Stochastické programovanie a jeho využitie v ekonomike

Stochastic Programming and Its Use in Economics

Richard Martinus[1]

Abstrakt
Tento článok sa zaoberá problematikou stochastického programovania, ktoré je dôležitým nástrojom pre riešenie rozhodovacích problémov v prostrediach s neistotou a náhodnosťou. V práci bude poskytnutý všeobecný prehľad pravdepodobnostných rozdelení a metód používaných v stochastickom programovaní, s dôrazom na techniky ako L-Shaped a Dekompozícia bázy. Konkrétne bude vysvetlený prístup Monte Carlo, ktorý je široko využívaný na simuláciu náhodných procesov a odhadovanie hodnôt v stochastických problémoch. Tento prístup bude potom aplikovaný na riešenie príkladov, kde je demonštrovaná jeho efektívnosť a flexibilita. Okrem toho budú v práci predstavené príklady riešené aj inými metódami než Monte Carlo, pričom je použité normálne pravdepodobnostné rozdelenie na modelovanie neistoty a rizika v týchto príkladoch. Tento komplexný pohľad na problematiku stochastického programovania má za cieľ poskytnúť čitateľovi ucelené porozumenie rôznorodých metód a prístupov k riešeniu stochastických rozhodovacích problémov, pričom hlavným cieľom práce je ukážka aplikácií stochastického programovania v ekonomických príkladoch.

Kľúčové slová
Monte Carlo, Pravdepodobnosť, Python, Neistota

Abstract
This article deals with the issue of stochastic programming, which is an important tool for solving decision problems in environments with uncertainty and randomness. The thesis will provide a general overview of probabilistic distributions and methods used in stochastic programming, with an emphasis on techniques such as L-Shaped and Base Decomposition. Specifically, the Monte Carlo approach, which is widely used to simulate random processes and estimate values in stochastic problems, will be explained. This approach will then be applied to address examples where its effectiveness and flexibility are demonstrated. In addition, examples solved by methods other than Monte Carlo will be presented in the work, using the normal probabilistic distribution to model the uncertainty and risk in these examples. This comprehensive view of the issue of stochastic programming aims to provide the reader with a comprehensive understanding of common methods and approaches to solving stochastic decision-making problems, while the main goal of the work is to demonstrate applications of stochastic programming in economic examples.

Key words
Monte Carlo, Probability, Python, Uncertainty

JEL classification
C73, C8

[1] Ing. Richard Martinus, Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra operačného výskumu, Dolnozemská cesta 1 852 35 Bratislava, richard.misek@euba.sk.

1 Úvod

Stochastické programovanie je prístup na modelovanie optimalizačných problémov, ktoré zahŕňajú neistotu. Patria do skupiny matematického programovania a vieme pomocou nich modelovať úlohy na riešenie rozhodnutí, kde využívame simulácie na ich dosiahnutie. Tieto rozhodnutia sa zväčša týkajú problémov reálneho sveta, kde sa nachádzajú parametre, ktoré v momente rozhodnutia nie sú známe v čase. Príklady využitia v reálnom svete sú napríklad biológia, v ktorej je možné využiť stochastické programovanie na vytvorenie modelov resp. simulácií, ktoré zahŕňajú náhodnosť premenných ako sú genetické informácie alebo evolúcia spojená s dynamickosťou populácie. V informačných technológiách je možné využiť stochastické programovanie v oblasti kryptografie, umelej inteligencie alebo v strojovom učení. Ak by sme sa zamerali na obor financií, tu je možné náhodnosť modelovať pri cenách aktív alebo úrokových sadzbách. Simulácia znamená napodobňovať fungovanie reálneho systému pomocou počítačového modelu, pričom pod pojmom systém rozumieme určitú časť reálneho sveta, ktorú skúmame. Pri definovaní problému rozoznávame dva rôzne problémy (deterministický a stochastický), ktoré sa líšia v tom, ako sa správajú a ako sa s nimi pracuje. Aplikácie týchto problémov sú spísané v diele Comparison of deterministic and stochastic approaches to global optimization (Liberti & Kucherenko, 2005).

Deterministický problém je taký, kde môžeme presne vypočítať budúcu udalosť bez zapojenia náhodnosti. Ak je niečo deterministické, máme všetky potrebné údaje na predpovedanie (určenie) výsledku s istotou. Deterministické modely majú rôzne aplikácie vo vedeckom výskume a pri finančných trhoch. Aplikácie z praxe sú napríklad:

- Predpovedanie životnosti budov a ich komponentov je založené na regresnej analýze, ktorá môže popísať vzťah medzi degradačnými faktormi a stavom fasády (Silva, Brito & Gaspar 2016).
- Štúdie o vývoji klímy spočívajú vo vytváraní scenárov budúceho spoločenského vývoja, odhadovanie emisií skleníkových plynov. Tieto hodnoty sa vkladajú do matematických klimatických modelov, ktoré predpovedajú budúce otepľovanie a škody spôsobené týmto otepľovaním (Kung, Li & Lin, 2018).

Stochastický problém je taký, kde môžeme len odhadnúť pravdepodobnosť každého výsledku a rozhodnúť sa na základe tohto odhadu. Stochastické modely majú istú vrodenú náhodnosť, kde rovnaká sada parametrov a počiatočných podmienok povedie k súboru rôznych výstupov. Zatiaľ čo sa deterministické modely často používajú na predpovedanie budúcich investičných výnosov, stochastické modely sa používajú na riešenie neistôt vstupov. Príklady by mohli byť nasledovné:

- Modelovanie pohybu molekuly plynu - smer a rýchlosť pohybu molekuly plynu sú ovplyvnené náhodnými zrážkami s inými molekulami (Rudyak & Lezhnev, 2020).
- Modelovanie vo financiách - predpovedanie finančných trhov. Známe aplikácie sú simuláciou Monte Carlo, regresné modely a Markovove reťazce (Brooks, 2002).

Pri tvorbe simulačných modelov je často potrebné modelovať procesy ako náhodné premenné. Je to spôsobené tým, že vo väčšine reálnych prípadov (časové intervaly medzi príchodmi dvoch zákazníkov alebo čas obsluhy zákazníkov) procesy nie sú časovo konštantné. Modelovanie takejto udalosti je možné prostredníctvom vhodne zvoleného diskrétneho, alebo spojitého pravdepodobnostného rozdelenia. V modeloch obsluhy sa čas medzi príchodom zákazníkov a čas obsluhy môže modelovať ako konštanta, alebo ako náhodná premenná pomocou pravdepodobnostného rozdelenia. Na modelovanie času dodania sa najčastejšie používa gamma rozdelenie. V modeloch obnovy sa ako náhodná premenná môže modelovať čas zlyhania a najčastejšie sa tu používajú exponenciálne, gamma alebo Weibullovo rozdelenie. Ak sú k dispozícii obmedzené informácie (údaje) o procese, tak sa väčšinou používa rovnomerné rozdelenie, alebo trojuholníkové rozdelenie. Rovnomerné rozdelenie sa používa, ak vieme, že náhodná premenná sa nachádza v určitom intervale, ale nemáme žiadne ďalšie

informácie. Trojuholníkové rozdelenie sa používa v prípade, ak poznáme maximálnu, minimálnu a najpravdepodobnejšiu hodnotu (Prékopa, 1995).

2 Modely a metódy riešenia úloh stochastického programovania

V tejto kapitole sa budeme zaoberať modelmi a metódami riešenia stochastického programovania. Začiatok kapitoly bude sprevádzaný všeobecným popisom modelu lineárneho programovania, nasledovaný náhodnými procesmi a odhadom ich parametrov a základné typy spojitého rozloženia pravdepodobnosti.

2.1 Všeobecný popis modelu lineárneho programovania

Každý optimalizačný model je zložený z účelovej funkcie a obmedzujúcich podmienok, ktoré spolu tvoria dve hlavné časti. Účelová funkcia popisuje cieľ optimalizácie a obmedzujúce podmienky, zodpovedajú za náročnejšie dosiahnutie daných cieľov vďaka vytváraným obmedzeniam. Množina, ktorá spĺňa všetky podmienky sa nazýva množina prípustných riešení úlohy (X^P). Graficky sa dá množina X^P znázorniť ako *n*-dimenzionálna polyedrická množina, v ktorej *n* je počet premenných daného modelu. Pokiaľ existuje množina prípustných riešení, je následne vybrané riešenie, ktoré najlepšie vyhovuje zadanej účelovej funkcii. Toto riešenie je nazývané optimálne riešenie (x^{opt}). Optimálne riešenie úlohy lineárneho programovania (ďalej LP) sa vždy nachádza v jednom z vrcholov polyédru reprezentujúceho množinu X^P. Pokiaľ sa jedná o úlohu lineárneho programovania, potom sú všetky časti modelu popísané pomocou lineárnych rovníc a nerovníc. Všeobecný model LP je zachytený zápisom (1), kde n je počet premenných, m počet obmedzujúcich podmienok, c je vektor koeficientov účelovej funkcie rozmeru $n \times 1$, A je matika technických koeficientov rozmeru $m \times n$, a vektor pravých strán je označovaný ako b s rozmermi $m \times 1$. Platí, že $c, b, A \in$ R, kde R predstavuje reálne čísla.

$$z(x) = c^T x \rightarrow max \tag{1}$$

za podmienok:

$$Ax \leq b$$
$$x \in R_+^n$$

$$\max(c^T x \mid Ax \leq b, x \in R_+^n)$$

2.2 Náhodné procesy a odhad ich parametrov

Stanovenie alebo odhad rozdelení pravdepodobnosti je zásadným problémom, ktorému je nutné pri stochastickom modelovaní vždy čeliť. Vzhľadom k prítomnosti faktoru neistoty nie je možné stanoviť hodnoty premenných deterministicky, ale je možné tieto hodnoty aspoň odhadnúť spolu s pravdepodobnosťou, s ktorou nastanú.

Rozdelenie pravdepodobnosti náhodnej veličiny je pravidlo, ktoré určuje každej hodnote tejto veličiny, prípadne každému intervalu hodnôt pravdepodobnosť, s ktorou môže nastať. Jedná sa teda o zobrazenie $f : R \rightarrow < 0; 1 >$. Toto zobrazenie sa nazýva funkcia hustoty ak sa jedná o spojité náhodné veličiny alebo pravdepodobnostné funkcie, ak sa jedná o diskrétne náhodné veličiny. Primitívne funkcie k funkcii hustoty sa potom nazýva distribučná funkcie rozdelenia náhodnej veličiny $F(x)$ zobrazené v rovnici (2):

$$F(x) = \int_{-\infty}^{x} f\,(t)\,dt \tag{2}$$

Hodnota distribučnej funkcie v bode x udáva pravdepodobnosť, s ktorou náhodná premenná nenadobudne hodnôt vyšších než práve x.

2.3 Algoritmy pre optimalizáciu

Pre optimalizáciu dvoch alebo viacstupňových úloh existujú algoritmy ako je L-shaped metóda, alebo metóda dekompozície bázy.

Metóda L-shaped je založená na princípe rezania rozhodovacích stromov. Začína sa súborom ohraničujúcich podmienok, ktoré obmedzujú možné hodnoty rozhodovacích premenných. Potom je vykonaná optimalizácia pre prvý stupeň rozhodovania, pričom sa ignoruje stochastická povaha problému (predpokladá sa, že neexistujú žiadne neistoty). Na základe tohto výsledku sa generuje "suboptimálny" podproblém pre druhý stupeň. Podproblém pre druhý stupeň sa nazýva L-shaped podproblém a je to problém so zúženým priestorom možných riešení. Jeho cieľom je zlepšiť pôvodné riešenie získané pre prvý stupeň tým, že zohľadní stochastickú povahu. L-shaped podproblém sa rieši pomocou metódy rezania podľa rezov (cutting-plane method). Táto metóda postupne pridáva podmienky, známe ako rezy, ktoré odstránia neplatné časti priestoru riešení. Tieto rezy sa odvodzujú zo stochastických obmedzení modelu a aktualizujú sa postupne tak, aby sa získalo čo najlepšie riešenie. Proces rezania a aktualizácie rezov pokračuje, kým sa nedosiahne konvergencia a nie je možné nájsť lepšie riešenie alebo dosiahnuť požadovanú presnosť. Celkovo povedané, metóda L-shaped umožňuje efektívne riešenie dvojstupňových stochastických modelov tým, že kombinuje optimalizáciu na prvom stupni s postupným zlepšovaním riešenia pomocou rezy na druhom stupni, čím zohľadňuje stochastickú povahu problému (Pereira et al., 2020).

Druhou možnou metódou je princíp metódy dekompozície bázy, ktorý spočíva v rozložení pôvodného problému na menšie, menej zložité podproblémy. Tieto podproblémy sú následne riešené nezávisle a ich riešenia sú následne kombinované tak, aby poskytli riešenie pôvodného problému. Častá aplikácia v prípade, keď je možné rozdeliť model na dva alebo viac podproblémov, pričom každý z nich sa zaoberá jedným aspektom neistoty. Napríklad, v dvojstupňových stochastických modeloch by sa prvý stupeň mohol zaoberať rozhodnutiami priamočiaro, zatiaľ čo druhý stupeň by sa mohol zaoberať rozhodnutiami, ktoré sú ovplyvnené neistotou alebo stochastickými faktormi. Každý z týchto podproblémov je potom riešený samostatne. Po získaní riešení pre jednotlivé podproblémy sa tiež často vykonáva iteratívna aktualizácia, ktorá slúži na zlepšenie a konzistenciu riešení. Viac o tejto metóde je možné sa dozvedieť v dielach Decomposition Methods for Machine Learning with Small, Incomplete or Noisy Datasets (Caiafa et al., 2020) a Machine Learning-Based Epileptic Seizure Detection Methods Using Wavelet and EMD-Based Decomposition Techniques: A Review (Thangarajoo et al., 2021).

3 Modely a metódy riešenia stochastického programovania

Stochastické programovanie je oblasť matematickej optimalizácie, ktorá využíva znalosti teórie pravdepodobnosti a matematickej štatistiky pre zohľadnenie faktoru neistoty, ktorý je v modely prítomný (Dupáčová, 1986). Jedná sa o vedecký odbor, ktorý sa zaoberá náhodnými premennými (Prékopa, 1995). Pokiaľ by sme chceli formulovať všeobecnú formuláciu modelu stochastického programovania vyzerala by nasledovne:

$$f\ (x, c) \rightarrow min \tag{3}$$

za podmienok:

$$g_i(x, c) \leq 0, i = 1, \dots, n$$

kde x je rozhodovací vektor ($x \in R^n$) a c je náhodný vektor z pravdepodobnostného priestoru (Ω, β_Ω, P) a platí, že $c \in \Omega \subset R^n$.

Pokiaľ je modelovanie obmedzené iba na lineárne modely, v tomto prípade modely stochastického lineárneho programovania (SLP), sa dá všeobecne zapísať nasledovne:

$$z = c'^T x \rightarrow \min \tag{4}$$

za podmienok:

$$A'x \leq b'$$

Apostrof nad maticami koeficientov, vektor pravých strán a vektor koeficientov účelovej funkcie symbolizuje, že jeden alebo viac prvkov je zadaný pomocou stochastickej náhodnej premennej.

3.1 Klasifikácia modelov stochastického programovania

Pri stochastickom programovaní sa je možné stretnúť s viacerými kritériami, podľa ktorých sa dajú stochastické optimalizačné modely klasifikovať. Voľba konkrétneho modelu a postup jeho riešenia záleží na účelu a cieli modelovania a všeobecne na charaktere rozhodovacieho problému, ktorý má byť pomocou modelu riešený (Prékopa, 1995).

Prvým hľadiskom klasifikácie je čas, kedy je potrebné učiniť rozhodnutie. Prvou možnosťou je, že je možné počkať na realizáciu náhodného vektoru c a až potom na základe tejto informácie sa rozhodnúť. Tento model sa nazýva *wait-and-see*. Takýto prípad obsahuje náhodnú premennú a je obdobou deterministického rozhodovacieho problému (Prékopa, 1995). Druhá skupina úloh, v ktorý sa musí vykonať rozhodnutie x pred tým, ako je známa realizácia náhodného vektoru c sa nazývajú *here-and-now* úlohy. V ekonomickej praxi sú častejšie a jedná sa o výpočtovo zložitejšie problémy. Typickým príkladom z praxe je napr. problém voľby investičného portfólia kedy sú známe iba aktuálne nákupné alebo predajné ceny, ale tieto ceny sa budú ďalej v budúcnosti náhodne vyvíjať a meniť a výnos zo zvoleného portfólia preto nejde predom presne stanoviť. S tým, či sa rozhodnutie realizuje pred alebo po pozorovaní náhodnej premennej súvisí ďalšie delenie modelu SLP na statické a dynamické modely.

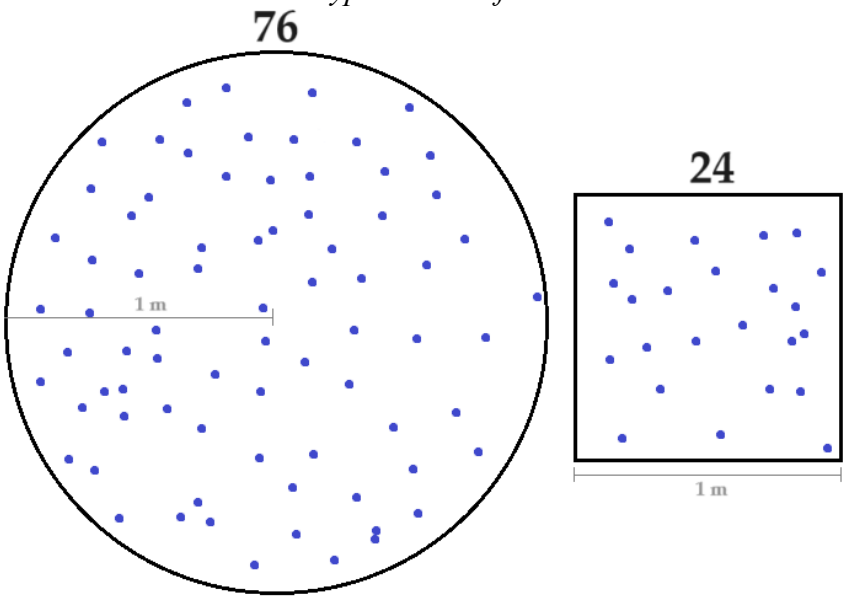
Statické modely sa používajú na predpovedanie statických výsledkov v danom momente, pričom sa najskôr realizuje rozhodnutie a potom až nasleduje pozorovanie (*here-and-now* úlohy). Statické modely sú jednostupňové (*single-stage*) Pri statických modeloch rozpoznávame najznámejšiu simuláciu Monte Carlo, ktorá predstavuje štatistický experiment umožňujúci riešiť úlohy deterministického alebo stochastického charakteru. Na experimentovanie je zostavená pravdepodobnostná úloha, ktorej výsledok riešenia má taktiež pravdepodobnostný charakter, čiže ide o štatistický odhad hodnôt jednotlivých veličín. Presnosť riešenia je závislá na počte opakovaných experimentov. Využitie simulácie je možné pri hľadaní hodnoty Ludolfovho čísla π. Na obrázku 1 si uveďme príklad výpočtu Ludolfovho čísla. Majme dve nádoby kruhovej a štvorcovej podstavy umiestnené vedľa seba spoločnej podložke (podstave). Nech je polomer kruhu a strana štvorca jeden meter. Pokiaľ by sme náhodne púšťali guľôčky zvrchu na podstavu, jednotlivé nádoby by sa pomaly plnili guľôčkami. Výsledok na výpočet Ludolfovho čísla spočíva v podiele počtu guľôčok v kruhu s počtom guľôčok v štvorci.

$$\pi \approx \frac{Počet\ guľôčok\ v\ kruhu}{Počet\ guľôčok\ v\ štvorci} \tag{5}$$

Čím vyšší počet iterácií (guľôčok) tým je vyššia presnosť experimentu a nastáva tu konvergencia k hodnote π. Pokiaľ by sme generovali 100 guľôčok, kde 76 by sa nachádzalo v kruhovej nádobe a 24 v štvorcovej nádobe výsledok by bol 3.167 (5). Ďalšie aplikácie statických modelov sú napríklad pri softvérovom modelovaní a dizajne, kde je modelovanie

využité na popis statickej štruktúry modelovaného systému alebo pri adaptívnom filtrovaní referenčného signálu.

Obr. 1: Výpočet Ludolfovho čísla



Zdroj: Vlastné spracovanie

Dynamické modely sa používajú na modelovanie systémov vyvíjajúcich sa v čase spojito, pričom stavy systému sa menia v diskrétnych časových okamihoch a aspoň jedno pozorovanie musí byť realizované pred vykonaním rozhodnutia (kombinácia *wait-and-see* úlohy s úlohou *here-and-now*). V praxi sa simulačné modelovanie uplatňuje na analýzu a optimalizáciu, napríklad výrobných procesov, organizáciu skladov, logistických procesov, plánovania výroby, finančného plánovania, riadenia vnútropodnikovej dopravy, riadenia a plánovania projektov. Simulácie teda použijeme ak skúmame vzťahy v rámci komplexných systémov. Pri skúmaní informačných, organizačných a environmentálnych zmien na vývoj systému. Pri zlepšovaní chápania systému, nakoľko získané informácie nám zo simulácie upresnia jeho správanie.

Stochastické modely sa teda dajú klasifikovať nasledovne:

Podľa počtu rozhodnutí a možnosti svoje rozhodnutia neskôr korigovať: jednostupňové a viacstupňové. Podľa počtu časových období, pre ktoré sa rozhodnutie prijíma: single-period a multi-period. Podľa toho, či sa pred rozhodovaním poznáme realizáciu náhodného vektoru: here-and-now a wait-and-see. Podľa linearity modelu: lineárne (SLP) a nelineárne.

Podľa typu náhodnej premennej: s diskrétnou náhodnou premennou a so spojitou náhodnou premennou. Postup pri riešení ľubovoľnej stochastickej optimalizačnej úlohy je nasledovný (Prékopa 1995):

1. Zostrojenie deterministického podkladového modelu. Ten slúži na ujasnenie si štruktúry cieľového modelu, prípadne pre získanie prvého odhadu optimálnej hodnoty účelovej funkcie.
2. Výber parametrov modelu, ktoré v modeli budú stochastické.
3. Stanovenie rozdelenia pravdepodobnosti jednotlivých stochastických premenných. Buď je možné využiť niektoré z „klasických“ rozdelení, u ktorých je známa rovnica distribučnej funkcie a funkcie hustoty a potom stačí iba odhadnúť parametre týchto rozdelení. Druhou možnosťou je stanoviť si vlastné empirické rozloženie pravdepodobnosti.
4. Stanovenie ďalších parametrov modelu.

5. Prevod stochastického modelu na jeho deterministický ekvivalent pomocou príslušných metód a algoritmov.
6. Optimalizácie deterministického modelu.

3.2 Využitie náhodnej zložky v stochastických modeloch

Prvý prípad ako môže náhodná zložka ovplyvňovať model je pri ovplyvnení koeficientov účelovej funkcie, pričom obmedzujúce podmienky sú stanovené deterministicky. Typickým príkladom takéhoto modelu je napr. úloha optimalizácie portfólia, kde sú stochastické výnosy z jednotlivých aktív a tiež riziko spojené s držaním týchto výnosov. Cieľom je maximalizovať úžitok majiteľa portfólia, ktorý rastie s celkovým výnosom z portfólia a ktorý v prípade risk-averse agenta klesá s rizikom portfólia viď Kall a Mayer (2011). Aby bolo možné realizovať optimalizáciu, je nutné zaviesť vhodné usporiadanie. Usporiadanie sa zavedie tým, že je účelová funkcia transformovaná do deterministickej podoby. Birge a Louveaux (1997), Kall a Mayer (2011) a Dupačová (1986) rozlišujú dve možnosti, ako ide zo stochastickej účelovej funkcie odstrániť náhodnú zložku. Prvou možnosťou je previesť účelovú funkciu na jej strednú (očakávanú) hodnotu podľa tzv. *E*-kritéria. Nevýhodou je, že v tomto prípade je využitá iba malá časť informácie o rozdelení pravdepodobnosti náhodnej premennej. Druhou možnosťou je do deterministického prepisu zahrnúť tiež riziko, ako tomu je napr. u modelov výberu portfólia. Riziko je možné modelovať rôznymi spôsobmi a pomocou rôznych ukazovateľov, vždy sa ale dá povedať, že pri odpore k riziku sa so zvyšujúcim sa ukazovateľom klesá celkový úžitok.

Ak je pri transformácii stochastickej účelovej funkcie do deterministickej formy zohľadnená stredná hodnota aj riziko vo forme napr. rozptylu, jedná sa o *mean-risk* modely.

Všeobecný zápis pomocou úžitkovej účelovej funkcie je nasledovný:

$$z(x) = E[f(x,c)] + \psi k \rightarrow max \tag{6}$$

kde $E[f(x,c)]$ je stredná hodnota pôvodnej funkcie, ψ je koeficient vyjadrujúci postoj k riziku (pre risk-averse agentov je hodnota ψ záporná, pre risk-lover agentov naopak kladná a pre agentov s neutrálnym vzťahom k riziku je ψ=0 hodnota účelovej funkcie je potom daná iba strednou hodnotou) a k je miera rizika.

Pri modeloch so stochastickou premennou v podmienkach vlastného obmedzenia, môžu byť náhodnou premennou nahradené koeficienty na ľavých stranách podmienok alebo konštanty na pravých stranách obmedzujúcich podmienok, prípadne sa môže jednať o kombináciu týchto prípadov. Model SLP so stochastickou premennou v podmienke vlastného obmedzenia má nasledovný všeobecný zápis:

$$z(x) = c^T x \rightarrow \min \tag{6}$$

za podmienok, že:

$$g_i\,(x,c) \leq 0, i = 1,2,\dots,n\,,$$

kde g_i je lineárna funkcia agregujúca ľavou aj pravou stranou obmedzujúcej podmienky a je ovplyvnená náhodnou zložkou c.

3.3 Viacstupňové stochastické modely

Hlavná myšlienka viacstupňových stochastických modelov bola predstavená pri klasifikácii modelov stochastického programovania.

Dvojstupňové stochastické programovanie sa vyznačuje rozhodnutiami, ktoré sú robené v dvoch fázach. Prvá fáza predstavuje rozhodnutia, ktoré sa musia urobiť predtým, ako sa stane nejaká náhodná udalosť (napríklad predaj tovaru pred tým, ako sa dozvieme o dopyte). Druhá fáza predstavuje rozhodnutia, ktoré sa robia po tom, ako sa táto náhodná udalosť stane (napríklad nákup dodatočného tovaru, keď je dopyt vyšší ako očakávaný). Príkladom môže byť farmárov problém, kde farmár musí rozhodnúť, koľko z každej plodiny vypestuje (prvý stupeň), a potom koľko z každej plodiny predá alebo kúpi (druhý stupeň). Všeobecný zápis dvojstupňovej úlohy je nasledovný:

$$z = c^T x + E(\min\{\, c'^T y \mid Ax + A'y \leq b\}) \rightarrow min, x \in X \qquad (7)$$

kde $x \in R^n$ je rozhodovací vektor prvého stupňa úlohy (*here-and-now* úloha) a $y \in R^n$ je rozhodovací vektor druhého stupňa, v ktorom je realizácia náhodného vektoru známa. c^T a A je vektor koeficientov účelovej funkcie, resp. matice ľavých strán obmedzení pre prvý stupeň, ich ekvivalenty sa symbolom „ ´ “ sa viažu k druhému stupňu. Množina X obsahuje všetky také vektory x, pre ktoré má úloha $\min\{c'^T y \mid Ax + A'y \leq b\}$ prípustné riešenie pre každú možnú hodnotu náhodného vektoru c. Ten obsahuje údaje o náhodných premenných, ktoré sú odhalené pred druhým stupňom (buď sú stochastické všetky parametre c´, A, A´, b alebo iba niektoré z nich). V prvom stupni je optimalizovaná jednak deterministická časť účelovej funkcie $c^T(x)$ plus očakávané optimum účelovej funkcie v druhom stupni $\min\{c'^T y\}$. Druhý stupeň potom určuje rozhodnutie y, ktoré bude prijaté, ak sú známe hodnoty náhodného vektoru c, a ak bolo skôr rozhodnutie x prijaté. Iný pohľad na druhý stupeň môže byť ten, že je vnímaný ako akási „oprava“ pri porušení pôvodných obmedzujúcich podmienok $Ax \leq b$, ktoré bolo pôvodne zaťažené náhodou. Ľavé strany porušených podmienok sú potom kompenzované (viď $A'y$ v obmedzujúcej podmienke) a táto kompenzácie vedie k penalizácii účelovej funkcie ($c'^T y$). Pre samostatnú optimalizáciou dvoch a viacstupňových úloh existuje mnoho algoritmov, ako napríklad „L-shaped“ metóda alebo metóda dekompozície bázy.

Viacstupňové stochastické programovanie je rozšírením dvojstupňového stochastického programovania na viacero stupňov alebo časových období. Každé obdobie môže mať vlastné rozhodnutia a náhodné udalosti. Príkladom môže byť plánovanie výroby a skladovania v továrni, kde sa v každom období musí rozhodnúť, koľko výrobkov vyrobiť, koľko skladovať a koľko predávať, pričom dopyt je náhodný a môže sa líšiť v každom období.

S rekurentnými rozhodnutiami sa vyznačujú rozhodnutiami, ktoré sa musia robiť opakovanie v čase, pričom každé rozhodnutie ovplyvňuje stav systému pre budúce rozhodnutia. Príkladom môže byť riadenie zásob, kde sa musí rozhodnúť, koľko objednať v každom období, pričom množstvo objednaného tovaru ovplyvňuje množstvo tovaru na sklade v budúcich obdobiach.

S časovým horizontom sa vyznačujú rozhodnutiami, ktoré musia byť urobené v rámci istého časového horizontu. Časový horizont môže byť konečný (napríklad plánovanie výroby na nasledujúci rok) alebo nekonečný (napríklad plánovanie dôchodkového fondu s neistým dátumom smrti). Príkladom môže byť plánovanie investícií, kde sa musí rozhodnúť, koľko investovať do aktív v každom období, pričom výnosy z aktív sú náhodné a môžu sa líšiť v každom období.

4 Dvojstupnňové stochastické programovanie s využitím programu MS Excel

V tomto príklade vytvárame simuláciu, kde predpokladáme existenciu apartmánového komplexu na Slovensku, kde je naplánovaná štvordňová konferencia a organizátori chcú čo najpresnejšie určiť, koľko hotelových izieb rezervovať, aby sa minimalizovali celkové náklady. Hostia budú ubytovaní v hoteli Slopartments, kde je maximálny počet hotelových apartmánov

je 400. Hotel Slopartments dal možnosť kvôli konferencii využiť zvýhodnenú cenu 145 € za noc ak sa apartmány rezervujú vopred. Za každý rezervovaný apartmán, ktorý nie je využitý musia organizátori platiť plnú sumu, čo činí 193 € za každý deň konferencie. Alternatívne, ak je viac uchádzačov o konferenciu ako rezervovaných apartmánov, organizátori musia zaplatiť 229 € za osobu na deň. Tieto náklady pokrývajú najbližší menej kvalitný hotel Spoor s cenou 161 € na deň plus transport z hotela na konferenciu v hodnote 68 €. Neistotou je, že organizátori nevedia koľko ľudí príde na konferenciu. Podľa organizátorov, je očakávaná návštevnosť vopred predpovedateľná a je znázornená v tabuľke 1.

Tab. 1: Predom známe informácie od usporiadateľov

Scenár (zajtrajšok)	**Pravdepodobnosť**	**Potrebné hotelové izby**
1	15%	250
2	25%	300
3	30%	350
4	20%	400
5	10%	450

Zdroj: Vlastné spracovanie

r = počet izieb, ktoré sa rezervujú dopredu v hoteli-1 (premenná z 1. stupňa)
o_i, u_i= nad-rezervácia, pod-rezervácia izieb podľa jednotlivých scenárov (i= 1, ..., 5) (2.stupeň)
Nad-rezervácia nastáva ak r je väčšie ako dopyt a pod-rezervácia ak r je menšie ako dopyt.
Formulácia matematického modelu je nasledovná:

$$\boldsymbol{min_z(u) = 145r + 0.15(48o_1 + 229u_1) + 0.25\,(48o_2 + 229u_2) + 0.30\,(48o_3 + 229u_3) + 0.20\,(48o_4 + 229u_4) + 0.10\,(48o_5 + 229u_5)}$$

kde 145 predstavuje cenu pred-rezervovaného apartmánu, desatinné hodnoty zodpovedajú pravdepodobnosti nastátia udalosti, 48 predstavuje dodatočné náklady ak nad-rezervujem, 229 ak pod-rezervujem. Účelová funkcia zobrazuje očakávané náklady. 3 predstavuje trojdňovú konferenciu.

Obmedzenia (premenná – prvý stupeň):

$$r \leq 400$$

Obmedzenia (premenné – druhý stupeň):

$$r - o_1 + u_1 = 250$$
$$r - o_2 + u_2 = 300$$
$$r - o_3 + u_3 = 350$$
$$r - o_4 + u_4 = 400$$
$$r - o_5 + u_5 = 450$$
$$r, o_i, u_i \geq 0$$

V programe MS Excel sme si zostrojili vyššie navrhnutý príklad. Žlté polia predstavujú polia, ktorých hodnota bola pôvodne nevyplnená. Oranžové pole predstavuje účelovú funkciu na výpočet očakávaných nákladov po druhom stupni modelu.

Obr. 2: Optimalizácia v MS Excel

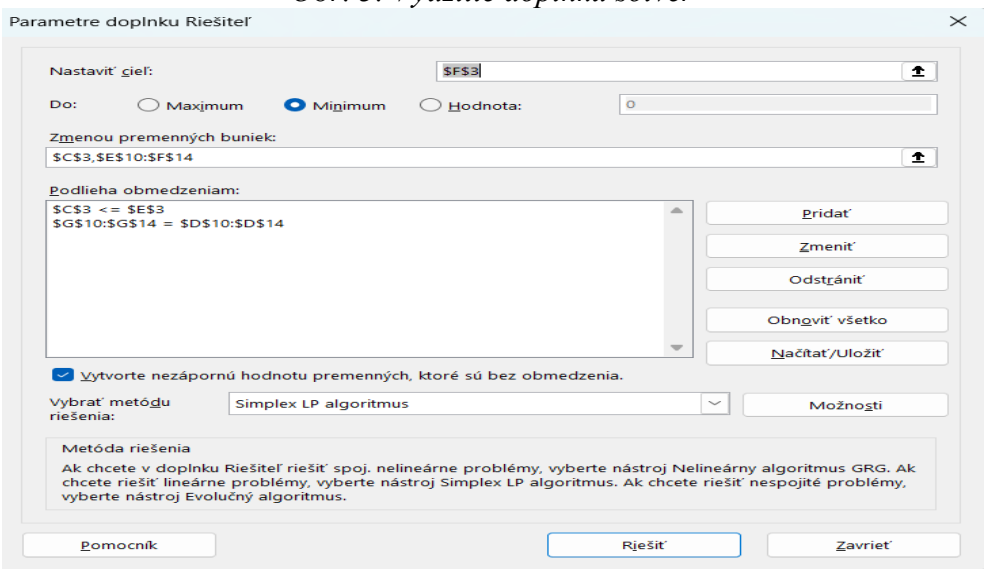
			Dostupných apartmánov	Očakávané náklady
Rezervovať pre dopyt v Slopartments	300	<=	400	221,240.00 €
Vopred rezervovanie Slopartments	145.00 €			
Plná cena apartmánu Slopartments	193.00 €			
Spoor hotel	161.00 €			
Cestovné zo Spoor	68.00 €			

Scenár	Pravdepodobnosť	Dopyt	Nepoužité izby (nad-rezervácia o)	Dodatočne požadované (pod-rezervácia u)	Rezervované - nadbytok + nedostatok
1	15%	250	50	0	250
2	25%	300	0	0	300
3	30%	350	0	50	350
4	20%	400	0	100	400
5	10%	450	0	150	450
		Náklady	48	229	

Zdroj: Vlastné spracovanie

Princíp výpočtu spočíva vo využití riešiteľa (doplnok solver), kde minimalizácia je vypočítaná v bunke F3 ako cieľová hodnota. Následne sú do modelu vnesené zmeny buniek žltej farby. Týmto interpretujeme nastavenie ideálneho rozloženie izieb. Na základe stanovených podmienok rezervovaných izieb, aby boli ich počet bol menší alebo rovný ako 400 dostupných apartmánov a aby sa rovnali počty návštevníkov (G stĺpec) a dopytu (D stĺpec). Po spustení riešiteľa lineárnym simplexovým algoritmom vidíme výsledok 300 izieb. Tento postup je zobrazený na obrázku 3.

Obr. 3: Využitie doplnku solver



Zdroj: Vlastné spracovanie

Tento výsledok vieme interpretovať nasledovne: neexistuje také riešenie, ktoré je možné učiniť dnes, aby bolo ideálne vo všetkých prípadoch. Tento dvojstupňový stochastický optimalizačný model udáva optimálnu hodnotu pre usporiadateľov takýchto typov konferencií očakávaný dlhodobý priemer minimalizujeme výdavky najviac ako je možné. Aktuálne je najlepšie pre usporiadateľov zarezervovať vopred 300 apartmánov. V 25 % prípadoch budeme v druhom scenári bez výdavkov. V ostatných prípadoch sme minimalizovali náklady tak, aby v každom možnom výsledku boli náklady minimálne. Na 15 % budeme mať rezervovaných 50 izieb naviac a v 60 % prípadoch sa bude musieť doobjednať hosťom ďalšie ubytovanie v hoteli Spoor. Zaujímavé si je všimnúť, že 30 % scenár s dopytom 350 nebol najlepším riešením tejto situácie. Očakávané náklady na štvordňovú konferenciu sa po minimalizácii dostávajú na hodnotu 221 240 €. Priebeh výpočtu príkladu je možné vidieť na obrázkoch 2 a 3.

5 Riešenie úlohy prístupom s normálnou distribúciou pravdepodobnosti s využitím Python

V tejto kapitole budú riešené úlohy v programovacom jazyku Python, kde bude využité pri zobrazení neistoty normálne pravdepodobnostné rozdelenie.

5.1 Úloha farmár

Pri tejto úlohe sme vychádzali zo skonštruovaných príkladov uvádzaných v dielach Linear programming with Python and Pulp (Parganiha, 2018) a Crop profit optimization for farmers (Romero & Smith, 2016). V tomto príklade uvažujeme o poľnohospodárovi, ktorý má 100 árov pôdy a môže pestovať pšenicu alebo kukuricu. Farmár sa chce rozhodnúť, koľko árov každej plodiny vysadí, aby maximalizoval očakávaný zisk. Zisk z každej plodiny je však neistý vzhľadom na faktory, ako sú počasie, škodcovia a trhové ceny. Túto neistotu modelujeme tak, že predpokladáme, že zisky na ár pšenice a kukurice sú náhodné premenné, ktoré sa riadia normálnym rozdelením. Stredná hodnota tohto rozdelenia predstavuje očakávaný zisk na ár a štandardná odchýlka predstavuje variabilitu alebo riziko. Problém poľnohospodára je potom formulovaný ako stochastický optimalizačný problém. Cieľom je maximalizovať očakávaný celkový zisk, ktorý je súčtom očakávaných ziskov z každej plodiny. Rozhodovacími premennými sú počet árov, ktoré sa majú osiať pre každú plodinu. Medzi obmedzenia problému patrí obmedzenie týkajúce sa pôdy, ktoré stanovuje, že celkový počet vysadených akrov nesmie prekročiť celkovú dostupnú pôdu, a obmedzenie nezápornosti, ktoré stanovuje, že počet akrov vysadených pre každú plodinu musí byť nezáporný. Na riešenie tohto problému používame prístup simulácie Monte Carlo. Táto metóda je štatistickým experimentom, kde presnosť výsledku stúpa s počtom iterácií a popisujeme ho v kapitole 3.1. To zahŕňa vytvorenie mnohých scenárov pre zisky na aker pre každú plodinu, riešenie optimalizačného problému pre každý scenár a následné spriemerovanie výsledkov. Riešenie nám poskytne priemerný počet akrov, ktoré treba osiať pre každú plodinu, čím sa maximalizuje očakávaný celkový zisk, pričom sa zohľadní neistota v ziskoch na aker pre pšenicu a kukuricu.

Matematická formulácia modelu:

$$E[Z] = E[c_1]x_1 + E[c_2]x_2$$

Obmedzenia:

$$x_1 + x_2 \leq 100$$
$$x_1, x_2 \geq 0$$

Nech x_1 a x_2 sú premenné reprezentujúce áre, na ktorých sa pestuje pšenica a kukurica,. c_1a c_2 predstavujú náhodné premenné reprezentujúce zisky na ár z pšenice a kukurice. Pri týchto premenných využívame normálne rozdelenie s priemerom μ_1 pre pšenicu a μ_2 pre kukuricu a štandardnou odchýlkou pre pšenicu σ_1 a kukuricu σ_2. $E[c_1]$ a $E[c_2]$ sú očakávané zisky v prepočte na ár pre pšenicu a kukuricu. Očakávaný zisk je priemerom normálneho rozdelenia, kde $E[c_1] = \mu_1$ a $E[c_2] = \mu_2$. Táto matematická formulácia cieli na nájdenie počtu árov na zasadenie každej z plodín a maximalizácie celkového zisku pomocou využitia neistoty v ziskoch na ár pre pšenicu a kukuricu.

Obr. 4: Optimalizácia výsevu plodín

```python
# Importujeme potrebné knižnice
import numpy as np
from scipy.optimize import linprog

# Nastavíme priemerný zisk a štandardnú odchýlku na akejkoľvek ploche pre pšenicu a kukuricu
mu = np.array([50, 30])  # Priemerný zisk
sigma = np.array([10, 5])  # Štandardné odchýlky

# Nastavíme počet scenárov, ktoré chceme generovať
num_scenarios = 1000

# Nastavíme seed pre generátor náhodných čísel pre reprodukovateľnosť
np.random.seed(0)

# Generujeme scenáre pre zisk na akejkoľvek ploche pomocou normálneho rozdelenia
profits = np.random.normal(mu, sigma, (num_scenarios, 2))

# Inicializujeme polia na ukladanie výsledkov
x1_values = np.zeros(num_scenarios)  # Akre pšenice pre každý scenár
x2_values = np.zeros(num_scenarios)  # Akre kukurice pre každý scenár

# Cyklus pre každý scenár
for i in range(num_scenarios):
    # Nastavíme koeficienty cieľovej funkcie pre aktuálny scenár
    # Poznámka: Používame záporné hodnoty, pretože linprog robí minimalizáciu
    c = -profits[i]

    # Nastavíme koeficienty pre nerovnosti
    A = [[1, 1]]  # Koeficienty pre obmedzenie pôdy

    # Nastavíme pravú stranu nerovností
    b = [100]  # Celková dostupná pôda

    # Nastavíme hranice pre rozhodovacie premenné
    x0_bounds = (0, None)  # Akre pšenice musia byť >= 0
    x1_bounds = (0, None)  # Akre kukurice musia byť >= 0

    # Riešime lineárny programovací problém pre aktuálny scenár
    res = linprog(c, A_ub=A, b_ub=b, bounds=[x0_bounds, x1_bounds], method='highs')

    # Uložíme riešenie pre aktuálny scenár
    x1_values[i] = res.x[0]
    x2_values[i] = res.x[1]

# Vypočítame priemerné hodnoty zo všetkých scenárov
x1_avg = x1_values.mean()
x2_avg = x2_values.mean()

# Vytlačíme priemerné hodnoty
print('Priemerný počet akrov pšenice na výsadbu:', x1_avg)
print('Priemerný počet akrov kukurice na výsadbu:', x2_avg)
```

```
Priemerný počet akrov pšenice na výsadbu: 96.3
Priemerný počet akrov kukurice na výsadbu: 3.7
```

Zdroj: Vlastné spracovanie

5.2 Úloha alokácie portfólia

Téma investícií je široko využívaná za pomoci stochastických optimalizačných prístupov. V táto téma bola skúmaná vo viacerých dielach, napríklad aj v diele A hybrid combinatorial approach to a two-stage stochastic portfolio optimization model with uncertain asset prices (Cui, Bai & Ding, 2020). Dvojstupňové stochastické modely boli popísané v kapitole 3.3. Existujú viaceré prístupy na optimalizáciu a odhad ideálneho portfólia. Alternatívna metóda na hľadanie váh portfólia môže predstavovať minimalizačná funkcia, ktorú bude v našom príklade v kóde predstavovať knižničná funkcia basinhopping. Optimalizačný kód vyberá optimálne váhy portfólia a spolu s nimi vypíše aj očakávaný výnos portfólia a riziko resp. štandardnú odchýlku portfólia. Výpočet úlohy je znázornený na obrázku 6. V tomto príklade budeme pracovať s troma aktívami, kde očakávané výnosy sú 8 %, 12 %, 10 % ročne. Štandardné odchýlky pohybu cien troch spoločností sú 10 %, 15 % a 12 %. V tejto úlohe sa budeme snažiť o optimalizáciu váh jednotlivých cenných papierov za pomoci využitia minimalizácie rizika portfólia.

Po inicializácii očakávaných výnosov a štandardných odchýlok sme vytvorili korelačnú maticu. Pri minimalizácii rizika je nutné mať vytvorenú funkciu na výpočet rizika čo zabezpečuje funkcia portfolio_risk, pričom táto funkcia dostáva ako vstupné argumenty váhy a kovariančnú maticu. Simulácia minimalizácie portfólia je zabezpečovaná pomocou funkcie basinhopping. Táto metóda kombinuje lokálnu optimalizáciu s náhodnými krokmi v priestore váh portfólia. Nachádza sa tu viacero argumentov, ktorými sa dá dodefinovať napríklad dodatočné argumenty, ktoré budú použité pri iteráciách minimalizačného algoritmu. Ďalej sú

tu definované obmedzenia, ktoré určujú aké hodnoty môžu mať optimalizačné parametre (váhy). Počet váh musí byť rovný jednej. Výsledok váh je zaznačený do premennej optimal_weights a vypísaný s očakávaným výnosom a rizikom portfólia. Tento kód minimalizuje riziko pre tri aktíva. Obdobný postup je možné aplikovať aj na reálne aktíva.

Obr. 6: Optimalizácia portfólia pomocou očakávaných výnosov a štandardnej odchýlky

```python
import numpy as np
from scipy.optimize import minimize
from scipy.optimize import basinhopping

# Predpokladané výnosy a štandardné odchýlky jednotlivých aktív
expected_returns = [0.08, 0.12, 0.10]  # Predpokladané očakávané výnosy
std_deviations = [0.10, 0.15, 0.12]     # Štandardné odchýlky výnosov

# Korelačná matica medzi výnosmi aktív (v tomto príklade uvádzame úplne nezávislé aktíva)
correlation_matrix = np.eye(3)          # Jednotková matica

# Funkcia na výpočet rizika portfólia
def portfolio_risk(weights, cov_matrix):
    return np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))

# Funkcia na minimalizáciu rizika portfólia
def min_risk(weights):
    return portfolio_risk(weights, np.diag(std_deviations))

# Funkcia na generovanie náhodných váh portfólia
def generate_random_weights():
    #Generuje náhodné váhy pre aktíva v portfóliu s použitím Dirichletovho rozdelenia.
    return np.round(np.random.dirichlet(np.ones(len(expected_returns)), size=1)[0], 4)

# Konštraint pre sumu váh
constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})

# Simulované minimalizácie rizika portfólia

#metóda basinhopping na minimalizáciu rizika portfólia - kombinuje lokálnu optimalizáciu s náhodnými krokmi
#v priestore váh portfólia, čo zahŕňa stochastický prvok

#minimizer_kwargs={"constraints": constraints} definuje dodatočné argumenty,ktoré budú použité pri volaní interného
#minimizačného algoritmu počas simulácie, constraints je dodatočný parameter - zoznam obmedzení, ktoré určujú, aké hodnoty
#môžu mať optimalizačné parametre (v tomto prípade váhy aktív v portfóliu).
#definujeme, že súčet váh musí byť rovný 1, čo zaručuje, že celková alokácia v portfóliu je 100%.

result = basinhopping(min_risk, x0=generate_random_weights(), minimizer_kwargs={"constraints": constraints})

# Optimalne váhy portfólia
optimal_weights = np.round(result.x, 4)

# Výpočet očakávaného výnosu portfólia
expected_return = np.round(np.dot(optimal_weights, expected_returns), 4)

# Výpočet rizika portfólia (štandardnej odchýlky) zaokrúhlené na 4 desatinné miesta
# Diagonálna matica zo vstupnej matice - np.diag vráti vektor obsahujúci prvky na diagonále tejto matice.
risk = np.round(portfolio_risk(optimal_weights, np.diag(std_deviations)), 4)

print("Optimalne váhy portfólia:", optimal_weights)
print("Očakávaný výnos portfólia:", expected_return)
print("Riziko portfólia (štandardná odchýlka):", risk)

```

```
Optimalne váhy portfólia: [0.4    0.2667 0.3333]
Očakávaný výnos portfólia: 0.0973
Riziko portfólia (štandardná odchýlka): 0.2
```

Zdroj: Vlastné spracovanie

6 Záver

V závere práce sme sa zamerali na štúdium stochastického programovania a jeho aplikáciu v rôznych oblastiach rozhodovania s neistotou. Práca bola zameraná na vysvetlenie základných pravdepodobnostných rozdelení a metód stochastického programovania, s dôrazom na využitie normálneho pravdepodobnostného rozdelenia. V rámci práce sme popísali význam metód ako L-Shaped a Dekompozícia bázy a aplikáciu metódy L-shaped v riešení komplexných rozhodovacích problémov. Okrem toho sme použili prístup Monte Carlo na simuláciu náhodných procesov a odhadovanie hodnôt v stochastických problémoch, čo mi umožnilo analyzovať a riešiť problémy s neistotou. Dôležitou súčasťou práce bolo tiež ukázať praktické príklady, kde boli využité normálne pravdepodobnostné rozdelenie na modelovanie neistoty a rizika. Tieto príklady umožnili ilustrovať a aplikovať teoretické poznatky na reálne situácie, čo posilnilo pochopenie problematiky stochastického programovania. Širokú použiteľnosť stochastického prístupu pri ekonomických aplikáciách vidíme v mnoho dielach ako sú napríklad A stochastic programming approach to multicriteria portfolio optimization (Şakar & Köksalan, 2013) alebo An interactive approach to stochastic programming-based portfolio optimization (Köksalan & Şakar, 2016). Stochastický prístup je vhodné použiť v prípade, kde hodnoty nie sú isté (v prípade neistoty). Toto je ideálne pre prácu s finančnými trhmi, kde cena nie je vopred známa a je tak možné vytvoriť predpovede potenciálnej ceny aktív. Celkovo považujem prácu za dôležitý príspevok k pochopeniu a aplikácii stochastického programovania

v praxi. Verím, že obsiahnuté poznatky a príklady poskytnú čitateľom užitočné nástroje na riešenie rozhodovacích problémov v prostrediach s neistotou a náhodnosťou.

Príspevok bol spracovaný v rámci riešenia grantovej úlohy VEGA – 1/0120/23 „Environmentálne modely ako nástroj ekologicko-ekonomických rozhodnutí“.

Literatúra

1. Birge, J.R., & Louveaux, F. (1997). Introduction to Stochastic Programming. Springer-Verlag, New York.
2. Brooks, S. (2002). Markov chain Monte Carlo method and its application. *The Statistician*, *47*(1), 69–100.
3. Caiafa, C. F., Solé-Casals, J., Marti-Puig, P., Zhe, S., & Tanaka, T. (2020). Decomposition methods for machine learning with small, incomplete or noisy datasets. *Applied Sciences*, *10*(23), 8481.
4. Cui, T., Bai, R., Ding, S., Parkes, A. J., Qu, R., He, F., & Li, J. (2020). A hybrid combinatorial approach to a two-stage stochastic portfolio optimization model with uncertain asset prices. *Soft Computing*, *24*, 2809-2831.
5. Dupačová, J. (1986). Stability in stochastic programming with recourse. Contaminated distributions. *Stochastic Programming 84 Part I*, 133-144.
6. Kall, P., & Mayer, J. Stochastic linear programming. Models, theory, and computation, 2011. *Springer US, Boston*, *10*, 978-1.
7. Köksalan, M., & Şakar, C. T. (2016). An interactive approach to stochastic programming-based portfolio optimization. *Annals of Operations Research*, *245*, 47-66.
8. Kung, C. C., Li, H., & Lin, R. (2018). Bioenergy strategies under climate change: A stochastic programming approach. *Journal of cleaner production*, *188*, 290-303.
9. Liberti, L., & Kucherenko, S. (2005) Comparison of deterministic and stochastic approaches to global optimization. *International Transactions in Operational Research*, *12*, 263-285.
10. Parganiha, K. (2018). Linear programming with Python and PULP. *International Journal of Industrial Engineering Research and Development*, *9*(3), 1–8.
11. Pereira, G. C., Yoshida, M. I., LeBoulluec, P., Lu, W.-T., Alves, A. P., & Avila, A. F. (2020). Application of artificial intelligence models for predicting time-dependent spring-back effect: The L-shape case study. *Composites Science and Technology*, *199*, 108251.
12. Prékopa, A. (1995). *Stochastic Programming. Optimizations and Programming*. Springer Science & Business Media.
13. Romero, J., & Smith, K. (2016, April). Crop profit optimization for farmers. In *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)* (pp. 289-291). IEEE.
14. Rudyak, V. Ya. & Lezhnev, E.V. (2020). Stochastic molecular modeling the transport coefficients of rarefied gas and gas nanosuspensions. *Nanosystems: physics, chemistry, mathematics*, *11*(3), 285-293.
15. Silva, A., De Brito, J., & Gaspar, P. L. (2016). *Methodologies for service life prediction of buildings: with a focus on façade claddings*. Springer (pp.67-162).
16. Şakar, C., & Köksalan, M. (2013). A stochastic programming approach to multicriteria portfolio optimization. *Journal of Global Optimization*, 57, 299-314.
17. Thangarajoo, R. G., Reaz, M. B. I., Srivastava, G., Haque, F., Ali, S. H. M., Bakar, A. A. A., & Bhuiyan, M. A. S. (2021). Machine learning-based epileptic seizure detection methods using wavelet and EMD-based decomposition techniques: A review. *Sensors*, *21*(24), 8485.