbb{E}[\delta \mathcal{D}(Y^H, \mathcal{F}^{H-1}) - \mathbb{E}(Y^H, \mathcal{F}^{H-1})]$$

$$\begin{aligned} \mathbf{1}^T x^0 & = 1, \\ \mathbf{1}^T x^t & = (\mathbf{1} + \xi^t)^T x^{t-1}, \quad t = 1, \dots, H-1, \\ (x^t & \geq 0, \quad t = 0, \dots, H-1) \end{aligned} \tag{2.17}$$

Úlohu (1.3) je možné prepísať na tvar

$$\begin{aligned}
 \min_{x,a,z} \quad & \sum_{t=0}^{H-2} \sum_{k=1}^{K^t} c^t p_k^t \left(\sum_{j \in \{k\}^+} (pc(j)(\mathbf{1} + \xi_j^{t+1})^T x_k^t) - a_k^t + \frac{1}{\alpha^t} \sum_{j \in \{k\}^+} pc(j) z_j^{t+1} \right) + \\
 & + \sum_{k=1}^{K^{H-1}} c^{H-1} p_k^{H-1} \left(\sum_{j \in \{k\}^+} (pc(j)(\delta - 1)(\mathbf{1} + \xi_j^H)^T x_k^{H-1}) - \delta a_k^{H-1} + \frac{\delta}{\alpha^{H-1}} \sum_{j \in \{k\}^+} pc(j) z_j^H \right) \\
 -a_k^t + (\mathbf{1} + \xi_j^{t+1})^T x_k^t + z_j^{t+1} \geq & 0, \quad k = 1, \dots, K^t, \quad t = 0, \dots, H-1, \quad j \in \{k\}^+, \\
 z_j^{t+1} \geq & 0, \quad k = 1, \dots, K^t, \quad t = 0, \dots, H-1, \quad j \in \{k\}^+ \quad (2.18) \\
 \mathbf{1}_M^T x^0 = & 1, \\
 (\mathbf{1}_M + \xi_j^t)^T x_k^{t-1} = & \mathbf{1}_M^T x_j^t, \quad k = 1, \dots, K^t, \quad t = 1, \dots, H-1, \quad j \in \{k\}^+, \\
 (x^t \geq & 0, \quad t = 1, \dots, H-1),
 \end{aligned}$$

v ktorom už v ohraničeniach nevystupuje ohraničenie na minimálny výnos portfólia na konci sledovaného obdobia, ale sa zahrnie do účelovej funkcie pomocou parametra δ . Poznamenávame, že úloha v tvare (1.3) sa štandardne definuje s ohraničením na minimálnu očakávanú hodnotu portfólia v tvare $\mathbb{E}[(\mathbf{1} + \xi^H)^T x^{H-1}, \mathcal{F}^0] \geq \mu$. Na takto definovanú úlohu nemožno aplikovať horeuvedený prepis, a teda takto definovaná úloha nie je riešiteľná pomocou dekompozičných algoritmov L-shaped a CVaRmin metódy. Z tohto dôvodu sme boli nútení mierne preformulovať túto úlohu na tvar (1.3), a teda sme sa rozhodli uvažovať ohraničenie v tvare $\mathbb{E}[(\mathbf{1} + \xi^H)^T x^{H-1}, \mathcal{F}^{H-1}] \geq \mu$. Takto zadefinovaná úloha je postačujúcou na vyhodnotenie efektívnosti a výhodnosti použitia jednotlivých algoritmov na riešenie úloh typu (2.18) a štrukturálne podobných úloh. Takto prepísaná úloha sa už poľahky dá naformulovať v tvare rekurzív (2.11)–(2.13). Úloha (2.13) nadobúda tvar:

$$\begin{aligned}
 \min_{x^0, a^0} \quad & (\mathbf{1} + r^0)^T x^0 - a^0 + \mathcal{Q}^1(x^0, a^0) \\
 W^0 x^0 = & h^0, \\
 (x^0 \geq & 0),
 \end{aligned} \tag{2.19}$$

kde

$$W^0 = (\mathbf{1}_M^T), \quad h^0 = 1, \quad y^0 = ((x^0)^T, a^0)$$

Úlohy (2.12) pre $t = 1, \dots, H - 1$ sa prepíšu do tvaru:

$$\begin{aligned} Q^t \left(x_{\{k\}^-}^{t-1}, a_{\{k\}^-}^{t-1}, \xi_k^t \right) &= \min_{y_k^t} (c_k^t)^T y_k^t + Q^{t+1}(x_k^t, a_k^t) \\ W^t y_k^t &= -T_k^{t-1} y_{\{k\}^-}^{t-1}, \\ (x_k^t \geq 0), z_k^t &\geq 0, d_k^t \geq 0 \end{aligned} \quad (2.20)$$

kde

$$\begin{aligned} Q^{t+1}(x_k^t, a_k^t) &= \sum_{j \in \{k\}^+} p_j^{t+1} Q^{t+1}(x_k^t, a_k^t, \xi_j^{t+1}), \\ y_k^t &= \left((x_k^t)^T, a_k^t, z_k^t, d_k^t \right), \quad c_k^t = \left((\mathbf{1} + r_k^t)^T, -1, \frac{1}{\alpha}, 0 \right) \text{ pre } t = 1, \dots, H - 2, \\ c_k^{H-1} &= \left((\delta - 1)(\mathbf{1} + r_k^{H-1})^T, -\delta, \frac{1}{\alpha}, 0 \right) \\ W^t &= \begin{pmatrix} \mathbf{1}_M^T & 0 & 0 & 0 \\ \mathbf{0}_M^T & 0 & -1 & 1 \end{pmatrix}, \quad T_k^{t-1} = \begin{pmatrix} -(\mathbf{1} + \xi_k^t)^T & 0 & 0 & 0 \\ -(\mathbf{1} + \xi_k^t)^T & 1 & 0 & 0 \end{pmatrix}. \end{aligned}$$

Premenné d_k^t , $k = 1, \dots, K^t$ sú pomocnými premennými podobne ako v podkapitole 2.1.3 a vektor r_k^t predstavuje očakávaný výnos v stupni $t + 1$ ak v stupni t nastala realizácia k . Terminálna podmienka (2.11) nadobúda tvar:

$$\begin{aligned} Q^H \left(x_{\{k\}^-}^{H-1}, a_{\{k\}^-}^{H-1}, \xi_k^H \right) &= \min_{y_k^H} (c_k^H)^T y_k^H \\ W^H y_k^H &= -T_k^{H-1} y_{\{k\}^-}^{H-1}, \\ z_k^H &\geq 0, d_k^H \geq 0 \end{aligned} \quad (2.21)$$

kde

$$\begin{aligned} c_k^H &= \left(\frac{\delta}{\alpha}, 0 \right), \quad W^H = (-1, 1), \quad T_k^{H-1} = -(\mathbf{1} + \xi_k^H)^T, \quad y_k^H = (z_k^H, d_k^H). \\ h_k^t &= 0, \quad t = 1, \dots, H, \quad k = 1, \dots, K^t. \end{aligned}$$

Poznamenávame, že kompenzačné matice W^t , $t = 0, \dots, H$ úlohy v tvare (2.19)–(2.21) nespĺňajú podmienku úplných kompenzačných matíc:

$$\forall f \in \mathbf{R}^{m_t} \quad \exists y \geq 0 : W^t y = f, \quad t = 0, \dots, H$$

a teda vo všeobecnosti nepatria do skupiny jednoduchých alebo úplných kompenzačných matíc, čiže riešenie zo stupňa $t - 1$ nebude vždy prípustným v stupni t . V prípade kompenzačných matíc W^H sa jedná vždy o jednoduché kompenzačné matice, nakoľko nadobúdajú tvar totožný s tvarom uvedeným v podkapitole 2.1.3. Problém nastáva v prípade kompenzačných matíc W^t , $t = 1, \dots, H - 1$ pri uvažovaní nezápornosti váh v portfóliu. Vtedy kompenzačné matice nespĺňajú podmienky uvedené v Leme na

strane 24 a teda ani podmienku úplných kompenzačných matíc. Túto skutočnosť ako aj skutočnosť, že v prípade uvažovania shortsellingu sa naopak jedná o úplné kompenzačné matice, je možné veľmi jednoducho a priamočiaro overiť, či už priamo z podmienky úplných kompenzačných matíc alebo pomocou algoritmu uvádzanom v [14].

Týmto sme prepísali viac-periódovú úlohu optimalizácie portfólia v tvare (1.3) na tvar deterministického ekvivalentu (2.11)–(2.13). V ďalšej kapitole sa budeme venovať dekompozičným algoritmom určeným na riešenie jedno-periódových a viac-periódových úloh optimalizácie portfólia, ktorých testovanie je cieľom tejto práce a bude hlavnou náplňou Kapitoly 4.

Kapitola 3

Algoritmy na riešenie jedno-periódovej a viac-periódovej základnej úlohy optimalizácie portfólia

Náplňou tejto kapitoly bude predstavenie a vysvetlenie fungovania algoritmov na riešenie úloh odvodených v predchádzajúcej kapitole. Budú to algoritmy, ktoré efektívne využívajú špeciálnu blokovú štruktúru úloh zobrazených na Obr. 2.1 a Obr. 2.2. Ako sme uviedli, riešenie úloh priamou metódou (riešenie deterministického ekvivalentu) je pri veľkom počte scenárov časovo a pamäťovo náročné. V prípade viac-periódových úloh optimalizácie portfólia je použitie priamych metód častokrát z dôvodu obrovských rozmerov úlohy nemožné. Postupne si predstavíme rôzne modifikácie tzv. L-shaped metódy pre dvoj-stupňové SLÚ s pevnou kompenzáciou. Konkrétne to budú jednorezová, viacrezová a regularizovaná L-shaped metóda. Neskôr uvedieme aj metódu špeciálne vyvinutú na riešenie úloh s CVaR v účelovej funkcii. Jedná sa o tzv. CVaRmin metódu, ktorá využíva dobré vlastnosti druhostupňových úloh, čím sa výrazne zjednoduší implementácia metódy ako aj samotná efektívnosť metódy. Táto metóda je modifikáciou jednorezovej L-shaped metódy. Keďže v práci uvažujeme aj viacrezovú a regularizovanú L-shaped metódu, rozhodli sme sa odvodiť aj viacrezovú verziu metódy CVaRmin. Uvedieme aj zovšeobecnené algoritmy jednorezovej a viacrezovej L-shaped metódy na riešenie viac-stupňových SLÚ s pevnou kompenzáciou, tzv. jednorezovú a viacrezovú verziu vnorenej L-shaped metódy. Nakoniec, použitím rovnakých princípov aké boli použité pri odvodení CVaRmin metódy, modifikujeme a zefektívňujeme vnorenú verziu L-shaped metódy.

3.1 Jednorezová L-shaped metóda

L-shaped metóda je založená na Bendersovej dekompozícii [4] a prvýkrát bola uvedená v [30]. Najprv uvedieme algoritmus metódy a potom si podrobnejšie vysvetlíme jeho jednotlivé časti. Algoritmus uvádzame podľa [6].

Algoritmus jednerezovej L-shaped metódy

Krok 0: (Inicializácia)Prirad' $r = s = \nu = 0$.**Krok 1: (Riešenie hlavnej úlohy (3.1)–(3.5))**Prirad' $\nu = \nu + 1$, potom rieš úlohu:

$$\min_{x, \theta} z = c^T x + \theta \quad (3.1)$$

$$Ax = b, \quad (3.2)$$

$$D_l x \geq d_l, \quad l = 1, \dots, r, \quad (3.3)$$

$$E_l x + \theta \geq e_l, \quad l = 1, \dots, s, \quad (3.4)$$

$$x \geq 0, \quad \theta \in \mathbf{R}. \quad (3.5)$$

Nech (x^ν, θ^ν) je riešenie úlohy (3.1)–(3.5). Ak ešte neboli pridané rezy optimality ($s = 0$), prirad' $\theta^\nu = -\infty$ a θ^ν sa nebude uvažovať pri výpočte x^ν .

Krok 2: (Kontrola prípustnosti v druhostupňových úlohách (3.6))Ak platí $x \in K_2$, pokračuj na Krok 3.

Inak, pridaj rez optimality (3.3) a choď na Krok 1.

Krok 3: (Riešenie druhostupňových úloh - pridávanie rezov optimality)Pre všetky scenáre $k = 1, \dots, K$ rieš úlohu:

$$\begin{aligned} \min_y w &= q_k^T y \\ Wy &= h_k - T_k x^\nu, \\ y &\geq 0. \end{aligned} \quad (3.6)$$

Nech π_k^ν je vektor riešenia duálnej úlohy k (3.6) pre scenár k . Potom definuj:

$$E_{s+1} = \sum_{k=1}^K p_k (\pi_k^\nu)^T T_k, \quad (3.7)$$

$$e_{s+1} = \sum_{k=1}^K p_k (\pi_k^\nu)^T h_k. \quad (3.8)$$

Nech $w^\nu = e_{s+1} - E_{s+1} x^\nu$. Ak platí $w^\nu \leq \theta^\nu$, Koniec. x^ν je optimálne riešenie. Inak, prirad' $s = s + 1$, pridaj rez optimality (3.4) a choď na Krok 1.

3.2 Princíp fungovania L-shaped metódy

Algoritmus L-shaped metódy je založený na postupnej (iteračnej) lineárnej aproximácii člena $\mathcal{Q}(x)$, ktorý je vo všeobecnosti nelineárny. Táto aproximácia sa realizuje a postupne zlepšuje pridávaním rezov optimality (3.4) v každej makroiterácii ν . Ako sme už uviedli, algoritmus využíva špeciálnu blokovú štruktúru deterministického ekvivalentu zobrazenú na Obrázku 2.1. Jedna makroiterácia pozostáva z postupného riešenia hlavnej úlohy, v ktorej parametre s a r reprezentujú počet doposiaľ pridaných rezov optimality a rezov prípustnosti. Nasleduje kontrola prípustnosti riešenia hlavnej úlohy v druhostupňových úlohách. Tieto úlohy sa zvyknú nazývať aj podúlohami. V prípade zistenia neprípustnosti v niektorej z podúloh sa pridá rez prípustnosti (3.3) a opäť sa rieši hlavná úloha. V opačnom prípade sa pokračuje riešením podúloh a následnou kontrolou optimality. Pod kontrolou optimality sa myslí kvalita aproximácie. Ak je aproximácia dostatočne dobrá, algoritmus končí. Ak nie, pridá sa rez optimality a opäť sa rieši hlavná úloha. Teraz pristúpime k podrobnejšiemu popisu krokov 2 a 3, aby sme ozrejmili spôsob výpočtu rezov optimality a rezov prípustnosti. Podotýkame, že úlohy, ktorými sa budeme zaoberať, sú SLÚ s jednoduchou kompenzáciou a preto, ako už bolo uvedené, prípustné riešenie z hlavnej úlohy je vždy prípustné v podúlohách. To znamená, že krok 2 je možné v tomto prípade vynechať.

V kroku 2 sa kontroluje prípustnosť podúloh a v prípade neprípustnosti sa pridávajú rezy prípustnosti. Prípustnosť sa kontroluje riešením nasledovnej úlohy pre všetky $k = 1, \dots, K$:

$$\begin{aligned} \min_{v^+, v^-} w &= e^T v^+ + e^T v^- \\ Wy + Iv^+ - Iv^- &= h_k - T_k x^\nu, \\ y \geq 0, v^+ \geq 0, v^- &\geq 0. \end{aligned} \quad (3.9)$$

Podotýkame, že úloha (3.9) je vždy prípustná a riešenie hlavnej úlohy je prípustné, ak pre optimálnu hodnotu w úlohy (3.9) platí $w = 0$. V prípade $w > 0$ nastáva neprípustnosť. Úloha (3.9) sa rieši postupne pre všetky k . Ak pre nejaké k nastáva $w > 0$, potom definujeme:

$$D_{r+1} = (\sigma_k^\nu)^T T_k, \quad (3.10)$$

$$d_{r+1} = (\sigma_k^\nu)^T h_k, \quad (3.11)$$

pridáme rez prípustnosti v tvare (3.3), priradíme $r = r + 1$ a vrátime sa na krok 1 (riešime hlavnú úlohu s novým ohraničením). Vektor σ_k^ν reprezentuje vektor riešenia duálnej úlohy k úlohe (3.9). Ak pre všetky k platí $w = 0$, pokračujeme na krok 3.

Ďalej ozrejmíme spôsob výpočtu rezov prípustnosti. Uvažujme duálnu úlohu k úlohe (3.9) v tvare:

$$\begin{aligned}
\max_{\sigma^\nu} & (h_k - T_k x^\nu)^T \sigma^\nu \\
& (\sigma^\nu)^T W \leq 0, \\
& (\sigma^\nu)^T I \leq 1, \\
& -(\sigma^\nu)^T I \leq 1.
\end{aligned} \tag{3.12}$$

Ak pre nejaké k nastáva $w > 0$, tak pre hodnotu účelovej funkcie úlohy (3.12) dostávame tiež $(h_k - T_k x^\nu)^T \sigma^\nu > 0$. Tento fakt vyplýva zo silnej vety o dualite, uvedenej napr. v [21], ktorá hovorí, že ak má jedna z úloh optimálne riešenie, tak má aj druhá a hodnoty účelových funkcií sa v optime rovnajú. Definujme si množinu podľa [6]: $posW = \{t | Wy = t, y \geq 0\}$, ktorá reprezentuje všetky pravé strany ohraničení druhostupňových úloh, ktoré sa dajú vyjadriť pomocou nezápornej kombinácie stĺpcov matice W . V prípade neprípustnosti pre nejaké k musí platiť $h_k - T_k x^\nu \notin posW$. V takomto prípade musí existovať nadrovina oddeľujúca $h_k - T_k x^\nu$ a $posW$, pre ktorú platí $\sigma^T t \leq 0$ pre všetky $t \in posW$ a $(h_k - T_k x^\nu)^T \sigma > 0$. Takáto nadrovina je v kroku 2 reprezentovaná práve vektorom riešenia duálnej úlohy. Vidieť, že v prípade neprípustnosti, vektor σ_k^ν spĺňa potrebné podmienky. Teda na to, aby prvostupňové riešenie bolo prípustné, je potrebné aby patrilo do $posW$ a teda musí platiť $(h_k - T_k x^\nu)^T \sigma^\nu \leq 0$, čo zodpovedá rezom prípustnosti s koeficientami v tvare (3.10) a (3.11). Z tohto je zrejmé, že pridaním ohraničení v tvare (3.3) sa vylúčia neprípustné prvostupňové rozhodnutia.

Ostáva ozrejmiť spôsob výpočtu rezov optimality. Opäť je užitočné zdefinovať duálnu úlohu. Duálna úloha k úlohe (3.6) má tvar:

$$\begin{aligned}
\max_{\pi^\nu} & (h_k - T_k x^\nu)^T \pi^\nu, \\
& (\pi^\nu)^T W \leq q_k.
\end{aligned} \tag{3.13}$$

Pre optimálne riešenia z duality opäť vyplýva:

$$Q(x^\nu, \xi_k) = (h_k - T_k x^\nu)^T \pi_k^\nu.$$

Z konvexnosti funkcie $Q(x^\nu, \xi_k)$ a taktiež zo slabej vety o dualite dostávame vzťah pre ľubovoľné x :

$$Q(x, \xi_k) \geq (h_k - T_k x)^T \pi_k^\nu.$$

Pre funkciu $Q(x)$ platí:

$$Q(x_\nu) = \mathbb{E}[Q(x_\nu, \xi)] = \sum_{k=1}^K Q(x_\nu, \xi_k) = \sum_{k=1}^K (h_k - T_k x_\nu)^T \pi_k^\nu$$

a pre ľubovoľné x

$$\mathcal{Q}(x) \geq \sum_{k=1}^K (h_k - T_k x)^T \pi_k^\nu.$$

Výraz na pravej strane je zjavne lineárny v x a teda aproximácia funkcie $\mathcal{Q}(x)$ zodpovedá už spomínanej aproximácii pomocou lineárnych oporných nadrovín. Úloha

$$\begin{aligned} \min_x z &= c^T x + \mathcal{Q}(x) \\ Ax &= b, \\ x &\geq 0 \end{aligned}$$

je ekvivalentná s úlohou

$$\begin{aligned} \min_x z &= c^T x + \theta \\ \mathcal{Q}(x) &\leq \theta, \\ Ax &= b, \\ x &\geq 0, \end{aligned} \tag{3.14}$$

z čoho následne dostávame nerovnosť, ktorá zodpovedá rezom optimality (3.4):

$$\begin{aligned} \theta \geq \sum_{k=1}^K (h_k - T_k x)^T \pi_k^\nu &= \sum_{k=1}^K p_k (\pi_k^\nu)^T h_k - \sum_{k=1}^K p_k T_k x^T \pi_k^\nu \\ &\Downarrow \\ E_l x + \theta &\geq e_l. \end{aligned}$$

Na záver ešte objasníme ukončovaciu podmienku. Z hore uvedeného je zjavné, že pre optimálne riešenie (x^ν, θ^ν) platí $\theta^\nu = \mathcal{Q}(x^\nu)$, čo zodpovedá ukončovacej podmienke $\theta^\nu \geq \mathcal{Q}(x^\nu)$. V prípade, kde nastáva $\theta^\nu < \mathcal{Q}(x^\nu)$, žiaden z doposiaľ pridaných rezov optimality nezaručuje $\mathcal{Q}(x) \leq \theta$ a preto je potrebné pridať ďalší rez optimality, a teda riešenie (x^ν, θ^ν) ešte nie je optimálne. Týmto sme vysvetlili spôsob fungovania L-shaped algoritmov (jednorezovej aj viacrezovej verzie) podľa [6].

3.3 Viacrezová L-shaped metóda

Viacrezová L-shaped metóda sa od jednorezovej L-shaped metódy líši len v spôsobe pridávania rezov optimality. Ako sme uviedli, v jednorezovej verzii sa pre všetky scenáre pridáva stále len jeden rez optimality. V prípade viacrezovej verzii sa bude pridávať jeden rez optimality pre každý scenár vypočítaný podľa (3.22) a (3.23). Algoritmus uvádzame podľa [6] ale prvýkrát bol v takejto podobe uvedený v [5].

Algoritmus viacrezovej L-shaped metódy

Krok 0: (Inicializácia)

Prirad' $r = \nu = 0$ a $s_k = 0$ pre všetky $k = 1, \dots, K$.

Krok 1: (Riešenie hlavnej úlohy (3.15)–(3.19))

Prirad' $\nu = \nu + 1$, potom rieš úlohu:

$$\min_{x, \theta} z = c^T x + \sum_{k=1}^K \theta_k \quad (3.15)$$

$$Ax = b, \quad (3.16)$$

$$D_l x \geq d_l, \quad l = 1, \dots, r, \quad (3.17)$$

$$E_{l_k} x + \theta_k \geq e_{l_k}, \quad l_k = 1, \dots, s_k, \quad k = 1, \dots, K, \quad (3.18)$$

$$x \geq 0, \quad \theta_k \in \mathbf{R}, \quad k = 1, \dots, K. \quad (3.19)$$

Nech $(x^\nu, \theta_1^\nu, \dots, \theta_K^\nu)$ je riešenie úlohy (3.15-3.19). Ak ešte pre nejaké k neboli pridané rezy optimality ($s_k = 0$), prirad' $\theta_k^\nu = -\infty$ a θ_k^ν sa neuvažuje pri výpočte x^ν .

Krok 2: (Kontrola prípustnosti v druhostupňových úlohách (3.20))

Ak platí $x \in K_2$, pokračuj na Krok 3.

Inak, pridaj rez prípustnosti (3.17) a choď na Krok 1.

Krok 3: (Riešenie druhostupňových úloh - pridávanie rezov optimality)

Pre všetky scenáre $k = 1, \dots, K$ rieš úlohu:

$$\begin{aligned} \min_y w &= q_k^T y \\ Wy &= h_k - T_k x^\nu, \\ y &\geq 0. \end{aligned} \quad (3.20)$$

Nech π_k^ν je vektor riešenia duálnej úlohy k (3.20) pre scenár k . Ak platí

$$\theta_k^\nu < p_k(\pi_k^\nu)(h_k - T_k x^\nu), \quad (3.21)$$

potom definuj

$$E_{s_k+1} = p_k(\pi_k^\nu)^T T_k, \quad (3.22)$$

$$e_{s_k+1} = p_k(\pi_k^\nu)^T h_k. \quad (3.23)$$

Prirad' $s_k = s_k + 1$ a pridaj rez optimality (3.22) a (3.23). Ak podmienka (3.21) nie je splnená pre žiadne $k = 1, \dots, K$, Koniec, x^ν je optimálne riešenie. Inak choď na Krok 1.

Pridávanie jedného rezu optimality pre každý scenár má za následok rýchlejšiu

aproximáciu funkcie $\mathcal{Q}(x)$, a preto vo všeobecnosti viacrezová verzia konverguje v menšom počte makroiterácií, čo ale nemusí znamenať aj kratší výpočtový čas, keďže hlavná úloha v tomto prípade môže nadobúdať výrazne väčšie rozmery a tým značne spomaliť optimalizačný proces. To, ktorú verziu je výhodnejšie použiť, závisí na charaktere konkrétnej úlohy ([11]).

3.4 Regularizovaná L-shaped metóda

Algoritmus bol pôvodne uvedený v [25] a my sme ho uvádzame podľa [6]. Regularizovaná L-shaped metóda je modifikáciou viacrezovej L-shaped metódy. Hlavným rozdielom je pridanie kvadratického regulačného členu $\frac{\beta}{2}\|x - a^\nu\|^2$ k účelovej funkcii. a^ν je zvolený prípustný vektor (prípustné riešenie) a parameter β slúži na zmenu váhy regulačného členu v priebehu procesu. Základnou myšlienkou algoritmu je vygenerovanie postupnosti vektorov $\{a^\nu\}$ konvergujúcej k optimálnemu riešeniu. Spôsob určovania vektora $a^{\nu+1}$ prebieha nasledovne. V prípade tzv. nulových krokov, v ktorých nastáva neprípustnosť riešenia x^ν alebo riešenie x^ν je horšie (t.j. to, pre ktoré je hodnota účelovej funkcie väčšia) ako a^ν , sa v ďalšej iterácii použije vektor $a^{\nu+1} = a^\nu$. V opačnom prípade dostávame vektor riešenia, ktorý je lepší ako a^ν a teda v nasledujúcej iterácii sa použije $a^{\nu+1} = x^\nu$. Z tohto je zjavné, že postupnosť hodnôt účelovej funkcie $c^T a^\nu + \mathcal{Q}(a^\nu)$ daná vektormi a^ν je nerastúca a vektor a^ν sa dá interpretovať ako najlepšie riešenie po iteráciu ν . Algoritmus končí vtedy, keď je hodnota účelovej funkcie rovnaká pre riešenie $(x^\nu, \theta_1^\nu, \dots, \theta_K^\nu)$ úlohy (3.24)-(3.28) a aj pre vektor a^ν . Spôsob aktualizácie parametra β uvádzame podľa [6]. V prípade nulových krokov sa parameter β zdvojnásobí, čím sa zvýši váha regulačného členu. Zväčšenie parametra β zabezpečí, že optimálne riešenie hlavnej úlohy bude z bližšieho okolia bodu a^ν , ktorý je prípustný a doposiaľ najlepší, a teda možnosť neprípustného alebo horšieho riešenia sa zníži. V opačnom prípade sa β zmenší na polovicu, čím sa zníži váha regulačného členu a zvýši sa dôraz na minimalizáciu účelovej funkcie. Podotýkame, že algoritmus je citlivý na voľbu vektora a^1 . Pri testovaní tejto metódy bolo ukázané, že pre väčšinu úloh pri vhodnej voľbe vektora a^1 dosahuje regularizovaná metóda lepšie výsledky ako viacrezová alebo jednorezová metóda. Na druhej strane pri nevhodne zvolenom vektore a^1 sa môže výpočtový čas niekoľkonásobne zvýšiť. Viac o voľbe vektora a^1 sa dá nájsť v [6],[25],[26].

Na záver časti o algoritmoch L-shaped metódy ešte uvedieme vety hovoriace o konvergencii algoritmov.

Veta 5 ([6], str. 198)

Nech ξ je konečná náhodná premenná. Potom algoritmus L-shaped metódy (jednorezová a viacrezová verzia) konverguje v konečnom počte makroiterácií k optimálnemu riešeniu, ak toto existuje, v opačnom prípade vykazuje neprípustnosť úlohy (2.7).

Konštrukčný dôkaz je uvedený v [6].

Veta 6 ([6], str. 208)

Ak má pôvodná úloha (2.7) riešenie, tak algoritmus konverguje v konečnom počte makroiterácií k optimálnemu riešeniu. V opačnom prípade generuje postupnosť prípustných vektorov $\{a^\nu\}$ takých, že $Q(a^\nu) \rightarrow -\infty$ pre $\nu \rightarrow \infty$.

Konstruktívny dôkaz je uvedený v [6].

Algoritmus regularizovanej L-shaped metódy**Krok 0: (Inicializácia)**

Prirad' $r = \nu = 0$ a $s_k = 0$ pre všetky $k = 1, \dots, K$, zvol' prípustný bod a^1 a prirad' $\beta = 1$.

Krok 1: (Riešenie hlavnej úlohy (3.24)–(3.28))

Prirad' $\nu = \nu + 1$, potom rieš úlohu:

$$\min_{x, \theta} z = c^T x + \sum_{k=1}^K \theta_k + \frac{\beta}{2} \|x - a^\nu\|^2 \quad (3.24)$$

$$Ax = b, \quad (3.25)$$

$$D_l x \geq d_l, \quad l = 1, \dots, r, \quad (3.26)$$

$$E_{l_k} x + \theta_k \geq e_{l_k}, \quad l_k = 1, \dots, s_k, \quad k = 1, \dots, K, \quad (3.27)$$

$$x \geq 0, \quad \theta_k \in \mathbf{R}, \quad k = 1, \dots, K. \quad (3.28)$$

Nech $(x^\nu, \theta_1^\nu, \dots, \theta_K^\nu)$ je riešenie úlohy (3.24-3.28). Ak ešte pre nejaké k neboli pridané rezy optimality ($s_k = 0$), prirad' $\theta_k^\nu = -\infty$ a θ_k^ν sa neuvažuje pri výpočte x^ν . Ak $c^T x^\nu + \mathbf{1}_K^T \theta^\nu = c^T a^\nu + Q(a^\nu)$, Koniec, a^ν je optimálne riešenie.

Krok 2: (Kontrola prípustnosti v druhostupňových úlohách (3.20))

Ak platí $x \in K_2$, pokračuj na Krok 3. Inak, pridaj rez optimality (3.26), prirad' $a^{\nu+1} = a^\nu$ (nulový krok neprípustnosti), $\beta = 2\beta$ a choď na Krok 1.

Krok 3: (Riešenie druhostupňových úloh - pridávanie rezov optimality)

Pre všetky scenáre $k = 1, \dots, K$ rieš úlohu (3.20) a vypočítaj $Q(x^\nu, \xi_k)$. Ak platí podmienka (3.21), prirad' $s_k = s_k + 1$ a pridaj rez optimality (3.22),(3.23).

Krok 4: (presný krok)

Ak podmienka (3.21) nie je splnená pre žiadne $k = 1, \dots, K$, prirad' $a^{\nu+1} = x^\nu$ (presný krok), $\beta = \frac{\beta}{2}$ a choď na Krok 1.

Krok 5: (odhadnutý krok)

Ak platí $c^T x^\nu + Q(x^\nu) \leq c^T a^\nu + Q(a^\nu)$, prirad' $a^{\nu+1} = x^\nu$ (odhadnutý krok), $\beta = \frac{\beta}{2}$ a choď na Krok 1. Inak, prirad' $a^{\nu+1} = a^\nu$ (nulový krok prípustnosti), $\beta = 2\beta$ a choď na Krok 1.

3.5 CVaRmin metóda

V tejto časti uvedieme algoritmus vyvinutý špeciálne na riešenie optimalizačných úloh s CVaR v účelovej funkcii. Algoritmus uvedieme podľa [2] s malou modifikáciou, pretože autori v [2] definovali CVaR pre funkciu strát a my sme definovali CVaR pre funkciu výnosov. Pozmeníme aj poradie jednotlivých krokov, aby korešpondovali s krokmi predchádzajúcich algoritmov. Taktiež v algoritme použijeme značenie totožné so značením použitým v predchádzajúcich kapitolách a nie značenie použité autormi v [2], aby sme sa vyhli nezrovnalostiam spôsobeným nejednoznačnosťou zápisu. Diskrétna úloha optimalizácie portfólia (1.6) z Kapitoly 1 je ekvivalentná s úlohou v tvare:

$$\begin{aligned} \min_{x,a} \quad & x^T r - a + \frac{1}{\alpha} \mathbb{E}(Q(x, a, \xi)) \\ & x^T r \geq \mu, \\ & \mathbf{1}_M^T x \leq 1, \\ & (x \geq 0), \end{aligned}$$

s podúlohami v tvare:

$$\begin{aligned} Q(x, a, \xi) = \min_z \quad & \\ & z \geq a - x^T \xi, \\ & z \geq 0. \end{aligned} \tag{3.29}$$

Duálna úloha k úlohe (3.29) má tvar:

$$\begin{aligned} \max_{\pi} \quad & (a - x^T \xi) \pi \\ & 0 \leq \pi \leq 1. \end{aligned} \tag{3.30}$$

Všimnime si, že úloha (3.30) je veľmi jednoduchá a má riešenie $\pi = 0$ alebo $\pi = 1$ v závislosti od znamienka výrazu $(a - x^T \xi)$. Takýto typ riešenia je značným plusom z hľadiska implementácie algoritmu, pretože namiesto riešenia podúloh pre všetky scenáre sa stačí pozrieť na znamienko výrazu $(a - x^T \xi)$ a určiť hodnotu π . Možnosť tohto zjednodušenia predstavuje výraznú úsporu výpočtového času. Pripomíname, že sa jedná o SLÚ s jednoduchou kompenzáciou, v ktorých je riešenie hlavnej úlohy vždy prípustné v podúlohách, a preto sa krok 2 z predchádzajúcich algoritmov v tomto prípade vynechá. Toto opäť predstavuje značnú úsporu výpočtového času, pretože odpadne potreba riešenia úlohy (3.9) pre všetky scenáre. Označme \mathcal{K}^* označuje množinu indexov tých scenárov, pre ktoré majú korešpondujúce podúlohy duálne riešenia rovné jednej $\pi_k' = 1$. Algoritmus CVaRmin vyzerá nasledovne:

Algoritmus CVaRmin metódy

Krok 0: (Inicializácia) Prirad' $s = \nu = 0$.

Krok 1: (Riešenie hlavnej úlohy (3.31)–(3.35))

Prirad' $\nu = \nu + 1$ potom rieš úlohu :

$$\min_{x, a, \theta} \quad x^T r - a + \frac{1}{\alpha} \theta \quad (3.31)$$

$$x^T r \geq \mu, \quad (3.32)$$

$$\mathbf{1}_M^T x \leq 1, \quad (3.33)$$

$$\theta \geq \hat{p}_l a - \hat{\xi}_l^T x, \quad l = 1, \dots, s, \quad (3.34)$$

$$\theta \geq 0, \quad (x \geq 0). \quad (3.35)$$

Nech $(x^\nu, a^\nu, \theta^\nu)$ je riešenie úlohy (3.31)–(3.35).

Definuj $\mathcal{K}^* = \{k | 1 \leq k \leq N, (a - x^T \xi_k) > 0\}$.

Krok 3: (Pridávanie rezov optimality)

Definuj $w^\nu = \sum_{k \in \mathcal{K}^*} p_k (a - x^T \xi_k)$.

Ak platí $w^\nu - \theta^\nu \leq 0$, Koniec, x^ν je optimálne riešenie. Inak, definuj

$\hat{p}_{s+1} = \sum_{k \in \mathcal{K}^*} p_k$, $\hat{\xi}_{s+1} = \sum_{k \in \mathcal{K}^*} p_k \xi_k$, prirad' $s = s + 1$, pridaj rez optimality v tvare (3.34) a choď na Krok 1.

Z dôvodu nezahŕňať pomocné premenné do úlohy optimalizácie portfólia, definujeme hlavnú úlohu v trochu odlišnom tvare ako je vo všeobecnosti definovaná. Jedná sa o prvostupňové ohraničenia na rozpočet a minimálny požadovaný výnos, ktoré sú v tvare nerovností, možnosť shortsellingu a následné porušenie nezápornosti vektora váh. Všetky tieto modifikácie sa dajú pri implementácii algoritmu prepísať na požadovaný tvar spôsobom, aký sme uviedli v Kapitole 1.

Jedna makroiterácia pozostáva z riešenia hlavnej úlohy (s už len rezmi optimality). Nasleduje kontrola optimality, ktorá je rovnaká ako v prípade predchádzajúcich algoritmov. Stačí si uvedomiť, že:

$$h_k = 0, \quad \forall k; \quad T_k = -a + x^T \xi_k; \quad \pi_k^\nu = 1, \quad \forall k \in \mathcal{K}^*; \quad \pi_k^\nu = 0, \quad \forall k \notin \mathcal{K}^*, \quad (3.36)$$

$$E_{s+1} = \sum_{k=1}^K p_k (\pi_k^\nu)^T T_k, \quad e_{s+1} = \sum_{k=1}^K p_k (\pi_k^\nu)^T h_k.$$

Potom:

$$e_{s+1} - E_{s+1} - \theta^\nu \leq 0$$

⇕

$$\sum_{k \in \mathcal{K}^*} p_k (a - x^T \xi_k) - \theta^\nu \leq 0.$$

Proces pridávania rezov optimality je s použitím (3.36) tiež ekvivalentný so spôsobom v predchádzajúcich algoritmoch. Keďže tento algoritmus je len zovšeobecnením jednorezovej L-shaped metódy, o konvergencii algoritmu hovorí Veta 5. Zovšeobecnená verzia vety pre tento prípad je uvedená v [2]. Ako uvedieme neskôr v Kapitole 4, zo vzájomného porovnania jednotlivých verzií L-shaped metódy najlepšie výsledky dosahovala viacrezová verzia. To nás motivovalo k uvažovaniu o viarezovej verzii CVaRmin. Viacrezová verzia CVaRmin nie je v literatúre uvádzaná, tak sme ju odvodili podľa vzoru jednorezovej CVaRmin.

3.6 Viacrezová CVaRmin metóda

Algoritmus viacrezovej CVaRmin metódy

Krok 0: (Inicializácia) Prirad' $\nu = 0$ a $s_k = 0$ pre všetky $k = 1, \dots, K$.

Krok 1: (Riešenie hlavnej úlohy (3.37)–(3.41))

Prirad' $\nu = \nu + 1$ potom rieš úlohu :

$$\min_{x, a, \theta} \quad x^T r - a + \frac{1}{\alpha} \sum_{k=1}^K \theta_k \quad (3.37)$$

$$x^T r \geq \mu, \quad (3.38)$$

$$\mathbf{1}_M^T x \leq 1, \quad (3.39)$$

$$\theta_k \geq \hat{p}_{l_k} a - \hat{\xi}_{l_k}^T x, \quad l_k = 1, \dots, s_k, \quad k = 1, \dots, K, \quad (3.40)$$

$$\theta_k \geq 0 \quad k = 1, \dots, K, \quad (x \geq 0). \quad (3.41)$$

Nech $(x^\nu, a^\nu, \theta_1^\nu, \dots, \theta_K^\nu)$ je riešenie úlohy (3.37)–(3.41).

Krok 3: (Pridávanie rezov optimality)

Pre všetky scenare $k = 1, \dots, K$ urči znamienko výrazu $(a - x^T \xi_k)$ a hodnotu π_k^ν .

Ak platí:

$$\theta_k^\nu < p_k(\pi_k^\nu)(a - x^T \xi_k), \quad (3.42)$$

potom definuj

$$\hat{p}_{s_k+1} = p_k \pi_k^\nu, \quad (3.43)$$

$$\hat{\xi}_{s_k+1} = p_k \pi_k^\nu \xi_k. \quad (3.44)$$

Prirad' $s_k = s_k + 1$ a pridaj rez optimality v tvare (3.40). Ak podmienka (3.42) nie je splnená pre žiadne $k = 1, \dots, K$, Koniec, x^ν je optimálne riešenie. Inak choď na Krok 1.

Táto metóda je odvodená z viacrezovej L-shaped metódy. Využitím (3.36), jednoduchšej kompenzačnej matice a miernej modifikácie hlavnej úlohy (3.31)–(3.35) je z uvedeného algoritmu zrejmé, že sa v podstate jedná o zjednodušený algoritmus viacrezovej L-shaped metódy. Poznamenávame, že zo vzťahov v kroku 3 vyplýva, že pre každý scenár môžeme pridať maximálne jeden rez optimality. To je spôsobené tým, že hodnota duálnych riešení je vždy $\pi_k' = 0$ alebo $\pi_k' = 1$. V prípade $\pi_k' = 0$ sa nepridáva žiaden rez optimality a v prípade $\pi_k' = 1$ sa pridá rez optimality, ak je splnená podmienka (3.42). Pre každý scenár môžeme nanajvýš pridať jeden rez optimality korešpondujúci s $\pi_k' = 1$. Z toho je zrejmé, že táto metóda musí konvergovať v dvoch makroiteráciách. V prvej makroiterácii sa pridávajú rezy optimality pre všetky scenáre, v ktorých je to potrebné a v druhej už dostávame optimálne riešenie. Toto pozorovanie, ako uvedieme v Kapitole 4, sa aj potvrdilo. Na záver ešte poznamenávame, že konvergenciu aj tejto metódy potvrdzuje Veta 5.

3.7 Priame metódy

V tejto časti veľmi stručne popíšeme priame metódy, ktorých verzie priamo implementované v Matlabe použijeme pri testovaní v Kapitole 4. Konkrétne to budú simplexová metóda a metóda vnútorného bodu. Použijeme ich len na jedno-periódové úlohy optimalizácie portfólia, pretože v prípade viac-periódovej úlohy optimalizácie portfólia je v dôsledku obrovských rozmerov a pamäťových nárokov použitie týchto metód nemožné.

3.7.1 Simplexová metóda

Simplexovú metódu ako prvý rozpracoval Dantzing v roku 1947 a prvýkrát bola publikovaná v [10]. Je to základná metóda pre riešenie úloh lineárneho programovania. Uvedieme len základnú myšlienku simplexovej metódy uvádzanú napr. v [21]. Podrobné informácie o simplexovej metóde možno nájsť v [14] a [21].

Uvažujme úlohu v tvare (2.1), ktorý považujeme za všeobecný tvar úloh lineárneho programovania. Ďalej predpokladáme bez ujmy na všeobecnosti, že množina prípustnosti úlohy (2.1) je neprázdna a obsahuje vrcholy, pričom jeden z nich poznáme. Metóda začína v známom vrchole množiny prípustnosti, označme ho y^0 . Myšlienka metódy spočíva v nájdení takej hrany množiny prípustnosti $\{y^0 + \lambda d\}$, ktorá vedie do nižších hodnôt účelovej funkcie, t.j. $c^T d < 0$ pre $\lambda > 0$. Ak je táto hrana úsečkou $[y^0, y^1]$, tak minimum účelovej funkcie sa nadobúda v bode y^1 . Týmto spôsobom vieme vytvoriť postupnosť bodov pre ktoré platí $c^T y^0 > c^T y^1 > \dots > c^T y^n$ a ak pre bod y^n už nie je možné nájsť hranu spĺňajúcu potrebné vlastnosti, tak y^n je optimálnym riešením. Ak pre bod y^n je možné nájsť úsečku vedúcu do nižších hodnôt účelovej funkcie ale táto úsečka je polpriamkou, tak nastáva neohraničenosť.

3.7.2 Metóda vnútorného bodu

Metódy vnútorného bodu boli vyvinuté v 60-tych a na začiatku 70-tych rokov na riešenie nelineárnych úloh s ohraničeniami v tvare nerovností. Aj v tomto prípade uvedieme len základnú myšlienku (algoritmus) primárno-duálnej metódy vnútorného bodu a pre bližšie informácie ohľadom metód vnútorného bodu čitateľa odkazujeme napr. na [8],[14],[24].

Duálna úloha k úlohe (2.1) má tvar:

$$\begin{aligned} \min_{y,s} \quad & b^T y \\ A^T y + s &= c, \\ s &\geq 0. \end{aligned} \tag{3.45}$$

Označme množinu prípustných riešení primárnej úlohy (2.1) a množinu prípustných riešení duálnej úlohy (3.45) ako:

$$\mathcal{P} = \{x | Ax = b, x \geq 0\} \quad \mathcal{D} = \{(y, s) | A^T y + s = c, s \geq 0\}.$$

Ďalej zavedme označenie $w = (x, y, s)$ zodpovedajúce primárno-duálnemu bodu a $\mathcal{W} = \mathcal{P} \times \mathcal{D}$ reprezentujúce primárno-duálnu množinu prípustnosti. Karush-Kuhn-Tuckerove podmienky optimality pre primárno-duálny pár w sú dané sústavou rovníc:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ XS\mathbf{1}_n &= 0, \\ (x, s) &\geq 0, \end{aligned} \tag{3.46}$$

kde $X, S \in \mathbf{R}^{n \times n}$ sú diagonálne matice s prvkami x_i , respektíve s_i pre $i = 1, \dots, n$ na diagonále. Všeobecný algoritmus metódy vnútorného bodu vyzerá nasledovne. Na začiatku predpokladáme, že máme k dispozícii nejaký striktné prípustný primárno-duálny bod, t.j. bod spĺňajúci $w \in \{w \in \mathcal{W} | (x, s) > 0\}$, ktorý je štartovacím bodom metódy. V každej iterácii k , pomocou nejakej metódy voľnej optimalizácie, nájdeme približné riešenie systému rovníc (3.46). Najčastejšie používanou metódou je Newtonova metóda a teda pre nový bod nájdený touto metódou platí $w^{k+1} = w^k + \alpha_k \Delta w^k$, kde Δw^k označuje smer a α_k dĺžku kroku Newtonovej metódy. V súčasnosti existuje veľké množstvo variant metódy vnútorného bodu. Tieto varianty sa môžu líšiť napr. v spôsobe výpočtu smeru Δw^k , v spôsobe výpočtu dĺžky kroku α_k , v ukončovacej podmienke atď.

3.8 Vnorená jednorezová L-shaped metóda

Vnorená jednorezová L-shaped metóda je určená na riešenie viac-stupňových SLÚ s pevnou kompenzáciou a je zovšeobecnením jednorezovej L-shaped metódy. Algoritmus uvádzame podľa [6], ale prvýkrát bol uvedený v [19]. Pred uvedením samotného algoritmu je potrebné zdefinovať úlohu, ktorú budeme riešiť v každom stupni a každom uzle. Táto úloha je akýmsi ekvivalentom hlavnej úlohy v prípade algoritmov na riešenie dvoj-stupňových SLÚ s pevnou kompenzáciou.

Pre každý stupeň $t = 0, \dots, H$ a každý uzol $k = 1, \dots, K^t$ definujeme hlavnú úlohu v tvare:

$$\min_{x_k^t, \theta_k^t} (c_k^t)^T x_k^t + \theta_k^t \quad (3.47)$$

$$W^t x_k^t = h_k^t - T_k^{t-1} x_{\{k\}^-}^{t-1}, \quad (3.48)$$

$$D_{k,l}^t \geq d_{k,l}^t, \quad l = 1, \dots, r_k^t, \quad (3.49)$$

$$E_{k,l}^t + \theta_k^t \geq e_{k,l}^t, \quad l = 1, \dots, s_k^t, \quad (3.50)$$

$$x_k^t \geq 0, \quad (3.51)$$

kde $x_{\{k\}^-}^{t-1}$ je priebežné riešenie úlohy (3.47)–(3.51) pre daný uzol, t.j. doposiaľ najlepšie riešenie. Pre prvý stupeň $t = 0$ ohraničenie (3.48) má tvar $W^0 x_1^0 = h_1^0$ a predstavuje počiatočné podmienky úlohy v tvare (2.10) alebo v tvare (2.13). V stupni $t = H$ pre všetky $k = 1, \dots, K^H$ definuje účelovú funkciu bez člena θ_k^H a taktiež bez ohraničení (3.49) a (3.50). Ďalej je užitočné zdefinovať množinu A_k^t ako množinu nasledovníkov uzla $k = 1, \dots, K^t$ v stupni $t = 0, \dots, H - 1$. Hlavnú úlohu (3.47)–(3.51) pre daný stupeň t a uzol k budeme podľa [6] označovať ako $NLDS(t, k)$.

Algoritmus vnorenej jednorezovej L-shaped metódy

Krok 0: (Inicializácia)

Polož $t = 0$ a $k = 1$ (koreň stromu scenárov), pre všetky t a k prirad' $r_k^t = s_k^t = 0$ a pridaj ohraničenie $\theta_k^t = 0$ do $NLDS(t, k)$. Nastav $smer = Vpred$ a $\nu = 1$.

Krok 1: Riešenie hlavnej úlohy (3.47)–(3.51)

Rieš $NLDS(t, k)$.

Ak je neprípustná a $t = 0$, potom je celá úloha (2.11)–(2.13) neprípustná.

Ak je neprípustná a $t > 0$, potom prirad' $r_{\{k\}^-}^{t-1} = r_{\{k\}^-}^{t-1} + 1$. Vypočítaj koeficienty rezov optimality ako:

$$D_{\{k\}^-, r_{\{k\}^-}^{t-1}}^{t-1} = (\pi_k^t)^T T_k^{t-1}, \quad (3.52)$$

$$d_{\{k\}^-, r_{\{k\}^-}^{t-1}}^{t-1} = (\pi_k^t)^T T_k^t + (\rho_k^t)^T d_k^t. \quad (3.53)$$

kde π_k^t je vektor duálnych riešení prislúchajúcich ohraničeniam (3.48) a ρ_k^t je

vektor duálnych riešení prislúchajúcich ohraničeniam (3.49). Pridaj rez prípustnosti s koeficientami (3.52) a (3.53) do $NLDS(t-1, \{k\}^-)$.

Polož $t = t-1, k = \{k\}^-$ a choď na Krok 1.

Ak je prípustná aktualizuj hodnoty riešenia x_k^t, θ_k^t a ulož hodnoty duálnych riešení prislúchajúcich ohraničeniam (3.48)–(3.50) ako $(\pi_k^t, \rho_k^t, \sigma_k^t)$.

Ak $k < K^t$, polož $k = k+1$ a choď na Krok 1.

Inak ($k = K^t$), ak $t = 0$, nastav $smer = Vpred$.

Ak $t < H$ a $smer = Vpred$, polož $t = t+1, k = 1$ a choď na Krok 1.

Ak $t = H$, nastav $smer = Vzad$. Choď na Krok 2.

Krok 2: Výpočet a pridávanie rezov optimality (3.49)

Pre všetky uzly $k = 1, \dots, K^{t-1}$ v $t-1$ vypočítaj:

$$E_{k, (s_k^{t-1}+1)}^{t-1} = \sum_{j \in A_k^t} \frac{p_j^t}{p_k^{t-1}} (\pi_j^t)^T T_j^{t-1}, \quad (3.54)$$

a

$$e_{k, (s_k^{t-1}+1)}^{t-1} = \sum_{j \in A_k^t} \frac{p_j^t}{p_k^{t-1}} \left[(\pi_j^t)^T T_j^{t-1} + \sum_{i=1}^{r_j^t} (\rho_{ji}^t)^T d_{ji}^t + \sum_{i=1}^{s_j^t} (\sigma_{ji}^t)^T e_{ji}^t \right], \quad (3.55)$$

kde ρ_{ji}^t a σ_{ji}^t sú i -te skalárne zložky vektorov ρ_j^t a σ_j^t . Súčasná podmienená hodnota účelových funkcií podúloh v A_k^t je daná ako $\bar{\theta}_k^{t-1} = e_{k, s_k^{t-1}+1}^{t-1} - E_{k, s_k^{t-1}+1}^{t-1} x_k^{t-1}$.

Ak sa ohraničenie $\theta_k^{t-1} = 0$ nachádza v $NLDS(t-1, k)$, nahraď ho ohraničeným (3.49) s koeficientami (3.54) a (3.55) a priraď $s_k^{t-1} = 1$.

Ak $\bar{\theta}_k^{t-1} > \theta_k^{t-1}$, pridaj ohraničenie (3.49) s koeficientami (3.54) a (3.55) do $NLDS(t-1, k)$ a priraď $s_k^{t-1} = s_k^{t-1} + 1$.

Ak $t = 1$ a do $NLDS(0, 1)$ neboli pridané žiadne ohraničenia, Koniec, x_1^0 je optimálne riešenie.

Inak, polož $t = t-1, k = 1$. Ak $t = 0$, priraď $smer = Vpred$ a $\nu = \nu + 1$. Choď na Krok 1.

V ďalšom texte bližšie popíšeme princíp fungovania uvedeného algoritmu. Algoritmus najprv prejde strom smerom *Vpred*, t.j. od koreňa k listom, pričom v každom uzle rieši prislúchajúcu hlavnú úlohu. V prípade prípustnosti sa riešenia x_k^t posielajú do hlavných úloh prislúchajúcim nasledovníkom uzla k v stupni $t+1$, kde vystupujú v ohraničeniach (3.48). V prípade hlásenia neprípustnosti môžu nastať dve situácie. Ak je neprípustnosť v prvom stupni, t.j. neprípustná je hlavná úloha prislúchajúca koreňu stromu scenárov, potom je celá úloha (2.11)–(2.13) neprípustná. V opačnom prípade sa do hlavnej úlohy prislúchajúcej predchodcovi uzla k v stupni $t-1$ pridá rez prípustnosti s koeficientami v tvare (3.52) a (3.53), čím sa vylúči riešenie spôsobujúce neprípustnosť. Tvar koeficientov rezu prípustnosti (3.49) je odvodený analogickým spôsobom ako v prípade L-shaped metódy pre dvoj-stupňové SLÚ s pevnou kompenzáciou, a

teda neprípustnosť je daná takými duálnymi riešeniami $\pi_k^t, \rho_k^t \geq 0$, ktoré spĺňajú:

$$(\pi_k^t)^T W^t + (\rho_k^t)^T W_k^t \leq 0, \quad (\pi_k^t)^T (h_k^t - T_k^{t-1} x_{\{k\}^-}^{t-1}) + (\rho_k^t)^T d_k^t > 0.$$

Na to, aby riešenie $x_{\{k\}^-}^{t-1}$ bolo prípustné, musí platiť:

$$(\pi_k^t)^T (h_k^t - T_k^{t-1} x_{\{k\}^-}^{t-1}) + (\rho_k^t)^T d_k^t \leq 0,$$

čo zodpovedá koeficientom rezov prípustnosti v tvare (3.52) a (3.53). Po pridaní rezu prípustnosti sa algoritmus vráti do uzla $\{k\}^-$ v stupni $t-1$ a opätovne rieši prislúchajúcu hlavnú úlohu s už pridaným rezom prípustnosti.

Akonáhle algoritmus dosiahne posledný list stromu scenárov, smer sa zmení na *Vzad* a začnú sa riešiť hlavné úlohy smerom od listov ku koreňu. Duálne riešenie $(\pi_k^t, \rho_k^t, \sigma_k^t)$ slúži na výpočet rezov optimality v hlavnej úlohe prislúchajúcej predchodcovi uzla k v stupni t . Odvodenie rezov optimality je opäť podobné prípadu z podkapitoly 3.2. Pre $t = 0, \dots, H$, $k = 1, \dots, K^t$ a ľubovoľné $x_{\{k\}^-}^{t-1}$ z duality vyplýva:

$$Q^t(x_{\{k\}^-}^{t-1}, \xi_{k \in A_{\{k\}^-}^t}^t) \geq (\pi_k^t)^T T_k^{t-1} x_{\{k\}^-}^{t-1} + \sum_{i=1}^{r_k^t} (\rho_{ki}^t)^T d_{ki}^t + \sum_{i=1}^{s_k^t} (\sigma_{ki}^t)^T e_{ki}^t,$$

z toho dostávame:

$$Q^t(x_{\{k\}^-}^{t-1}) \geq \sum_{k \in A_{\{k\}^-}^t} \frac{p_k^t}{p_{\{k\}^-}^{t-1}} \left[(\pi_k^t)^T T_k^{t-1} x_{\{k\}^-}^{t-1} + \sum_{i=1}^{r_k^t} (\rho_{ki}^t)^T d_{ki}^t + \sum_{i=1}^{s_k^t} (\sigma_{ki}^t)^T e_{ki}^t \right],$$

a použitím vzťahu $\theta_k^{t-1} \geq Q^t(x_{\{k\}^-}^{t-1})$ dostávame tvar rezov optimality (3.50):

$$\theta_k^{t-1} \geq -E_{\{k\}^-}^{t-1} x_{\{k\}^-}^{t-1} + e_{\{k\}^-}^{t-1}.$$

Ukončovacia podmienka je totožná s podmienkou uvedenou v podkapitole 3.2. Pod jednou makroiteráciou budeme rozumieť jeden prechod algoritmu od koreňa stromu k listom a späť. Vo všeobecnosti existuje viacero pravidiel postupu algoritmu po strome scenárov. Tieto pravidlá sa zvyknú označovať pojmom *sekvenčný protokol*. Nami použitý sekvenčný protokol sa nazýva rýchlo vpred - rýchlo vzad (fast forward - fast back) a ako sa uvádza v [6], tak viaceré experimenty potvrdili, že tento sekvenčný protokol je najefektívnejší.

Nevýhodou uvedeného algoritmu je skutočnosť, že pri pridaní rezu prípustnosti si algoritmus nepamätá, v ktorom uzle nastala neprípustnosť a zbytočne prechádza časť stromu druhýkrát. Tomuto problému sa dá predísť zavedením zásobníka, do ktorého sa ešte pred vrátením do uzla $\{k\}^-$ v $t-1$ uloží súčasná pozícia, čiže uzol k , stupeň t a smer postupu. Následne v prípade prípustnosti sa algoritmus najprv pozrie do zásobníka a v prípade, že je neprázdny vyberie z neho uložený smer postupu a pozí-

ciu, z ktorej pokračuje ďalej. V opačnom prípade algoritmus postupuje na ďalší uzol podľa uvedených pravidiel. Implementácia zásobníka ako aj otestovanie efektívnosti jeho použitia sú uvedené v [11].

3.9 Vnorená viacrezová L-shaped metóda

Modifikácia vnorenej jednorezovej L-shaped metódy na viacrezovú verziu sa urobí podľa rovnakého princípu ako v prípade metód z podkapitol 3.1 a 3.3.

Pre každý stupeň $t = 0, \dots, H - 1$ a každý uzol $k = 1, \dots, K^t$ definujeme hlavnú úlohu v tvare:

$$\min_{x_k^t, \theta_{k,j}^t} (c_k^t)^T x_k^t + \sum_{j \in A_k^t} \theta_{k,j}^t \quad (3.56)$$

$$W^t x_k^t = h_k^t - T_k^{t-1} x_{\{k\}^-}^{t-1}, \quad (3.57)$$

$$D_{k,l}^t \geq d_{k,l}^t, \quad l = 1, \dots, r_k^t, \quad (3.58)$$

$$E_{k,l_j}^t + \theta_{k,j}^t \geq e_{k,l_j}^t, \quad l_j = 1, \dots, s_{k,j}^t, \quad j \in A_k^t, \quad (3.59)$$

$$x_k^t \geq 0. \quad (3.60)$$

Namiesto jedného rezu optimality pre všetkých nasledovníkov uzla k v stupni t pridávame jeden rez optimality pre každé $j \in A_k^t$. Každému nasledovníkovi prislúcha samostatný člen $\theta_{k,j}^t$ v účelovej funkcii. Hlavná úloha pre $t = H$ ostáva bez zmeny. Po zadefinovaní hlavných úloh je algoritmus vnorenej viacrezovej L-shaped metódy takmer totožný s algoritmom uvedeným v podkapitole 3.8. Rozdiel je len v spôsobe pridávania rezov optimality, ktorý prebieha nasledovne:

Pre všetky $k = 1, \dots, K^{t-1}$ v stupni $t - 1$ a pre všetky $j \in A_k^{t-1}$ vypočítaj:

$$E_{k,(s_{k,j}^{t-1}+1)}^{t-1} = \frac{p_j^t}{p_k^{t-1}} (\pi_j^t)^T T_j^{t-1}, \quad (3.61)$$

a

$$e_{k,(s_{k,j}^{t-1}+1)}^{t-1} = \frac{p_j^t}{p_k^{t-1}} \left[(\pi_j^t)^T T_j^{t-1} + \sum_{i=1}^{r_j^t} (\rho_{ji}^t)^T d_{ji}^t + \sum_{i=1}^{s_j^t} (\sigma_{ji}^t)^T e_{ji}^t \right], \quad (3.62)$$

Ak platí $\theta_{k,j}^{t-1} < e_{k,(s_{k,j}^{t-1}+1)}^{t-1} - E_{k,(s_{k,j}^{t-1}+1)}^{t-1} x_k^{t-1}$ alebo ak sa ohraničenie $\theta_{k,j}^{t-1} = 0$ nachádza v hlavnej úlohe (3.56)–(3.60), potom pridaj rez optimality v tvare (3.59) s koeficientami (3.61)–(3.62) a priraď $s_{k,j}^{t-1} = s_{k,j}^{t-1} + 1$, respektíve $s_{k,j}^{t-1} = 1$.

Ná záver časti o vnorenej L-shaped metóde uvádzame vetu zaručujúcu konvergenciu uvedených algoritmov.

Veta 7 ([6], str. 268)

Ak všetky náhodné vektory ξ^t , $t = 1, \dots, H$ pochádzajú z konečného diskrétného rozdelenia a všetky vektory x^t , $t = 0, \dots, H$ majú konečné horné ohraničenie, potom algoritmus vnorenej L-shaped metódy (jednorezová a viaczrezová verzia) konverguje k optimálnemu riešeniu úlohy (2.11)–(2.13) v konečnom počte iterácií.

Konstruktívny dôkaz je uvedený v [6].

3.10 Zefektívnenie jedno a viaczrezovej vnorenej L-shaped metódy

V tejto podkapitole stručne popíšeme jednoduchú modifikáciu obidvoch verzií vnorenej L-shaped metódy, ktorá, ako uvidíme v Kapitole 4, predstavuje značné zefektívnenie optimalizačného procesu. Využívať budeme skutočnosti popísané v podkapitole 3.5. Vychádzať budeme zo skutočnosti, že kompenzačné matice zodpovedajúce poslednému stupňu H patria do skupiny jednoduchých kompenzačných matic, t.j. $W^H = (-1, 1)$ a teda modifikáciu budeme realizovať pre riešenie $NLDS(H-1, k)$ pre $k = 1, \dots, K^{H-1}$. Postup je nasledovný. Po dosiahnutí stupňa $H-1$, pri prechode algoritmu stromom scenárov v doprednom smere, pre každé $k = 1, \dots, K^{H-1}$ riešime úlohu $NLDS(H-1, k)$ a jej riešenie určí znamienko výrazu $(a_k^{H-1} - (\mathbf{1} + \xi_j^H)^T x_k^{H-1})$. Podľa znamienka vieme určiť hodnotu duálneho riešenia π_j^H ($\pi_j^H = 0$ alebo $\pi_j^H = 1$) pre všetkých nasledovníkov uzla k , t.j. $j \in \{k\}^+$ a už v tomto kroku pridať rezy optimality podľa princípu opísaného v podkapitole 3.5 (v prípade modifikácie jednorezovej vnorenej L-shaped metódy) alebo v podkapitole 3.6 (v prípade modifikácie viaczrezovej vnorenej L-shaped metódy), t.j. algoritmus vnorenej L-shaped metódy sa zjednoduší tým, že bude prechádzať strom scenárov len po stupeň $H-1$ s tým, že pri smere prechodu vpred bude v stupni $H-1$ implementovaná uvedená modifikácia. Touto modifikáciou, namiesto riešenia úlohy $NLDS(H, k)$ pre všetky realizácie ξ^H , t.j. K^H krát a následného výpočtu rezov optimality stačí určiť hodnotu π_j^H pre všetky $j \in \{k\}^+$, $k = 1, \dots, K^{H-1}$ a pridať rezy optimality. Týmto sa zvýši efektívnosť vnorených verzií L-shaped metódy.

Na záver tejto kapitoly poznamenávame, že existuje väčšie množstvo metód vyvinutých na riešenie stochastických lineárnych úloh s pevnou kompenzáciou. Viaceré sú však len miernymi modifikáciami metód uvedených v tejto kapitole. Napríklad ekvivalentom k simplexovej metóde je duálna simplexová metóda ([21]). V prípade dekompozičných metód je ekvivalentom Bendersovej dekompozície Dantzing-Wolfeho dekompozícia ([6]), ktorá je založená na linearizácii duálnej úlohy k úlohe (2.2). Prehodením regularizačného člena z účelovej funkcie v prípade regularizovanej L-shaped metódy do ohraničení vznikne trieda tzv. *Bundle trust region methods*. Týchto metód existuje viacero variant a líšia sa v spôsobe aktualizovania parametra β alebo použitím rôznych druhov noriem. V prípade regularizovanej L-shaped metódy sa používa len euklidovská

norma, ale v prípade Bundle trust region methods sú často používanými normami $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$. Viac o tomto type metód sa dá nájsť v [17]. V súčasnosti je predmetom rozsiahleho výskumu aj iný typ dekompozičných metód, než aké sme uviedli v tejto kapitole. Sú to dekompozičné metódy založené na metóde vnútorného bodu. Jedná sa o rozsiahlu triedu dekompozičných metód, o čom svedčí aj veľké množstvo článkov publikovaných v posledných rokoch. Zaujateho čitateľa odkazujeme na články [3], [8], [9], [18], [31] a ich zoznamy literatúr.

Kapitola 4

Numerické výsledky a implementácia

Cieľom tejto kapitoly bude testovanie algoritmov popísaných v Kapitole 3. V prvej časti tejto kapitoly sa budeme venovať algoritmom na riešenie jedno-periódovej úlohy optimalizácie portfólia a náplňou druhej časti budú algoritmy na riešenie viac-periódovej úlohy optimalizácie portfólia. Všetky uvedené algoritmy boli implementované v Matlabe R2011a a následné testovanie bolo realizované na notebooku Toshiba Satellite L650 s procesorom Intel Core i3 M350 2.27GHz, pamäťou (RAM) 4GB a operačným systémom Windows 7 Home Premium. Zameriame sa na testovanie závislosti výpočtového času jednotlivých metód od rôznych parametrov. Konkrétne to budú parameter K zodpovedajúci počtu realizácií náhodného vektora reprezentujúceho výnosy aktív v portfóliu, parameter M určujúci počet aktív v portfóliu a parameter α zodpovedajúci zvolenej pravdepodobnosti pri rizikovej miere Conditional Value-at-Risk. Uvažovať budeme optimalizáciu portfólia s povoleným ($sh = \text{áno}$) a aj s nepovoleným short-sellingom ($sh = \text{nie}$), kde v ďalšom texte bude sh označovať shortselling.

4.1 Jedno-periódová úloha optimalizácie portfólia

V tejto časti sa zameriame na porovnanie numerických výsledkov získaných pomocou jedno a viaczrovej L-shaped metódy, jedno a viaczrovej CVaRmin metódy, regularizovanej L-shaped metódy a priamych metód implementovaných v Matlabe, t.j. simplexová metóda a metóda vnútorného bodu. Očakávame vyššiu efektívnosť oboch verzií metódy CVaRmin, pretože sú špeciálne vyvinuté na riešenie optimalizačných úloh s CVaR v účelovej funkcii. V Kapitole 3 sme uviedli, že viaczrové verzie musia pre tento typ úloh skonvergovať v dvoch makroiteráciách a z tohto dôvodu sa dá predpokladať výhodnosť týchto metód. V prípade regularizovanej verzie L-shaped metódy sa podľa viacerých experimentov potvrdila vyššia efektívnosť tejto metódy ([6]) v porovnaní s ostatnými verziami L-shaped metódy. V našom prípade sa ale táto skutočnosť s určitosťou nepotvrdí, pretože minimálny počet makroiterácií potrebných na skonvergovanie tejto metódy je 2, čo je počet, ktorý viaczrová verzia dosiahne v každom prípade. Z tohto dôvodu by viaczrová verzia L-shaped metódy mala byť najefektívnej-

šou verziou L-shaped metódy a preto sa dá očakávať, že nami navrhnutá viacrezová verzia CVaRmin metódy bude najefektívnejšou metódou spomedzi všetkých metód testovaných v tejto podkapitole.

Pre účel testovania sme použili historické údaje o 60-tich spoločnostiach. Zoznam spoločností je uvedený v prílohe A. Každá spoločnosť reprezentuje jedno aktívum a o vektore výnosov aktív predpokladáme, že pochádza z viacrozmerneho normálneho rozdelenia so strednou hodnotou \bar{r} a kovariančnou maticou Σ , ktoré sme odhadli z historických údajov získaných z *finance.yahoo.com*. Použili sme týždenné údaje za obdobie január 2006 až december 2011. Scenáre k budeme generovať z rozdelenia $\xi_k \sim \mathcal{N}(\bar{r}, \Sigma)$, $k = 1, \dots, K$. Pravdepodobnosť nastatia bude pre všetky scenáre rovnaká, a to $p_k = \frac{1}{K}$, $k = 1, \dots, K$.

Algoritmy boli implementované presne podľa popisu uvedeného v Kapitole 3 s výnimkou ukončovacej podmienky, ktorú bolo potrebné upraviť na tvar:

$$e_{s+1} - E_{s+1}x^\nu - \theta^\nu \leq \varepsilon$$

v prípade jednorezovej L-shaped metódy a na tvar:

$$e_{s_k+1} - E_{s_k+1}x^\nu - \theta_k^\nu \leq \varepsilon, \quad k = 1, \dots, K$$

v prípade viacrezovej L-shaped metódy. Podobným spôsobom sa upravili aj ukončovacie podmienky pri ostatných algoritmoch. Parameter ε sa nazýva ukončovacia tolerancia a počas celého testovania bude nastavený na hodnote $\varepsilon = 10^{-6}$.

Ako je možné vidieť z Obr. 2.1, matica ohraničení úlohy (1.6) je veľmi riedka, t.j. obsahuje veľa nulových prvkov. Z tohto dôvodu sme pri implementácii použili na ukladanie vstupných údajov algoritmov funkciu *sparse*, ktorá ignoruje nulové prvky, čím sa výrazne znížia pamäťové nároky úlohy. Pre lepšiu názornosť uvádzame v Tab. 4.1 veľkosti úloh použitých pri testovaní (rôzne hodnoty parametra K) a ich pamäťové nároky v prípade použitia a nepoužitia funkcie *sparse* pre $M = 10$.

K	riadky	stĺpce	RAM	sRAM
100	102	111	90.58KB	20.45KB
500	502	511	2.05MB	0.1MB
1000	1002	1011	8.1MB	0.2MB
2000	2002	2011	32.21MB	0.4MB
5000	5002	5011	200.52MB	1MB
10000	10002	10011	800.72MB	2MB

Tabuľka 4.1: Veľkosti úloh

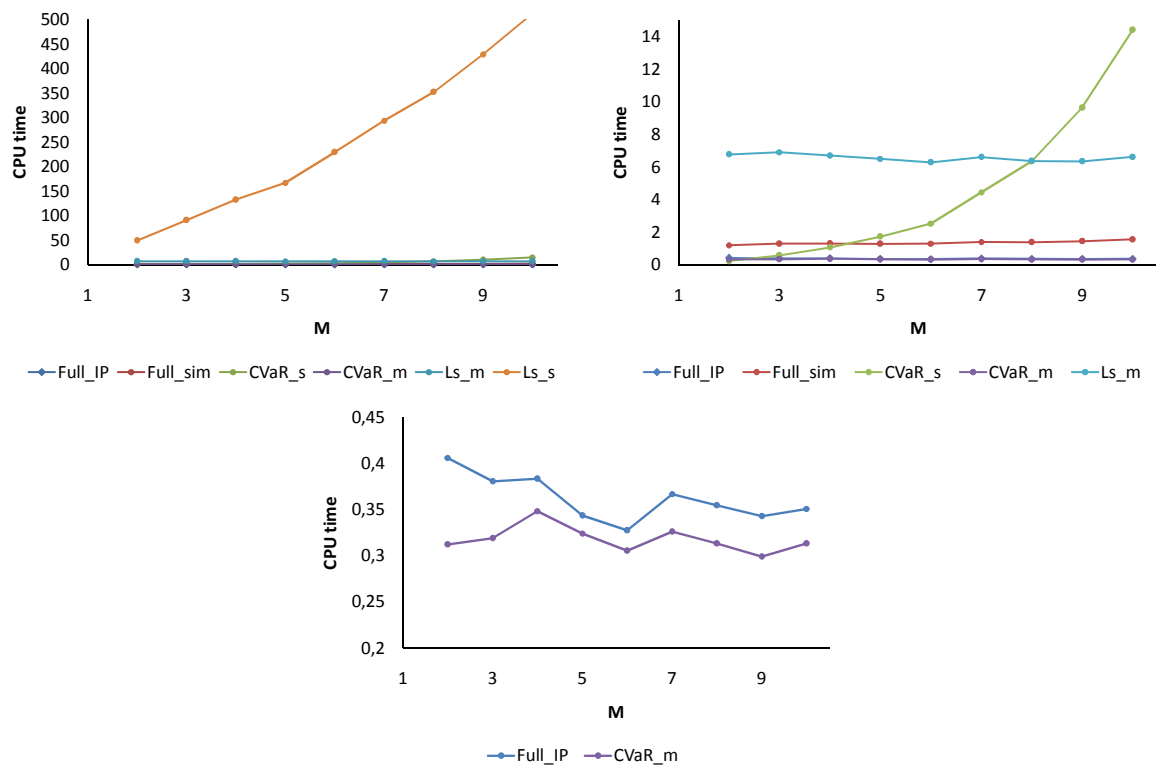
Riadky a stĺpce v Tab. 4.1 zodpovedajú rozmerom matice ohraničení pre jednotlivé hodnoty K , RAM reprezentuje pamäť potrebnú na uloženie matice ohraničení a v stĺpci sRAM je uvedená potrebná pamäť pri použití funkcie *sparse*. V Tab. 4.1 je zreteľná výrazná úspora pamäti pri použití funkcie *sparse*.

Testovanie bude pozostávať z viacerých častí, ktoré budeme nazývať numerický experiment. Sled numerických experimentov bude nasledovný. Najprv sa zameriame na závislosť výpočtového času od parametra M . Tejto časti budú zodpovedať dva numerické experimenty. Postupne budeme uvažovať optimalizačné úlohy s menším počtom aktív, t.j. $M \in \{2, \dots, 10\}$ a úlohy s väčším počtom aktív, t.j. $M \in \{10, 20, 30, 40, 50, 60\}$. Ďalšie dva numerické experimenty budú sledovať závislosť výpočtového času od parametra K postupne pre počty aktív $M = 5$ a $M = 60$, kde $K \in \{100, 500, 1000, 2000, 5000, 10000\}$. Ku každému numerickému experimentu priložíme jeden obrázok a jednu tabuľku. Obrázok bude obsahovať niekoľko grafov porovnávajúcich výpočtový čas jednotlivých metód, pričom porovnanie na každom ďalšom grafe bude bez výrazne horších metód, čím sa docielu podrobné porovnanie výpočtového času najefektívnejších metód. Na x -ovej osi grafov budú uvedené hodnoty príslušného parametra a na y -ovej osi výpočtový čas (CPU time) v sekundách. Výpočtové časy v grafoch budú priemerami z niekoľkých spustení. Konkrétny počet spustení bude špecifikovaný pri príslušnom experimente a pod spustením rozumieme jedno vygenerovanie realizácií výnosov a následné spustenie uvažovaných metód. Tabuľka bude obsahovať výpočtový čas najrýchlejšej metódy a pre ostatné metódy uvedieme, koľkonásobne bol ich čas horší v porovnaní s najrýchlejšou metódou. Taktiež bude obsahovať aj hodnoty ostatných (netestovaných) parametrov použitých pri príslušnom numerickom experimente. Ostatné údaje budú uvedené v tabuľkách v tabuľkovej prílohe. Tieto tabuľky budú pre každú uvažovanú metódu obsahovať priemerný, minimálny a maximálny výpočtový čas, priemerný, minimálny a maximálny počet makroiterácií potrebných na skonvergovanie a priemernú a maximálnu odchýlku. Pod odchýlkou budeme rozumieť vzdialenosť (euklidovskú normu) riešenia príslušnej metódy od riešenia priamej metódy vnútorného bodu, ktorá, ako uvidíme neskôr, sa ukázala byť najspoľahlivejšou. Ďalším numerickým experimentom bude testovanie závislosti výpočtového času od parametra α . Pre tento experiment priložíme jeden obrázok obsahujúci šesť grafov prislúchajúcich jednotlivým metódam. V každom numerickom experimente uvažujeme $sh = \text{áno}$ a aj $sh = \text{nie}$. Označenie jednotlivých metód bude počas celého testovania nasledovné: Full_IP - metóda vnútorného bodu, Full_sim - simplexová metóda, CVaR_s - jednorezová CVaRmin metóda, CVaR_m - viacrezová CVaRmin metóda, Ls_s - jednorezová L-shaped metóda, Ls_m - viacrezová L-shaped metóda. Celé testovanie realizujeme bez uvažovania regularizovanej L-shaped metódy, pretože, ako uvedieme neskôr v numerickom experimente v podkapitole 4.1.4, táto metóda je nevhodnou pre použitie na optimalizáciu úlohy (1.6). Týmto sme popísali proces testovania a v ďalšom texte priložíme a stručne popíšeme výsledky jednotlivých numerických experimentov. Začneme s testovaním závislosti výpočtového času od počtu aktív.

4.1.1 Závislosť výpočtového času od parametra M

Pri tomto testovaní sme v prípade Ls_s uskutočnili 20 spustení a pre ostatné metódy sme realizovali 50 spustení. Menší počet spustení pre Ls_s, ako uvidíme na Obr. 4.1,

má za následok výrazne vyššia časová náročnosť optimalizácie pomocou tejto metódy.



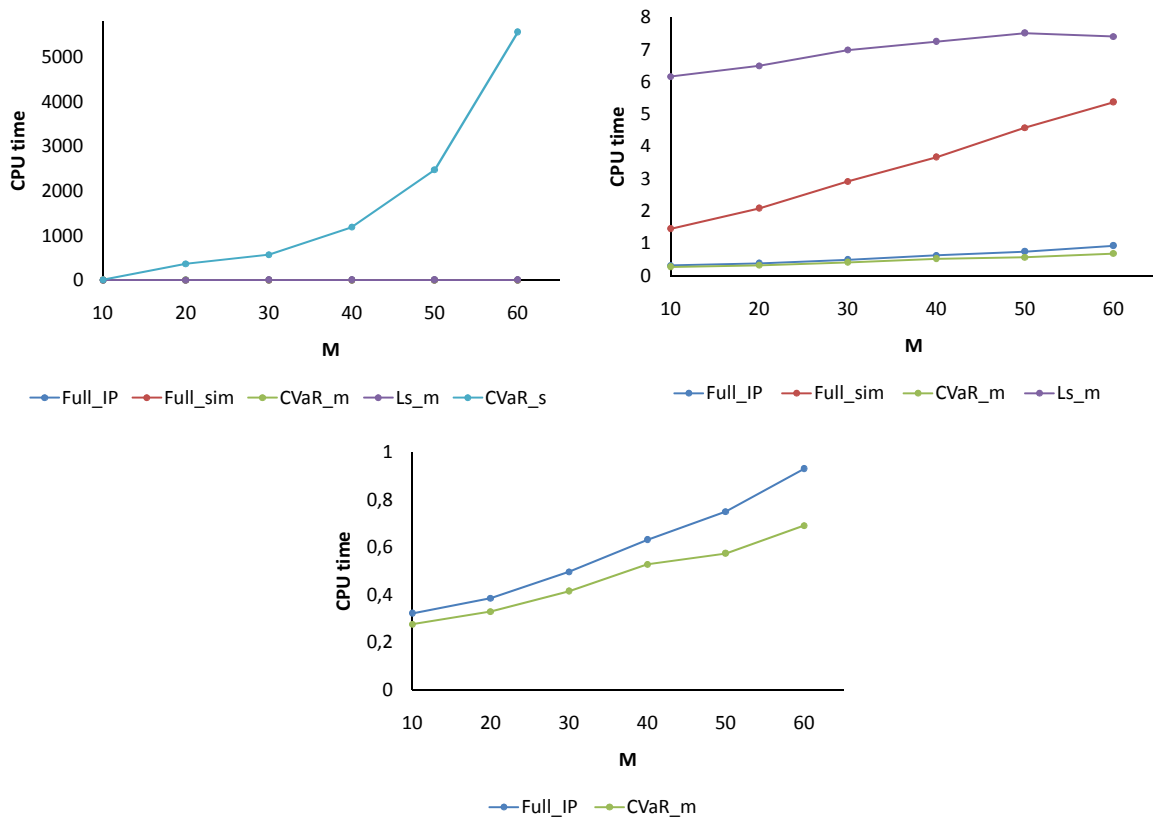
Obr. 4.1: Výpočtový čas v závislosti od počtu aktív

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$						
M	CVaR_m	Full_IP	Full_sim	Ls_m	CVaR_s	Ls_s
2	0.3122	1.31	3.82	21.86	0.69	151.79
3	0.3190	1.22	4.08	22.04	1.79	273.73
4	0.3484	1.14	3.83	19.93	3.13	349.22
5	0.3241	1.03	3.80	19.39	5.13	521.07
6	0.3054	1.03	4.03	19.70	7.89	766.34
7	0.3260	1.14	4.29	20.52	13.78	851.56
8	0.3133	1.10	4.25	19.75	19.72	1106.09
9	0.2989	1.12	4.65	20.64	31.40	1442.12
10	0.3133	1.12	4.92	21.12	46.08	1561.15

Tabuľka 4.2: Porovnanie výpočtového času pre rôzne M

Z tab. 4.2 a Obr. 4.1 vidieť, že najrýchlejšou bola CVaR_m, ktorej čas sa pohyboval v rozmedzí 0.2989s – 0.3484s. Výrazne najpomalšou bola Ls_s, ktorej časy konvergenzie boli od 151.79 do 1561.15 násobne väčšie v porovnaní s CVaR_m. Pri $M = 10$ skonvergovanie v priemere trvalo približne 8 minút. V prípade jednorezových verzií je jasne viditeľná rastúca závislosť výpočtového času od M , t.j. sú citlivé na M . Naopak,

výpočtový čas ostatných metód nezávisí od M . Obr. 4.2 a Tab. 4.3 zobrazujú výsledky experimentu pre väčšie hodnoty parametra M . Tento experiment sme realizovali bez Ls_s , kvôli veľkej citlivosti metódy na veľkosť parametra M . Počet spustení sme zvolili 10 pre $CVaR_s$ a 50 pre ostatné metódy.



Obr. 4.2: Výpočtový čas v závislosti od počtu aktív

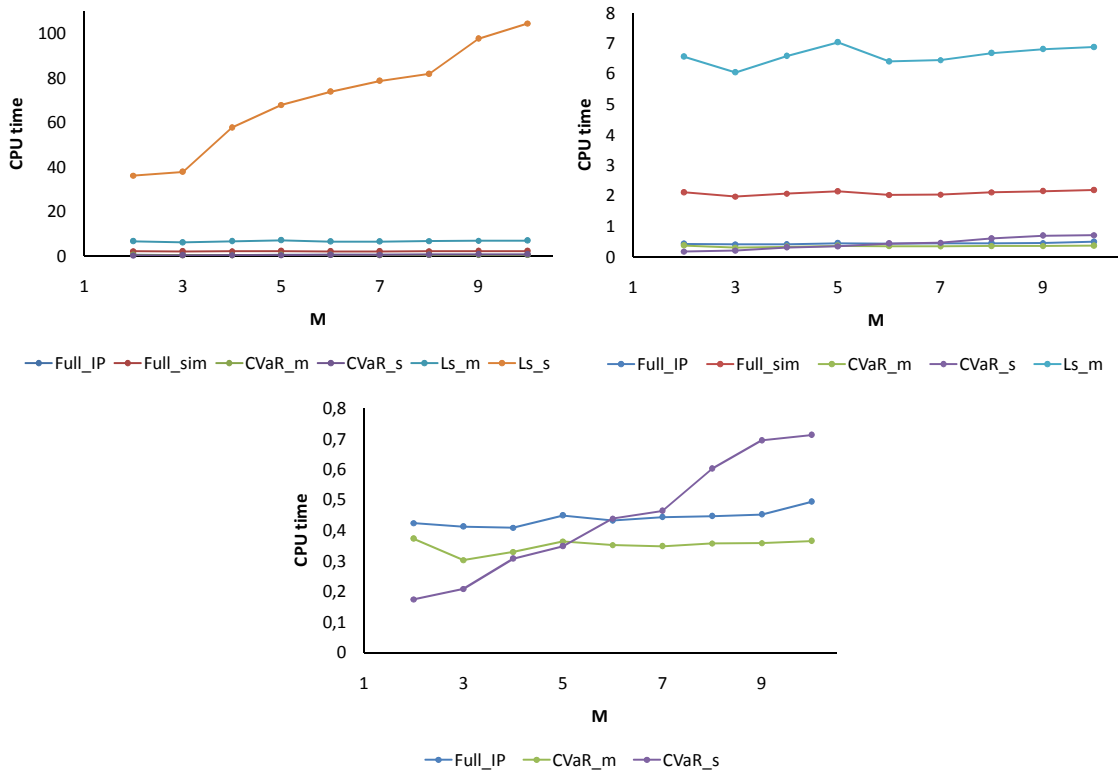
$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$					
M	CVaR_m	Full_IP	Full_sim	Ls_m	CVaR_s
10	0.2758	1.14	5.27	23.26	47.21
20	0.3290	1.18	6.35	20.27	1169.51
30	0.4148	1.19	7.04	17.27	1383.63
40	0.5278	1.25	6.96	13.58	2283.57
50	0.5737	1.38	7.99	13.48	4427.94
60	0.6912	1.42	7.77	11.37	8332.58

Tabuľka 4.3: Porovnanie výpočtového času pre rôzne M

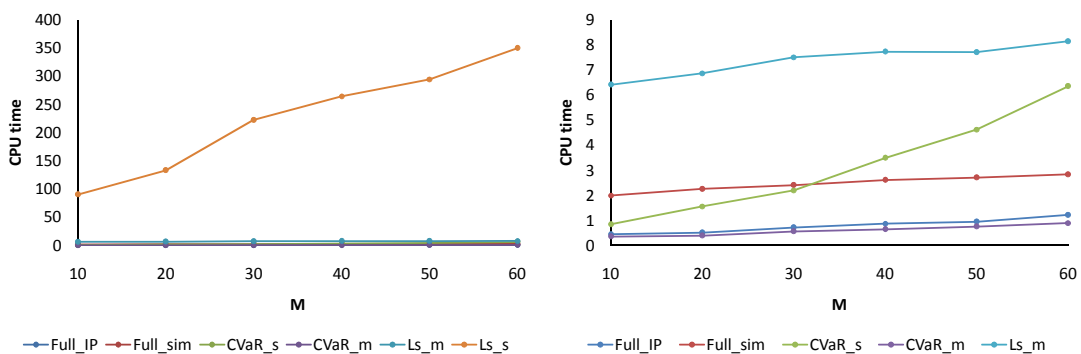
Týmto experimentom sa nám potvrdila veľká citlivosť $CVaR_s$ metódy na veľkosť parametra M , kde pri $M = 60$ metóda konvergovala v priemere za 98 minút. Táto metóda bola najpomalšou s časom konvergenencie, ktorý je takmer v každom prípade niekoľko tisíc násobne väčší než je čas najlepšej metódy. Najefektívnejšou metódou

bola CVaR_m, ktorá konvergovala v čase z intervalu $0.2758s - 0.6912s$ v závislosti na parametri M . V tomto prípade sa dá pozorovať určitá závislosť u všetkých metód. Najcitlivejšou metódou je CVaR_s (spolu s Ls_s) a najmenej citlivou je Ls_m.

V ďalšom texte zopakujeme predchádzajúce testovania pre $sh = \text{nie}$. Počet spustení sme zvolili 50 pre všetky metódy.



Obr. 4.3: Výpočtový čas v závislosti od počtu aktív



Obr. 4.4: Výpočtový čas v závislosti od počtu aktív

Na Obr. 4.3 a v Tab. 4.4 sú zobrazené výsledky pre menšie hodnoty parametra M a na Obr. 4.4 a v Tab. 4.5 sú uvedené výsledky numerického experimentu pre väčšie

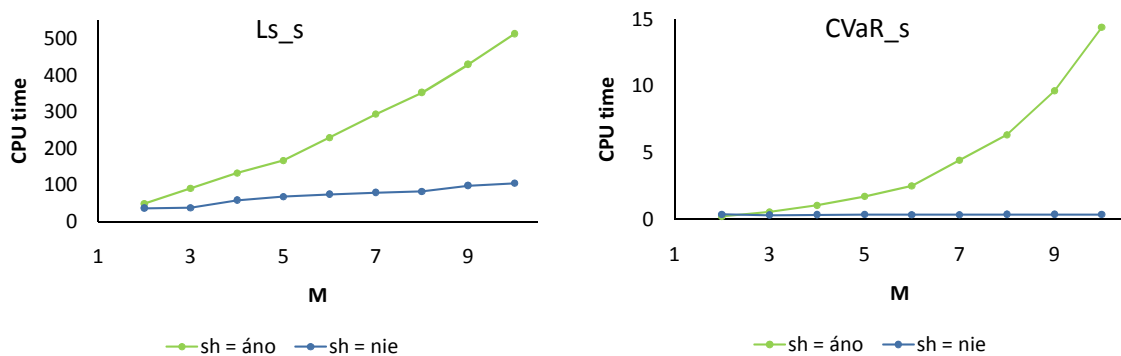
$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$						
M	CVaR_m	Full_IP	CVaR_s	Full_sim	Ls_m	Ls_s
2	0.3727	1.14	0.47	5.67	17.61	92.40
3	0.3028	1.36	0.69	6.53	19.97	125.17
4	0.3294	1.24	0.93	6.29	19.98	177.53
5	0.3640	1.23	0.96	5.89	19.32	182.58
6	0.3522	1.23	1.25	5.77	18.18	206.43
7	0.3482	1.27	1.33	5.86	18.52	229.52
8	0.3574	1.25	1.68	5.92	18.69	227.84
9	0.3585	1.26	1.94	6.01	18.98	261.28
10	0.3652	1.35	1.95	5.99	18.83	278.81

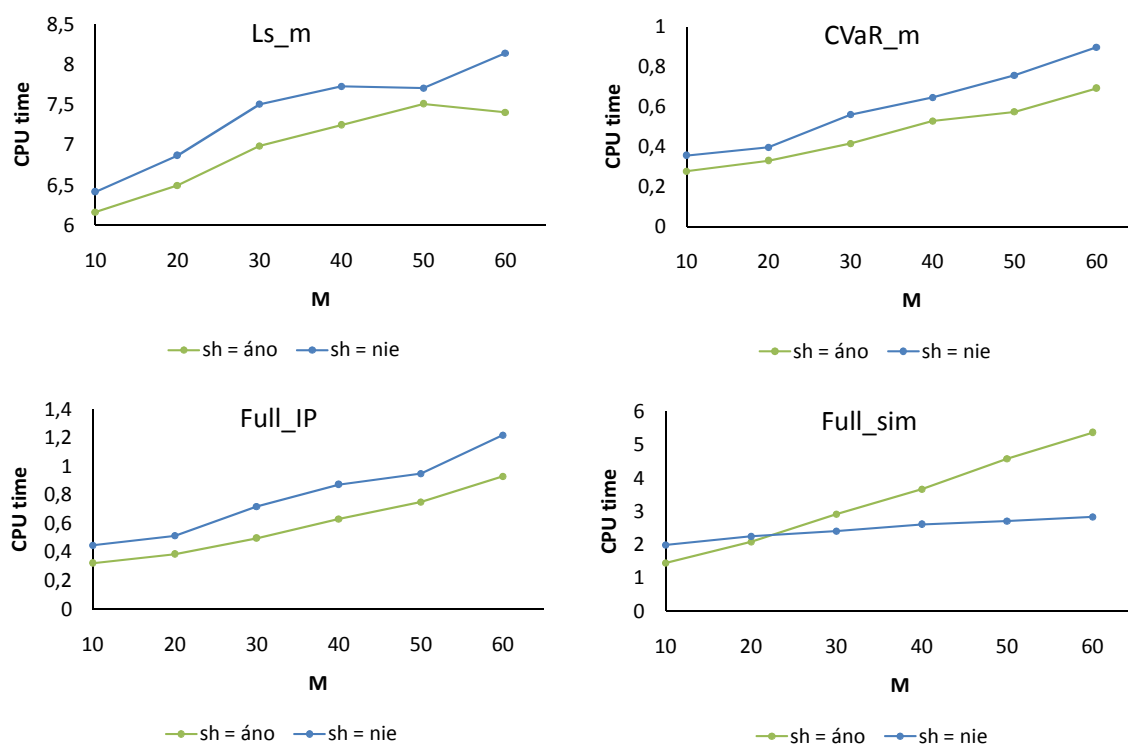
Tabuľka 4.4: Porovnanie výpočtového času pre rôzne M

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$						
M	CVaR_m	Full_IP	CVaR_s	Full_sim	Ls_m	Ls_s
10	0.3546	1.26	2.52	5.62	19.18	255.21
20	0.3955	1.30	3.90	5.70	18.21	337.19
30	0.5599	1.28	3.90	4.30	13.48	397.43
40	0.6455	1.35	5.34	4.05	12.13	409.66
50	0.7567	1.25	6.20	3.58	10.79	389.25
60	0.8964	1.36	7.28	3.16	9.28	390.53

Tabuľka 4.5: Porovnanie výpočtového času pre rôzne M

hodnoty M . V prípade $sh = \text{nie}$ je najefektívnejšou metódou CVaR_m a najhoršou Ls_s, ktorá je aj najcitlivejšou metódou na veľkosť parametra M . Najmenej citlivou metódou sa v tomto prípade zdá byť Full_sim. Z porovnania výsledkov numerických experimentov je zrejmé, že v prípade jednorezových verzií, pri podmienke nezápornosti váh v portfóliu ($sh = \text{nie}$), sa jedná o výrazné zefektívnenie optimalizačného procesu a zníženie závislosti (citlivosti) od parametra M . Metóda CVaR_s sa stáva porovnateľnou s dvoma doposiaľ najlepšími metódami (CVaR_m a Full_IP). Zavedenie ohraničenia nezápornosti váh, v prípade ostatných metód, nespôsobilo žiadne výrazné zmeny v efektívnosti optimalizácie, čo je graficky znázornené na Obr. 4.5.



Obr. 4.5: Porovnanie výpočtového času pre $sh = \text{áno}$ a $sh = \text{nie}$

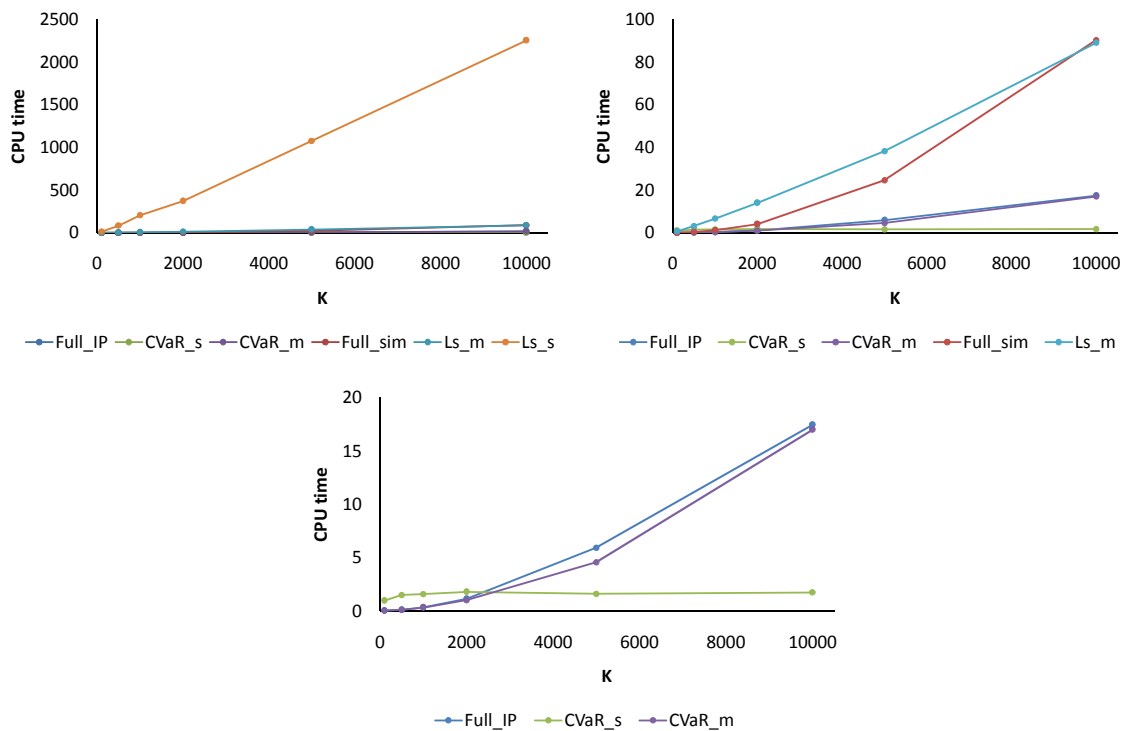
4.1.2 Závislosť výpočtového času od parametra K

V tomto experimente sa pozrieme na závislosť výpočtového času od parametra K . Ako možno vidieť z Obr. 4.6 a Tab. 4.6, až na metódu CVaR_s sme dostali rastúcu závislosť výpočtového času. Metóda CVaR_s je v tomto prípade, pri väčších počtoch scenárov, výrazne efektívnejšia než všetky ostatné metódy. Jej čas konvergencie sa v závislosti od K pohyboval medzi 0.9774 a 1.7548.

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$						
K	CVaR_s	CVaR_m	Full_IP	Full_sim	Ls_m	Ls_s
100	0.9774	0.07	0.05	0.06	0.69	13.96
500	1.5095	0.08	0.09	0.26	2.14	55.77
1000	1.5975	0.20	0.22	0.77	4.21	124.33
2000	1.8098	0.57	0.62	2.14	7.60	195.97
5000	1.6212	2.85	3.71	14.48	23.59	641.73
10000	1.7548	10.22	10.55	51.38	52.34	1277.21

Tabuľka 4.6: Porovnanie výpočtového času pre rôzne K

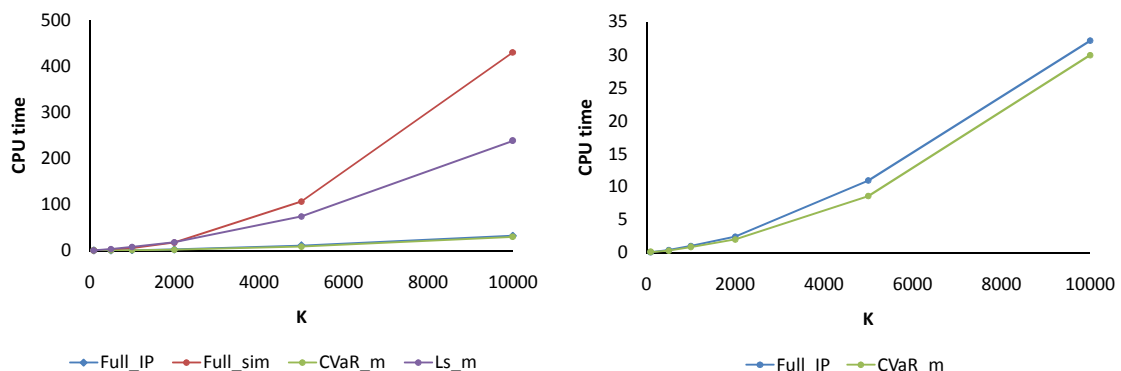
Najpomalšou je Ls_s, ktorá, ako môžeme vidieť v Tab. 4.6, je opäť mnohonásobne pomalšou v porovnaní s ostatnými metódami. V prípade $K = 10000$ potrebovala na



Obr. 4.6: Výpočtový čas v závislosti od počtu scenárov

zkonvergovanie v priemere 40 minút. Počet spustení sme zvolili rovnako ako v prvom experimente, teda 20 pre Ls_s a 50 pre ostatné metódy.

Na Obr. 4.7 a v Tab. 4.7 sú zobrazené výsledky rovnakého numerického experimentu ako v predchádzajúcom prípade, len hodnota parametra M sa zmenila na 60. V tomto experimente neuvažujeme metódy Ls_s a $CVaR_s$, nakoľko sa ukázalo, že sú veľmi citlivé na veľkosť parametra M , čo by predstavovalo obrovskú časovú náročnosť tohto experimentu. Počet spustení bol v tomto prípade rovnaký pre všetky metódy (50).



Obr. 4.7: Výpočtový čas v závislosti od počtu scenárov

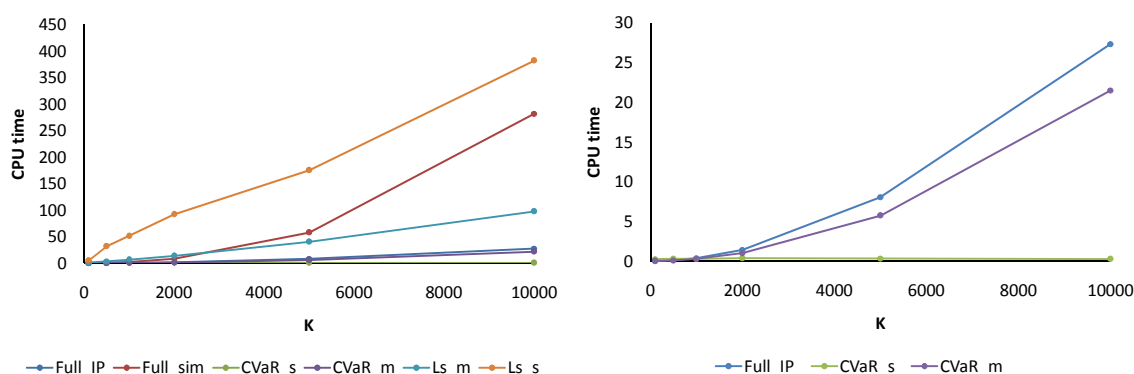
Najrýchlejšou metódou bola znova $CVaR_m$ a jej výpočtový čas sa pre rôzne K

$M = 60, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$				
K	CVaR_m	Full_IP	Ls_m	Full_sim
100	0.0934	0.80	7.08	5.58
500	0.3068	1.32	11.06	6.73
1000	0.8527	1.21	9.17	6.42
2000	2.0072	1.22	9.07	8.64
5000	8.5951	1.27	8.59	12.35
10000	29.9441	1.07	7.94	14.34

Tabuľka 4.7: Porovnanie výpočtového času pre rôzne K

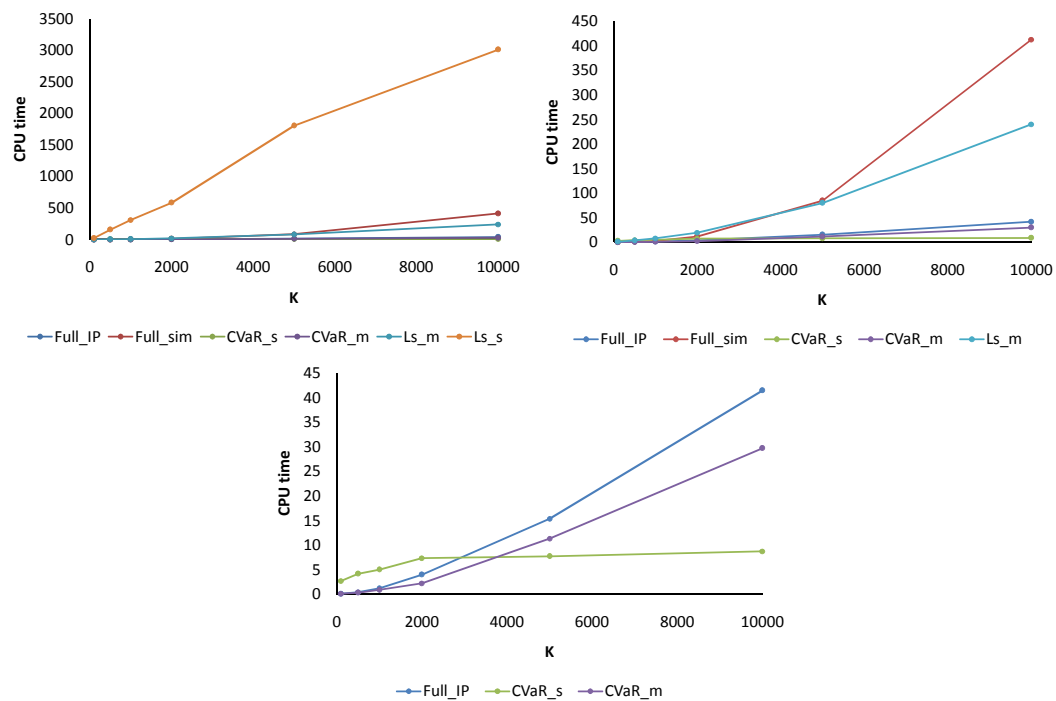
nachádza v intervale $0.0934s - 29.9441s$. Najdlhšie (takmer 8 min pri $K = 10000$) konvergovala metóda Full_sim.

Rovnaké dve testovania sme urobili aj pre prípad zavedenia ohraničení nezápornosti na váhy v portfóliu. Tu sme nevyklúčovali jednorezové verzie, pretože sa ukázalo, že pri nepovolení shortsellingu nie je citlivosť na parameter M až taká výrazná. Na Obr. 4.8, Obr. 4.9, v Tab. 4.8 a v Tab. 4.9 sú uvedené výsledky pre obe testovania. Počet spustení bol pre všetky metódy v oboch experimentoch nastavený na 50, len v prípade Ls_s a $M = 60$ sme zvolili hodnotu 10. V obidvoch experimentoch sa ukázala ako jasne najefektívnejšia metóda CVaR_s, ktorá skonvergovala za $0.2884s$ pri $K = 10000$ a $M = 5$, respektíve za $8.7161s$ pri $K = 10000$ a $M = 60$. Výpočtový čas druhej najrýchlejšej metódy (CVaR_m) pri tých istých hodnotách M a K bol 74.49 , respektíve 3.41 krát väčší. Najhoršou metódou, podobne ako v predchádzajúcich experimentoch, bola metóda Ls_s. Na Obr. 4.10 je graficky znázornené porovnanie výpočtového času pre $sh = \text{áno}$ a $sh = \text{nie}$. V prípade jednorezových verzií znova dostávame výraznú časovú úsporu pri zavedení ohraničení nezápornosti váh. V prípade priamych metód je situácia opačná, t.j. väčšiu efektívnosť dosahujú tieto metódy pri povolení shortsellingu. Viacrezové verzie sú, rovnako ako pri parametri M , nezávislé na type ohraničenia váh.



Obr. 4.8: Výpočtový čas v závislosti od počtu scenárov

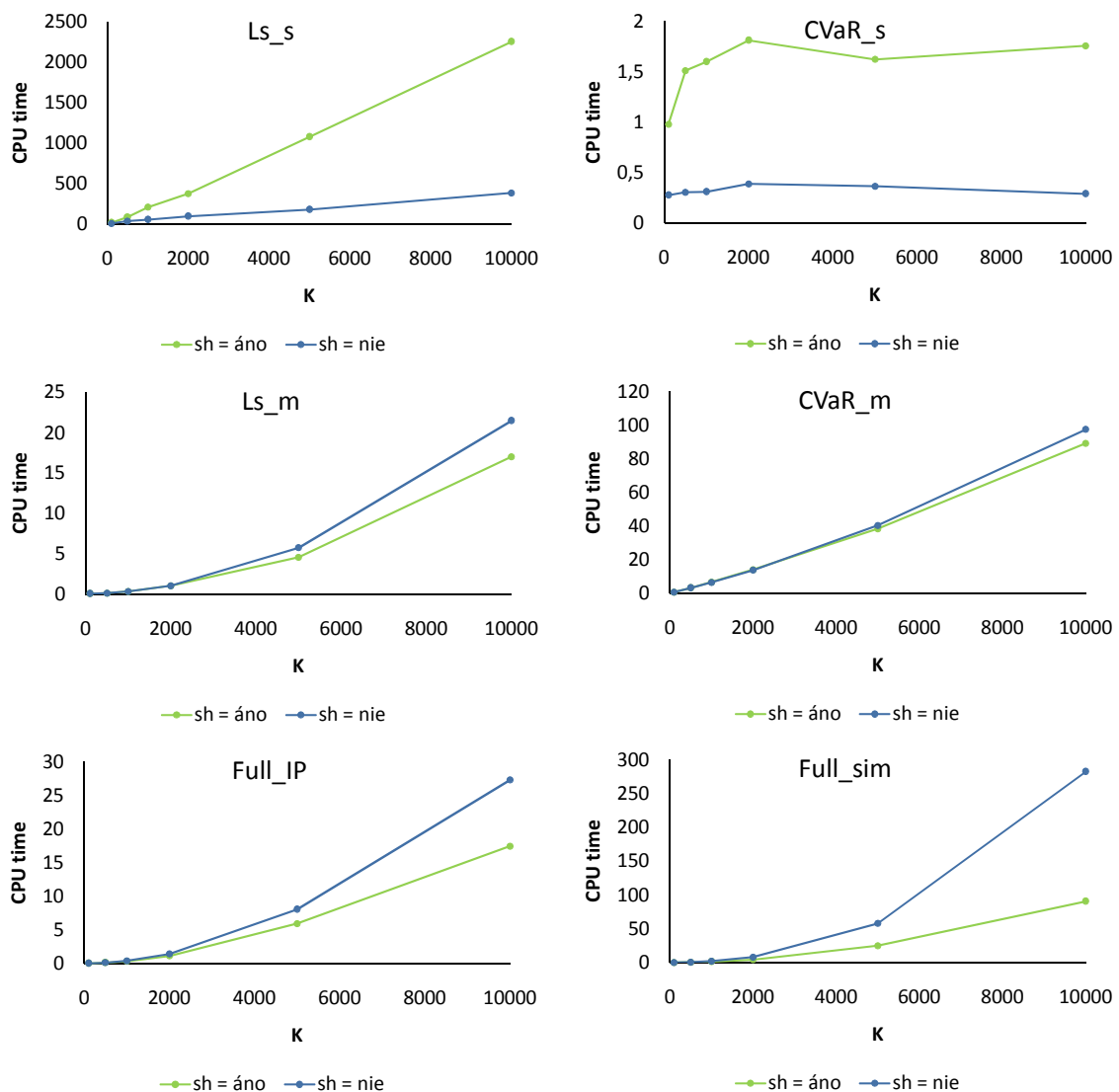
$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = nie$						
K	CVaR_s	CVaR_m	Full_IP	Ls_m	Full_sim	Ls_s
100	0.2744	0.19	0.14	2.46	0.30	18.09
500	0.3038	0.42	0.52	10.60	2.03	102.60
1000	0.3078	1.03	1.49	21.43	6.78	167.09
2000	0.3863	2.68	3.81	36.00	18.94	232.75
5000	0.3630	15.38	22.77	110.94	152.84	467.85
10000	0.2884	73.44	91.23	340.60	953.78	1284.97

Tabuľka 4.8: Porovnanie výpočtového času pre rôzne K 

Obr. 4.9: Výpočtový čas v závislosti od počtu scenárov

$M = 60, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = nie$						
K	CVaR_s	CVaR_m	Full_IP	Ls_m	Full_sim	Ls_s
100	2.7022	0.04	0.03	0.27	0.04	8.55
500	4.1629	0.08	0.10	0.84	0.19	38.06
1000	5.0191	0.18	0.23	1.57	0.55	61.86
2000	7.3320	0.30	0.54	2.62	1.59	79.60
5000	7.7392	1.46	1.98	10.31	10.96	233.32
10000	8.7161	3.41	4.76	27.54	47.35	345.45

Tabuľka 4.9: Porovnanie výpočtového času pre rôzne K

Obr. 4.10: Porovnanie výpočtového času pre $sh = \text{áno}$ a $sh = \text{nie}$

Pri testovaní závislosti výpočtového času od parametra K sme nastavili maximálnu testovanú hodnotu na 10000. Táto hodnota bola poslednou, pre ktorú sme dokázali na nami použitom notebooku bez dodatočných nastavení využívania pamäte v Matlabe alebo v operačnom systéme, riešiť optimalizačné úlohy pomocou priamych metód a viacrezových verzií (viacrezové verzie pridávajú pre každý scenár jeden rez optimality a preto je v konečnom dôsledku veľkosť hlavnej úlohy podobná s veľkosťou deterministického ekvivalentu). V prípade jednorezových verzií sú možnosti použitia oveľa väčšie. V Tab. 4.10 a 4.11 sú uvedené výsledky numerického experimentu pre veľké hodnoty parametra K . Uvažujeme len jednorezovú verziu CVaRmin metódy, pretože z predošlých testovaní vidieť, že Ls_s je veľmi neefektívnou metódou.

Z tabuliek vidieť, že jednorezová CVaRmin metóda je použiteľná a veľmi efektívna aj pri obrovských počtoch scenárov. V prípade povolenia shortsellingu ($sh = \text{áno}$) je

$M = 10, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$						
K	CPU time			Iterácie		
	priem.	min	max	priem.	min	max
10000	0.9650	0.2933	2.2421	36	14	67
50000	1.1677	0.6219	1.7995	38	21	56
100000	1.3549	1.0125	1.8065	38	30	47
500000	3.0360	2.5270	3.8807	38	31	46
1000000	5.1752	4.3483	6.2505	37	31	50
2000000	9.7621	8.1637	11.5004	39	31	46
5000000	20.6453	17.7399	26.0382	34	29	45
10000000	42.3758	37.8262	47.7627	35	31	49

Tabuľka 4.10: Výpočtový čas CVaRmin v závislosti od K

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$						
K	CPU time			Iterácie		
	priem.	min	max	priem.	min	max
10000	2.1472	1.3531	2.9890	75	54	88
50000	2.6363	2.1531	3.2722	81	70	89
100000	2.8805	2.3717	3.3455	82	75	91
500000	5.2191	4.6912	5.8257	82	75	89
1000000	8.2630	7.2692	9.1090	82	72	90
2000000	13.9408	11.7551	14.8696	82	70	87
5000000	31.0357	28.6271	33.3411	82	75	89
10000000	62.7287	56.6899	69.5264	81	76	88

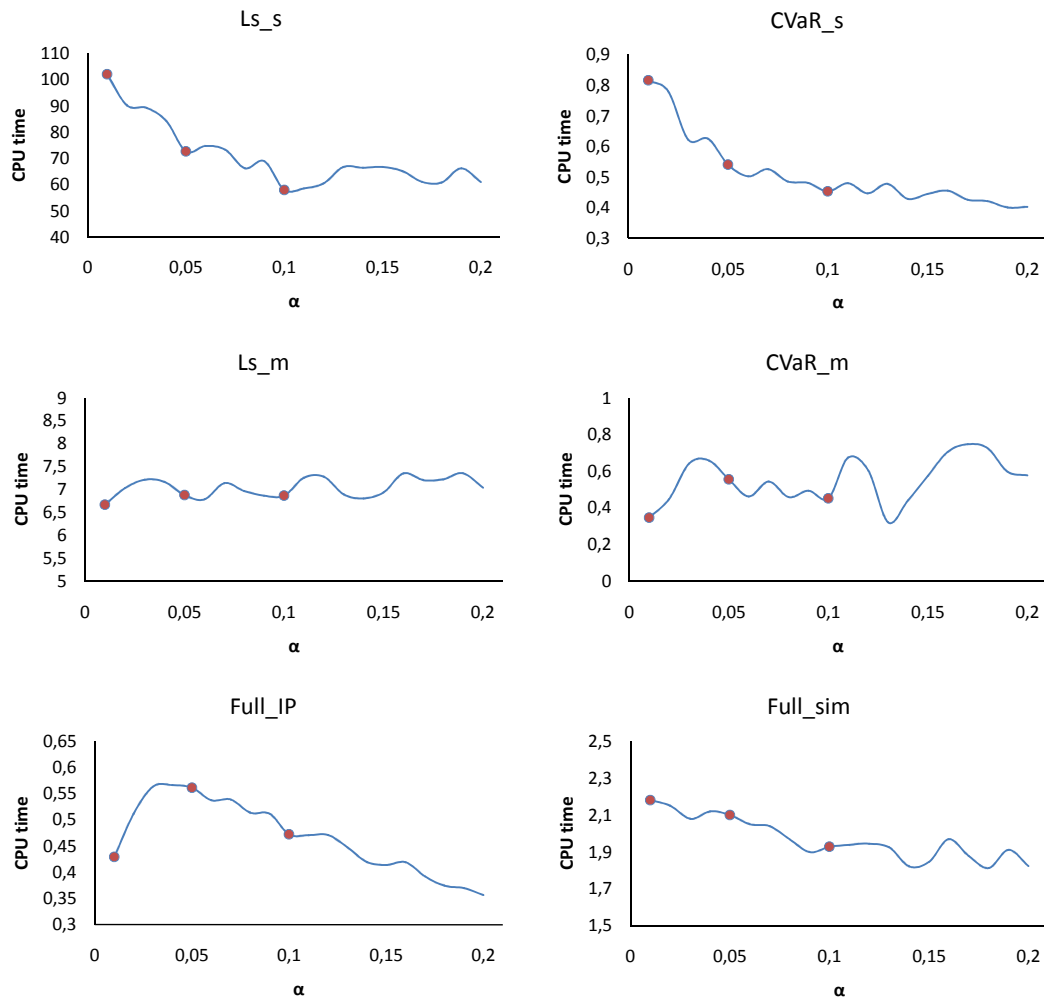
Tabuľka 4.11: Výpočtový čas CVaRmin v závislosti od K

jej použitie obmedzené len pre veľmi malé počty aktív v portfóliu (Obr. 4.2), ale v opačnom prípade, ako sa ukázalo v predošlom testovaní, sa dá táto metóda efektívne použiť na optimalizáciu portfólia s väčším počtom aktív. Pre $K = 10000000$ (maximum pre nami použitý notebook) metóda skonvergovala za 42.3758s, respektíve 62.7287s, čo je čas, ktorý nedosiahli niektoré metódy ani pri $K = 10000$.

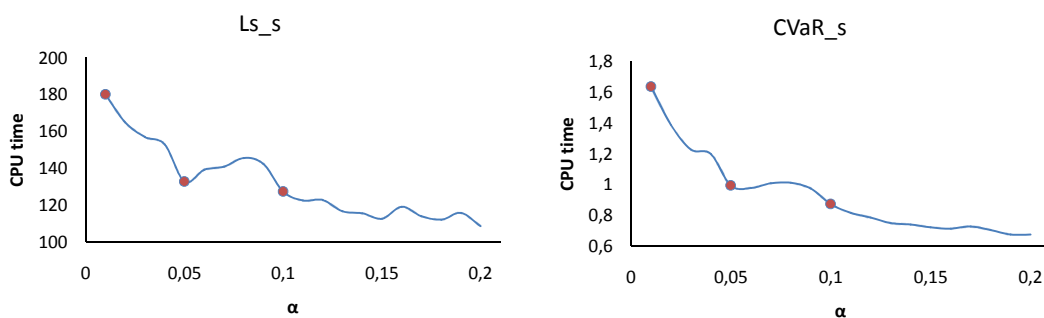
4.1.3 Závislosť výpočtového času od parametra α

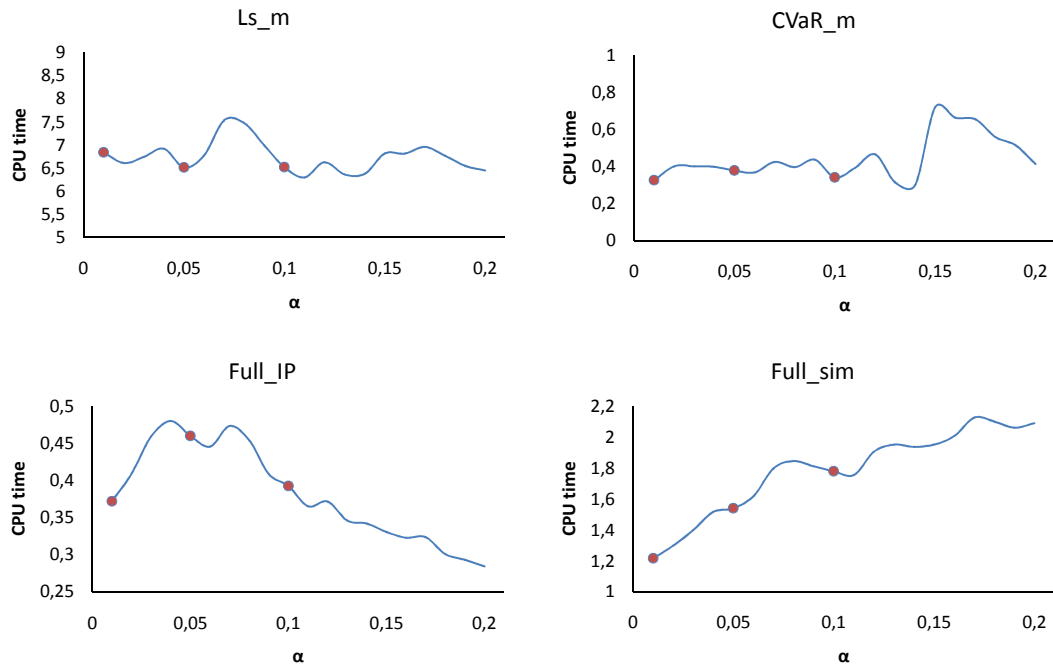
Ďalej nás bude zaujímať, či majú rôzne hodnoty parametra α vplyv na výpočtový čas jednotlivých metód. Na Obr. 4.11 ($sh = \text{áno}$) a 4.12 ($sh = \text{nie}$) sú graficky zobrazené závislosti výpočtového času od parametra α , ktorého hodnoty uvažujeme z intervalu 0.01 – 0.2. Na každom z grafov sú zvýraznené často používané hodnoty α , t.j. 0.01, 0.05 a 0.1. Počet spustení pre tento numerický experiment bol zvýšený, v snahe dostať čo najhladšiu krivku závislosti výpočtového času, na hodnotu 100 pre všetky metódy. Z grafov vidieť, že pre jednorezové verzie dostávame v oboch prípadoch klesajúcu závislosť. Rovnakú závislosť v oboch prípadoch dostávame aj pre metódu Full_IP, kde po hodnotu 0.05 výpočtový čas rastie a od tejto hodnoty klesá. V prípade viaczových

verzií sme dostali veľmi „divoké“ krivky a teda nie je možné určiť akúkoľvek tendenciu výpočtového času. Zaujímavá závislosť je viditeľná v prípade simplexovej metódy, kde pri povolení shortsellingu dostávame klesajúcu závislosť a v opačnom prípade ide o rastúcu závislosť.



Obr. 4.11: Porovnanie výpočtového času pre α , $sh = \text{áno}$, $K = 1000$, $M = 5$, $\mu = 0.0015$, $\varepsilon = 0.000001$





Obr. 4.12: Porovnanie výpočtového času pre α , $sh = \text{nie}$, $K = 1000$, $M = 10$, $\mu = 0.0015$, $\varepsilon = 0.000001$

4.1.4 Regularizovaná L-shaped metóda

Doposiaľ sme sa nevenovali regularizovanej verzii L-shaped metódy. Teraz uvedieme dôvody, prečo sme túto metódu vynechali z predošlého testovania. Pre použitie tejto metódy je potrebné na začiatku optimalizačného procesu zvoliť štartovací bod. V nasledujúcom texte porovnáme výpočtový čas regularizovanej metódy pre rôzne štartovacie body s viacrezovou verzou L-shaped metódy, keďže regularizovaná metóda bola odvodená práve z tejto metódy. V príkladoch z literatúry ([6],[25]) sa uvažujú jednoduché

$K = 500$, $M = 5$, $\alpha = 0.01$, $\mu = 0.0015$, $\varepsilon = 0.000001$, $sh = \text{áno}$							
Štart. bod	Čas (sec.)			Iterácie			neskonverg.
	priem.	min	max	priem.	min	max	
a_1	160.6	31.6	275.6	13	4	19	3
a_2	189.9	44.9	282.4	16	5	20	4
a_3	192.8	48.7	276.4	16	6	20	4
a_4	119.8	35.4	248.2	11	4	16	2
a_5	153.6	40.0	259.6	13	5	18	1
a_6	160.2	31.3	276.8	14	4	20	3
a_7	179.6	80.3	302.0	16	8	22	5
a_8	180.6	47.4	264.8	15	6	19	5
Ls_m	3.2453	3.1704	3.5510	2	2	2	0

Tabuľka 4.12: Porovnanie regularizovaných metód

štartovacie body, napr. vektor jednotiek, núl atď. a preto aj my budeme uvažovať štartovacie body tohto typu. V Tab. 4.12 sú uvedené výsledky numerického experimentu, ktorým sa potvrdilo naše očakávanie, a to, že regularizovaná metóda nebude efektívnejšia ako viacrezová verzia L-shaped metódy. Uvažované štartovacie body sú $a_1 = 0_M^T$, $a_2 = 1_M^T$, $a_3 = -1_M^T$, $a_4 = \frac{1}{M}1_M^T$, $a_5 = -\frac{1}{M}1_M^T$, $a_6 = +(-)\frac{1}{M}1_M^T$ a $a_7 = +(-)1_M^T$. Označenie $+(-)$ znamená, že jednotlivé prvky štartovacieho bodu majú znamienko podľa znamienka očakávaných výnosov, t.j. záporné očakávané výnosy znamenajú záporné váhy a opačne, a_8 reprezentuje štartovací bod zodpovedajúci riešeniu len s jedným scenárom, ktorý zodpovedá očakávaným výnosom. V riadkoch a_1 až a_8 sú uvedené výsledky regularizovanej L-shaped metódy s príslušným štartovacím bodom a riadok Ls_m zodpovedá viacrezovej L-shaped metóde. V stĺpci neskonverg. je uvedený počet neskonvergovaní, pričom pod neskonvergovaním sa rozumie aj dosiahnutie hornej hranice makroiterácií, ktorá bola nastavená na hodnote 30. Počet spustení sa rovnal 20.

Z Tab. 4.12 je zjavné, že regularizovaná metóda je pre tento typ úloh výrazne horšia než viacrezová verzia, hoci vo všeobecnosti bolo experimentmi preukázané ([6]), že je efektívnejšia v porovnaní s obidvoma verziami L-shaped metódy. Dokonca je horšou aj v porovnaní s Ls_s, ktorá v predchádzajúcich experimentoch dosahovala najhoršie výsledky. Za povšimnutie stojí aj veľký rozdiel medzi minimálnym a maximálnym výpočtovým čas (v porovnaní s Ls_m), čo svedčí o veľkej citlivosti na to, aké scenáre sa vygenerujú a taktiež aj počet neskonvergovaní, kde v niektorých prípadoch dosiahla miera neskonvergovania až 25%. Toto pozorovanie sme považovali za postačujúce na to, aby sme túto metódu ďalej pri numerických experimentoch neuvažovali .

4.1.5 Zhrnutie

Pred zhrnutím ešte uvedieme posledné pozorovanie, ktorým bude úspešnosť alebo spoľahlivosť jednotlivých metód, t.j. v koľkých percentách z celkového počtu spustení jednotlivé metódy neskonvergovali. Celkovo sme realizovali 39170 spustení, t.j. približne 6500 pre každú z metód a výsledky tohto pozorovania sú zobrazené v Tab. 4.13. Aj v tomto prípade rozdelíme pozorovanie na dve časti, t.j. pre prípad povolenia a aj nepovolenia shortsellingu, pretože ako vidieť z Tab. 4.13, úspešnosť niektorých metód sa pre jednotlivé prípady výrazne líši.

Medzi vcelku spoľahlivé metódy môžeme zaradiť viacrezové verzie (CVaR_m, Ls_m) a metódu vnútorného bodu. Tieto metódy neskonvergovali len v 0.165%, respektíve 0.03%. Menej spoľahlivými metódami sú simplexová metóda (1.93%) a jednorezové verzie (1.09% CVaR_s, 1.5% Ls_s).

Na koniec testovania metód pre riešenie jedno-periódovej úlohy optimalizácie portfólia si zhrnieme jednotlivé experimenty. Za najuniverzálnejšie možno označiť viacrezové verzie, hlavne kvôli tomu, že tieto metódy neprejavili žiadnu výraznú citlivosť pri parametri M , t.j. je možné ich použiť pri optimalizačných úlohách s väčším počtom aktív. Typ ohraničenia na váhy taktiež nemá vplyv na výpočtový čas ako vidieť z Obr. 4.5 a 4.10. V prípade parametra α nebolo možné určiť žiaden typ závislosti, a teda

shortselling	Spustení			Neskon.			Neskon./Spustení(%)		
	áno	nie	celkom	áno	nie	celkom	áno	nie	celkom
Full_IP	3350	3350	6700	0	2	2	0	0.06	0.03
Full_sim	3350	3350	6700	25	104	129	0.75	3.11	1.93
CVaR_s	3210	3750	6960	34	42	76	1.06	1.12	1.09
CVaR_m	3350	3350	6700	5	6	11	0.15	0.18	0.165
Ls_s	2300	3110	5410	54	27	81	2.35	0.87	1.5
Ls_m	3350	3350	6700	5	6	11	0.15	0.18	0.165

Tabuľka 4.13: Úspešnosť jednotlivých metód

sa dá povedať, že tento parameter nemá vplyv na výpočtový čas viacrezových verzií. V prospech univerzálnosti tiež hovorí aj počet neskonvergovaní, ktorý je v oboch prípadoch ($sh = \text{áno}$ a $sh = \text{nie}$) veľmi nízky a takmer rovnaký. Jedinou veľkou nevýhodou týchto metód je skutočnosť, že je ich možné použiť len pre počet scenárov do hodnoty 10000. Z tejto dvojice metód dosahovala výrazne lepšie výsledky metóda CVaR_min a preto je možné túto metódu označiť za najvhodnejšiu pre optimalizačné úlohy s väčším počtom aktív a s povoleným shortsellingom.

Priame metódy dosiahli dobré výsledky, keď takmer vo všetkých experimentoch boli lepšie ako obidve verzie L-shaped metódy. Metóda vnútorného bodu dokonca dosahovala vo všetkých experimentoch porovnateľnú efektívnosť s CVaR_m metódou a tak častokrát bola druhou najefektívnejšou metódou. Bola aj najspoľahlivejšou, kde v prípade povolenia shortsellingu skonvergovala vždy a opačnom prípade neskonvergovala len štyrikrát. Táto štatistika neplatí v prípade simplexovej metódy, ktorá bola najnespoľahlivejšou metódou, keď pri nepovolení shortsellingu neskonvergovali až 3.11% testovaných úloh, celkovo to bolo 1.93%.

Jednerezové verzie patrili v testovaní medzi najcitlivejšie metódy, kde v každom experimente bolo možné pozorovať nejakú závislosť výpočtového času. Metóda Ls_s bola jednoznačne najhoršou v každom numerickom experimente a spolu s regularizovanou metódou je nevhodná na riešenie úloh typu (1.6). Na druhej strane, pri metódach CVaR_s je zaujímavým javom minimálna závislosť na počte scenárov. Táto skutočnosť spolu s vedomosťou, že metóda pridáva len jeden rez optimality pre všetky scenáre, dovoľujú jej použitie aj pri obrovskom počte scenárov. V prípade menšieho počtu aktív v portfóliu je táto metóda výrazne efektívnejšia ako CVaR_m a pri zavedení ohraničenia nezápornosti na váhy dominuje aj pre väčšie hodnoty parametra M . V týchto prípadoch je najvhodnejšou voľbou použitie metódy CVaR_s. Jedinou nevýhodou je len veľká citlivosť na M pre $sh = \text{áno}$, čo je aj prípad kedy je vhodnejšie použiť CVaR_m. Za pozornosť stojí aj sedemkrát väčší počet neskonvergovaných úloh v porovnaní s CVaR_m.

Čo sa týka našich očakávaní, potvrdila sa výrazná výhodnosť metód špeciálne vyvinutých na riešenie optimalizačných úloh s CVaR v účelovej funkcii, kde jednerezová CVaRmin metóda je jednoznačne najefektívnejšou metódou až na prípad povolenia

shortsellingu a väčší počet aktív v portfóliu, kedy je najlepšie použiť nami odvodenú viacrezovú CVaRmin metódu. Z časti sa potvrdila výhodnosť viacrezových metód, kde viacrezová verzia L-shaped metódy bola stále lepšou v porovnaní s jednorezovou verziou a naša CVaR_m prekonala CVaR_s práve v prípade povolenia shortsellingu a väčšieho M . Na druhej strane sa nepotvrdila väčšia efektívnosť dekompozičných metód, kde priame metódy častokrát výrazne prekonali verzie L-shaped metódy.

Pre kompletnosť uvádzame v prílohe B tabuľky pre numerické experimenty s parametrami M a K . Konkrétne uvádzame tabuľky pre závislosť výpočtového času Ls_s od menšieho počtu aktív, pre ostatné metódy uvádzame závislosť výpočtového času od väčšieho počtu aktív. Taktiež pre všetky metódy uvádzame závislosť výpočtového času od K . Všetky tieto závislosti uvádzame aj pre povolený shortselling ($sh = \text{áno}$) a nepovolený shortselling ($sh = \text{nie}$).

4.2 Viac-periódová úloha optimalizácie portfólia

V tejto časti porovnáme numerické výsledky získané pomocou jednorezovej a viacrezovej vnorenej L-shaped metódy a modifikovanej jedno a viacrezovej vnorenej L-shaped metódy. Naše očakávania sú v tomto prípade podobné tým z predchádzajúcej časti a to, že viacrezové verzie budú efektívnejšie než jednorezové a taktiež, že modifikované verzie budú výhodnejšie než pôvodné verzie vnorenej L-shaped metódy. Pri očakávaníach o výhodnosti viacrezových verzií treba však brať do úvahy skutočnosť, že kompenzačné matice vo všeobecnosti nepatria do skupiny úplných kompenzačných matíc, čo môže mať za dôsledok pridávanie rezov prípustnosti a z toho plynúce zpomalenie optimalizačného procesu.

Pre účel testovania sme použili tie isté historické údaje ako v predchádzajúcom testovaní, t.j. jednotlivé realizácie náhodného vektora výnosov pochádzajú z rozdelenia $\xi_k^t \sim \mathcal{N}(\bar{r}, \Sigma)$, $k = 1, \dots, K^t$, $t = 1, \dots, H$. Strom scenárov budeme generovať nasledovne. Pre každý uzol v každom stupni (okrem koreňa stromu) vygenerujeme príslušnú realizáciu ξ_k^t . Pravdepodobnosti nastatia jednotlivých realizácií závisia od toho, v ktorom stupni sa príslušná realizácia nachádza, t.j. $p_k^t = \frac{1}{K^t}$, $k = 1, \dots, K^t$, $t = 1, \dots, H$. Ukončovacie podmienky boli upravené rovnakým spôsobom ako v predchádzajúcom testovaní a na ukládanie vstupných údajov sme použili funkciu sparse.

Testovať budeme postupne závislosť výpočtového času od počtu aktív M , od počtu scenárov K^H (počtu všetkých možných ciest od koreňa k listom stromu) a od parametra α . Opäť ku každému numerickému experimentu priložíme grafické porovnanie jednotlivých metód a tabuľku, v ktorej bude uvedené porovnanie modifikovaných metód s pôvodnými metódami. V každom numerickom experimente uvažujeme aj povolenie shortsellingu ($sh = \text{áno}$) a nepovolenie shortsellingu ($sh = \text{nie}$). Označenie metód bude nasledovné: VLs_s - jednorezová vnorená L-shaped metóda, MLs_s - modifikovaná jednorezová vnorená L-shaped metóda, VLs_m - viacrezová vnorená L-shaped metóda, MLs_m - modifikovaná viacrezová vnorená L-shaped metóda.

Pred samotným testovaním uvádzame veľkosti riešených úloh. Parameter K reprezentuje počet realizácií náhodného vektora výnosov v danom uzle, t.j. počet nasledovníkov daného uzla (pre všetky uzly v stupni $t = 0, \dots, H - 1$ uvažujeme rovnakú hodnotu parametra K). Uvádzame aj veľkosť uvažovaných úloh pre testovanie závislosti výpočtového času od parametra M , pretože v prípade viac-periódovej úlohy optimalizácie portfólia má hodnota parametra M značný vplyv na veľkosť úlohy. Pri testovaní závislosti od parametra K uvažujeme $H = 5$, t.j. päť časových období a pri testovaní závislosti od parametra M uvažujeme $H = 4$, t.j. štyri časové periódy. Riadky a stĺpce v Tab. 4.14 zodpovedajú rozmerom matice ohraničení úlohy (1.7) a stĺpec K^H označuje počet scenárov v príslušnej úlohe.

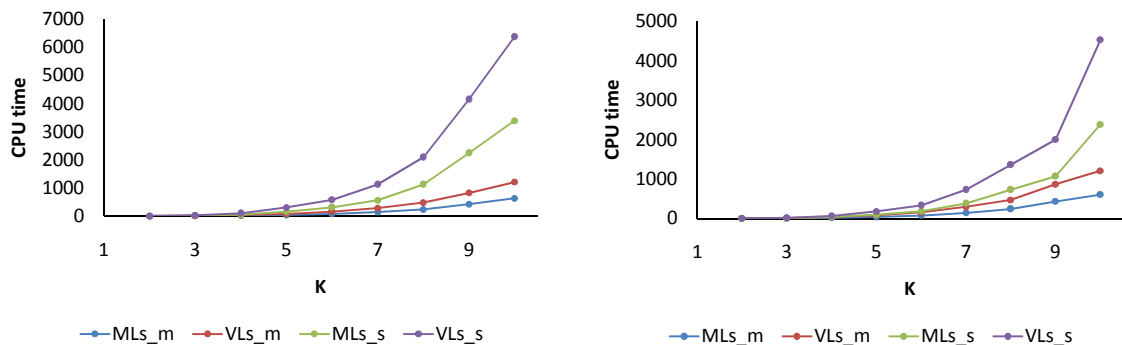
$H = 5, M = 5$				$H = 4, K = 10, K^H = 10000$		
K	Riadky	Stĺpce	K^H	M	Riadky	Stĺpce
2	93	248	32	2	12221	14443
3	484	1089	243	5	12221	17776
4	1705	3410	1024	8	12221	21109
5	4686	8591	3125	11	12221	24442
6	10885	18660	7776	14	12221	27775
7	22408	36413	16807	17	12221	31108
8	42129	65534	32768	20	12221	34441
9	73810	110715	59049			
10	122221	177776	100000			

Tabuľka 4.14: Rozmery matice ohraničení testovaných úloh pre rôzne hodnoty K a pre rôzne hodnoty M

4.2.1 Závislosť výpočtového času od parametra K

Na Obr. 4.13 je uvedené grafické porovnanie závislosti výpočtového času od parametra K . Na grafe vľavo je porovnanie pre prípad $sh = \text{áno}$ a na grafe vpravo je porovnanie pre prípad $sh = \text{nie}$. V oboch prípadoch bola jasne najefektívnejšou metódou MLs_m, ktorá pri $K = 10$, čo je 100000 scenárov, skonvergovala v priemere za 629.13s pri $sh = \text{áno}$, respektíve za 605.51s pri $sh = \text{nie}$. Najpomalšou metódou bola VLs_s, ktorá pri tom istom počte scenárov potrebovala na skonvergovanie v priemere 6366.01s pri $sh = \text{áno}$, respektíve 4528.35s pri $sh = \text{nie}$. Z oboch grafov na Obr. 4.13 je zreteľná výhodnosť modifikovaných verzií vnorenej L-shaped metódy. Táto výhodnosť je uvedená v Tab. 4.15 a 4.16, kde stĺpec označený zefekt. reprezentuje percentuálne zefektívnenie optimalizačného procesu v prípade použitia modifikovanej metódy v porovnaní s pôvodnou metódou. Pri viacrezovej verzii sme dosiahli zefektívnenie na úrovni 48% až 54% a v prípade jednorezovej verzie to bolo 45% až 50%. Na Obr. 4.14 je porovnanie výpočtového času pre $sh = \text{áno}$ a $sh = \text{nie}$, kde v prípade viacrezových verzií sa nedá pozorovať žiadna citlivosť na hodnotu parametra sh . To isté sa nedá povedať o

jednerezových verziách, ktoré pri $sh = \text{nie}$ konvergujú výrazne rýchlejšie než pri $sh = \text{áno}$. Počet spustení v tomto numerickom experimente bol 10.



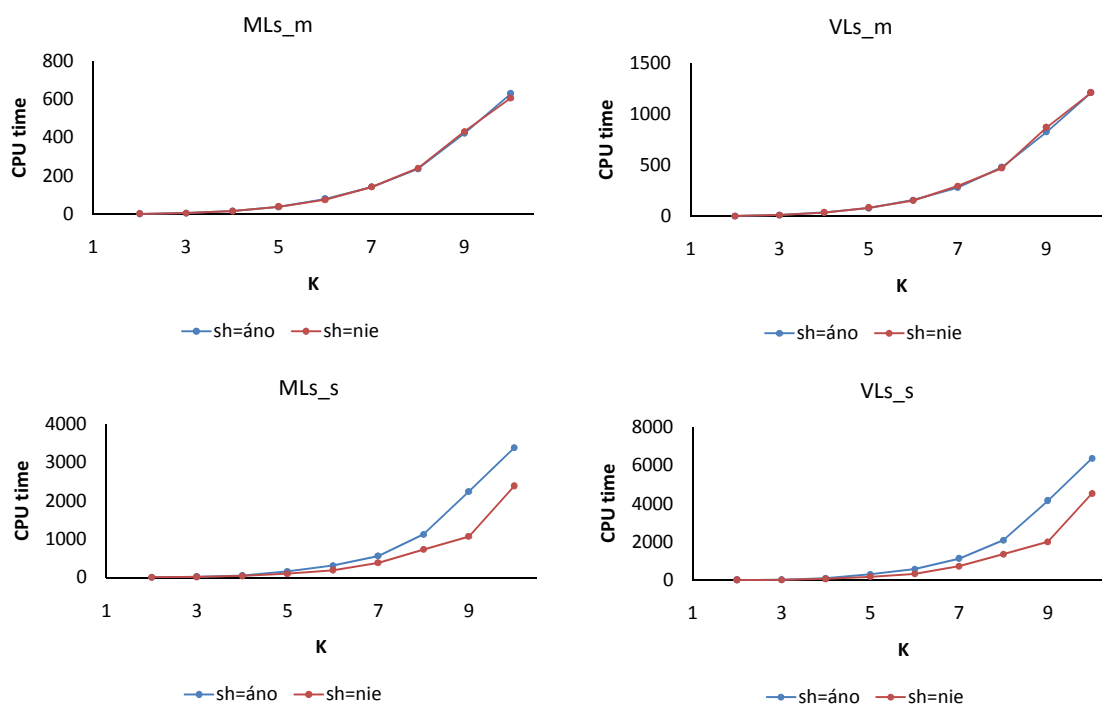
Obr. 4.13: Závislosť výpočtového času od K pre $sh = \text{áno}$ (vľavo) a $sh = \text{nie}$ (vpravo)

$M = 5, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{áno}$						
K	MLs_m	VLs_m	zefekt.	MLs_s	VLs_s	zefekt.
2	1.36	2.87	0.53	2.56	4.71	0.46
3	5.78	11.84	0.51	14.47	26.47	0.45
4	16.91	36.53	0.54	50.95	99.35	0.49
5	37.89	78.82	0.52	154.30	302.43	0.49
6	79.54	157.50	0.49	309.46	578.69	0.47
7	142.75	279.79	0.49	559.44	1124.48	0.50
8	236.77	480.64	0.51	1123.30	2089.33	0.46
9	422.25	823.39	0.49	2240.84	4145.55	0.46
10	629.13	1207.94	0.48	3386.17	6366.01	0.47

Tabuľka 4.15: Porovnanie modifikovaných a pôvodných verzií vnorenej L-shaped metódy pre $sh = \text{áno}$

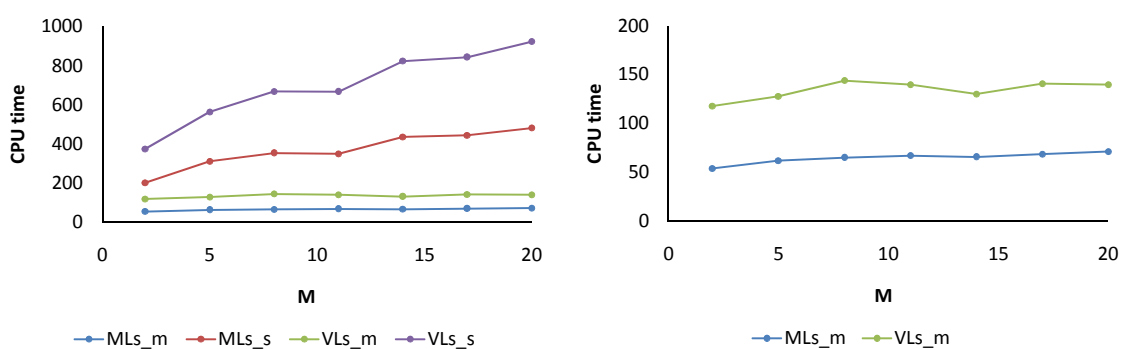
$M = 5, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{nie}$						
K	MLs_m	VLs_m	zefekt.	MLs_s	VLs_s	zefekt.
2	1.30	2.58	0.49	1.58	2.93	0.46
3	5.63	12.15	0.54	9.33	18.28	0.49
4	16.11	35.12	0.54	35.89	67.12	0.47
5	38.65	83.10	0.53	96.15	178.46	0.46
6	75.68	153.63	0.51	183.18	331.93	0.45
7	142.65	293.86	0.51	381.18	728.82	0.48
8	239.91	470.22	0.49	728.88	1359.35	0.46
9	431.76	867.85	0.50	1069.04	1999.11	0.47
10	606.51	1206.95	0.50	2383.34	4528.35	0.47

Tabuľka 4.16: Porovnanie modifikovaných a pôvodných verzií vnorenej L-shaped metódy $sh = \text{nie}$

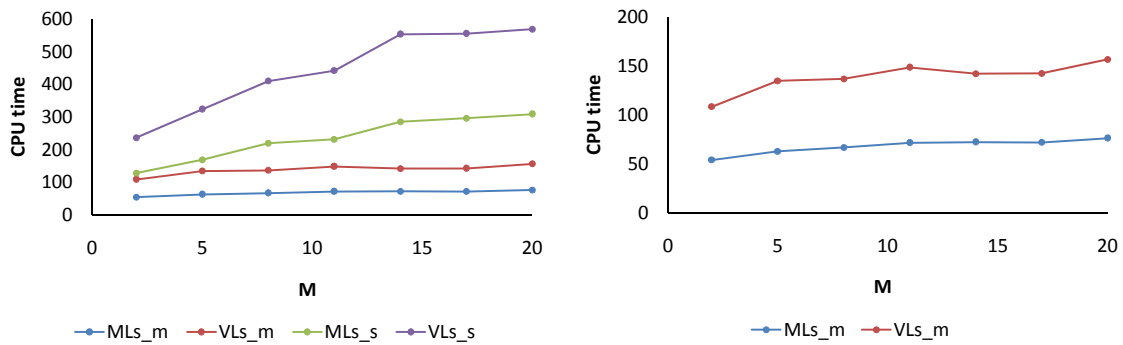
Obr. 4.14: Porovnanie výpočtového času pre $sh = \text{áno}$ a $sh = \text{nie}$

4.2.2 Závislosť výpočtového času od parametra M

V tomto numerickom experimente budeme sledovať závislosť výpočtového času od parametra $M \in \{2, \dots, 20\}$. Z Obr. 4.15 a 4.16 vidieť, že dostávame veľmi podobné výsledky ako pri jedno-periódových úlohách, t.j. jednorezové verzie sú podstatne citlivejšie na zmenu hodnoty parametra M než viacrezové verzie.

Obr. 4.15: Závislosť výpočtového času od M pre $sh = \text{áno}$

Aj v tomto numerickom experimente bola narychlejšou metóda MLs_m, ktorej výpočtový čas takmer nezávisí na hodnote parametra M a pohybuje sa medzi 53s až 72s. Najneefektívnejšou je opäť VLS_s, ktorá je veľmi citlivá na zmenu hodnoty parametra M a v najhoršom prípade potrebovala k skonvergovaniu 922.88s. Modi-

Obr. 4.16: Závislosť výpočtového času od M pre $sh = \text{nie}$

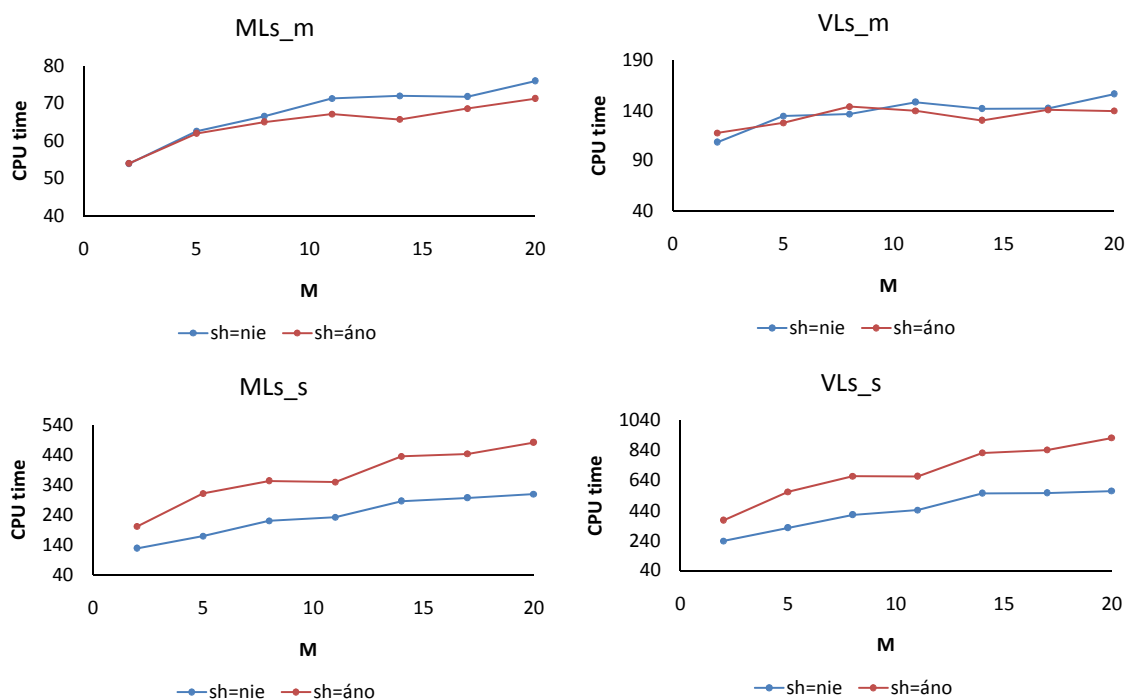
fikované verzie sú aj v tomto prípade výrazne efektívnejšie a miera zefektívnenia sa pohybuje, podobne ako pri parametri K , medzi 49% až 55% pri viacrezovej verzii a medzi 45% až 48% pri jednorezovej verzii. Toto pozorovanie je zobrazené v Tab. 4.17 a 4.18. Na Obr. 4.17 prikladáme porovnanie výpočtového času jednotlivých metód pre $sh = \text{áno}$ a $sh = \text{nie}$. Počet spustení bol nastavení na hodnotu 10.

$K = 10, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{áno}$						
M	MLs_m	VLs_m	zefekt.	MLs_s	VLs_s	zefekt.
2	53.96	117.63	0.54	200.81	373.50	0.46
5	61.94	127.60	0.51	310.82	563.20	0.45
8	64.98	143.87	0.55	352.68	667.62	0.47
11	67.13	139.67	0.52	348.92	666.43	0.48
14	65.69	130.07	0.49	434.99	823.01	0.47
17	68.61	140.65	0.51	443.20	842.68	0.47
20	71.23	139.60	0.49	480.88	922.88	0.48

Tabuľka 4.17: Porovnanie modifikovaných a pôvodných verzií vnorenej L-shaped metódy pre $sh = \text{áno}$

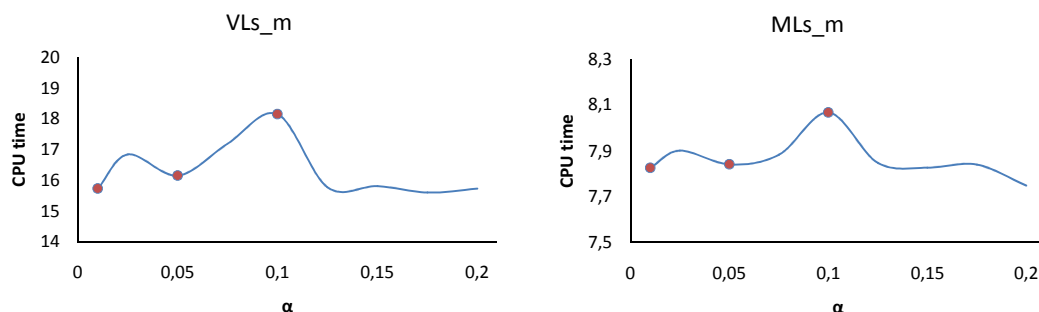
$K = 10, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{nie}$						
M	MLs_m	VLs_m	zefekt.	MLs_s	VLs_s	zefekt.
2	53.85	108.25	0.50	127.77	236.38	0.46
5	62.56	134.51	0.53	168.57	323.65	0.48
8	66.57	136.47	0.51	219.57	410.60	0.47
11	71.31	148.32	0.52	231.67	442.49	0.48
14	71.98	141.79	0.49	285.52	553.91	0.48
17	71.77	142.10	0.49	296.37	555.69	0.47
20	75.94	156.44	0.51	308.55	569.27	0.46

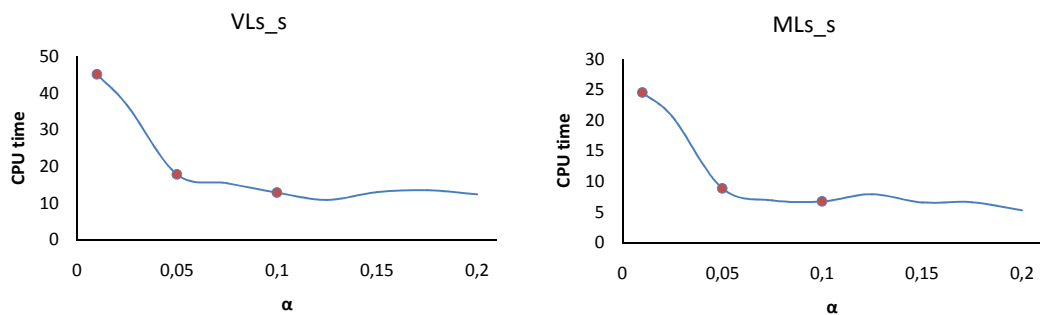
Tabuľka 4.18: Porovnanie modifikovaných a pôvodných verzií vnorenej L-shaped metódy $sh = \text{nie}$

Obr. 4.17: Porovnanie výpočtového času pre $sh = \text{áno}$ a $sh = \text{nie}$

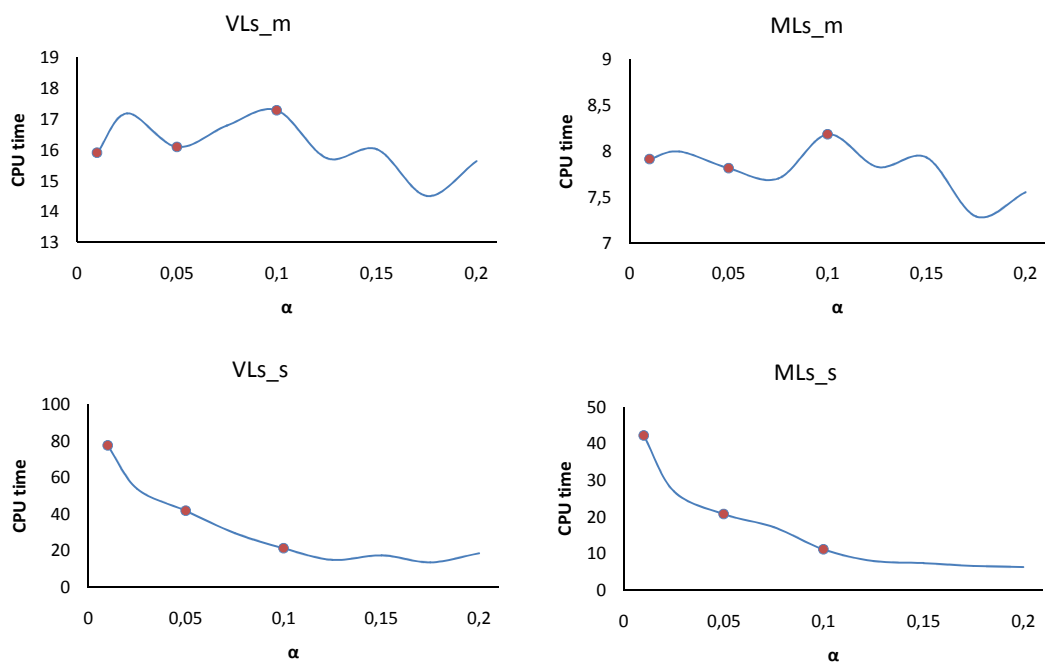
4.2.3 Závislosť výpočtového času od parametra α

V tejto časti budeme sledovať závislosť výpočtového času od parametra α , ktorého hodnoty uvažujeme z intervalu 0.01 – 0.2. Na každom z grafov na Obr. 4.18 a 4.19 sú zvýraznené často používané hodnoty α , t.j. 0.01, 0.05 a 0.1. Počet spustení pre tento numerický experiment bol 20. Z Obr. 4.18 a 4.19 vidieť, že pre jednorezové aj viaczrezové verzie dostávame veľmi podobné výsledky ako pri testovaní z podkapitoly 4.1, t.j. pre jednorezové verzie dostávame v oboch prípadoch klasickú závislosť a v prípade viaczrezových verzií je opäť náročné určiť nejaký druh závislosti.





Obr. 4.18: Porovnanie výpočtového času pre α , $sh = \text{áno}$, $K = 5$, $M = 5$, $H = 4$, $\delta = 2$, $\varepsilon = 0.000001$



Obr. 4.19: Porovnanie výpočtového času pre α , $sh = \text{nie}$, $K = 5$, $M = 5$, $H = 4$, $\delta = 2$, $\varepsilon = 0.000001$

4.2.4 Zhrnutie

Na záver tohto testovania zhrnieme jednotlivé pozorovania a vyhodnotíme efektívnosť a vhodnosť použitia jednotlivých metód. Najprv vyhodnotíme zefektívnenie jednorezovej a viaczovej vnorenej L-shaped metódy modifikáciou uvedenou v podkapitole 3.10. Vo všetkých numerických experimentoch dosiahla miera zefektívnenia aspoň 48% pri viaczovej verzii a aspoň 45% pri jednorezovej verzii, čo považujeme za veľmi dobrý výsledok našej modifikácie. Viaczové verzie, podobne ako v predchádzajúcom testovaní, neprejavili žiadnu výraznú citlivosť na hodnoty parametrov M , α a sh a v priebehu testovania neskonvergovalo len 0.7% testovaných úloh. Tieto skutočnosti môžeme považovať za postačujúce na prehlásenie viaczových metód za výrazne univerzálnejšie

v porovnaní s jednorezovými verziami. Pri jednorezových verziách sa potvrdila, aj v predchádzajúcom testovaní odpozorovaná, vysoká citlivosť na všetky testované parametre. Miera neskonvergovania v prípade jednorezových verzií dosiahla až 4%. Tieto skutočnosti robia z jednorezových verzií (modifikovaná a pôvodná) nevhodnú a neefektívnu variantu pre riešenie úloh v tvare (2.18). Pre úlohy v tvare (2.18) odporúčame použiť modifikovanú viacrezovú vnorenú L-shaped metódu, pretože vo všetkých numerických experimentoch jasne prevýšila ostatné metódy.

V prílohe C sa nachádzajú dodatočné tabuľky obsahujúce ďalšie údaje z numerických experimentov, v ktorých sme testovali závislosť výpočtového času od parametrov M a K . Štruktúra tabuliek je nasledovná. V prvých troch stĺpcoch tabuľky je uvedený priemerný, minimálny a maximálny výpočtový čas príslušnej modifikovanej metódy. Ďalej nasleduje priemerný, minimálny a maximálny výpočtový čas príslušnej pôvodnej metódy a napokon je uvedený priemerný, minimálny a maximálny počet makroiterácií potrebných na skonvergovanie. Informácia o makroiteráciách je spoločná pre obe metódy, pretože modifikovaná verzia je len zjednodušením pôvodnej verzia a tak počet makroiterácií a riešenie sú vždy úplne rovnaké.

Záver

Cieľom tejto práce bolo vyhodnotiť efektívnosť použitia rôznych algoritmov pre riešenie jedno-periódovej a viac-periódovej úlohy optimalizácie portfólia s rizikovou mierou Conditional Value-at-Risk. Jednotlivé algoritmy sme implementovali v Matlabe a následne sme ich efektívnosť vyhodnotili na základe rozsiahleho numerického experimentu.

Na riešenie jedno-periódovej úlohy optimalizácie portfólia sme použili jednorezovú, viacrezovú a regularizovanú verziu L-shaped metódy. Ďalej sme použili CVaRmin metódu a nami odvodenú viacrezovú verziu CVaRmin metódy. Do testovania sme zahrnuli aj metódu vnútorného bodu a simplexovú metódu, ktoré sú priamo implementované v Matlabe. Testovanie potvrdilo naše očakávania o vyššej efektívnosti CVaRmin metód, kde nami navrhnutá viacrezová CVaRmin metóda bola jasne najefektívnejšou pri úlohách s väčším počtom aktív a uvažovaním shortsellingu. V ostatných prípadoch bola výrazne najefektívnejšou jednorezová CVaRmin metóda. Taktiež sa potvrdili naše očakávania ohľadom výhodnosti použitia viacrezových metód, ktoré konvergovali vždy v dvoch makroiteráciách a jasne prevýšili jednorezové verzie uvažovaných metód vo všetkých testovaných aspektoch. Na základe uvedeného považujeme CVaRmin metódy za najvhodnejšiu voľbu pre riešenie jedno-periódovej úlohy optimalizácie portfólia. Naopak, jednorezovú a regularizovanú verziu L-shaped metódy považujeme za najnevhodnejšie varianty a z tohto dôvodu ich použitie neodporúčame. Dobré výsledky dosiahla metóda vnútorného bodu, ktorá bola častokrát druhou najrýchlejšou metódou a zároveň bola najspoľahlivejšou, pričom neskonvergovala len v 0.03% testovaných úloh.

V prípade riešenia viac-periódovej úlohy optimalizácie portfólia sme uvažovali jednorezovú a viacrezovú vnorenú L-shaped metódu. V obidvoch verziách sme navrhli modifikáciu umožňujúcu zefektívnenie procesu optimalizácie pomocou týchto metód. V testovaní sa potvrdila efektívnosť našej modifikácie, kde v prípade viacrezovej verzie dosiahla miera zefektívnenia 48% až 55% a v prípade jednorezovej verzie to bolo 45% až 50%. Aj v tomto prípade sa potvrdilo naše očakávanie a viacrezové verzie výrazným spôsobom prevýšili jednorezové. Z uvedených skutočností odporúčame použiť modifikovanú viacrezovú vnorenú L-shaped metódu.

Literatúra

- [1] ACERBI, C., TASCHE, D.: *Expected Shortfall: a natural coherent alternative to Value at Risk*, Economic notes, Vol. 31, Issue 2, 379-388, 2002.
- [2] BAY, A.K., MAYER, J.: *Computational Aspects of Minimizing Conditional Value at Risk*, Computational Management Science, Vol. 3, Issue 1, 3-27, 2006.
- [3] BELKELAAR, A., GROMICHO, J.A.S., KOUWENBERG, R., ZHANG, S.: *A Primal-Dual Decomposition Algorithm for Multistage Stochastic Convex Programming*, Mathematical programming, Vol. 104, Issue 1, 153-177, 2005.
- [4] BENDERS, J.F.: *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik, Vol. 4, 238-252, 1962.
- [5] BIRGE, J.R., LOUVEAUX, F.V.: *A multicut algorithm for two-stage stochastic linear programs*, European Journal of Operations Research, Vol. 34, 384-392, 1988.
- [6] BIRGE, J.R., LOUVEAUX, F.V.: *Introduction to Stochastic Programming*, Second Edition, Springer, 2011.
- [7] CAMPANA, A.: *On Tail Value-at-Risk for sums of non-independent random variables with a generalized Pareto distribution*, The Geneva Risk and Insurance Review, Vol. 32, Issue 2, 169-180, 2007.
- [8] COLOMBO, M.: *Advances in Interior Point Methods for Large-Scale Linear Programming*, Dissertation thesis, University of Edinburg, 2007.
- [9] COLOMBO, M., GROTHEY, A.: *A decomposition-based warm-start method for stochastic programming*, Technical Report ERGO 09-008, 2009.
- [10] DANTZING, G.B.: *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963.
- [11] DOBIAŠ, M.: *Vnorená L-shaped metóda s adaptívnou agregáciou rezov optimality*, Diplomová práca, FMFI UK, Bratislava, 2011.
- [12] FABIÁN, C.I.: *Handling CVaR Objectives and Constraints in Two-stage Stochastic Models*, European Journal of Operational Research, Vol. 191, Issue 3, 888-911, 2008.

- [13] HOLTON, A.G.: *History of Value-at-Risk: 1908-1998*, Contingency Analysis, 2002.
- [14] KALL, P.,MAYER, J.: *Stochastic Linear Programming: Models, Theory and Computation*, Springer, 2005.
- [15] KILIANOVÁ, S.: *Stochastic Dynamic Optimization Models for Pension Planning*, Dizertačná práca, FMFI UK, 2008.
- [16] LEAVENS, D.H.: *Diversification of Investments, Trusts and Estates*, Vol. 80, Issue 5, 469-473, 1945.
- [17] LINDEROTH, J., WRIGHT, S.: *Decomposition Algorithms for Stochastic Programming on a Computational Grid*, Optimization Technical Report 02-07, Computer Sciences Department, University of Wisconsin-Madison, 2002.
- [18] LIU, X., SUN, J.: *A new decomposition technique in solving multistage stochastic linear programs by infeasible interior point methods*, Journal of Global Optimization, Vol. 28, Issue 2, 2004.
- [19] LOUVEAUX, F.V.: *A solution method for multistage stochastic programs with recourse with application to an energy investment problem*, Operations Research, Vol. 28, 889–902, 1980.
- [20] PFLUG, G.CH., RÖMISCH, W.: *Modeling, Measuring and Managing Risk*, World Scientific, 2007.
- [21] PLESNÍK, J., DUPAČOVÁ, J., VLACH, M.: *Lineárne Programovanie*, Alfa, Bratislava, 1990.
- [22] ROCKEFELLAR, R.T., URYASEV, S.: *Optimalization of Conditional Value at Risk*, The Journal of Risk, Vol. 2, No. 3, 21-41, 2000.
- [23] ROCKEFELLAR, R.T., URYASEV, S.: *Conditional Value at Risk for General Loss Distributions*, Research Report 2001-5, ISE Dept., University of Florida, 2001.
- [24] ROOS, C., TERLAKY, T., VIAL, J.P.: *Interior Point Methods for Linear Optimization*, Springer, 2005.
- [25] RUSZCZYŃSKI, A.: *A regularized decomposition for minimizing a sum of polyhedral functions*, Mathematical Programming, Vol. 35, 309–333, 1986.
- [26] RUSZCZYŃSKI, A.: *Regularized decomposition of stochastic programs: algorithmic techniques and numerical results*, Working Paper, WP-93-21, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1993.
- [27] SARYKALIN, S., SERRAINO, G., URYASEV, S.: *Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization*, Tutorial in operations research, INFORMS 2008.

-
- [28] SELEČENIOVÁ, R.: *Podmienená Hodnota Rizika (CVaR)*, Bakalárska práca, FMFI UK, Bratislava, 2010.
- [29] SHAPIRO, A., DENTCHEVA, D., RUSZCZYŃSKI, A.: *Stochastic Programming: Modeling and Theory*, SIAM, 2009.
- [30] VAN-SLYKE, R., WETS, R.J-B.: *L-shaped linear programs with application to optimal control and stochastic programming*, SIAM Journal on Applied Mathematics, Vol. 17, 638–663, 1969.
- [31] ZHAO, G.: *A Log-Barrier method with Benders decomposition for solving two-stage stochastic linear programs*, Mathematical programming, Vol. 90, Issue 3, 507-536, 2001.

Priloha A

AAPL	Apple Inc.	HRS	Harris Corp.
ACA.PA	Credit Agricole S.A	HSP	Hospira Inc.
AGN.AS	Aegon	INTU	Intuit Inc.
ALPA	L'Air Liquide SA	KLAC	KLA-Tencor Corporation
AIZ	Assurant In.	MAS	Masco Corporation
ALTR	Altera Corp.	MC.PA	LVMH Moet Hennessy Louis Vuitton
ALU	Alcatel-Lucent S.A	MEO.DE	METRO AG
ARG	Airgas Inc.	MUV2.DE	M
BAYN	Bayer Inc.	NBR	Nabors Industries, Ltd.
BEAM	Beam Inc.	NEM	Newmont Mining Corp.
BN.PA	Danone	NOA3.DE	NOKIA
BNP.PA	BNP Paribas SA	NVDA	NVIDIA Corporation
BRCM	Broadcom Corp.	OR.PA	L'Oreal SA
C	Citigroup Inc.	PCL	Plum Creek Timber Co. Inc.
CNP	Center Point Energy Inc.	PHIA.AS	ROY.PHILIPS ELECTR
CNX	Consol Energy, Inc.	PWR	Quanta Services Inc.
COG	Cabot Oil & Gas Corporation	PX	Praxair Inc.
CS.PA	Axa	REP.MC	Repsol YPF SA
CTL	CenturyLink Inc.	RNO.PA	Reanult SA
DBK.DE	Deutsche Bank AG	RWE.DE	RWE AG
DRI	Darden Restaurant Inc.	SAN.PA	Sanofi
DTE.DE	Deutsche Telecom AG	SAP.DE	SAP AG
DV	DeVry Inc.	SGO.PA	Compagnie de Saint-Gobain
EA	Electronic Arts Inc.	SIE.DE	Siemens AG
F	Ford Motor Co.	SLB	Schlumberger Limited
FII	Federated Investors Inc.	SLM	SLM Corporation
FISV	Fiserv Inc.	TXN	Texas Instruments Inc.
GE	General Electric Company	USB	U.S. Bancorp.
GILD	Gilead Sciences Inc.	WY	Weyerhaeuser Co.
GOOG	Google Inc.	ZION	Zions Bancorp.

Príloha B

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$								
M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
2	47.39	38.46	60.11	16	13	20	$3.92E - 04$	$2.01E - 03$
3	87.32	75.12	106.37	30	25	35	$5.54E - 04$	$2.45E - 03$
4	121.67	95.02	148.58	41	32	48	$3.32E - 04$	$3.04E - 03$
5	168.88	147.25	200.24	58	51	67	$6.30E - 04$	$2.51E - 03$
6	234.04	209.06	267.08	82	75	91	$3.91E - 04$	$1.04E - 03$
7	277.61	254.27	295.21	95	88	98	$6.31E - 04$	$2.09E - 03$
8	346.54	297.20	384.78	119	106	127	$3.90E - 04$	$1.36E - 03$
9	431.05	353.46	541.72	149	124	180	$4.28E - 04$	$5.75E - 04$
10	489.11	439.08	558.04	167	154	185	$3.72E - 04$	$6.85E - 04$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$								
M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
2	34.44	24.36	43.52	12	8	14	$8.52E - 05$	$7.52E - 04$
3	37.90	23.84	44.64	13	7	15	$1.69E - 04$	$1.61E - 03$
4	58.48	27.19	88.93	21	9	31	$2.37E - 04$	$8.59E - 04$
5	66.46	27.76	115.83	24	10	41	$7.00E - 05$	$4.12E - 04$
6	72.90	24.37	112.88	23	8	40	$8.24E - 05$	$3.73E - 04$
7	79.92	28.93	137.66	28	9	48	$1.64E - 04$	$7.79E - 04$
8	81.43	38.81	129.11	31	14	46	$4.61E - 05$	$4.98E - 04$
9	93.67	51.75	155.91	32	15	54	$1.39E - 04$	$5.90E - 04$
10	101.82	45.42	149.41	36	16	52	$1.31E - 04$	$3.21E - 04$

Tabuľka 4.19: Jednerezová L-shaped metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$								
M	Full_IP			Full_sim			Odchýlka	
	Čas (sec.)			Čas (sec.)			priem.	max
	priem.	min	max	priem.	min	max		
10	0.314	0.231	0.459	1.453	1.225	1.827	$4.82E - 06$	$1.57E - 04$
20	0.388	0.273	0.606	2.089	1.119	2.499	$5.66E - 06$	$6.27E - 05$
30	0.494	0.407	0.619	2.920	2.431	3.609	$3.99E - 06$	$3.70E - 05$
40	0.660	0.505	0.761	3.673	2.754	4.247	$2.83E - 06$	$4.49E - 05$
50	0.792	0.638	1.037	4.584	3.526	5.864	$4.52E - 06$	$2.68E - 05$
60	0.982	0.807	1.278	5.370	4.574	7.010	$6.43E - 06$	$3.30E - 05$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$								
M	Full_IP			Full_sim				
	Čas (sec.)			Čas (sec.)			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	0.447	0.310	0.668	1.993	1.855	2.420	$2.84E - 05$	$9.20E - 04$
20	0.514	0.337	0.786	2.254	1.932	3.430	$2.01E - 06$	$2.01E - 04$
30	0.717	0.455	1.400	2.407	2.025	3.271	$5.28E - 05$	$6.12E - 04$
40	0.871	0.533	1.744	2.614	2.230	3.702	$4.12E - 07$	$8.06E - 05$
50	0.946	0.628	1.466	2.709	2.195	3.595	$3.90E - 06$	$7.27E - 05$
60	1.219	0.817	2.030	2.832	2.315	3.672	$1.92E - 06$	$8.97E - 05$

Tabuľka 4.20: Full_IP a Full_sim pre $sh = \text{áno}$ a $sh = \text{nie}$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$								
M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	13.02	10.75	17.90	187	168	201	$8.38E - 06$	$2.71E - 04$
20	384.77	358.01	436.92	552	530	601	$1.05E - 05$	$8.34E - 03$
30	573.93	539.46	620.21	1032	985	1083	$3.03E - 05$	$2.19E - 04$
40	1205.27	1067.52	1283.30	1628	1526	1718	$3.70E - 05$	$6.66E - 04$
50	2540.31	2489.76	2576.06	2198	2158	2314	$2.49E - 05$	$4.98E - 04$
60	5759.48	5646.33	5839.02	2988	2945	3018	$9.46E - 05$	$5.88E - 04$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$								
M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	0.893	0.215	2.183	36	15	68	$3.82E - 04$	$4.27E - 04$
20	1.542	0.563	3.956	51	27	89	$1.15E - 04$	$9.83E - 04$
30	2.186	0.714	5.231	64	27	101	$2.18E - 04$	$5.67E - 04$
40	3.447	1.196	7.681	79	44	124	$1.93E - 04$	$7.11E - 04$
50	4.692	1.271	9.434	82	39	127	$3.45E - 04$	$8.29E - 04$
60	6.524	1.886	12.753	95	42	136	$2.84E - 04$	$5.88E - 04$

Tabuľka 4.21: Jednerezová CVaRmin metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$								
M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	0.276	0.169	0.540	2	2	2	$1.50E - 05$	$4.27E - 04$
20	0.329	0.204	0.838	2	2	2	$8.90E - 05$	$2.08E - 03$
30	0.415	0.256	0.777	2	2	2	$1.86E - 05$	$1.87E - 04$
40	0.528	0.352	1.207	2	2	2	$6.62E - 05$	$1.04E - 03$
50	0.574	0.381	1.140	2	2	2	$9.60E - 05$	$7.63E - 04$
60	0.691	0.436	1.484	2	2	2	$1.36E - 04$	$1.26E - 03$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$

M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	0.355	0.199	0.640	2	2	2	$1.22E - 05$	$3.72E - 04$
20	0.396	0.236	0.733	2	2	2	$1.89E - 05$	$1.87E - 04$
30	0.560	0.282	1.307	2	2	2	$7.85E - 05$	$2.07E - 03$
40	0.646	0.350	1.199	2	2	2	$5.33E - 05$	$1.25E - 03$
50	0.767	0.408	1.671	2	2	2	$8.88E - 05$	$3.06E - 03$
60	0.896	0.436	1.741	2	2	2	$6.51E - 05$	$1.31E - 03$

Tabuľka 4.22: Viacrezová CVaRmin metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$

M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	6.414	6.091	6.950	2	2	2	$1.50E - 05$	$4.27E - 04$
20	6.669	6.322	7.419	2	2	2	$8.90E - 05$	$2.08E - 03$
30	7.165	6.762	7.812	2	2	2	$1.86E - 05$	$1.87E - 04$
40	7.177	6.764	8.092	2	2	2	$6.62E - 05$	$1.04E - 03$
50	7.736	7.161	8.876	2	2	2	$9.60E - 05$	$7.63E - 04$
60	7.861	7.272	10.348	2	2	2	$1.36E - 04$	$1.26E - 03$

$K = 1000, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$

M	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
10	6.800	6.372	7.494	2	2	2	$1.22E - 05$	$3.72E - 04$
20	7.202	6.776	7.777	2	2	2	$1.89E - 05$	$1.87E - 04$
30	7.549	6.936	8.406	2	2	2	$7.85E - 05$	$2.07E - 03$
40	7.831	7.284	8.677	2	2	2	$5.33E - 05$	$1.25E - 03$
50	8.165	7.557	9.810	2	2	2	$8.88E - 05$	$3.06E - 03$
60	8.334	7.421	9.328	2	2	2	$6.51E - 05$	$1.31E - 03$

Tabuľka 4.23: Viacrezová L-shaped metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$

K	Full_IP			Full_sim				
	Čas (sec.)			Čas (sec.)			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
100	0.044	0.037	0.079	0.062	0.049	0.079	$6.12E - 06$	$8.45E - 04$
500	0.131	0.088	0.248	0.398	0.375	0.512	$7.61E - 07$	$1.22E - 05$
1000	0.348	0.253	0.509	1.232	1.117	1.665	$2.63E - 07$	$4.81E - 06$
2000	1.127	0.784	1.772	3.870	3.413	4.542	$1.45E - 05$	$8.37E - 05$
5000	6.014	3.738	10.36	23.47	20.64	28.97	$6.44E - 04$	$2.49E - 02$
10000	18.52	13.03	21.86	90.17	81.28	98.62	$4.81E - 04$	$9.59E - 03$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = nie$									
K	Full_IP			Full_sim					
	Čas (sec.)			Čas (sec.)			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
100	0.039	0.032	0.066	0.081	0.061	0.104	$7.15E - 09$	$1.26E - 08$	
500	0.158	0.111	0.243	0.618	0.552	0.749	$2.26E - 09$	$7.26E - 06$	
1000	0.455	0.301	0.694	2.087	1.886	2.238	$9.64E - 05$	$7.17E - 03$	
2000	1.471	0.904	2.865	7.315	7.008	8.711	$2.91E - 07$	$1.80E - 05$	
5000	8.266	5.381	13.75	55.48	53.07	59.20	$4.62E - 05$	$6.94E - 04$	
10000	26.31	22.64	38.12	275.07	271.15	279.82	$2.18E - 05$	$9.88E - 05$	

Tabuľka 4.24: Full_IP a Full_sim pre $sh = \text{áno}$ a $sh = nie$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$									
K	Čas (sec.)			Iterácie			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
	100	0.977	0.644	1.296	42	32	55	$3.58E - 06$	$5.64E - 04$
500	1.506	0.817	2.219	60	39	76	$4.54E - 04$	$7.21E - 04$	
1000	1.598	0.707	1.982	61	41	69	$2.62E - 04$	$6.84E - 03$	
2000	1.810	0.963	2.486	66	45	77	$5.48E - 04$	$3.95E - 03$	
5000	1.621	0.881	2.378	69	35	75	$2.64E - 04$	$1.79E - 03$	
10000	1.755	1.605	2.249	69	57	76	$8.14E - 04$	$2.58E - 03$	

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = nie$									
K	Čas (sec.)			Iterácie			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
	100	0.275	0.108	0.661	16	8	30	$4.58E - 06$	$2.47E - 05$
500	0.304	0.125	0.695	16	9	35	$4.98E - 05$	$1.85E - 04$	
1000	0.308	0.164	0.569	18	9	31	$2.75E - 04$	$4.99E - 04$	
2000	0.386	0.154	0.994	21	8	42	$8.65E - 04$	$1.25E - 03$	
5000	0.363	0.124	0.765	19	8	37	$3.38E - 04$	$1.34E - 03$	
10000	0.288	0.162	0.837	19	9	35	$4.45E - 04$	$6.71E - 04$	

Tabuľka 4.25: Jednerezová CVaRmin metóda pre $sh = \text{áno}$ a $sh = nie$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$									
K	Čas (sec.)			Iterácie			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
	100	13.64	11.27	16.77	42	34	52	$4.28E - 06$	$2.48E - 04$
500	84.18	40.29	98.14	55	27	63	$3.29E - 04$	$9.27E - 04$	
1000	198.61	155.36	261.48	62	49	74	$2.49E - 04$	$1.83E - 03$	
2000	354.67	260.17	475.34	60	51	77	$3.44E - 04$	$2.47E - 03$	
5000	1040.37	838.28	1147.41	64	51	71	$8.24E - 04$	$2.37E - 03$	
10000	2241.24	2159.34	2305.14	69	67	71	$6.21E - 04$	$1.94E - 03$	

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$									
K	Čas (sec.)			Iterácie			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
100	4.964	2.761	7.576	16	9	27	$8.61E - 06$	$7.86E - 05$	
500	31.17	19.42	49.43	19	11	32	$5.25E - 05$	$3.74E - 04$	
1000	51.43	39.61	61.83	18	13	23	$3.72E - 04$	$8.08E - 04$	
2000	89.91	58.74	136.51	18	11	30	$3.27E - 04$	$1.53E - 03$	
5000	169.83	124.34	237.29	17	12	27	$4.94E - 05$	$4.64E - 04$	
10000	370.59	262.18	482.42	20	13	26	$4.29E - 04$	$2.93E - 04$	

Tabuľka 4.26: Jednerezová L-shaped metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$									
K	Čas (sec.)			Iterácie			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
100	0.064	0.049	0.108	2	2	2	$3.45E - 05$	$4.95E - 04$	
500	0.124	0.087	0.181	2	2	2	$2.88E - 06$	$8.64E - 05$	
1000	0.322	0.204	0.495	2	2	2	$4.15E - 05$	$7.64E - 04$	
2000	1.034	0.684	1.674	2	2	2	$4.68E - 05$	$2.59E - 03$	
5000	4.616	1.574	6.784	2	2	2	$2.35E - 03$	$5.28E - 02$	
10000	17.94	9.24	38.76	2	2	2	$6.17E - 03$	$3.48E - 02$	

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$									
K	Čas (sec.)			Iterácie			Odchýlka		
	priem.	min	max	priem.	min	max	priem.	max	
100	0.051	0.044	0.119	2	2	2	$5.14E - 07$	$3.65E - 06$	
500	0.127	0.072	0.205	2	2	2	$4.96E - 05$	$2.69E - 04$	
1000	0.317	0.192	0.527	2	2	2	$1.62E - 06$	$2.24E - 05$	
2000	1.036	0.532	1.708	2	2	2	$3.67E - 04$	$4.36E - 03$	
5000	5.584	2.854	8.236	2	2	2	$4.26E - 05$	$7.28E - 04$	
10000	21.18	11.71	31.02	2	2	2	$5.25E - 06$	$3.12E - 05$	

Tabuľka 4.27: Viacrezová CVaRmin metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{áno}$								
K	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
100	0.675	0.644	0.795	2	2	2	$3.45E - 05$	$4.95E - 04$
500	3.227	3.018	3.557	2	2	2	$2.88E - 06$	$8.64E - 05$
1000	6.726	6.449	7.674	2	2	2	$4.15E - 05$	$7.64E - 04$
2000	13.75	13.46	15.59	2	2	2	$4.68E - 05$	$2.59E - 03$
5000	38.24	35.59	41.48	2	2	2	$2.35E - 03$	$5.28E - 02$
10000	91.84	81.43	118.17	2	2	2	$6.17E - 03$	$3.48E - 02$

$M = 5, \alpha = 0.01, \mu = 0.0015, \varepsilon = 0.000001, sh = \text{nie}$								
K	Čas (sec.)			Iterácie			Odchýlka	
	priem.	min	max	priem.	min	max	priem.	max
100	0.674	0.641	0.748	2	2	2	$5.14E - 07$	$3.65E - 06$
500	3.219	3.112	3.755	2	2	2	$4.96E - 05$	$2.69E - 04$
1000	6.595	6.451	7.049	2	2	2	$1.62E - 06$	$2.24E - 05$
2000	13.91	13.23	14.84	2	2	2	$3.67E - 04$	$4.36E - 03$
5000	40.27	37.39	42.71	2	2	2	$4.26E - 05$	$7.28E - 04$
10000	98.23	89.98	109.45	2	2	2	$5.25E - 06$	$3.12E - 05$

Tabuľka 4.28: Viacrezová L-shaped metóda pre $sh = \text{áno}$ a $sh = \text{nie}$

Príloha C

$H = 5, M = 5, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{áno}$									
K	MLs_m			VLs_m			Iterácie		
	Čas (sec.)			Čas (sec.)					
	priem.	min	max	priem.	min	max	priem.	min	max
2	1.36	1.23	1.46	2.87	2.52	3.12	2	2	2
3	5.78	5.15	6.71	11.84	10.71	13.70	2	2	2
4	16.91	16.21	18.63	36.53	34.70	38.94	2	2	2
5	37.89	36.27	40.31	78.82	74.35	83.45	2	2	2
6	79.54	72.79	84.04	157.50	144.86	169.76	2	2	2
7	142.75	131.28	153.48	279.79	256.00	303.89	2	2	2
8	236.77	227.47	243.93	480.64	457.21	497.61	2	2	2
9	422.25	416.01	432.24	823.39	807.05	856.43	2	2	2
10	629.13	612.42	660.49	1207.94	1175.85	1281.35	2	2	2

$H = 5, M = 5, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{áno}$									
K	MLs_s			VLs_s			Iterácie		
	Čas (sec.)			Čas (sec.)					
	priem.	min	max	priem.	min	max	priem.	min	max
2	2.56	1.95	2.80	4.71	3.65	5.27	4	3	4
3	14.47	13.18	16.32	26.47	23.60	30.20	5	5	5
4	50.95	32.84	58.56	99.35	63.71	115.37	6	4	7
5	154.30	134.41	188.39	302.43	262.10	365.48	8	7	10
6	309.46	255.76	363.15	578.69	475.72	689.99	9	7	10
7	559.44	445.11	677.87	1124.48	903.57	1342.18	9	7	10
8	1123.30	777.19	1461.33	2089.33	1453.35	2659.62	10	7	13
9	2240.84	1954.23	2515.94	4145.56	3595.78	4641.91	13	11	14
10	3386.18	3024.65	3958.25	6366.02	5625.85	7520.68	12	11	14

$H = 4, K = 10, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{áno}$									
M	MLs_m			VLs_m			Iterácie		
	Čas (sec.)			Čas (sec.)					
	priem.	min	max	priem.	min	max	priem.	min	max
2	53.96	53.10	54.86	117.63	114.17	122.34	2	2	2
5	61.94	59.61	66.52	127.60	116.83	135.00	2	2	2
8	64.98	61.69	66.94	143.87	137.85	147.52	2	2	2
11	67.13	66.24	68.17	139.67	137.21	142.47	2	2	2
14	65.69	62.12	68.08	130.07	121.14	133.43	2	2	2
17	68.61	68.20	68.91	140.65	138.44	141.96	2	2	2
20	71.23	68.79	76.06	139.60	134.84	149.08	2	2	2

$H = 4, K = 10, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{áno}$

M	MLs_s			VLs_s			Iterácie		
	Čas (sec.)			Čas (sec.)					
	priem.	min	max	priem.	min	max	priem.	min	max
2	200.81	191.27	202.28	373.50	357.67	383.53	8	7	8
5	310.82	280.22	339.95	563.20	518.97	634.00	11	10	12
8	352.68	308.62	390.05	667.62	570.94	764.49	12	11	14
11	348.92	312.27	385.56	666.43	592.07	729.48	12	11	13
14	434.99	362.86	489.07	823.01	676.74	925.79	12	12	14
17	443.20	420.13	452.72	842.68	806.72	868.96	12	10	13
20	480.88	471.01	490.75	922.88	908.71	956.90	12	12	13

$H = 5, M = 5, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{nie}$

K	MLs_m			VLs_m			Iterácie		
	Čas (sec.)			Čas (sec.)					
	priem.	min	max	priem.	min	max	priem.	min	max
2	1.30	1.25	1.35	2.58	2.46	2.72	2	2	2
3	5.63	5.17	6.13	12.15	11.32	13.13	2	2	2
4	16.11	15.27	16.64	35.12	33.30	36.45	2	2	2
5	38.65	34.85	47.43	83.10	74.22	101.50	2	2	2
6	75.68	70.17	79.42	153.63	141.03	163.60	2	2	2
7	142.65	135.93	145.51	293.86	277.29	304.12	2	2	2
8	239.91	230.86	246.17	470.22	454.79	482.49	2	2	2
9	431.76	415.36	443.57	867.85	834.87	896.01	2	2	2
10	606.51	598.31	630.74	1206.95	1187.56	1263.59	2	2	2

$H = 5, M = 5, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{nie}$

K	MLs_s			VLs_s			Iterácie		
	Čas (sec.)			Čas (sec.)					
	priem.	min	max	priem.	min	max	priem.	min	max
2	1.58	1.18	1.89	2.93	2.20	3.49	3	2	3
3	9.33	7.56	10.66	18.28	15.05	20.80	4	3	4
4	35.89	28.36	39.20	67.12	53.31	71.74	5	4	5
5	96.15	84.55	109.48	178.46	158.96	201.44	5	5	6
6	183.18	168.91	204.86	331.93	309.11	368.67	5	5	6
7	381.18	329.72	434.79	728.82	623.17	834.80	6	5	7
8	728.88	632.25	850.85	1359.35	1175.98	1574.06	7	6	8
9	1069.05	966.52	1153.42	1999.12	1797.74	2156.89	6	5	7
10	2383.34	2156.66	2567.36	4528.45	4076.09	4890.82	9	8	9

$H = 4, K = 10, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{nie}$

M	MLs_m			VLs_m			Iterácie		
	Čas (sec.)			Čas (sec.)			priem.	min	max
	priem.	min	max	priem.	min	max			
2	53.85	52.17	55.08	108.25	105.91	111.25	2	2	2
5	62.56	61.06	64.07	134.51	130.67	138.38	2	2	2
8	66.57	65.67	67.46	136.47	133.32	140.32	2	2	2
11	71.31	65.63	81.01	148.32	133.89	169.31	2	2	2
14	71.98	67.29	76.62	141.79	131.22	154.01	2	2	2
17	71.77	68.94	76.55	142.10	137.19	150.80	2	2	2
20	75.94	73.05	80.19	156.44	149.03	166.80	2	2	2

$H = 4, K = 10, \alpha = 0.01, \delta = 2, \varepsilon = 0.000001, sh = \text{nie}$

M	MLs_s			VLs_s			Iterácie		
	Čas (sec.)			Čas (sec.)			priem.	min	max
	priem.	min	max	priem.	min	max			
2	127.77	100.72	153.28	236.38	185.32	285.10	5	4	6
5	168.57	143.88	175.20	323.65	279.12	334.62	5	5	6
8	219.57	203.35	232.81	410.60	370.10	437.68	8	7	8
11	231.67	193.33	253.53	442.49	369.26	486.77	8	6	8
14	285.52	205.18	365.86	553.91	391.89	711.90	10	7	12
17	296.37	234.98	357.76	555.69	439.64	664.01	10	8	12
20	308.55	295.34	322.11	569.27	546.96	599.77	9	8	11