



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Tomáš Trégner

**Redukce scénářů v Monte Carlo
metodách v optimalizaci**

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: doc. RNDr. Ing. Miloš Kopa, Ph.D.

Studijní program: Matematika

Studijní obor: Obecná matematika

Praha 2017

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Redukce scénářů v Monte Carlo metodách v optimalizaci

Autor: Tomáš Trégner

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: doc. RNDr. Ing. Miloš Kopa, Ph.D., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Tato práce se zabývá redukcí scénářů při použití Monte Carlo metod. Hlavním cílem je posoudit, jaké výhody, či zlepšení nám může redukce scénářů poskytnout a zda nám může být v praxi užitečná. V práci budeme prezentovat výsledky získané pomocí vlastní implementace redukčního algoritmu v jazyku Python. Pro účely posouzení efektivity redukce scénářů byly vybrány dva konkrétní problémy. Prvním z nich je odhad konstanty π , který je pro tento účel vhodný zejména proto, že je znám přesný výsledek. Druhým problémem, na který se soustředíme, je pak výběr optimálního portfolia z daných akcií, který jsme vybrali proto, že se jedná o poměrně náročný a zajímavý problém umožňující posoudit časovou efektivitu metody redukce scénářů. Na základě našich výpočtů docházíme k závěru, že redukce scénářů může být užitečným nástrojem pro složité úlohy, je však třeba si dávat pozor na vhodnou volbu použité metriky.

Klíčová slova: redukce scénářů, Monte Carlo, optimalizace, výběr portfolia

Title: Scenario reduction in Monte Carlo methods in optimization

Author: Tomáš Trégner

Department: Department of Probability and Mathematical Statistics

Supervisor: doc. RNDr. Ing. Miloš Kopa, Ph.D., Department of Probability and Mathematical Statistics

Abstract: This work focuses on scenario reduction used in the Monte Carlo methods. Our main objective is to evaluate advantages and improvements that scenario reduction can provide us and whether it could be useful in practical applications. The results presented in this work were obtained using our own implementation of the reduction algorithm written in the Python programming language. Two specific problems were chosen to determine the efficiency of scenario reduction. The first problem is the estimation of π , which is especially suitable for our purposes because the exact result is known. The second problem we focus on is the optimal portfolio selection from given shares, which was chosen because it is quite demanding and interesting problem which allows us to assess the time efficiency of scenario reduction. Based on our results we conclude that scenario reduction could be useful for solving complicated problems, but we need to be careful about the choice of metrics.

Keywords: scenario reduction, Monte Carlo, optimization, portfolio selection

Rád bych poděkoval panu doktorovi Václavu Kozmíkovi za jeho trpělivost a cenné rady při psaní této práce. A také svým rodičům, za jejich neustávající podporu.

Obsah

Úvod	2
1 Monte Carlo metody a scénáře	3
1.1 Monte Carlo metody	3
1.2 Scénáře	4
1.3 Implementace redukčního algoritmu	6
2 Odhad čísla π	8
3 Výběr portfolia	10
3.1 Formulace problému	10
3.2 Model	11
3.3 Řešení metodou Monte Carlo	12
3.4 Odhady parametrů a generování scénářů	13
3.5 Výsledky	15
Závěr	19
Seznam použité literatury	20
Seznam tabulek	21
Seznam použitých zkratek	22

Úvod

V četných stochastických i deterministických optimalizačních úlohách, ale též v řadě výpočetních úloh, lze k řešení využít tzv. Monte Carlo metod. Těchto metod využíváme zejména v situacích, kdy nelze využít žádné jiné optimalizační metody, tedy pokud je úloha příliš složitá nebo například není přesně znám analytický předpis účelové funkce.

Standardní postup při použití metod Monte Carlo je založen na vygenerování velkého množství scénářů, které získáváme opakováním nějakého náhodného pokusu. Získané výsledky se následně vhodně interpretují či využijí k dalším výpočtům.

Přesnost výsledků do velké míry závisí na počtu opakování náhodného pokusu, tedy na počtu vygenerovaných scénářů. S množstvím scénářů však také roste výpočetní a paměťová složitost problému. Někdy se proto využívá metody redukce scénářů. Její princip je velice jednoduchý. Z velkého množství daných či vygenerovaných scénářů se vybere menší množství těch, které lze v nějakém smyslu považovat za reprezentativní. Těmto vybraným scénářům mohou být přiděleny váhy, například v podobě pravděpodobností. Vyšší pravděpodobnost pak znamená, že v nejbližším okolí bylo zredukováno více scénářů.

Právě metodou redukce scénářů se budeme zabývat v této bakalářské práci. V první kapitole bude zaveden termín scénář a popsána metoda redukce. Ve druhé kapitole bude ukázána aplikace této metody na nejznámější problém řešený metodou Monte Carlo - odhad hodnoty čísla π . V poslední kapitole pak bude ukázána aplikace metody na konkrétní optimalizační problém. Tímto problémem bude výběr portfolia z akcií sedmi společností, se kterými se obchoduje na pražské Burze cenných papírů.

U obou výše uvedených aplikací metody redukce scénářů mě budou zajímat zejména dva faktory:

- zda řešení získaná z n scénářů zredukováných z většího množství budou přesnější, než řešení získaná z n scénářů bez redukce
- a jaká bude časová náročnost výpočtu.

Skutečné řešení je přitom u první úlohy známo. U druhé úlohy není v našich silách skutečné řešení zjistit a proto budeme porovnávat směrodatnou odchylku řešení vzniklých bez redukce a s redukcí.

1. Monte Carlo metody a scénáře

1.1 Monte Carlo metody

Vznik metody Monte Carlo je datován do období druhé světové války, kdy John von Neumann a Stanislav Ulam využili ruletu k simulaci chování neutronů při průchodu různými látkami, jak je uvedeno v Fabian a Klumber (1998). Ruleta dala též podnět k pojmenování metody po hlavním městě Monackého knížectví, jež je známé pro hazardní hry a zejména ruletu. Fabian a Klumber (1998) dále uvádí, že metoda našla záhy uplatnění při řešení problémů ekonomických, technických, problémů z oblasti činnosti telefonních centrál, řízení dopravy, hromadné obsluhy, kontroly stavu zásob a dalších.

Postupně se objevily aplikace i v matematice samotné. Patrně nejznámější, nikoliv však nejdůležitější, aplikací metody Monte Carlo je odhad konstanty π , kterému se budeme blíže věnovat ve druhé kapitole a na kterém prověříme efektivitu metody redukce scénářů. Z dalších aplikací lze uvést výpočet určitých integrálů, a to včetně vícerozměrných, řešení systémů lineárních rovnic, hledání kořenů rovnic a výpočet hodnot funkcí. Využití našla metoda Monte Carlo i v optimalizaci, například v podobě metody stochastické aproximace či různých evolučních algoritmů.

Princip metody Monte Carlo je v Dupač (1962) popsán tak, že určitý problém se neřeší a často ani neformuluje matematicky, nýbrž řeší se vytvářením umělých náhodných výběrů a početními operacemi, většinou velmi jednoduchými, s těmito výběrovými hodnotami. Častým modelem užití, který budeme dále využívat i v této práci, je ten, který je popsán právě v Dupač (1962) a který nyní uvedeme jen v modernějším jazyce a značení:

„Řešením daného problému je střední hodnota veličiny \mathbf{X} , která může být i vícerozměrná a která je známou funkcí nenáhodných veličin (parametrů) i náhodných veličin: $\mathbf{X} = f(a, b, c, \dots; X_1, X_2, X_3, \dots)$. Hodnoty parametrů a, b, c, \dots jsou známy, rovněž je známo sdružené rozdělení náhodných veličin X_1, X_2, X_3, \dots . Pro libovolné výběrové hodnoty s_1, s_2, s_3, \dots náhodných veličin X_1, X_2, X_3, \dots lze vypočítat příslušnou hodnotu \mathbf{s} náhodné veličiny \mathbf{X} ; teoretický výpočet střední hodnoty $\mathbf{E}(\mathbf{X})$ je však prakticky neproveditelný. Přibližné určení $\mathbf{E}(\mathbf{X})$ se pak provádí tak, že se vytvoří výběrové hodnoty $s_{i,1}, s_{i,2}, s_{i,3}, \dots$ náhodných veličin $X_1, X_2, X_3, \dots, (i = 1, \dots, n)$, k nim se následně vypočítají výběrové hodnoty $s_i = f(a, b, c, \dots; s_{i,1}, s_{i,2}, s_{i,3}, \dots)$ a střední hodnota $\mathbf{E}(\mathbf{X})$ se odhaduje aritmetickým průměrem $\sum_{i=1}^n \frac{s_i}{n}$.“

V tomtéž článku dále získáme i orientační představu o přesnosti metody Monte Carlo:

„Řešení nalezená metodou Monte Carlo mají tedy povahu statistických odhadů. Jejich přesnost je zpravidla řádu $n^{-\frac{1}{2}}$, tedy poměrně malá. Náznorně to znamená, že musíme stonásobně zvýšit rozsahy výběrů, chceme-li výsledky zpřesnit o jedno desetinné místo; pro výpočty s velkými nároky na přesnost se tedy metody Monte Carlo nehodí.“

Z posledního uvedeného vyplývá, že například pro odhad konstanty π metoda Monte Carlo není vhodným nástrojem, protože dosáhneme jen značně omezené přesnosti. Cílem této bakalářské práce však není řešit jakýkoliv problém s vysokou

přesností, avšak demonstrovat, jaký efekt na přesnost má užití metody redukce scénářů.

Hlavní předností metody Monte Carlo je, že její přesnost nebývá ovlivněna počtem dimenzí, na rozdíl od jiných metod, které často podléhají tzv. prokletí dimenzionality. Například při numerickém řešení integrálů přesnost výpočtu vůbec nezávisí na dimenzi, jak je psáno ve Fabian a Kluiber (1998), narozdíl od kvadratických formulí, jež jsou nejznámějším způsobem, kterým se integrály numericky řeší.

V dnešní době je užití metody Monte Carlo úzce spjato s využitím počítačů. Podle Fabian a Kluiber (1998) závisí úspěch použití počítačů zejména na třech faktorech:

1. kvalitě generátoru náhodných, respektive pseudonáhodných čísel
2. výběru racionálního algoritmu výpočtu
3. kontrole přesnosti získaného výsledku

V práci se proto budeme snažit vždy uvádět, jak jsme postupovali, abychom zajistili dostatečnou úroveň výše zmíněných faktorů, zejména budeme podrobně popisovat způsob, jakým byla generována pseudonáhodná čísla.

1.2 Scénáře

Nyní se budeme zabývat ději, jejichž vývoj lze zcela popsat pomocí nějakého reálného náhodného vektoru $\mathbf{X} : (\Omega, \mathcal{A}, \mathbf{P}) \rightarrow (\mathbb{R}^k, \mathcal{B}^k)$. Například v úloze o výběru portfolia se jedná o vektor změny cen všech akcií.

Uvedeme několik důležitých definic. Tyto definice jsou volně inspirovány článkem Dupačová a kol. (2003), avšak jsou zjednodušeny na konečné problémy, aby nemusely být zaváděny zbytečně složité termíny, jako je Kantorovičův funkcionál. Nejprve zadefinujeme klíčový pojem této bakalářské práce, a sice pojem scénář.

Definice 1 (Scénář). *Nechť \mathbf{X} je náhodný vektor. Pak scénářem rozumíme libovolnou realizaci $\mathbf{s} = (s_1, \dots, s_k)^\top$ tohoto náhodného vektoru.*

Scénáře můžeme získávat různými způsoby, z nichž mezi nejběžnější patří reálná pozorování daného děje (například sledování skutečné změny cen akcií), vygenerování pomocí počítače (pokud známe nebo umíme odhadnout rozdělení vektoru \mathbf{X}) nebo expertní odhad (například scénáře inflace pro následující rok).

Definice 2 (Úplný systém scénářů). *Řekneme, že scénáře $\mathbf{s}_1, \dots, \mathbf{s}_n$ tvoří úplný systém scénářů, jestliže je každému scénáři \mathbf{s}_i přiřazena váha $p_i > 0$ tak, že $\sum_{i=1}^n p_i = 1$.*

Definice 3 (Zredukovaný podsystém scénářů). *Nechť $\mathcal{S} = \mathbf{s}_1, \dots, \mathbf{s}_n$ je úplný systém scénářů. Nechť $J \subset \{1, \dots, n\}$. Systém scénářů $\{\mathbf{s}_j : j \in \{1, \dots, n\} \setminus J\}$ nazveme zredukovaným podsystémem systému \mathcal{S} , jestliže každému scénáři $\mathbf{s}_j, j \in \{1, \dots, n\} \setminus J$ je přiřazena váha $q_j > 0$ tak, že $\sum_{j \in \{1, \dots, n\} \setminus J} q_j = 1$.*

Redukujeme-li tedy nějaký systém scénářů, vybereme některé scénáře, ty odebereme a změníme váhy tak, aby váhy vzniklého systému dávaly opět dohromady 1, tedy aby zredukovaný systém byl opět úplný. Váhy lze ovšem v takovém případě volit mnoha různými způsoby. Naším dalším cílem bude zjistit, který způsob zvolit, aby se zredukovaný podsystém pokud možno co nejméně lišil od původního systému. K tomu potřebujeme nějaký způsob, jak měřit jejich vzdálenost.

Definice 4 (Vzdálenost zredukovaného systému). *Nechť $\mathcal{S} = \mathbf{s}_1, \dots, \mathbf{s}_n$ je úplný systém scénářů. Nechť $J \subset \{1, \dots, n\}$ a $\mathcal{R} = \{\mathbf{s}_j : j \in \{1, \dots, n\} \setminus J\}$ je zredukovaný podsystém systému \mathcal{S} . Nechť $c : \mathbb{R}^k \times \mathbb{R}^k \rightarrow [0; \infty)$ je metrika. Pak funkcionál D definovaný předpisem*

$$D(J, \mathbf{q}) := \min \left\{ \sum_{i=1}^n \sum_{j \in \{1, \dots, n\} \setminus J} c(\mathbf{s}_i, \mathbf{s}_j) \eta_{i,j} : \eta_{i,j} \geq 0, \sum_{i=1}^n \eta_{i,j} = q_j, \right. \\ \left. \sum_{j \in \{1, \dots, n\} \setminus J} \eta_{i,j} = p_i, \forall i, j \right\}$$

nazýváme vzdáleností zredukovaného podsystému.

Uvedený vzorec pro vzdálenost vypadá dosti komplikovaně, ale ve skutečnosti tomu tak není. Veličina $c(\mathbf{s}_i, \mathbf{s}_j)$ udává „vzdálenost“ i -tého a j -tého scénáře a faktor $\eta_{i,j}$ udává „množství hmoty“ přesunuté z i -tého do j -tého scénáře, přičemž máme dáno, kolik „hmoty“ je ve kterém scénáři na počátku a kolik jí má být na konci. Dané minimum ve funkci D je tedy vlastně řešením dopravního problému.

Nyní nás zajímá, jakým způsobem pro danou množinu J zvolit váhy \mathbf{q} tak, aby vzdálenost $D(J, \mathbf{q})$ byla minimální. Na to dává odpověď Theorem 2 z Dupačová a kol. (2003):

Věta 1 (optimální váhy). *Je-li J dána, pak*

$$D_J = \min \{ D(J, \mathbf{q}) : q_j \geq 0, \sum_{j \notin J} q_j = 1 \} = \sum_{i \in J} p_i \min_{j \notin J} c(\mathbf{s}_i, \mathbf{s}_j)$$

Minima se nabývá v bodě $\bar{\mathbf{q}}$ splňujícím: $\bar{q}_j = p_j + \sum_{i \in J_j} p_i$, pro každé $j \notin J$, kde $J_j := \{i \in J : j = j(i)\}$ a $j(i) \in \arg \min_{j \notin J} c(\mathbf{s}_i, \mathbf{s}_j)$, pro každé $i \in J$.

Důkaz. Je uveden v Dupačová a kol. (2003) ve větší obecnosti pro $D(J, \mathbf{q})$ definovanou pomocí Kantorovičova funkcionálu $\hat{\mu}_c$. Použijeme-li však vyjádření $\hat{\mu}_c$ pro konečně mnoho scénářů uvedené na straně 495, dostaneme přesně funkci $D(J, \mathbf{q})$ z definice 4. □

Větu 1 lze interpretovat tak, že nejmenší vzdálenosti dosáhneme tehdy, pokud váhy scénářů, které zůstávají, nebudeme nikam přesouvat, a pravděpodobnosti scénářů, které zanikají, přesuneme vždy do nejbližšího zůstávajícího scénáře. Což není nic překvapivého.

Nyní máme návod, jak přerozdělit váhy, pokud je zadaná množina J . Zbývá tedy určit, jak volit množinu J . Pro účely této bakalářské práce se omezíme na případ, kdy vždy redukuje pouze jeden scénář naráz, tedy $|J| = 1$. V takovém případě:

$$D_J = D_{\{i\}} = \sum_{i \in J} p_i \min_{j \notin J} c(\mathbf{s}_i, \mathbf{s}_j) = p_i \min_{j \neq i} c(\mathbf{s}_i, \mathbf{s}_j)$$

Mezi systémy zredukovanými právě o jeden scénář nyní hledáme ten, který má nejmenší vzdálenost od původního systému ve smyslu definice 4. Řešíme tedy úlohu:

$$\min_{i \in \{1, \dots, N\}} D_{\{i\}} = \min_{i \in \{1, \dots, N\}} p_i \min_{j \neq i} c(\mathbf{s}_i, \mathbf{s}_j)$$

Najdeme-li scénář \mathbf{s}_i řešící tuto úlohu, přerozdělíme váhy dle následujícího pravidla:

$$q_j = p_j + p_i I_{\{j=l\}}, \forall j \neq i,$$

kde $l \in \arg \min_{j \neq i} c(\mathbf{s}_i, \mathbf{s}_j)$ je index vybraného scénáře, do kterého „přesuneme“ zredukovaný scénář.

1.3 Implementace redukčního algoritmu

Nyní, když už víme, jak volit scénáře k redukci a měnit váhy, můžeme zformulovat algoritmus pro redukci:

1. Najdi i minimalizující výraz $p_i \min_{j \neq i} c(\mathbf{s}_i, \mathbf{s}_j)$.
2. Odeber scénář \mathbf{s}_i .
3. Najdi nejmenší l takové, že $l \in \arg \min_{j \neq i} c(\mathbf{s}_i, \mathbf{s}_j)$. (K danému scénáři \mathbf{s}_i z předchozího kroku může existovat více scénářů, které jsou od něj stejně vzdálené ve smyslu metriky c . Aby byl algoritmus deterministický, vybereme tedy ten z nich, který má nejnižší index, a tento index označíme l .)
4. $p_l := p_l + p_i$
5. Pokud je počet zbývajících scénářů vyšší, než cílový počet scénářů, vrať se k 1.

Jako metriku c jsme v obou případech zvolili euklidovskou vzdálenost. Důvody této volby popíšeme u jednotlivých problémů.

Při praktické realizaci jsme zavedli pole, ve kterém si pamatujeme ke každému scénáři jeho nejbližší scénář, vzdálenost k němu a též hodnotu výrazu $p_i \min_{j \neq i} c(\mathbf{s}_i, \mathbf{s}_j)$, kterou jsme nazvali důležitostí. Díky tomu při hledání scénáře, který budeme odebírat, stačí projít jedno pole délky n a není již třeba nic počítat. Po odebrání scénáře je sice vždy potřeba příslušná pole aktualizovat, přitom však stačí přepočítat pouze nejbližší scénáře ke scénářům, jejichž dosavadním nejbližším scénářem byl ten odebraný. Přemýšleli jsme i o udržování matice vzdáleností mezi všemi scénáři, avšak taková matice by pro větší (a tedy zajímavé) množství scénářů byla příliš velká a nevešla by se do operační paměti používaného počítače. Vlastní redukce pak je rozdělena do dvou procedur - jedna vytváří výše uvedená pomocná pole a druhá provádí vlastní redukci. Zdrojové kódy obou procedur i dalších pomocných procedur lze nalézt v příloze.

Musíme podotknout, že takto implementovaná redukce v jazyce Python je bohužel pomalá. Je to proto, že obsahuje výrazné množství cyklů. Pro srovnání uvedeme, že v Mathematice byly výpočty ještě zhruba pětkrát delší, avšak při

použití jazyku Pascal jsme dosáhli výsledků výrazně lepších. Stejně množství scénářů, konkrétně 4000, prošlo na tomtéž počítači procedurou „InicializujScenare“ v jazyce Python za 84 s a v jazyku Pascal za necelé 2 s. Jazyk Python byl zvolen z toho důvodu, že se v něm lépe pracuje s velkými poli a že balíček NumPy poskytuje velké množství předprogramovaných funkcí zejména pro optimalizaci. Lze ovšem předpokládat, že pokud by byl zvolen kompilovaný jazyk, rychlost programu by mohla být výrazně vyšší.

2. Odhad čísla π

V této kapitole uvedeme, jak dopadl náš pokus použít metodu redukce scénářů při řešení klasického Monte Carlo problému, tedy odhadu čísla π . Jak jsme již konstatovali v sekci 1.1, metoda Monte Carlo není pro tento účel příliš vhodná, jelikož ke zvýšení přesnosti o jedno desetinné místo je třeba stokrát zvýšit rozsah výběru. Naším cílem je však prokázat, zda metoda redukce scénářů může zvýšit přesnost výpočtu, nikoliv odhadnout π naprosto přesně.

V této kapitole abstrahujeme i od časové náročnosti výpočtu. Zpracování dat při odhadu čísla π je velice jednoduché, mnohem jednodušší, než proces redukce scénářů. Je tedy výrazně efektivnější využít větší rozsah výběru přímo k výpočtu, než aplikovat metodu redukce scénářů.

Proč se tímto problémem tedy vůbec zabýváme, když nezískáme jeho přesné řešení ani ho nedokážeme spočítat rychle? Nabízí nám totiž pro demonstraci funkčnosti redukce scénářů dvě výhody. Zaprvé je známé jeho řešení, a tedy je snadné porovnat, který z výsledků je přesnější. Zadruhé vygenerované scénáře se využívají velice jednoduchým a čitelným způsobem a případné rozdíly by se proto mohly dobře projevit.

Odhad konstanty π bez redukce scénářů provádíme tak, že generujeme dvojice nezávislých náhodných veličin X_i, Y_i s rovnoměrným rozdělením na intervalu $[0; 1]$, tedy body ve čtverci $[0; 1]^2$. Dále zavedeme náhodnou veličinu $Z_i = \mathbb{I}_{[X_i^2 + Y_i^2 \leq 1]}$, tedy indikátor, zda vygenerovaný bod leží ve čtvrtkruhu se středem v bodě $[0; 0]$ a poloměrem 1. Pak $Z_i \sim \text{Alt}(p)$, kde $p = \frac{\pi}{4}$ je poměr obsahu čtvrtkruhu a obsahu celého čtverce. Silný zákon velkých čísel (SZVČ) nám pak říká, že $\bar{Z}_n = \frac{1}{n} \sum_{i=1}^n Z_i \xrightarrow{P} \mathbb{E} Z_1 = p = \frac{\pi}{4}$. Za odhad čísla π použijeme tedy hodnotu $\hat{\pi}_n = 4\bar{Z}_n$.

Použijeme-li redukci scénářů, je situace o něco složitější. Dané náhodné veličiny Z_i mají k sobě totiž navíc váhy $p_i = \frac{n_i}{n}$, kde n_i je počet scénářů, které jsou po redukci reprezentovány i -tým scénářem. Počty n_i , a tedy i váhy p_i , jsou však závislé na hodnotách všech náhodných veličin $X_j, Y_j, j \in \{1, \dots, n\}$, a tedy jsou samy náhodnými veličinami. Za odhad potom používáme $\hat{\pi}_n = 4 \sum_{i=1}^n p_i Z_i$. Náhodné veličiny p_i nejsou nezávislé (jejich součet je roven jedné), takže na ně nemůžeme použít SZVČ a není jasné, jestli $\hat{\pi}_n \xrightarrow{P} \pi$. Pokud však scénáře po redukci dobře reprezentují původní scénáře (což bychom v této práci pomocí experimentů rádi ukázali), pak: $\hat{\pi}_n = \frac{4}{n} \sum_{i=1}^n n_i Z_i \sim \frac{4}{n} \sum_{i=1}^n Z_i \xrightarrow{P} 4 \mathbb{E} Z_1 = 4p = \pi$. Praktickou implementaci lze najít v příloze.

Pro účely této bakalářské práce byly provedeny dvě sady po deseti experimentech. V prvních deseti byl odhad π vypočten na základě 2500 vygenerovaných scénářů. Ve druhých deseti pokusech bylo vždy vygenerováno 25000 scénářů, které byly následně zredukovány na 2500, jež se použily pro odhad.

Velichiny X_i, Y_i byly generovány pomocí procedury „random.rand“, která dle dokumentace ke generování používá Mersenne Twister. Ten je v současnosti jedním z nejpoužívanějších generátorů náhodných čísel. Autoři o něm v článku Matsumoto a Nishimura (1998) uvádí, že jeho perioda je $2^{19937} - 1$ a splňuje několik statistických testů náhodnosti včetně diehard testu. Považujeme ho tudíž za vhodný i pro naše účely.

Semínko pro generování bylo vždy generováno náhodně pomocí internetového

generátoru náhodných čísel dostupného na <http://www.itnetwork.cz/javascript-online-generator-nahodnych-random-cisel-se-zvolitelnym-rozsahem>.

Pro redukci byla zvolena jako metrika c euklidovská vzdálenost v \mathbb{R}^2 . Tuto volbu považujeme za nejpřirozenější, jelikož v tomto případě vygenerované scénáře skutečně mají geometrický význam a euklidovská vzdálenost od bodu 0 se navíc objevuje i v samotné metodě pro odhad konstanty π .

Výsledky pokusů shrnuje tabulka 2.1. Jelikož Python pracuje ve dvojkové soustavě a s omezenou přesností, výsledky, které zobrazuje, jsou v podobě hodnoty datového typu float, která je nejbližší k číslu, které je uvedené v tabulce. Číslo z tabulky je přitom správnou hodnotou, jelikož počáteční váhy scénářů jsou 0,0004, při 2500 počátečních scénářích, respektive 0,00004, při 25000 počátečních scénářích, a získaný odhad π musí být násobkem těchto vah.

Tabulka 2.1: Odhady konstanty π

Pokus	1	2	3	4	5
Bez redukce	3,152	3,1504	3,152	3,0944	3,1392
S redukcí	3,14368	3,13168	3,14944	3,14704	3,1536
Pokus	6	7	8	9	10
Bez redukce	3,1456	3,16	3,1488	3,1536	3,1664
S redukcí	3,14464	3,14112	3,13328	3,12608	3,15008

Pro lepší představu uvádíme několik základních statistik, které nám pomohou porovnat kvalitu obou odhadů. Pro výpočet absolutní odchylky jsme použili hodnotu čísla π zaokrouhlenou na pět desetinných míst, tedy $\pi \doteq 3,14159$. Pro výběrový průměr je použit vzorec $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$, pro směrodatnou odchylku je použit vzorec $S_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2}$, který považujeme za nejtradičnější a můžeme jej použít následně i u výběru portfolia, a pro průměrnou absolutní odchylku od hodnoty π vzorec $MAD = \frac{1}{n} \sum_{i=1}^n |X_i - \pi|$, který považujeme v této situaci za nejnázornější způsob, jak vyjádřit chybu odhadu.

Tabulka 2.2: Vybrané statistiky odhadů π

	Bez redukce	S redukcí
Výběrový průměr	3,14624	3,142064
Výběrová směrodatná odchylka	0,019657	0,008988
Průměrná absolutní odchylka	0,014566	0,007314

Tabulka 2.2 ukazuje, že s metodou redukce scénářů skutečně dosahujeme lepších výsledků. Výběrová směrodatná odchylka i průměrná absolutní odchylka od skutečné hodnoty π klesly zhruba na polovinu. Jak jsme konstatovali v sekci 1.1, přesnost metody Monte Carlo je zpravidla řádu $n^{-\frac{1}{2}}$, při desetinásobném zvětšení rozsahu výběru bez redukce tedy lze očekávat zhruba třikrát přesnější výsledek. Vidíme tedy, že redukcí dojde k nějaké ztrátě informace, avšak významná část se jí zachová.

3. Výběr portfolia

V poslední kapitole ilustruji užití metody redukce scénářů na problém výběru portfolia z akcií sedmi významných společností, které jsou obchodovány na pražské Burze cenných papírů. Konkrétně se bude jednat o akcie společností ČEZ, Erste group bank (EGB), Fortuna (FT), Komerční banka (KB), Phillip Morris (PM), Pegas Nonwovens (PN) a Unipetrol (UP). Tyto společnosti byly vybrány z důvodu, že jejich akcie jsou na trhu již delší dobu (všechny alespoň od roku 2012) a jsou obchodovány v objemu, který lze v ČR považovat za významný. Po úvaze nakonec nebyly použity akcie společnosti O2, jelikož tato společnost v průběhu období, z něhož bylo odhadováno rozdělení růstu akcií, prošla rozpadem na dvě společnosti, což mělo na růst jejich akcií zásadní vliv.

3.1 Formulace problému

Investor chce vybrat vhodné portfolio z akcií výše uvedených společností. Má daný požadavek, aby střední hodnota výnosu dosahovala alespoň hodnoty m , a chce vybrat takové portfolio, které mu přináší nejmenší možné riziko, které zde budeme reprezentovat mírou CVaR. Definici této rizikové míry přebíráme z prezentace k přednášce analýza investic, dostupné na stránkách paní profesorky Dupacové (<http://www.karlin.mff.cuni.cz/~dupacova/downloads/rizika.pdf>):

Definice 5 (CVaR). *Nechť T je daný budoucí časový horizont, $\alpha \in (0,1]$ je zvolená hladina a X reprezentuje náhodnou ztrátu s distribuční funkcí $G(x)$ uvažovaného portfolia. Pak $CVaR_\alpha(X)$ je průměrná ztráta ze $100(1-\alpha)\%$ nejhorších případů ztrát portfolia.*

Označme $\mathbf{z} = (z_1, \dots, z_7)^\top$ množství finančních prostředků investovaných do akcií jednotlivých společností. Celkové množství investovaných finančních prostředků označme W_0 . Dále označme $\mathbf{R} = (R_1, \dots, R_7)^\top$ náhodný vektor změny cen jednotlivých akcií. Veličiny R_i udávají podíl cen akcií po uplynutí investičního období a na jeho počátku. Tedy například hodnota 1,05 by znamenala, že cena dané akcie za investiční období vzrostla na svůj 1,05-násobek, neboli vzrostla o 5%. Pak řešíme následující optimalizační úlohu:

$$\begin{aligned} \min_{\mathbf{z}} \quad & CVaR_\alpha(\mathbf{z}^\top(\mathbf{1}_7 - \mathbf{R})) \\ \text{s.t.} \quad & \mathbf{z} \geq \mathbf{0}_7 \\ & \mathbf{z}^\top \mathbf{1}_7 = W_0 \\ & \mathbf{E}[\mathbf{z}^\top \mathbf{R}] \geq m. \end{aligned}$$

Zde (i v dalším textu) $\mathbf{1}_n = (1, \dots, 1)^\top$ značí sloupcový vektor jedniček délky n , $\mathbf{0}_n = (0, \dots, 0)^\top$ je nulový vektor délky n a značení $\mathbf{x} \geq \mathbf{y}$ znamená, že $x_i \geq y_i, \forall i$.

Rizikovou míru CVaR lze, jak je dokázáno v Rockafellar a Uryasev (2000), spočítat dle následujícího vzorce:

$$CVaR_\alpha(X) = \min_{a \in \mathbb{R}} \left\{ a + \frac{1}{1-\alpha} \mathbf{E}[X - a]^+ \right\}$$

Danou optimalizační úlohu můžeme tedy přepsat následujícím způsobem:

$$\begin{aligned} \min_{z,a} \quad & \left\{ a + \frac{1}{1-\alpha} \mathbb{E} [z^\top (\mathbf{1}_7 - \mathbf{R}) - a]^+ \right\} \\ \text{s.t.} \quad & a \in \mathbb{R}, z \geq \mathbf{0}_7 \\ & z^\top \mathbf{1}_7 = W_0 \\ & \mathbb{E} [z^\top \mathbf{R}] \geq m. \end{aligned}$$

3.2 Model

Abychom mohli úlohu řešit, musíme předpokládat něco o rozdělení vektoru \mathbf{R} . Některé z těchto předpokladů nejsou úplně reálné a proto výsledky této práce nelze brát jako investiční doporučení. Cílem těchto předpokladů je vytvořit řešitelnou matematickou úlohu, na které bude možné otestovat metodu redukce scénářů.

Dříve než zformulujeme předpoklady našeho modelu, potřebujeme zavést několik definic důležitých rozdělení. První dvě z nich, definující normální rozdělení, přejímáme z knihy Anděl (2011), třetí, definující mnohorozměrné log-normální rozdělení, je přejata z knihy Mardia a kol. (1979).

Definice 6 (Normální rozdělení). *Náhodná veličina X má normální rozdělení s parametry $\mu \in \mathbb{R}$ a $\sigma^2 > 0$, píšeme $X \sim \mathcal{N}(\mu, \sigma^2)$, je-li její hustota rovna:*

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

Rozdělení $\mathcal{N}(0,1)$ pak nazýváme normovaným normálním rozdělením.

Definice 7 (Mnohorozměrné normální rozdělení). *Mějme náhodný vektor $\mathbf{X} = (X_1, \dots, X_k)^\top$. Nechť $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^\top$ je daný vektor a $\boldsymbol{\Sigma} = (\sigma_{ij})$ symetrická pozitivně semidefinitní matice typu $k \times k$. Řekneme, že \mathbf{X} má k -rozměrné normální rozdělení s parametry $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, jestliže pro libovolný vektor $\mathbf{c} \in \mathbb{R}^k$ platí: $\mathbf{c}^\top \mathbf{X} \sim \mathcal{N}(\mathbf{c}^\top \boldsymbol{\mu}, \mathbf{c}^\top \boldsymbol{\Sigma} \mathbf{c})$.*

Definice 8 (Mnohorozměrné log-normální rozdělení). *Řekneme, že náhodný vektor $\mathbf{X} = (X_1, \dots, X_k)^\top$ má mnohorozměrné log-normální rozdělení s parametry $\boldsymbol{\mu}$ a $\boldsymbol{\Sigma}$, píšeme $\mathbf{X} \sim \mathcal{LN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, jestliže platí $\mathbf{X} = \exp(\mathbf{Y})$, kde $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.*

V případě vektorů přitom exponenciální funkci chápeme po složkách, tedy $\exp((a_1, \dots, a_k)^\top) = (\exp(a_1), \dots, \exp(a_k))^\top$.

Nyní všechny předpoklady zformulujeme:

1. Budoucí výnosy mají stejné rozdělení jako minulé výnosy.
2. Výplata dividendy neovlivňuje výrazným způsobem cenu akcie a dividenda se nezapočítává do výnosu.
3. Vektor \mathbf{R} má log-normální rozdělení a jeho parametry lze odhadnout z historických dat.

4. Do všech akcií můžeme investovat libovolnou částku, nejsme omezeni tím, že bychom museli investovat násobky cen jednotlivých akcií.
5. Investujeme tak malé částky, že nemůžeme ovlivnit cenu akcie.

První a třetí předpoklad jsou důležité proto, abychom vůbec mohli úlohu nějakým způsobem řešit, proto jsou tyto předpoklady ve finančních modelech obecně celkem běžné, ačkoliv je známo, že nejsou zcela reálné. Druhý předpoklad činíme kvůli zjednodušení odhadu parametrů z historických dat. Jsme si vědomi toho, že tento předpoklad má vliv na optimální výběr portfolia, protože například akcie společnosti ČEZ si lidé často kupují zejména kvůli dividendám. Nicméně nově vzniklá úloha je pro demonstraci efektivity metody redukce scénářů stejně dobrá, jako úloha původní. Samozřejmě změny cen akcií v době kolem výplaty dividendy nemají stejné rozdělení jako v ostatní dny, ale zanedbáním tohoto faktoru se již nedopouštíme výrazně větší chyby, než samotným předpokladem shodnosti minulých a budoucích výnosů. Čtvrtý předpoklad nám zaručí, že se nebudeme muset zabývat celočíselnou optimalizací, která by byla výrazně náročnější, než optimalizace neceločíselná.

V rámci modelu si ještě stanovme, že budeme pracovat s investičním obdobím jednoho týdne a hladinou spolehlivosti $\alpha = 0,95$.

3.3 Řešení metodou Monte Carlo

Problém řešíme metodou Monte Carlo. Scénáře jsou v tomto případě realizace náhodné veličiny \mathbf{R} . Způsob jejich generování a odhad parametrů rozdělení vektoru \mathbf{R} je popsán v následujícím oddíle. Scénáře označujeme $\mathbf{s}_1, \dots, \mathbf{s}_n$, přičemž jednotlivé scénáře mají složky $\mathbf{s}_i = (s_{i,1}, \dots, s_{i,7})^\top$, a pravděpodobnosti těchto scénářů označme $\mathbf{p} = (p_1, \dots, p_n)$. Dále označme $d_{i,j} = 1 - s_{i,j}$ ztráty z jednotlivých akcií při jednotlivých scénářích a $\mathbf{d}_i = \mathbf{1}_7 - \mathbf{s}_i$. Pomocí scénářů můžeme nyní z řešené stochastické optimalizační úlohy vytvořit úlohu lineárního programování, která má následující tvar:

$$\begin{aligned} \min_{\mathbf{z}, a, \mathbf{b}} \quad & \left\{ a + \frac{1}{1 - \alpha} \sum_{i=1}^n p_i b_i \right\} \\ \text{s.t.} \quad & a \in \mathbb{R}, \mathbf{z} \geq \mathbf{0}_7, \mathbf{b} \geq \mathbf{0}_n \\ & \mathbf{d}_i^\top \mathbf{z} - a \leq b_i, i = 1, \dots, n \\ & \mathbf{z}^\top \mathbf{1}_7 = W_0 \\ & \sum_{i=1}^n p_i \sum_{j=1}^7 s_{i,j} z_j \geq m. \end{aligned}$$

Tuto úlohu jsme již schopni vyřešit pomocí simplexu, v Pythonu jsme použili předdefinovanou funkci „linprog“. Tato funkce řeší dle dokumentace úlohu lineárního programování tvaru:

$$\begin{aligned}
& \min_{\mathbf{x}} \quad \mathbf{c}^\top \mathbf{x} \\
& \text{s.t.} \quad \mathbf{A}_{ub} \mathbf{x} \leq \mathbf{b}_{ub} \\
& \quad \quad \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq} \\
& \quad \quad x \in \text{bounds}.
\end{aligned}$$

Chceme-li tedy přepsat úlohu do takového tvaru, dostaneme následující vektory a matice:

$$\begin{aligned}
\mathbf{x} &= (z_1, \dots, z_7, a, b_1, \dots, b_n)^\top \\
\mathbf{c} &= (0, \dots, 0, 1, \frac{p_1}{1-\alpha}, \dots, \frac{p_n}{1-\alpha})^\top \\
\mathbf{A}_{ub} &= \left(\begin{array}{c|c|c} \mathbf{D} & -\mathbf{1}_n & -\mathbf{I}_n \\ \hline -\mathbf{p}^\top \mathbf{S} & 0 & \mathbf{0}_n^\top \end{array} \right) \\
\mathbf{b}_{ub} &= (0, \dots, 0, -m)^\top \\
\mathbf{A}_{eq} &= (1, \dots, 1, 0, 0, \dots, 0)^\top \\
\mathbf{b}_{eq} &= (W_0),
\end{aligned}$$

kde \mathbf{D} je matice, jejíž řádky jsou vektory \mathbf{d}_i^\top , \mathbf{S} je matice, jejíž řádky jsou vektory \mathbf{s}_i^\top a \mathbf{I}_n je jednotková matice řádu n . Hranice jsou pro všechny proměnné $[0; \infty)$, s výjimkou proměnné a , která má hranice neomezené. Implementaci v jazyce Python lze najít v přílohách.

Pro redukci scénářů jsme jako metriku c zvolili opět klasickou euklidovskou vzdálenost. To považujeme za nejvhodnější, jelikož vektor \mathbf{R} reprezentuje relativní změnu ceny, a vektor \mathbf{z} reprezentuje přímo částku, kterou do dané akcie investujeme a nikoliv množství akcií, které kupujeme. Nemusíme se tedy zabývat tím, zda jsou některé akcie důležitější než jiné a přidělovat jim dle toho nějaké váhy, jelikož všechny mají stejné měřítko.

3.4 Odhady parametrů a generování scénářů

Nyní popíšeme způsob, jakým byly odhadnuty parametry rozdělení vektoru \mathbf{R} , a jak byly generovány scénáře s tímto rozdělením. Předpokládáme, že R má log-normální rozdělení. Parametry tohoto rozdělení odhadujeme z historických dat z období od 3.12.2012 do 28.11.2016, která jsme získali na stránkách Burzy cenných papírů Praha (<https://www.pse.cz/>). Použité údaje lze najít v přílohách. Jelikož pracujeme s investičním obdobím délky jednoho týdne, používáme vždy poměry dvou po sobě jdoucích pondělních závěrečných kurzů. Pokud na některé pondělí připadl státní svátek a burza byla zavřena, údaj z předchozího týdne na tento týden i z tohoto týdne na týden následující vynecháváme. Celkem tak máme k dispozici 189 údajů, ze kterých odhadujeme hodnoty parametrů $\boldsymbol{\mu}$ a $\boldsymbol{\Sigma}$. Získané údaje nejprve zlogaritmujeme. Tím získáme 189 realizací (scénářů) $\mathbf{q}_1, \dots, \mathbf{q}_{189}$ náhodného vektoru s rozdělením $\mathbf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Z těchto dat pak získáme následující odhady hledaných parametrů: $\hat{\mu}_j = \bar{q}_j = \frac{1}{189} \sum_{i=1}^{189} q_{i,j}$, $j = 1, \dots, 7$ a $\hat{\Sigma}_{j,k} = \frac{1}{189} \sum_{i=1}^{189} (q_{i,j} - \bar{q}_j)(q_{i,k} - \bar{q}_k)$, $j, k = 1, \dots, 7$. Vyšlo nám tedy (zaokrouhleno

na 5 desetinných míst nebo jednu platnou číslici, přesnější údaje se nacházejí v příloze):

$$\hat{\boldsymbol{\mu}} = (-0,00299; -0,00062; 0,00020; 0,00092; 0,00277; 0,00092; -0,00004)^\top$$

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 0,00130 & 0,00059 & 0,00020 & 0,00049 & 0,000140 & 0,00007 & 0,00006 \\ 0,00059 & 0,00245 & 0,00008 & 0,00053 & 0,00020 & 0,00010 & 0,00026 \\ 0,00020 & 0,00008 & 0,00156 & -0,00002 & -0,00003 & 0,00005 & 0,00020 \\ 0,00049 & 0,00053 & -0,00002 & 0,00118 & 0,00018 & 0,00013 & 0,00005 \\ 0,000140 & 0,00020 & -0,00003 & 0,00018 & 0,00050 & -0,000004 & 0,00002 \\ 0,00007 & 0,00010 & 0,00005 & 0,00013 & -0,000004 & 0,00033 & 0,00005 \\ 0,00006 & 0,00026 & 0,00020 & 0,00005 & 0,00002 & 0,00005 & 0,00064 \end{pmatrix}$$

Při generování veličiny s mnohorozměrným log-normálním rozdělením generujeme veličinu s normálním rozdělením s danými parametry a pak ji převedeme dle definice 8. Pro generování mnohorozměrné normální rozdělení uijeme následující větu převzatou z Anděl (2011):

Věta 2. *Nechť náhodné veličiny X_1, \dots, X_r jsou nezávislé a každá z nich má rozdělení $N(0,1)$. Označme $\mathbf{X} = (X_1, \dots, X_r)^\top$. Nechť $\mathbf{Y} = (Y_1, \dots, Y_r)^\top \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $h(\boldsymbol{\Sigma}) = r \geq 1$. Budiž $\mathbf{B}_{k \times r}$ taková matice hodnosti r , že $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top$. Pak $\boldsymbol{\mu} + \mathbf{B}\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.*

Důkaz. Viz Lemma 4.9 na straně 65 knihy Anděl (2011). □

Jelikož matice $\hat{\boldsymbol{\Sigma}}$ je symetrická a pozitivně semidefinitní, matici \mathbf{B} získáme jako Choleského rozklad matice $\hat{\boldsymbol{\Sigma}}$. Její tvar lze najít v přílohách.

Při znalosti matice \mathbf{B} a věty 2 stačí vygenerovat k nezávislých veličin s rozdělením $N(0,1)$. Pomocí procedury „random.rand“, kterou jsem užívali i při odhadu π , vygenerujeme náhodné veličiny s rovnoměrným rozdělením na $[0; 1]$ (náhodné semínko opět generujeme pomocí téhož internetového generátoru náhodných čísel). Tyto pak převedeme na veličiny s normovaným normálním rozdělením pomocí polární metody, popsané v knize Ross (2006). Algoritmus je následující:

1. Vygeneruj náhodná čísla U_1 a U_2 s rovnoměrným rozdělením na $[0; 1]$.
2. Polož $V_1 = 2U_1 - 1$, $V_2 = 2U_2 - 1$, $S = V_1^2 + V_2^2$.
3. Pokud $S > 1$, vrať se ke kroku 1
4. Vrať nezávislé normované normální náhodné veličiny:

$$X = \sqrt{\frac{-2 \log(S)}{S}} V_1, \quad Y = \sqrt{\frac{-2 \log(S)}{S}} V_2$$

Implementaci polární metody v jazyce Python lze najít v příloze.

3.5 Výsledky

Nyní zbývá již jen prezentovat dosažené výsledky. Danou optimalizační úlohu jsme řešili nejprve desetkrát bez redukce užitím 10000 scénářů. Následně jsme desetkrát generovali 30000 scénářů, které jsme zredukovali na 10000, a ty jsme použili k řešení optimalizační úlohy.

Pro řešení úlohy jsme zvolili, že chceme investovat 10000 Kč. Jako rozumný požadavek konzervativního investora jsme vyhodnotili zisk po jednom týdnu 10 Kč, tedy 0,1% týdně, což po roce vychází na zisk přibližně 533 Kč, tedy mírně přes 5%.

Získané výsledky lze najít v tabulkách 3.1 (výsledky bez redukce) a 3.2 (výsledky s redukcí). Hodnoty uvedené v obou tabulkách jsou hodnoty proměnných z_1, \dots, z_7 , tedy množství finančních prostředků v korunách investovaných do jednotlivých akcií. V posledním sloupci pak uvádíme hodnotu účelové funkce, tedy rizikové míry CVaR. Tyto výsledky, včetně náhodných semínek použitých pro získání jednotlivých výsledků, lze najít i v přílohách.

Tabulka 3.1: Výsledky optimalizace bez redukce scénářů

Pokus	ČEZ	EGB	FT	KB	PM	PN	UP	CVaR
1	175	0	662	316	3068	4333	1445	241
2	188	0	643	349	3159	4157	1504	243
3	191	0	764	228	3234	4183	1400	238
4	177	0	661	184	3116	4282	1580	234
5	250	0	739	207	3209	3896	1699	237
6	171	0	765	263	3195	3980	1625	235
7	149	0	779	283	3278	3920	1591	237
8	142	0	666	261	3100	4213	1618	236
9	210	0	670	342	3171	4065	1542	241
10	120	0	706	142	3331	4278	1423	235

Tabulka 3.2: Výsledky optimalizace s redukcí scénářů

Pokus	ČEZ	EGB	FT	KB	PM	PN	UP	CVaR
1	214	0	594	149	3085	4529	1429	235
2	38	0	586	215	3261	4318	1583	227
3	136	0	565	239	3260	4183	1617	227
4	211	0	570	85	3333	4252	1549	234
5	135	0	674	240	3109	4230	1612	231
6	108	0	675	196	3159	4243	1619	230
7	73	0	633	200	3207	4381	1505	229
8	73	0	754	294	3212	4249	1418	228
9	120	0	704	183	3273	4223	1497	229
10	214	0	633	216	3132	4282	1522	227

V tabulce 3.3 pak uvádíme porovnání průměrných množství finančních prostředků investovaných do jednotlivých akcií dle vzorce $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$, výběro-

vých směrodatných odchylek spočtených dle vzorce $S_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2}$.

Tabulka 3.3: Porovnání vybraných statistik výsledků optimalizace

	Red.	ČEZ	EGB	FT	KB	PM	PN	UP	CVaR
\bar{X}_n	Ne	177,3	0	705,5	257,5	3186,1	4130,7	1542,7	237,7
	Ano	132,2	0	638,8	201,7	3203,1	4289	1535,1	229,7
S_n	Ne	36,56	0	51,73	68,14	81,36	157,13	97,94	3,02
	Ano	63,45	0	62,46	56,31	80,45	100,62	74,27	2,87

Narozdíl od úlohy s odhadem π zde bohužel neznáme skutečné řešení této optimalizační úlohy. Jako skutečné řešení nelze ani vzít řešení s velkým množstvím scénářů, jelikož velikost matice \mathbf{A}_{ub} roste kvadraticky s počtem scénářů a byť je z velké části jednotkovou maticí, zabírá v paměti příliš velké místo a proto na dostupných počítačích s použitým programátorským řešením úlohu nelze vyřešit pro více než 20000 scénářů, což nemůžeme považovat za významně přesnější řešení než pokusná řešení. Nicméně stále můžeme porovnat směrodatné odchylky našich odhadů, které o přesnosti výsledků také mnohé vypovídají. V příkladu s odhadem π navíc klesala střední absolutní odchylka od skutečné hodnoty přibližně stejně rychle jako směrodatná odchylka, takže lze předpokládat, že mezi těmito veličinami existuje silná závislost.

Pomineme-li EGB, do níž naše výpočty nikdy nedoporučily investovat, pak u dvou společností vyšla směrodatná odchylka nižší bez použití redukce (ČEZ a FT), u jedné společnosti jsou odchylky takřka totožné (PM) a u zbylých tří společností je směrodatná odchylka nižší v případě, že redukci použijeme. Při řešení optimalizační úlohy je pro nás důležitá zejména stabilita účelové funkce, v tomto případě rizikové míry CVaR, a i ta má směrodatnou odchylku nižší, pokud redukci použijeme. Tedy výsledky s redukcí se zdají být opět lepší, než bez redukce. Rozdíl zde není tak markantní, jako u úlohy s odhadem π , což je pravděpodobně způsobeno dvěma vlivy: jedná se o výrazně složitější problém a rozdíl v počtu scénářů není tak velký.

Bohužel výpočet redukce například z 50000 scénářů na 10000 by byl již velmi časově náročný. Rozhodli jsme se však provést ještě jednu sadu experimentů k ověření, zda výsledky s redukcí jsou opravdu přesnější než bez ní. Tentokrát jsme desetkrát spočetli tutéž úlohu jako výše s 1000 scénáři a následně s 10000 scénáři zredukovanými na 1000. To už je dostatečně velký rozdíl na to, aby se chování redukce projevilo, ačkoliv řešení založená na tak malém množství scénářů jsou samozřejmě velice nepřesná. Neuvádíme zde již kompletní výsledky, ty lze najít v přílohách, uvádíme pouze výběrové průměry a směrodatné odchylky (tabulka 3.4).

Ve druhé sadě experimentů tedy vyšly při výpočtech s redukcí již všechny směrodatné odchylky nižší (mnohdy velmi výrazně) než ve výpočtech bez redukce. Můžeme tedy říci, že redukce scénářů výsledek stabilizovala. Nicméně budeme-li považovat průměrné hodnoty z tabulky 3.3 (tedy ty spočtené z 10000 použitých scénářů) za přibližně správné, vidíme, že hodnoty spočtené přímo z tisíce scénářů bez redukce jsou těmto výsledkům (s výjimkou EGB) blíže než výsledky spočtené pomocí redukce. Řešení s redukcí mají obecně tendenci investovat více peněz do akcií se silným potenciálem k růstu a méně diverzifikovat portfolio (což

Tabulka 3.4: Výsledky optimalizace s 1000 použitými scénáři

Red.	ČEZ	EGB	FT	KB	PM	PN	UP	CVaR	
\bar{X}_n	Ne	252,5	36,8	544,5	199,4	3256	4078,2	1632,2	232,5
	Ano	55,3	0	539,6	7,3	3383,2	4762,8	1251,9	196,2
S_n	Ne	245,4	76,5	220,5	142,2	393,1	448,5	333,8	6,5
	Ano	101,8	0	142,9	15,5	193,8	290,9	331,4	5,4

vidíme v sadě s 1000 i 10000 použitými scénáři). Tyto výsledky jsou tedy zřetelně stabilnější, avšak zdá se, že od optimálního portfolia jsou ve skutečnosti vzdálenější. Rozhodli jsme se to ověřit ještě jedním způsobem. Vzali jsme všechna čtyři průměrná řešení z obou sad experimentů a spočetli jejich průměrný výnos a CVaR na 1000000 scénářů (ve všech případech stejných). Výsledky jsou shrnuty v tabulce 3.5:

Tabulka 3.5: Výnos a CVaR průměrných řešení na 1000000 scénářů

Počet scénářů	Výnos	CVaR
10000	10014,8973	238,7396967
30000 zredukováno na 10000	10015,17696	238,7575317
1000	10014,48208	239,471065
10000 zredukováno na 1000	10016,14485	240,8766223

Vidíme, že řešení s 10000 scénáři a 30000 scénáři zredukovánými na 10000 jsou si velmi blízká, nicméně řešení s redukcí má o trochu vyšší výnos i CVaR. Porovnáme-li řešení s 1000 scénáři a s 10000 scénáři zredukovánými na 1000, vidíme, že opět řešení s redukcí má vyšší výnos i CVaR, avšak rozdíl je tentokrát výraznější. Potvrzuje se nám tedy domněnka, že přestože řešení získaná pomocí redukce scénářů jsou stabilnější, k optimálním řešením mají ve skutečnosti dále. Vyšší hodnotu účelové funkce není v tomto případě možné nijak kompenzovat vyšším výnosem.

Redukce má tedy pravděpodobně tendenci řídit se podle složek s vyššími výnosy. Domníváme se, že je to způsobeno tím, že složky s vyššími hodnotami více ovlivňují euklidovskou vzdálenost a proto se redukce řídí více podle nich. Je tedy možné, že pokud bychom zvolili jinou metriku, která by například pomocí vah jednotlivých složek zohledňovala jejich průměrné hodnoty, výsledky získané pomocí redukce by byly přesnější. Nemáme tedy důvod metodu redukce scénářů zavrhnout (mimo jiné i proto, že při odhadu π fungovala velice dobře), spíše se zamyslet nad její lepší implementací, zejména pak nad výběrem vhodné metriky.

Podívejme se ještě na časové srovnání výsledků s redukcí a bez redukce. Na používaném počítači nám samotná optimalizační úloha s 10000 scénáři zabrala přibližně 25 minut, redukce z 10000 scénářů na 1000 a následná optimalizace kolem 30 minut a redukce z 30000 scénářů na 10000 a následná optimalizace trvala kolem 5 hodin. Stále se tedy více vyplatí použít rovnou 10000 scénářů, než se zabývat jejich redukcí na nižší počet. Nicméně tento rozdíl již není velký a víme, že redukcí by bylo možné naprogramovat efektivněji, zatímco předdefinovanou funkci pro simplex již pravděpodobně příliš zefektivnit nepůjde. Tedy pro tento problém by již skutečně redukce scénářů mohla být vhodným nástrojem a

pro složitější optimalizační úlohy, než je lineární programování, může přinést významnou časovou úsporu. Navíc jsme sami narazili na to, že s naší implementací algoritmů není možné použít více než 20000 scénářů. V takových případech může být redukce scénářů dokonce jedinou možností, jak zužtkovat velké množství scénářů.

Závěr

V této práci jsme nejprve představili metody Monte Carlo a rozebrali jejich silné a slabé stránky. Následně jsme se zabývali scénáři v těchto metodách a s pomocí teoretických výsledků jsme došli až k tomu, jak by měl vypadat redukční algoritmus. Dále jsme popsali naši implementaci tohoto algoritmu. Domníváme se, že by tento algoritmus mohl být implementován i lépe, výrazně by mohlo pomoci použití kompilovaného programovacího jazyka, čímž by se pravděpodobně zrychlil průběh cyklů.

Dále jsme se zabývali již konkrétními problémy, nejprve to byl odhad konstanty π . U tohoto problému nám nešlo o to, zda se redukce vyplatí z časového hlediska, ale pouze o to, zda použitím scénářů vzniklých redukcí získáme přesnější výsledky, než kdybychom použili jen scénáře přímo vygenerované. A porovnáním směrodatných odchylek i absolutních odchylek od skutečné hodnoty π nám opravdu vyšlo, že při redukci z desetinásobného množství scénářů získáme přibližně dvakrát přesnější výsledek než bez redukce. To lze považovat za velmi dobrý výsledek, jelikož přímým použitím desetinásobného množství scénářů bychom získali přibližně třikrát přesnější výsledek.

Nakonec jsme se věnovali výběru optimálního portfolia. Stanovili jsme si přitom minimální požadovaný zisk a hledali jsme, jak tohoto zisku dosáhnout s co nejnižším rizikem. Tento problém je poněkud náročnějšího charakteru a tak jsme se museli dopustit několika zjednodušení, která vedou k tomu, že získané výsledky nelze použít k výběru reálného portfolia na burze, nicméně jsme získali optimalizační úlohu, na které jsme mohli redukci testovat.

Provedli jsme dvě sady po dvaceti pokusech, vždy deset s redukcí a deset bez redukce. Došli jsme k závěru, že použití metody redukce scénářů snižuje směrodatnou odchylku a výsledek je tedy stabilnější. Nicméně se zdá, že výsledky získané metodou redukce scénářů vedou k investování vyšších částek do akcií s vysokým výnosem a nižší diverzifikaci portfolia, než by bylo optimální. Domníváme se však, že se nejedná o chybu metody redukce scénářů jako takové, nýbrž o ne zcela vhodnou volbu metriky použité při redukci. Co se týká časového srovnání, redukce scénářů stále drobně zaostává za přímým výpočtem, avšak tento rozdíl již není velký a při lepší implementaci bychom přímý výpočet pravděpodobně překonali.

Celkově se tedy domníváme, že při vhodné implementaci a s volbou vhodné metriky může být metoda redukce scénářů užitečným nástrojem při řešení náročnějších úloh.

Seznam použité literatury

- ANDĚL, J. (2011). *Základy matematické statistiky*. Třetí vydání. Matfyzpress. ISBN 978-80-7378-162-0.
- DUPAČ, V. (1962). Metody Monte Carlo. *Aplikace matematiky*, **7**.
- DUPAČOVÁ, J., GRÖWE-KUSKA, N. a RÖMISCH, W. (2003). Scenario reduction in stochastic programming, An approach using probability metrics. *Mathematical Programming*, **95**, 493–511.
- FABIAN, F. a KLUIBER, Z. (1998). *Metoda Monte Carlo a možnosti jejího uplatnění*. 1. vydání. Prospektrum, Praha. ISBN 8071750581.
- MARDIA, K., KENT, J. a BIBBY, J. (1979). *Multivariate analysis*. Probability and mathematical statistics. Academic Press. ISBN 9780124712508.
- MATSUMOTO, M. a NISHIMURA, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, **8**, 3–30.
- ROCKAFELLAR, R. T. a URYASEV, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, **2**(3), 21–41.
- ROSS, S. M. (2006). *Simulation*. Elsevier, fourth edition. ISBN 978-0-12-598063-0.

Seznam tabulek

2.1	Odhady konstanty π	9
2.2	Vybrané statistiky odhadů π	9
3.1	Výsledky optimalizace bez redukce scénářů	15
3.2	Výsledky optimalizace s redukcí scénářů	15
3.3	Porovnání vybraných statistik výsledků optimalizace	16
3.4	Výsledky optimalizace s 1000 použitými scénáři	17
3.5	Výnos a CVaR průměrných řešení na 1000000 scénářů	17

Seznam použitých zkratek

SZVČ	Silný zákon velkých čísel
EGB	Erste group bank
FT	Fortuna
KB	Komerční banka
PM	Phillip Morris
PN	Pegas Nonwovens
UP	Unipetrol