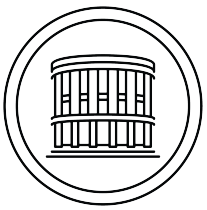
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



RIEŠENIE STOCHASTICKÝCH ÚLOH OPTIMÁLNEHO RIADENIA S VYUŽITÍM APROXIMAČNÝCH METÓD

DIPLOMOVÁ PRÁCA

2026 Marek PATZELT

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

RIEŠENIE STOCHASTICKÝCH ÚLOH OPTIMÁLNEHO RIADENIA S VYUŽITÍM APROXIMAČNÝCH METÓD

DIPLOMOVÁ PRÁCA

Študijný program:	Ekonomická a finančná matematika
Študijný odbor:	Matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	doc. RNDr. Zuzana Chladná, Dr.

Bratislava 2026 **Marek PATZELT**





Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

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

Názov: Riešenie stochastických úloh optimálneho riadenia s využitím aproximačných metód
Solving stochastic optimal control problem via approximation methods

Anotácia: Aproximatívne dynamické programovanie predstavuje efektívnu výpočtovú techniku na riešenie veľkorozmerných úloh diskrétneho stochastického dynamického programovania.

Cieľ: Naštudovať a prehľadne spracovať rôzne techniky aproximatívneho dynamického programovania. Implementovať a porovnať vybrané metódy na praktickom probléme.

Vedúci: doc. RNDr. Zuzana Chladná, Dr.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: doc. Mgr. Igor Melicherčík, PhD.

Dátum zadania: 09.01.2025

Dátum schválenia: 15.01.2025

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

.....
študent

.....
vedúci práce

Čestné vyhlásenie Čestne vyhlasujem, že celú diplomovú prácu na tému „Riešenie stochastických úloh optimálneho riadenia s využitím aproximačných metód", vrátane všetkých jej príloh a obrázkov, som vypracoval samostatne, a to s použitím literatúry uvedenej v priloženom zozname a nástrojov umelej inteligencie. Vyhlasujem, že nástroje umelej inteligencie som použil v súlade s príslušnými právnymi predpismi, akademickými právami a slobodami, etickými a morálnymi zásadami za súčasného dodržania akademickej integrity a že ich použitie je v práci vhodným spôsobom označené.

Miesto a dátum: ..... Titul, meno a priezvisko: .....

Poďakovanie Moje poďakovanie patrí vedúcej diplomovej práce doc. RNDr. Zuzane Chladnej, Dr. za obrovskú trpezlivosť, ochotu a pomoc počas celého procesu písania tejto práce. Ďakujem aj Mgr. Adamovi Fajkusovi za poskytnutie trhových dát.

Abstrakt v štátnom jazyku

PATZELT, Marek: Riešenie stochastických úloh optimálneho riadenia s využitím aproximačných metód [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Zuzana Chladná, Dr., Bratislava, 2026, 64 s.

Táto práca sa zaoberá problematikou riešenia úloh stochastického optimálneho riadenia, ktoré sú v reálnych podmienkach často výpočtovo nezvládnuteľné kvôli tzv. problému dimenzionality. Hlavným cieľom práce je preskúmať a implementovať aproximačné techniky, ktoré umožňujú nájsť dostatočne presné suboptimálne stratégie v rozumnom výpočtovom čase. Pozornosť venujeme najmä aproximácii hodnotovej funkcie pomocou lineárnej regresie a jej porovnanie s prístupom Model Predictive Control (MPC).

V prvej časti práce navrhneme algoritmus určený na riešenie stochastickej úlohy o inkrementálnom batohu. Efektivitu navrhnutého algoritmu a kvalitu získanej stratégie porovnávame s výsledkami dosiahnutými pomocou MPC prístupu. Druhá časť práce je venovaná praktickej aplikácii navrhnutých metód v oblasti energetiky. Presnejšie, zameriavame sa na optimálne obchodovanie s batériovým úložiskom na intradennom trhu, kde rozhodnutia o nabíjaní, vybíjaní alebo nečinnosti závisia od aktuálneho stavu batérie a vývoja cien elektriny.

Práca potvrdzuje, že aplikácia teoretických postupov pri riešení úloh veľkých rozmerov z praxe nie je priamočiara a vyžaduje hľadanie kompromisu medzi výpočtovou náročnosťou a mierou presnosti získaného riešenia. Výsledky naznačujú, že vhodne zvolené aproximačné techniky dokážu efektívne prekonať obmedzenia tradičných metód pri optimalizácii komplexných ekonomických systémov.

Kľúčové slová: Stochastické optimálne riadenie, dynamické programovanie, aproximačné dynamické programovanie, MPC, inkrementálny problém batoha, obchodovanie s batériou

Abstract

PATZELT, Marek: Solving stochastic optimal control problem via approximation methods [Master's Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: doc. RNDr. Zuzana Chladná, Dr., Bratislava, 2026, 64 p.

This thesis addresses the problem of solving stochastic optimal control tasks, which are often computationally intractable in real-world settings due to the curse of dimensionality. The primary objective of this work is to investigate and implement approximation techniques that enable the derivation of sufficiently accurate suboptimal strategies within a reasonable computational time. Particular attention is devoted to the approximation of the value function using linear regression, along with a comparison to the Model Predictive Control (MPC) approach.

In the first part of the thesis, an algorithm is proposed for solving a stochastic incremental knapsack problem. The performance of the proposed algorithm and the quality of the resulting strategies are evaluated and compared with those obtained using the MPC approach. The second part of the thesis focuses on a practical application of the proposed methods in the field of energy systems. Specifically, the problem of optimal trading with a battery storage system on the intraday electricity market is considered, where decisions regarding charging, discharging, or idling depend on the current state of the battery and the evolution of electricity prices.

The thesis demonstrates that applying theoretical methods to large dimensional problems from practice is not straightforward and requires finding an appropriate trade off between computational complexity and the accuracy of the obtained solution. The results suggest that suitably chosen approximation techniques can effectively overcome the limitations of traditional methods in optimizing complex economic systems.

Keywords: Stochastic optimal control, dynamic programming, approximate dynamic programming, MPC, incremental knapsack problem, battery trading

Obsah

Úvod **10**

1 Stochastické optimálne riadenie **13**

1.1 Formulácia úlohy 13

1.2 Rovnica dynamického programovania 14

1.3 Problém dimenzionality 14

1.4 Techniky aproximačného dynamického programovania 16

1.5 Model Predictive Control 17

2 Problém batoha **19**

2.1 Matematická formulácia problému 19

2.2 Výpočtová zložitosť algoritmu 20

2.3 Inkrementálny problém batoha 21

2.4 Formulácia problému 21

2.5 Problém dimenzionality 24

2.6 Lineárna aproximácia hodnotovej funkcie 25

2.7 Algoritmus hľadania stratégie 31

2.8 Riešenie problému inkrementálneho batoha pomocou MPC 33

3 Numerická analýza problému **36**

3.1 Ilustračný príklad 36

3.2 Porovnanie algoritmov 38

3.3 Porovnanie úspešnosti stratégií 41

4 Obchodovanie s batériou **43**

4.1 Formulácia problému obchodovania s batériou 43

4.2 Problém dimenzionality pri obchodovaní s batériou 46

4.3 Lineárny odhad hodnotovej funkcie 47

4.4 Parametre modelu 51

4.5 Modelovanie cien 51

4.6 Výpočet riadiacich premenných 54

4.7 Výsledky 54

Záver **57**

A Príloha A **62**

Príloha A **62**

Úvod

V súčasnom svete rýchleho technologického pokroku a automatizácie sa schopnosť efektívne riadiť komplexné systémy stáva konkurenčnou výhodou. Rozhodovacie problémy, akými sú napríklad riadenie autonómnych vozidiel, optimalizácia portfólia, obchodovanie s batériou na trhu s elektrickou energiou alebo regulácia chemických procesov, vykazujú sekvenčný charakter: rozhodnutia prijaté v jednom časovom okamihu ovplyvňujú nielen okamžitý výnos, ale aj budúci vývoj systému a možnosti ďalšieho rozhodovania.

Z matematického hľadiska spadajú uvedené rozhodovacie problémy pod teóriu optimálneho riadenia. Za zakladateľa tejto disciplíny je považovaný americký matematik Richard Bellman, ktorý v roku 1957 sformuloval tzv. *Princíp optimality*. Tento koncept umožňuje rozložiť komplexný viacetapový problém na sekvenciu jednoduchších, rekurzívne riešiteľných podúloh. Bellmanov princíp optimality položil teoretické základy metódy dynamického programovania. Viac podrobností možno nájsť napríklad v [8].

Aj keď je predstavená metóda výpočtu optimálneho riadenia teoreticky dobre prepracovaná, už aj samotný Bellman identifikoval slabinu pri aplikácii na reálne inžinierske a ekonomické úlohy. Tento fenomén, známy ako *problém dimenzionality* (angl. *curse of dimensionality*) [10], vyjadruje skutočnosť, že s rastúcim počtom stavových premenných narastá výpočtová a pamäťová náročnosť riešenia exponenciálne. V dôsledku toho sa hľadanie optimálneho riešenia stáva pre systémy s vysokou dimenziou v praxi nerealizovateľným.

Táto výpočtová náročnosť sa ešte prehlbuje v prípadoch, keď je časový vývoj systému vystavený náhodným vplyvom vonkajšieho prostredia. Riešenie stochastických úloh optimálneho riadenia tak naráža nielen na rozmer stavového priestoru, ale aj na matematickú komplexnosť spojenú s modelovaním neurčitosti a výpočtom očakávaných hodnôt budúcich výnosov.

Súčasne s rozvojom výpočtovej techniky postupne vznikali viaceré prístupy, ako sa s problémom dimenzionality vysporiadať. Ich cieľom je nájsť optimálny kompromis medzi mierou presnosti získaného riešenia a výpočtovou náročnosťou (časovou aj pamäťovou), potrebnou na jeho dosiahnutie. V predkladanej práci sa zameriavame na tzv. *aproximačné metódy* [10], často označované aj ako *aproximačné dynamické programovanie* (ADP). Podstata týchto metód spočíva v konštrukcii vhodnej aproximácie: buď samotnej stratégie

riadenia (*policy approximation*) alebo hodnotovej funkcie (*value function approximation*), ktorá predstavuje kľúčový prvok pri hľadaní optimálneho riadenia. Hlavnou motiváciou tohto prístupu je získať dostatočne presnú stratégiu riadenia s výrazne nižšími výpočtovými nárokmi, čo umožňuje riešenie úloh, ktoré sú klasickými metódami nezvládnuteľné.

Okrem aproximačného dynamického programovania v práci aplikujeme aj metódou *Model Predictive Control (MPC)*. Tento prístup bol prvýkrát predstavený v práci [11]. Myšlienka MPC prístupu vychádza z opakovaného riešenia optimalizačnej úlohy na kratšom predikčnom horizonte, pričom sa v každom kroku realizuje iba prvé riadenie. Výhodou MPC prístupu je, že umožňuje zohľadniť budúci vývoj systému bez potreby riešiť problém na celom časovom horizonte. V tejto práci slúži ako alternatívny aproximačný prístup, ktorý umožňuje porovnať kvalitu rozhodnutí získaných rôznymi metódami.

Hlavným cieľom tejto diplomovej práce je preskúmať možnosti a efektivitu aproximačných metód pri riešení stochastických úloh optimálneho riadenia. Ako reprezentatívny modelový príklad uvádzame tzv. *inkrementálny problém batoha*, ktorý rozširuje klasickú formuláciu tohto problému o dynamický časový horizont a stochasticky sa meniacu kapacitu. Tento typ úloh, analyzovaný napríklad v práci [2], predstavuje ideálnu testovaciu platformu: má jednoduchú ekonomickú interpretáciu, no zároveň v dôsledku exponenciálneho nárastu stavového priestoru názorne ilustruje limity presných výpočtových metód a ukazuje tak potrebu nasadenia pokročilých aproximačných algoritmov.

Druhým príkladom vychádzajúcim zo súčasnej praxe v oblasti energetiky je problém obchodovania s batériou na krátkodobom trhu s elektrickou energiou. Ide o úlohu stochastického optimálneho riadenia, v ktorej rozhodnutia o nabíjaní, vybíjaní alebo nečinnosti závisia od aktuálneho stavu batérie a vývoja cien elektriny. Modelovanie cien pritom predstavuje samostatnú výzvu, keďže vykazujú vysokú volatilitu, sezónnosť, skokové zmeny, ale aj tendenciu návratu k dlhodobému priemeru. Hľadanie optimálnej stratégie navyše sťažuje štruktúra vnútrodenného (*intraday*) trhu, ktorý umožňuje obchodovať v 96 pätnásťminútových intervaloch dňa súčasne. Táto diskretizácia vedie k extrémnemu nárastu dimenzie priestoru stavov a riadení. V práci navrhneme heuristický algoritmus, ktorý vychádza z aproximačných techník predstavených v [10] a porovnáme jeho výkonnosť s naivnou stratégiou, pri ktorej sa voľba stratégie riadi len okamžitým ziskom v danom čase.

Diplomová práca je rozdelená do štyroch kapitol. V prvej kapitole definujeme kľúčové pojmy stochastického optimálneho riadenia a predstavíme vybrané techniky aproximačného dynamického programovania. Druhá kapitola je venovaná problému inkrementálneho batoha a konštrukcii algoritmu na jeho riešenie. Získané výsledky analyzujeme a vzájomne porovnávame v tretej kapitole. V poslednej, štvrtej kapitole predstavíme matematickú formuláciu problému obchodovania s batériou. V závere zhrnieme výsledky numerických experimentov z obchodovania podľa navrhnutej aproximačnej stratégie.

1 Stochastické optimálne riadenie

Stochastické optimálne riadenie sa zaoberá problémami sekvenčného rozhodovania v podmienkach neistoty. V každej časovej etape systém prechádza do nového stavu v závislosti od aktuálneho stavu a zvoleného riadenia, pričom proces je riadený stochastickou stavovou rovnicou. Táto práca analyzuje modely s diskrétnym časom, t.j. úlohy, v ktorých vývoj skúmaného systému môžeme sledovať a riadiť v diskrétnych časových etapách. Cieľom je nájsť optimálne riadenie v každej časovej perióde tak, aby očakávaný súčet odmien počas daného počtu časových etáp bol maximálny.

1.1 Formulácia úlohy

Uvažujme diskrétny dynamický systém definovaný na konečnom časovom horizonte $t \in \{0, 1, \ldots, T\}$. Stav systému v čase t je opísaný vektorom $x_t \in \mathcal{X}_t$, kde $\mathcal{X}_t$ predstavuje príslušný stavový priestor. V každom kroku ovplyvňujeme stav systému hodnotou riadenia u_t z množiny prípustných riadení $\mathcal{U}_t(x_t)$, ktorá je podmienená aktuálnym stavom systému. Neurčitosť, vyplývajúcu napríklad z vonkajšieho šumu alebo exogénnych šokov, modelujeme pomocou náhodnej premennej z_t. Dynamiku vývoja tohto systému popisuje stavová rovnica v tvare:

$$\begin{aligned} x_{t+1} &= f_t(x_t, u_t, z_t)\,, \qquad t = 0, 1, \ldots, T-1, \\ x_0 &= a_0, \end{aligned} \tag{1}$$

kde f_t označuje prechodovú funkciu a a_0 reprezentuje počiatočný stav systému. Z rovnice (1) vyplýva, že stav systému v čase $t+1$ je plne určený aktuálnym stavom x_t, zvoleným riadením u_t a realizáciou náhodnej premennej z_t. Za predpokladu, že z_t je postupnosť nezávislých náhodných premenných (nezávislých od minulých stavov aj od jej predchádzajúcich realizácií), systém opísaný rovnicou (1) má Markovovskú vlastnosť [9]. To znamená, že aktuálny stav x_t v sebe nesie všetky relevantné informácie z predchádzajúcich časových období potrebné na určenie budúceho vývoja.

Riadiacu premennú v čase t uvažujeme v tvare spätnej väzby $u_t = v_t(x_t)$. Postupnosť spätných väzieb $(v_0, v_1, \ldots, v_{T-1})$ budeme nazývať *stratégiou*. Cieľom úlohy stochastického optimálneho riadenia je nájsť optimálnu stratégiu $\mathcal{V} = \{v_0, v_1, \ldots, v_{T-1}\}$, kde $v_i : \mathcal{X}_i \to \mathcal{U}_i$, ktorá maximalizuje očakávanú hodnotu súčtu výnosov z jednotlivých časových období.

Úlohu formálne zapíšeme nasledovne:

$$\max_{\mathcal{V}\in S} \mathbb{E}\left[\sum_{t=0}^{T-1} f_t^0(x_t, u_t, z_t)\right], \tag{2}$$

kde funkcia $f_t^0(x_t, u_t, z_t)$ určuje hodnotu okamžitého výnosu v čase t. Stredná hodnota $\mathbb{E}$ sa uvažuje vzhľadom na spoločné rozdelenie pravdepodobnosti náhodných premenných $z_0, \ldots, z_{T-1}$ a S reprezentuje triedu všetkých prípustných stratégií.

1.2 Rovnica dynamického programovania

Rovnica dynamického programovania (RDP) umožňuje rozdeliť optimalizačný problém počas viacerých periód na postupnosť jednoperiódových rozhodovacích úloh. Základom RDP je Bellmanov princíp optimality, podľa ktorého má optimálne riadenie tú vlastnosť, že bez ohľadu na počiatočný stav a počiatočné rozhodnutie musia byť zostávajúce rozhodnutia optimálne vzhľadom na stav, v ktorom sa nachádzame.

Pri formulácii RDP vychádzame z publikácie [8]. Hodnotovú funkciu pre úlohu z časti 1.1 definujeme ako maximálny očakávaný budúci výnos od času t do konca horizontu pri počiatočnom stave x_t, t. j.

$$V_t(x_t) = \max_{u\in\mathcal{U}(x_t)} \mathbb{E}\left[\sum_{\tau=t}^{T-1} f_\tau^0(x_\tau, u_\tau, z_\tau)\right]. \tag{3}$$

Nutnou a postačujúcou podmienkou na nájdenie optimálnej stratégie $\hat{\mathcal{V}} = \{\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_{T-1}\}$ je nasledujúca RDP

$$\begin{aligned} V_t(x_t) &= \max_{u_t\in\mathcal{U}(x_t)} \left[f_t^0(x_t, u_t, z_t) + \mathbb{E}\left[V_{t+1}(x_{t+1})\right]\right], \\ V_T(x_T) &= 0, \qquad t = 0, \ldots, T-1. \end{aligned} \tag{4}$$

Optimálnu stratégiu $\hat{v}(x_t)$ v čase t nájdeme nasledovne

$$\hat{v}(x_t) = \arg\max_{u_t\in\mathcal{U}(x_t)} \left[f_t^0(x_t, u_t, z_t) + \mathbb{E}\left[V_{t+1}(x_{t+1})\right]\right]. \tag{5}$$

1.3 Problém dimenzionality

Rovnica dynamického programovania (4) umožňuje nájsť optimálnu stratégiu postupným spätným výpočtom hodnotovej funkcie $V_t(x_t)$ pre jednotlivé časové periódy. Tento prístup

však vyžaduje, aby sme hodnotovú funkciu $V_t(x_t)$ vypočítali pre všetky možné stavy systému x_t a pre každý z nich uvážili všetky prípustné riadenia $u_t \in \mathcal{U}(x_t)$, kde $\mathcal{U}(x_t)$ je množina prípustných riadení pre daný stav x_t. Pri výpočte očakávanej hodnoty je potrebné uvažovať všetky možné realizácie náhodnej premennej $z_t \in \mathcal{Z}_t$, ktoré ovplyvňujú nasledujúci stav x_{t+1} a tým aj hodnotu $V_{t+1}(x_{t+1})$.

So zvyšujúcou sa dimenziou uvedených premenných rastú výpočtové nároky. Navyše, rekurentný spôsob výpočtu vyžaduje držať v pamäti hodnotovú funkciu pre ľubovoľný stav, čím rastie aj pamäťová zložitosť. Tento jav je známy ako *problém dimenzionality*.

V praktických úlohách sa s problémom dimenzionality môžeme stretnúť na troch úrovniach:

1. **Stavová premenná** môže byť vektor, kde $x_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,n_1})$. Nech každé $x_{t,i}$ môže nadobudnúť l_1 hodnôt. Potom počet možných stavov x_t je $l_1^{n_1}$, tj. $|\mathcal{X}_t| = l_1^{n_1}$.

2. **Riadiaca premenná** tiež môže byť vektor, kde $u_t = (u_{t,1}, u_{t,2}, \ldots, u_{t,n_2})$. Nech každé $u_{t,i}$ môže nadobudnúť l_2 hodnôt. Potom počet možných riadení u_t je $l_2^{n_2}$, tj. $|\mathcal{U}_t| = l_2^{n_2}$.

3. **Náhodná premenná** môže byť spojitá alebo diskrétna. V tejto práci budeme uvažovať diskrétnu náhodnú premennú, tj. $z_t = (z_{t,1}, z_{t,2}, \ldots, z_{t,n_z})$. Nech každá zložka $z_{t,i}$ môže nadobudnúť maximálne l_3 hodnôt. Potom počet možných stavov náhodnej premennej z_t je $l_3^{n_z}$, tj. $|\mathcal{Z}_t| = l_3^{n_z}$.

Zatiaľ čo niektoré optimalizačné úlohy čelia problému dimenzionality len v jednej z uvedených oblastí, iné narážajú na tento problém vo viacerých, prípadne vo všetkých troch dimenziách súčasne. Ak sú niektoré premenné spojitého charakteru, vyžadujú diskretizáciu: hustota delenia priamo úmerne zvyšuje rozsah stavového priestoru, čo vedie k extrémnej výpočtovej náročnosti.

Výpočet hodnotovej funkcie v rámci jednej časovej periódy vyžaduje vyhodnotenie všetkých prípustných konfigurácií (x_t, u_t, z_t) patriacich do stavovo-rozhodovacieho priestoru definovaného ako

$$\Omega_t = \{(x_t, u_t, z_t) \mid x_t \in \mathcal{X}_t, u_t \in \mathcal{U}(x_t), z_t \in \mathcal{Z}_t\}. \tag{6}$$

Výpočtová zložitosť je v tomto prípade priamo úmerná mohutnosti množiny Ω_t. Pri uvažovanej diskretizácii možno celkovú náročnosť vyjadriť v tvare:

$$\mathcal{O}(|\mathcal{X}_t| \cdot |\mathcal{U}(x_t)| \cdot |\mathcal{Z}_t|) = l_1^{n_1} \cdot l_2^{n_2} \cdot l_3^{n_z}.$$

Ako vidíme, s narastajúcim počtom stavových, riadiacich a stochastických premenných sa výpočtová náročnosť riešenia zvyšuje exponenciálne.

Problém dimenzionality predstavuje hlavné obmedzenie pri spätnom riešení pomocou rovnice dynamického programovania. To podnietilo hľadanie tzv. aproximačných metód, ktoré umožňujú nájsť síce len približné riešenie, no s podstatne menšou výpočtovou náročnosťou.

1.4 Techniky aproximačného dynamického programovania

Jednou z možností, ako sa vysporiadať s problémom dimenzionality, je použitie aproximačných techník. Prirodzene, použitie takýchto techník nemusí viesť k nájdeniu presného riešenia a spravidla neposkytuje záruku optimálnosti. Predstavujú však kompromis medzi presnosťou riešenia a výpočtovou zložitosťou.

Metódy aproximačného dynamického programovania sa vo všeobecnosti členia na dva základné smery. Prvý prístup je založený na aproximácii priamo v priestore stratégií, resp. spätných väzieb [10]. V tejto práci sa však zameriavame na druhý spôsob, ktorý spočíva v aproximácii hodnotovej funkcie. Pri hľadaní stratégie naďalej vychádzame z rekurentného vzťahu 4, v ktorom je však presná hodnotová funkcia nahradená jej aproximáciou $\bar{V}_t(x_t)$. Pôvodná RDP sa tak transformuje do nasledujúceho tvaru

$$V_t(x_t) = \max_{u_t \in \mathcal{U}(x_t)} \left[f_t^0(x_t, u_t, z_t) + \mathbb{E}\left[\bar{V}_{t+1}(x_{t+1})\right] \right], \tag{7}$$

kde $\bar{V}_{t+1}(x_{t+1})$ reprezentuje aproximáciu hodnotovej funkcie $V_{t+1}(x_{t+1})$.

Jednou z možností, ako hľadať vhodnú aproximáciu hodnotovej funkcie, je využitie lineárnej kombinácie bázických funkcií [10]. Základná myšlienka spočíva vo vyjadrení zložitej alebo neznámej funkcie ako kombinácie jednoduchších, vopred definovaných funkcií. Hodnotovú funkciu potom aproximujeme vo forme lineárnej sumy vážených bázických funkcií nasledovne

$$\bar{V}_t(x_t) = \sum_{i=1}^{m} \theta_{t,i} \phi_i(x_t), \tag{8}$$

kde $\phi_i(x_t)$ sú zvolené bázické funkcie a $\theta_{t,i}$ sú koeficienty, ktoré je potrebné odhadnúť. Odhad robíme postupne od konca pre $t = T-1, \ldots, 1$.

Presnosť, ktorá sa dá dosiahnuť pomocou aproximácie hodnotovej funkcie $\bar{V}_t(x_t)$ závisí od viacerých faktorov, najmä od konkrétneho problému a od zvolenej metódy hľadania aproximácie hodnotovej funkcie. Napríklad pre inkrementálny problém batoha, ktorému sa venujeme v časti 2.3, existuje aproximačný algoritmus, ktorý pre ľubovoľné $\varepsilon > 0$ nájde riešenie s hodnotou aspoň $(1-\varepsilon)$-násobku optimálnej hodnoty v kvázi-polynomiálnom čase [4].

1.5 Model Predictive Control

Alternatívnym prístupom k aproximačnému riešeniu rovnice (7) je použitie metódy známej pod názvom Model Predictive Control (MPC). MPC bolo prvýkrát spomenuté v roku 1978 [11] a rýchlo sa stalo populárne, na začiatku hlavne v chemickom priemysle [6]. MPC využívame ako alternatívnu aproximačnú techniku, ktorej sa v práci venujeme.

Ide o metódu riadenia, pri ktorej sa v každom čase t nerieši celý pôvodný problém až do konca horizontu T ako je to pri (34), ale iba optimalizačná úloha na kratšom horizonte dĺžky H.

V čase t teda zohľadňujeme aj predikciu budúcich časov $t, t+1, \ldots, t+H-1$, kde H je vopred zvolená dĺžka predikčného horizontu. Pre aktuálny stav x_t riešime úlohu

$$\max_{u_t,\ldots,u_{t+H-1}} \sum_{\tau=t}^{t+H-1} f_\tau^0(x_\tau, u_\tau, z_\tau) + \mathbb{E}\left[V_{t+H}(x_{t+H})\right], \tag{9}$$

za podmienok daných stavovou rovnicou

$$x_{\tau+1} = f_\tau(x_\tau, u_\tau, z_\tau), \qquad \tau = t, \ldots, t+H-1. \tag{10}$$

Výsledkom optimalizácie je postupnosť aproximačných riadiacich premenných

$$(u_t, u_{t+1}, \ldots, u_{t+H-1}).$$

V skutočnosti sa však aplikuje iba prvá aproximačná riadiaca premenná u_t. Následne sa systém posunie do stavu x_{t+1} a celý výpočet sa opakuje pre čas $t+1$. Tento postup sa označuje ako *rolling horizon*.

Nevýhodou MPC je, že so zvyšujúcou sa hodnotou parametra H môže výpočtová náročnosť rásť exponenciálne [7]. Zároveň nie je jednoznačné, či vyššia hodnota H vedie

k lepším výsledkom. V práci [7] autori problém s nárastom výpočtovej náročnosti riešia pomocou aproximačného algoritmu, ktorý umožňuje efektívne využitie MPC aj pre väčšie hodnoty H.

2 Problém batoha

Úloha o batohu (Knapsack problem) predstavuje klasický optimalizačný problém, ktorého cieľom je vybrať takú kombináciu predmetov, ktorá maximalizuje celkovú hodnotu pri dodržaní kapacitného obmedzenia. Vzhľadom na svoju štruktúru slúži ako základný stavebný blok pre komplexnejšie modely v oblasti logistiky, financií a správy portfólií. Uplatnenie nachádza aj v netradičných oblastiach, napríklad pri výbere optimálnej kombinácie opatrení na elimináciu šikany na pracovisku, kde jednotlivé opatrenia predstavujú položky s rôznymi nákladmi a prínosmi [1].

2.1 Matematická formulácia problému

Uvažujme klasický problém batoha s n predmetmi. Každý predmet $i \in \{1, \ldots, n\}$ je charakterizovaný dvojicou parametrov: váhou $w_i > 0$ a hodnotou $p_i > 0$. Batoh má konštantnú kapacitu c. Cieľom úlohy je vybrať takú podmnožinu predmetov, ktorej celková hodnota je maximálna, pričom súčet váh vybratých predmetov neprekročí kapacitu c.

Predstavený optimalizačný problém môžeme formulovať ako úlohu optimálneho riadenia v diskrétnom čase. Jednotlivé časové etapy $i = 1, \ldots, n$ prislúchajú rozhodovaciemu procesu o vložení i-teho predmetu. Rozhodnutie o (ne)vložení i-teho predmetu reprezentujeme binárnou riadiacou premennou $u_i \in \{0, 1\}$:

$$u_i = \begin{cases} 1, & \text{ak predmet } i \text{ vložíme do batoha,} \\ 0, & \text{inak.} \end{cases}$$

Stav systému x_i definujeme ako aktuálne zaplnenie batoha pred rozhodnutím o vložení i-teho predmetu, $i = 1, \ldots, n$. Dynamiku zmeny stavu popíšeme rovnicou

$$x_{i+1} = x_i + u_i w_i, \quad x_1 = 0, \quad i = 1, \ldots, n$$

kde x_1 predstavuje počiatočný stav prázdneho batoha.

Cieľom je maximalizovať hodnotu batoha, čo vyjadríme ako

$$\max_{u_i \in \{0,1\}} \sum_{i=1}^{n} p_i u_i,$$

pričom riešenie musí spĺňať koncové ohraničenie na stavovú premennú, ktoré predstavuje kapacitný limit batoha:

$$x_{n+1} \leq c.$$

Problém batoha môžeme sumárne reprezentovať v tvare:

$$\begin{aligned} &\max \sum_{i=1}^{n} p_i u_i \\ \text{za podmienok: } &x_{i+1} = x_i + u_i w_i, \quad i = 1, \ldots, n \\ &x_1 = 0 \\ &x_{n+1} \leq c \\ &u_i \in \{0, 1\}, \quad i = 1, \ldots, n \end{aligned} \tag{11}$$

2.2 Výpočtová zložitosť algoritmu

Na rozdiel od stochastickej RDP uvedenej v časti 1.2 budeme v tejto časti uvažovať deterministickú úlohu, v ktorej prechod medzi stavmi nezávisí od náhodnej premennej. Nutnou a postačujúcou podmienkou na nájdenie riešenia úlohy (11) je rekurentná RDP v tvare

$$V_i(x_i) = \max_{u_i \in \{0,1\}} \{p_i u_i + V_{i+1}(x_{i+1})\}, \tag{12}$$

$$V_{n+1}(x_{n+1}) = 0, \quad \text{pre všetky } x_{n+1} \leq c, \tag{13}$$

kde $V_i(x_i)$ predstavuje hodnotovú funkciu, tj. maximálnu hodnotu batoha, ktorý môžeme dosiahnuť od predmetu i až po predmet n, ak je aktuálny stav (zaplnenie) batoha x_i.

Riešenie rekurentnej rovnice (12) vyžaduje pre každý predmet ($i = 1, \ldots, n$) a pre každý možný stav zaplnenia batoha x_i nájsť optimálne riadenie u_i. Celková výpočtová zložitosť tohto postupu je určená počtom operácií v dvoch vnorených cykloch: vonkajší cyklus prechádza postupne všetkých n predmetov, zatiaľ čo vnútorný cyklus iteruje cez všetky celočíselné kapacity od 0 po c. Počet operácií potrebných na nájdenie optimálneho riešenia je teda priamo úmerný súčinu $n \times (c+1)$, čo v notácii veľkého O zapisujeme ako $O(n \cdot (c+1))$. Je dôležité zdôrazniť, že v tomto prípade ide o tzv. *pseudopolynomiálnu zložitosť*. Hoci sa zložitosť javí ako lineárna vzhľadom na c, ak by bola kapacita c extrémne vysoká, algoritmus by napriek fixnému počtu predmetov n nemusel skončiť v „rozumnom“ čase. Túto závislosť si pre lepšiu predstavu ukážeme na praktickom príklade.

Uvažujme fixný počet $n = 400$ predmetov. Kapacitu batoha c postupne uvážime z množiny $\{25 \cdot 10^4, 25 \cdot 10^5, 25 \cdot 10^6\}$. Pre každú zvolenú hodnotu kapacity c odmeriame čas potrebný na nájdenie optimálneho riadenia. Výsledky uvádzame v Tabuľke 1. Na

Počet predmetov	Kapacita batoha	Zložitosť	Čas výpočtu
400	$25 \cdot 10^4$	10^8	1.30 s
400	$25 \cdot 10^5$	10^9	5.38 s
400	$25 \cdot 10^6$	10^{10}	43.41 s

Tabuľka 1: Závislosť výpočtového času od kapacity batoha pri 400 predmetoch.

výsledkoch uvedených v Tabuľke 1 môžeme pozorovať, že pri fixnom počte predmetov rastie výpočtový čas približne lineárne s narastajúcou kapacitou batoha. Tento výsledok je v súlade s teoretickou časovou zložitosťou $O(n \cdot (c + 1))$.

2.3 Inkrementálny problém batoha

Rozšírením problému batoha predstaveného v časti 2.1 sa zaoberá práca [2]. Autori v nej uvádzajú formuláciu tzv. *inkrementálneho problému batoha*, v ktorom sa kapacita batoha v každej časovej perióde náhodne zväčšuje o určitý prírastok (inkrement). Úloha tak získava nový rozmer – časový horizont. Cieľom je maximalizovať súčet hodnôt predmetov v batohu počas zvoleného časového obdobia, pričom daný predmet generuje odmenu v každej perióde, v ktorej sa v batohu nachádza. Súčasne sa predpokladá, že predmet vložený do batoha už nie je možné v nasledujúcich obdobiach vybrať.

Takto formulovaná optimalizačná úloha má viaceré ekonomické interpretácie pri rozhodovaní. Napríklad firma má k dispozícii obmedzený investičný rozpočet a v jednotlivých obdobiach zvažuje realizáciu rôznych projektov s rôznou výškou počiatočnej investície a očakávaným výnosom. Každé prijaté rozhodnutie znižuje disponibilný kapitál pre budúce projekty a zároveň ovplyvňuje budúcu ziskovosť firmy. Ďalším príkladom je riadenie projektov vo verejnom sektore, kde vláda alebo samospráva disponuje obmedzeným rozpočtom a postupne rozhoduje o financovaní infraštruktúrnych projektov. Keďže rozpočet je viazaný na viacročné obdobia, dnešné rozhodnutia obmedzujú rozsah budúcich investícií.

2.4 Formulácia problému

V tejto časti predstavíme matematickú formuláciu inkrementálneho problému batoha. Uvažujme T diskrétnych časových periód, n predmetov a batoh s premenlivou kapacitou

c_t v čase $t \in \{0, \ldots, T-1\}$. Každý predmet $i \in \{1, \ldots, n\}$ je charakterizovaný dvoma parametrami: váhou w_i a hodnotou p_i. Cieľom je maximalizovať celkovú hodnotu batoha počas všetkých T časových periód, pričom okamžitý prínos v t-tej perióde je daný súčtom hodnôt všetkých predmetov, ktoré budú v batohu na konci t-tej periódy.

Podobne ako v časti 2.1, aj inkrementálny problém batoha sformulujeme ako úlohu diskrétneho dynamického programovania. Avšak vzhľadom na prítomnosť náhodnej premennej na rozdiel od deterministickej úlohy (11), táto úloha bude stochastická.

Rozhodnutie v čase t o (ne)vložení predmetu i reprezentujeme binárnou riadiacou premennou $u_{t,i} \in \{0, 1\}$, $t = 0, \ldots, T-1$; $\quad i = 1, \ldots, n$. Pre lepšiu prehľadnosť budeme riadiacu premennú v čase t označovať aj vo vektorovom tvare $\mathbf{u_t} = (u_{t,1}, u_{t,2}, \ldots, u_{t,n})^\top$, kde i-ta zložka $u_{t,i}$ udáva, či predmet i v čase t do batoha vložíme ($u_{t,i} = 1$), alebo nie ($u_{t,i} = 0$).

Na popísanie aktuálneho stavu batoha v čase t definujeme tri stavové premenné. Prvou stavovou premennou c_t reprezentujeme kapacitu batoha v čase t. Na rozdiel od úlohy predstavenej v časti 2.1 sa v inkrementálnom probléme kapacita c_t v každej časovej perióde náhodne zväčšuje. Dynamiku vývoja kapacity modelujeme nasledujúcou stochastickou stavovou rovnicou:

$$\begin{aligned} c_{t+1} &= c_t + z_t, \quad t = 0, \ldots, T-1, \\ c_0 &= \bar{c}_0, \end{aligned} \tag{14}$$

kde z_t je náhodná premenná reprezentujúca prírastok kapacity batoha v čase t. Pre zjednodušenie predpokladáme, že z_t je diskrétna náhodná premenná, čo znamená, že aj samotná kapacita c_t nadobúda hodnoty z diskrétnej množiny stavov. Zo vzťahu (14) je zrejmé, že proces $\{c_t\}_{t=0}^{T-1}$ vykazuje Markovovskú vlastnosť, nakoľko pravdepodobnostné rozdelenie stavu c_{t+1} závisí výlučne od predchádzajúcej hodnoty c_t a náhodného prírastku z_t.

Druhou stavovou premennou je vektor $\mathbf{y_t^{(1)}} = (y_{t,1}^{(1)}, \ldots, y_{t,n}^{(1)})$, ktorým reprezentujeme aktuálne zloženie batoha v čase t. Presnejšie, i-ta zložka vektora $y_{t,i}^{(1)}$ indikuje, či sa i-ty predmet v čase t v batohu nachádza ($y_{t,i}^{(1)} = 1$), alebo nenachádza ($y_{t,i}^{(1)} = 0$). Zmenu stavu predmetu i, $i = 1, \ldots, n$, v batohu popíšeme nasledovne:

$$\begin{aligned} y_{t+1,i}^{(1)} &= y_{t,i}^{(1)} + u_{t,i}, \\ y_{t,i}^{(1)} &\in \{0, 1\}, \quad \text{pre } t = 0, \ldots, T; \quad i = 1, \ldots, n \\ y_{0,i}^{(1)} &= 0, \end{aligned} \tag{15}$$

Podmienky na stavovú premennú $y_{t,i}^{(1)}$ uvedené v (15) zabezpečujú, že predmety môžeme vložiť do batoha len raz, t.j. $y_{t,i}^{(1)} = 0$, ak predmet ešte nebol vložený do batoha do času t, resp. $y_{t,i}^{(1)} = 1$, ak predmet bol vložený do batoha do času t. Z formulácie podmienok v (15) a z obmedzenia na riadenie $u_{t,i} \in \{0, 1\}$ tiež vyplýva, že predmety, ktoré už boli do batoha vložené, ostávajú v batohu počas všetkých nasledujúcich časových periód. Počiatočná podmienka stanovuje, že v batohu v čase $t = 0$ nebol žiaden predmet.

Poslednou stavovou premennou $y_t^{(2)}$ vyjadríme celkovú váhu predmetov, ktoré sa nachádzajú v batohu v čase t. Dynamiku zaplnenia batoha popíšeme stavovou rovnicou v tvare:

$$\begin{aligned} y_{t+1}^{(2)} &= y_t^{(2)} + \sum_{i=1}^{n} w_i u_{t,i}, \quad t = 0, \ldots, T-1, \\ y_{t+1}^{(2)} &\leq c_t, \quad t = 0, \ldots, T-1, \\ y_0^{(2)} &= 0 \end{aligned} \tag{16}$$

Podmienky uvedené v (16) zabezpečujú, že rozhodnutia v čase t vedúce k stavu $y_{t+1}^{(2)}$, rešpektujú kapacitné obmedzenie c_t. Počiatočná podmienka udáva, že batoh je v čase $t = 0$ prázdny.

Inkrementálny problém batoha sumárne zapíšeme nasledovne:

$$\begin{aligned} \max \mathbb{E} & \left[\sum_{t=0}^{T-1} \sum_{i=1}^{n} p_i (y_{t,i}^{(1)} + u_{t,i}) \right] \\ & c_{t+1} = c_t + z_t, \quad t = 0, \ldots, T-1, \\ & y_{t+1,i}^{(1)} = y_{t,i}^{(1)} + u_{t,i}, \quad t = 0, \ldots, T-1, \quad i = 1, \ldots, n \\ & y_{t+1}^{(2)} = y_t^{(2)} + \sum_{i=1}^{n} w_i u_{t,i}, \quad t = 0, \ldots, T-1, \\ & y_{t,i}^{(1)} \in \{0, 1\}, \quad t = 0, \ldots, T, \quad i = 1, \ldots, n \\ & y_{t+1}^{(2)} \leq c_t, \quad t = 0, \ldots, T-1, \\ & u_{t,i} \in \{0, 1\}, \quad t = 0, \ldots, T-1, \quad i = 1, \ldots, n \\ & c_0 = \bar{c}_0. \\ & y_{0,i}^{(1)} = 0, \quad i = 1, \ldots, n, \\ & y_0^{(2)} = 0, \end{aligned} \tag{17}$$

Účelová funkcia reprezentuje celkovú očakávanú hodnotu batoha počas T časových periód.

Hodnotu i-teho predmetu p_i započítame v každej časovej perióde, počas ktorej je predmet i v batohu. V čase $t = 0$ najprv inicializujeme veľkosť batoha a stavové premenné $\mathbf{y_0^{(1)}}$ a $y_0^{(2)}$. Potom urobíme rozhodnutie $\mathbf{u_0}$ o vložení predmetov do batoha. Následne upravíme hodnotu stavových premenných $\mathbf{y_1^{(1)}}$ a $y_1^{(2)}$. Rovnako riadime systém počas časových období $t = 1, \ldots, T-1$. Kapacita c_t sa v čase $t = 0$ inicializuje na pevnú hodnotu, v každej ďalšej perióde sa zväčšuje o náhodný inkrement z_t.

2.5 Problém dimenzionality

Ako sme uviedli v časti 1.2, optimálne riadenie môžeme nájsť riešením RDP. V prípade inkrementálneho problému batoha je stav systému daný trojicou $(\mathbf{y_t^{(1)}}, y_t^{(2)}, c_t)$, preto RDP pre úlohu (17) má nasledujúci tvar:

$$V_t(\mathbf{y_t^{(1)}}, y_t^{(2)}, c_t) = \max_{\mathbf{u_t} \in \mathcal{U}(y_t^{(1)}, y_t^{(2)}, c_t)} \left[f_t^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + \mathbb{E} \left[V_{t+1}(\mathbf{y_{t+1}^{(1)}}, y_{t+1}^{(2)}, c_{t+1}) \right] \right], \quad t = 0, \ldots, T-1 \tag{18}$$

$$V_T(\mathbf{y_T^{(1)}}, y_T^{(2)}, c_T) = \begin{cases} 0, & \text{pre } (\mathbf{y_T^{(1)}}, y_T^{(2)}, c_T) \in \mathcal{X}_T, \\ -\infty, & \text{pre } (\mathbf{y_T^{(1)}}, y_T^{(2)}, c_T) \notin \mathcal{X}_T. \end{cases} \tag{19}$$

kde $f_t^0(\mathbf{y_t^{(1)}} + \mathbf{u_t}) = \sum_{i=1}^{n} p_i(y_{t,i}^{(1)} + u_{t,i})$ a $\mathcal{X}_T$ je množina prípustných stavov systému. Pre zvýšenie prehľadnosti zavedieme označenie $\mathbf{x_t} = (\mathbf{y_t^{(1)}}, y_t^{(2)}, c_t)$.

Nájdenie presného riešenia je vzhľadom na rozmer úlohy výpočtovo náročné. Výpočet hodnotovej funkcie $V_t(\mathbf{x_t})$ vyžaduje určiť jej hodnotu pre každý prípustný stav $\mathbf{x_t}$. Celkový počet stavov je daný súčinom počtu možných hodnôt jednotlivých zložiek stavového vektora: kapacity c_t, konfigurácie predmetov $\mathbf{y_t^{(1)}}$ a aktuálnej hmotnosti $y_t^{(2)}$. Pozrime sa na priestor stavových premenných podrobnejšie a pokúsime sa odhadnúť jeho veľkosť.

Ako sme už spomenuli, pre zjednodušenie modelujeme náhodný prírastok batoha pomocou diskrétnej náhodnej premennej z_t s hodnotami z množiny $\{z_{t,1}, z_{t,2}, \ldots, z_{t,n_z}\}$, kde $z_{t,1} < z_{t,2} < \cdots < z_{t,n_z}$. Stav $z_{t,k}$ nastáva s pravdepodobnosťou $q_{t,k} \geq 0$ pre $k = 1, \ldots, n_z$, pričom platí $\sum_{k=1}^{n_z} q_{t,k} = 1$. Z tohto predpokladu vyplýva, že premenná c_t nadobúda hodnoty z množiny $\{\bar{c}_0 + tz_{t,1}, \ldots, \bar{c}_0 + tz_{t,n_z}\}$. Počet možných stavov kapacity c_t je teda určený rozsahom hodnôt náhodnej premennej z_t a v čase t zodpovedá hodnote $t(z_{t,n_z} - z_{t,1}) + 1$.

Je zrejmé, že stavový priestor premennej $y_t^{(2)}$ (celková hmotnosť v batohu) je rádovo porovnateľný s rozsahom kapacity c_t.

Ostáva určiť rozmer priestoru stavovej premennej $\mathbf{y_t^{(1)}}$. Keďže každý z n predmetov sa v batohu v čase t buď nachádza, alebo nie, počet možných konfigurácií stavovej premennej $\mathbf{y_t^{(1)}}$ je 2^n.

Celkový počet stavov vektora $\mathbf{x_t}$ v každom kroku t je teda približne:$(t(z_{t,n_z} - z_{t,1}) + 1)^2 \cdot 2^n$. V O-notácii to zodpovedá zložitosti $O(t^2 \cdot n_z^2 \cdot 2^n)$.

Výpočtovú zložitosť opäť ilustrujeme na príklade. V Tabuľke 2 uvádzame počet možných stavov stavovej premennej $\mathbf{x_t}$ v čase $t = 8$ pre $n_z = 4$, $c_0 = 20$ a $z_t \in \{1, 2, 3, 4\}$. Pri takomto nastavení parametrov nadobúda hodnota stavovej premennej c_8 hodnoty $c_8 \in \{20 + 8, \ldots, 20 + 4 \cdot 8\}$, t.j. spolu 25 možných stavov.

Počet predmetov	Počet c_t	Počet $\mathbf{y_t^{(1)}}$	Počet možných stavov v čase $t = 8$
10	25	2^{10}	1 048 576
15	25	2^{15}	33 554 432
20	25	2^{20}	1 073 741 824
25	25	2^{25}	34 359 738 368

Tabuľka 2: Exponenciálny rast počtu možných stavov v čase $t = 8$.

Ako vidíme v Tabuľke 2 počet možných stavov pre dané nastavenie parametrov je viac ako 34 miliárd.

Uvedomme si ešte, že výpočtová zložitosť v jednej časovej perióde závisí aj od riadiacej premennej. Riadiaca premenná $u_{t,i} \in \{0, 1\}$, takže počet možných konfigurácií riadiacej premennej $\mathbf{u_t}$ je 2^n. Tento priestor vieme z časti zredukovať, ak si uvedomíme, že z formulácie úlohy (17) vyplýva, že hodnota stavovej premennej $y_{t,i}^{(1)} = 1$ implikuje, že $u_{t,i} = 0$. To znamená, že sú prípustné len tri kombinácie $(y_{t,i}^{(1)}, u_{t,i})$ v každom čase t. Výpočtová zložitosť v každom kroku t je teda približne $O(t^2 \cdot n_z^2 \cdot 3^n)$. Teda napríklad v čase $t = 8$ je pri 25 predmetoch potrebné urobiť $8^2 \cdot 4^2 \cdot 3^{25}$ výpočtov, čo je viac ako 8.7×10^{14}.

2.6 Lineárna aproximácia hodnotovej funkcie

Vzhľadom na veľkosť priestoru stavov v probléme inkrementálneho batoha, ktorý znemožňuje nájsť optimálne riadenie spätným riešením RDP (18) v reálnom výpočtovom čase,

budeme tento problém riešiť s využitím aproximačných techník predstavených v časti 1.4.

Hodnotovú funkciu $V_t(\mathbf{x_t})$ teda nebudeme počítať presne pre každé $\mathbf{x_t}$, ale budeme sa snažiť nájsť jej dostatočne dobrú aproximáciu $\bar{V}_t(\mathbf{x_t})$. Hľadanie aproximácie si rozdelíme do nasledujúcich dvoch krokov:

1. Pre daný čas t vypočítame hodnoty $\tilde{V}_t(\mathbf{x_t})$ pre vhodne zvolené hodnoty $\mathbf{x_t}$.

2. Na základe získaných dvojíc $[\mathbf{x_t}, \tilde{V}_t(\mathbf{x_t})]$ odhadneme vektor koeficientov θ (pozri (8)), čím získame odhad funkcie $\bar{V}_t(\mathbf{x_t})$ v čase t.

Jednotlivé kroky si teraz podrobnejšie opíšeme. V prvom kroku potrebujeme získať dostatočný počet dvojíc $[\mathbf{x_t}, \tilde{V}_t(\mathbf{x_t})]$. Vzhľadom na pomerne veľký priestor stavov má zmysel zaoberať sa nie náhodne vygenerovanými stavmi, ale uvážiť také, do ktorých sa systém môže dostať „s väčšou pravdepodobnosťou". Uvedomme si, že stavová premenná $\mathbf{x_t}$ obsahuje tri zložky $(\mathbf{y_t^{(1)}}, y_t^{(2)}, c_t)$. Kým zložky $y_t^{(2)}$ a c_t sú jednorozmerné premenné, $\mathbf{y_t^{(1)}}$ je n-rozmerná premenná. Ťažisko generovania stavovej premennej teda spočíva v „rozumnej" voľbe zložky $\mathbf{y_t^{(1)}}$. Za účelom generovania takýchto vektorov $\mathbf{y_t^{(1)}}$ sme navrhli nasledujúci heuristický Algoritmus 1.

Výpočet hodnôt $\tilde{V}_t(x_t)$ budeme, analogicky ako pri riešení RDP, vykonávať spätne od konca časového horizontu. V danom čase t vygenerujeme hodnotu c_t z prípustnej množiny stavov a k nej uvážime všetky hodnoty $y_t^{(2)} \in \{0, 1, \ldots, c_t - 1\}$. K zvolenému stavu $y_t^{(2)}$ nájdeme „najrozumnejší" vektor $\mathbf{y_t^{(1)}}$, t.j. nájdeme predmety, ktoré sú pri zvolenom c_t a $y_t^{(2)}$ „najpravdepodobnejšie" umiestnené v batohu.

Na ich nájdenie sme navrhli nasledujúci postup. Definujeme si množinu $\mathcal{G}_t$ ako množinu predmetov, ktoré nie sú v batohu. Množinu predmetov, ktoré ešte môžeme pridať do batoha, nazveme ako $\mathcal{F}_t$. Hodnotou s vyjadríme, akú váhu majú predmety, ktoré sme už pridali do batoha.

Začnime s prázdnym batohom. Postupne budeme do batoha vyberať „vhodné" predmety. Predmety budeme do batoha pridávať postupne: v každom kroku si z množiny $\mathcal{F}_t$ vyberieme predmet i^*, ktorý pridá batoha najväčšiu hodnotu. Inými slovami nájdeme predmet i^*, ktorý získame riešením úlohy

$$i^* = \arg\max_{i \in \mathcal{F}_t} f^0\left(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\right) + \mathbb{E}\left[\bar{V}_t(\mathbf{x_t})\right]. \tag{20}$$

Avšak riešenie (20) vyžaduje poznať hodnotovú funkcie $\bar{V}_t(\mathbf{x_t})$, ktorú v čase výpočtu ešte nemáme odhadnutú. K dispozícii však už máme aproximáciu pre nasledujúcu časovú periódu $t+1$. Výraz $\mathbb{E}\big[\bar{V}_t(\mathbf{x_t})\big]$ si vieme prepísať ako $f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]$, za predpokladu, že nič v čase t nepridáme do batoha. Preto namiesto výrazu (20) používame aproximáciu založenú na hodnote v ďalšom kroku a vyberáme predmet i^*, ktorý dostaneme vyriešením úlohy

$$i^* = \arg\max_{i \in \mathcal{F}_t} f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big) + f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big] \text{ [1]}. \tag{21}$$

Následne aktualizujeme aktuálnu váhu batoha aj množinu dostupných predmetov. Proces pokračuje, kým množina $\mathcal{F}_t$ nie je prázdna.

S pomocou Algoritmu 1 uvedeným postupom získame vektor $\mathbf{y_t^{(1)}}$, ktorý určuje, ktoré položky sa „pri danom c_t" nachádzajú v čase t v batohu. Dôležité je si uvedomiť, že pre dané c_t a $y_t^{(2)}$ Algoritmus 1 generuje len jeden „najpravdepodobnejší" vektor $\mathbf{y_t^{(1)}}$. Inými slovami, výber položiek v batohu je v našej aproximácii deterministicky určený hodnotou $y_t^{(2)}$.

Algoritmus 1 Výber položiek v čase $t-1$

1: **for** $y_t^{(2)} \in \{0, 1, \ldots, c_t - 1\}$ **do**
2: Inicializujeme $s = 0$, $\mathcal{G}_t = \{1, \ldots, n\}$, $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq y_t^{(2)}\}$ a $u_{t-1,i} = 0, \quad \forall i = 1, \ldots, n$
3: **while** $\mathcal{F}_t \neq \emptyset$ **do**
4: $i^* = \arg\max_{i \in \mathcal{F}_t} \Big[2f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big) + \mathbb{E}[\bar{V}_{t+1}(\mathbf{x_{t+1}})]\Big]$ [2]
5: $u_{t-1,i^*} = 1$
6: $s = s + w_{i^*}$
7: $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$
8: $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq y_t^{(2)}\}$
9: **end while**
10: **return** $\mathbf{y_t^{(1)}} = \mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}$
11: **end for**

[1] V ďalšom texte budeme súčet $f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big) + f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big)$ zapisovať v skrátenom tvare $2f^0\Big(\mathbf{y_{t-1}^{(1)}} + \mathbf{u_{t-1}}\Big)$.

V ďalšom kroku získame hodnoty $\tilde{V}_t(\mathbf{x_t})$. Pri ich počítaní však opäť narážame na problém dimenzionality riadiacej premennej, keďže je potrebné riešiť úlohu

$$\tilde{V}_t(\mathbf{x_t}) = \max_{i \in \mathcal{F}_t} \left[f_t^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + \mathbb{E}[\bar{V}_{t+1}(\mathbf{x_{t+1}})] \right]. \tag{22}$$

Na vyriešenie tohto problému navrhujeme aproximačný postup, v ktorom budeme hodnotovú funkciu $\tilde{V}_t(\mathbf{x_t})$ počítať spätne od konca, pričom $\tilde{V}_T(\mathbf{x_T}) = 0$.

Výpočet optimálnej hodnoty výrazu (22) by bol výpočtovo náročný, keďže by bolo potrebné preskúmať všetky možné kombinácie predmetov, ktoré je možné do batoha pridať. Preto hodnotu $\bar{V}_t(\mathbf{x_t})$ počítame pomocou aproximačného postupu uvedeného v Algoritme 2.

Algoritmus 2 postupne konštruuje výber predmetov, ktoré môžeme do batoha pridať v časovej perióde t. Na začiatku vychádzame zo stavu $\mathbf{x_t}$. Podobne ako v Algoritme 1, aj v Algoritme 2 postupne pridávame jednotlivé predmety. Avšak na rozdiel od Algoritmu 1, kde sme sa snažili batoh zaplniť až po danú váhu $y_t^{(2)}$, v tomto prípade nemusí byť optimálne pridať do batoha všetky predmety, ktoré sa ešte zmestia. Pridanie ďalšieho predmetu síce môže zvýšiť okamžitú odmenu $f^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right)$, ale zároveň môže znížiť hodnotu $\mathbb{E}[\bar{V}_{t+1}(\mathbf{x_{t+1}})]$. Preto môže nastať situácia, že je výhodnejšie nepridať už žiadny ďalší predmet, aj keby sa do batoha ešte zmestil.

Algoritmus 2 funguje nasledovne. V prvom kroku inicializujeme hodnotu $V^* = f_t^0\left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + \mathbb{E}\left[\bar{V}_{t+1}(\mathbf{x_{t+1}}) \right]$, pričom $u_{t,i} = 0 \ \forall i = 1, \ldots, n$ a $\mathbf{y_t^{(1)}}$ získame z Algoritmu 1. Zložky x_{t+1} dopočítame podľa vzťahov (17). Následne uvažujeme o pridaní ďalšieho predmetu z množiny $\mathcal{F}_t$. Z tejto množiny vyberieme predmet i^*, ktorý maximalizuje hodnotu

$$\max_{i \in \mathcal{F}_t} \left(f_t^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + \mathbb{E}\left[\bar{V}_{t+1}(\mathbf{x_{t+1}}) \right] \right). \tag{23}$$

Ak je hodnota (23) po pridaní predmetu i^* väčšia alebo rovná aktuálnej hodnote V^*, predmet pridáme do batoha a aktualizujeme príslušné premenné. V opačnom prípade Algoritmus 2 skončí a hodnota V^* sa uloží ako $\tilde{V}_t(\mathbf{x_t})$.

[2]Na pravej strane nie je nikde index i, avšak v skutočnosti tam je, ako i-ta hodnota vektora $\mathbf{y_{t-1}^{(1)}}$ a $\mathbf{u_{t-1}^{(1)}}$.

Algoritmus 2 Výpočet $\bar{V}_t(\mathbf{x_t})$

1: **for** $c_t \in \{\bar{c}_0 + tz_{t,1}, \ldots, \bar{c}_0 + tz_{t,n_z}\}$, $y_t^{(2)} \in \{0, 1, \ldots, c_t - 1\}$ **do**

2: Inicializujeme $s = y_t^{(2)}$, $u_{t,i} = 0, \quad \forall i = 1, \ldots, n$, $\mathbf{y_t^{(1)}}$ z Alg. 1,
$\mathcal{G}_t = \{i \in \{1, \ldots, n\} \mid y_{t,i}^{(1)} = 0\}$, $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq c_t\}$ a
$V^* = f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]$

3: **while** $\mathcal{F}_t \neq \emptyset$ **do**

4: $i^* = \begin{cases} \arg\max\limits_{i \in \mathcal{F}_t} \Big(f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]\Big), & \text{ak } \max\limits_{i \in \mathcal{F}_t} \Big(f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]\Big) \geq V^*, \\ \emptyset, & \text{inak} \end{cases}$

5: **if** $i^* = \emptyset$ **then**

6: **break**

7: **end if**

8: $u_{t,i^*} = 1$, $s = s + w_{i^*}$, $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$ a $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq c_t\}$

9: $V^* = f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]$

10: **end while**

11: $\tilde{V}_t(\mathbf{x_t}) = V^*$

12: **return** $\tilde{V}_t(\mathbf{x_t})$

13: **end for**

Pomocou navrhnutých Algoritmov 1 a 2 sa výpočtová zložitosť v každom kroku t znižuje z $O(t^2 \cdot n_z^2 \cdot 3^n)$ na $O(t^2 \cdot n_z^2 \cdot (n^2 + n))$. Počet možných stavov c_t a $y_t^{(2)}$ zostáva nezmenený a určuje faktor $t^2 \cdot n_z^2$.

Zvyšná časť zložitosti vyplýva z výberu predmetov. Pri výbere predmetu i^* prechádzajú Algoritmy 1 a 2 množinu $\mathcal{F}_t$, ktorá môže obsahovať najviac n prvkov. Po výbere jedného predmetu sa jeho veľkosť zníži najviac na $n - 1$, a tento postup sa opakuje. Celkový počet operácií je tak zhora ohraničený súčtom $\sum_{i=1}^{n} i = \frac{n^2+n}{2}$. Z toho vyplýva, že celková výpočtová zložitosť Algoritmov 1 a 2 je $O(t^2 \cdot n_z^2 \cdot (n^2 + n))$.

Po zostavení dvojíc $[\mathbf{x_t}, \tilde{V}_t(\mathbf{x_t})]$ môžeme pristúpiť k aproximácii hodnotovej funkcie $\bar{V}_t(\mathbf{x_t})$. Odhadneme ju v tvare (8). Pre inkrementálny problém batoha sme navrhli nasledujúcich šesť vysvetľujúcich bázických funkcií:

1. Konštantná funkcia $\phi_{t,1} = 1$, ktorá umožňuje zachytiť „základnú" úroveň hodnotovej funkcie.

2. Funkcia $\phi_{t,2} = y_t^{(2)}$, ktorá vyjadruje aktuálnu celkovú váhu predmetov v batohu.

3. Funkcia $\phi_{t,3} = c_t - y_t^{(2)}$, ktorá predstavuje zostávajúcu kapacitu batoha, ktorá určuje potenciál pre pridanie ďalších predmetov.

4. Funkcia $\phi_{t,4} = \left(c_t - y_t^{(2)}\right)^2$, ktorá zachytáva nelineárny vplyv zostávajúcej kapacity na hodnotu riešenia.

5. Funkcia $\phi_{t,5} = \max_{i \in \mathcal{P}_t} \left(\frac{p_i}{w_i}\right)$, ktorá reprezentuje potenciálnu hodnotu predmetov, ktoré sa do batoha aktuálne nezmestia, ale zmestili by sa pri miernom zvýšení kapacity. Množinu $\mathcal{P}_t$ definujeme ako

$$\mathcal{P}_t = \left\{ i \in \mathcal{G}_t \;\middle|\; c_t - y_t^{(2)} + \kappa^{(1)} \leq w_i \leq c_t - y_t^{(2)} + \kappa^{(2)} \right\},$$

   kde $\kappa^{(1)}$ a $\kappa^{(2)}$ sú vopred zvolené parametre.

6. Funkcia $\phi_{t,6} = \sum_{i=1}^{n} p_i y_{t,i}^{(1)}$, ktorá vyjadruje hodnotu predmetov v batohu, čím zachytáva dynamiku vývoja systému.

V ďalšom overíme štatistickú významnosť koeficientov $\theta_{t,i}$. Zvolili sme parametre $\kappa^{(1)} = 1$ a $\kappa^{(2)} = 3$. Testovanie vykonávame na prvej sade vygenerovaných predmetov uvedenej v časti 3.2, pričom pri iných realizáciách by sa výsledky mohli líšiť. Na tento účel sme pre čas $t = T - 1$ vykonali t-test významnosti parametrov v lineárnom regresnom modeli. Pre skoršie časy sme t-test nevykonávali, keďže odhady $\theta_{t,i}$ sú získané rekurzívne a môžu obsahovať chyby prenesené z odhadov v neskorších časoch.

Výsledky t-testu pre čas $t = T - 1$ sú uvedené v Tabuľke 3. Z Tabuľky 3 je zrejmé, že štatisticky nevýznamné sa ukázali byť koeficienty $\theta_{T-1,1}$ a $\theta_{T-1,2}$, ktoré zodpovedajú konštantnej bázickej funkcii a premennej x_t (aktuálne zaplnenie batoha). Naopak, ostatné koeficienty sú štatisticky významné na hladine významnosti 5 %.

Pri aproximácii hodnotovej funkcie $\bar{V}_t(x_t)$ sme sa rozhodli nepoužívať bázické funkcie, ktorých koeficienty sa ukázali ako štatisticky nevýznamné. Výnimku predstavuje konštantná bázická funkcia, ktorú sme v modeli ponechali. Dôvodom je skutočnosť, že hoci sa v poslednom čase $t = T - 1$ neukázala ako významná, v skorších časoch by mohla zachytávať očakávanú budúcu hodnotu, ktorá nie je plne vysvetlená ostatnými bázickými funkciami.

Premenná	Hodnota	p-value
$\theta_{T-1,1}$	0.275845	0.704
$\theta_{T-1,2}$	0.101622	0.164
$\theta_{T-1,3}$	3.714950	$< 10^{-6}$
$\theta_{T-1,4}$	-0.001780	$< 10^{-6}$
$\theta_{T-1,5}$	0.305709	0.0168
$\theta_{T-1,6}$	0.948390	$< 10^{-6}$

Tabuľka 3: Štatistická významnosť parametrov pre $t = T - 1$

2.7 Algoritmus hľadania stratégie

Po vypočítaní parametrov $\theta_{t,i}$ pre všetky $t = 0, 1, \ldots, T-1$ a $i = 1, 2, \ldots, n$ môžeme pristúpiť k samotnému hľadaniu stratégie $\mathbf{u_t}$. V každej časovej perióde t chceme pre aktuálny stav $\mathbf{x_t}$ vybrať stratégiu $\mathbf{u_t}$. To znamená vyriešiť úlohu

$$\arg \max_{\mathbf{u_t} \in U(\mathbf{x_t})} \left(f_t^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + \mathbb{E}\left[\bar{V}_{t+1}(\mathbf{x_{t+1}}) \right] \right). \tag{24}$$

Na výpočet $\mathbf{u_t}$ používame Algoritmus 3. Jeho princíp je podobný ako v prípade Algoritmu 2. Rozdiel je v tom, že Algoritmus 2 používame pri výpočte aproximovanej hodnotovej funkcie $\tilde{V}_t(\mathbf{x_t})$ a aplikujeme ho postupne na všetky uvažované stavy. Naproti tomu, Algoritmus 3 používame už len pre aktuálny stav $\mathbf{x_t}$, v ktorom sa práve nachádzame. Algoritmus 3 nájde stratégiu $\mathbf{u_t}$ pre daný stav $\mathbf{x_t}$.

Algoritmus 3 Výpočet $\mathbf{u_t}$

1: Inicializujeme $s = y_t^{(2)}$, $u_{t,i} = 0, \quad \forall i = 1, \ldots, n, \quad \mathcal{G}_t = \{i \in \{1, \ldots, n\} \mid y_{t,i}^{(1)} = 0\}$ a
$\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq c_t\}$, $V^* = f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]$

2: **while** $\mathcal{F}_t \neq \emptyset$ **do**

3: $i^* = \begin{cases} \arg\max\limits_{i \in \mathcal{F}_t} \Big(f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]\Big), & \text{ak } \max\limits_{i \in \mathcal{F}_t} \Big(f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]\Big) \geq V^*, \\ \emptyset, & \text{inak} \end{cases}$

4: **if** $i^* = \emptyset$ **then**

5: **return** $\mathbf{u_t}$

6: **end if**

7: $u_{t,i^*} = 1$, $s = s + w_{i^*}$, $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$ a $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq c_t\}$

8: $V^* = f^0\Big(\mathbf{y_t^{(1)}} + \mathbf{u_t}\Big) + \mathbb{E}\big[\bar{V}_{t+1}(\mathbf{x_{t+1}})\big]$

9: **end while**

10: **return** $\mathbf{u_t}$

2.8 Riešenie problému inkrementálneho batoha pomocou MPC

Problém inkrementálneho batoha budeme riešiť aj inou aproximačnou technikou - MPC, predstavenou v časti 1.5. Použijeme aproximáciu hodnotovej funkcie $\bar{V}_t(\mathbf{x_t})$ postupom uvedeným v časti 2.6. Pre MPC algoritmus zvolíme tri dopredné časové kroky. Časový horizont $H = 3$ vytvára dostatočný priestor na strategické vyčkávanie (tzv. lookahead bias), a zároveň sa vyhneme potrebe zaoberať sa zložitým pravdepodobnostným stromom pre vzdialenú budúcnosť.

Počítame postupne od času $t = 0$. V prvom kroku vygenerujeme hodnoty c_{t+1} a c_{t+2} podľa príslušného pravdepodobnostného rozdelenia (14). Následne miesto pôvodnej úlohy

$$\arg \max_{\mathbf{u_t} \in \mathcal{U}(\mathbf{x_t})} f_t^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + E(\bar{V}_{t+1}(\mathbf{x_{t+1}})). \tag{25}$$

budeme riešiť úlohu

$$\arg \max_{\mathbf{u_t} \in \mathcal{U}(\mathbf{x_t})} f_t^0 \left(\mathbf{y_t^{(1)}} + \mathbf{u_t} \right) + f_t^0 \left(\mathbf{y_{t+1}^{(1)}} + \mathbf{u_{t+1}} \right) + f_t^0 \left(\mathbf{y_{t+2}^{(1)}} + \mathbf{u_{t+2}} \right) + \mathbb{E}(\bar{V}_{t+3}(\mathbf{x_{t+3}})). \tag{26}$$

Tento prístup opäť naráža na problém dimenzionality. Ak by sme chceli úlohu (26) riešiť presne, museli by sme prehľadávať všetky možné kombinácie riadení $\mathbf{u_t}$, $\mathbf{u_{t+1}}$ a $\mathbf{u_{t+2}}$.

Na riešenie úlohy (26) sme navrhli nasledujúci heuristický algoritmus. Vychádza z nasledujúcej myšlienky: Ak sa kapacita batoha v čase zväčšuje, potom by optimálne riešenie nemalo odkladať vloženie predmetu na neskorší čas, pokiaľ bolo možné vložiť tento predmet už skôr. Uvažujme napríklad situáciu, keď $c_t = 20$, $c_{t+1} = 23$, $y_{t+1}^{(2)} = 15$ a $y_{t+2}^{(2)} = 20$. Z rozdielu $y_{t+2}^{(2)} - y_{t+1}^{(2)} = 5$ vidíme, že v čase $t+1$ bol do batoha pridaný jeden alebo viac predmetov s celkovou váhou 5. Tieto predmety bolo možné vložiť už v čase t, tým pádom ich odloženie do času $t+1$ nie je výhodné, pretože by sa ich hodnota započítavala o jednu periódu kratšie. Takéto riešenia preto v heuristike nepovažujeme za „rozumné".

Návrh heuristického algoritmu vyžaduje definovať nové pomocné premenné m_t^1, m_{t+1}^1 a m_{t+2}^1, ktoré budú určovať maximálnu úroveň zaplnenia batoha, ktorú považujeme za „rozumnú" v čase t, $t+1$ a $t+2$. Za „rozumné" hodnoty týchto premenných považujeme také, ktoré zodpovedajú prirodzenému postupnému zapĺňaniu batoha v čase, pričom nevznikajú situácie, ktoré by bolo možné dosiahnuť už v skorších časových periódach. Presnejšie, uvažujme nasledujúce hodnoty pomocných premenných: $m_t^1 \in \{y_t^{(2)}, \ldots, c_t\}$, $m_{t+1}^1 \in \{c_t + 1, \ldots, c_{t+1}\}$ a $m_{t+2}^1 \in \{c_{t+1} + 1, \ldots, c_{t+2}\}$.

Ďalej sa pozrieme, aké kombinácie $y_{t+1}^{(2)}, y_{t+2}^{(2)}, y_{t+3}^{(2)}$ môžu v priebehu riešenia nastať. Rozlíšime štyri nasledujúce situácie:

1. všetky predmety, ktoré sa rozhodneme pridať, vložíme do batoha už v čase t a v nasledujúcich dvoch časových periódach sa zloženie batoha nemení, tj. $y_{t+1}^{(2)} = y_{t+2}^{(2)} = y_{t+3}^{(2)}$,

2. zloženie batoha sa v posledných dvoch uvažovaných periódach už nemení, tj. $y_{t+1}^{(2)} < y_{t+2}^{(2)} = y_{t+3}^{(2)}$,

3. zloženie batoha sa mení vo všetkých troch uvažovaných periódach, tj. $y_{t+1}^{(2)} < y_{t+2}^{(2)} < y_{t+3}^{(2)}$,

4. zloženie batoha sa nemení v prvých dvoch uvažovaných periódach, tj. $y_{t+1}^{(2)} = y_{t+2}^{(2)} < y_{t+3}^{(2)}$.

V Algoritme 4 navrhneme riešenie pre prvý prípad. Ak platí $y_{t+1}^{(2)} = y_{t+2}^{(2)} = y_{t+3}^{(2)}$, tj. zloženie batoha sa v periódach t, $t+1$ a $t+2$ nemení, tak sa nemení ani okamžitá odmena z týchto troch periód, tj. jej hodnota bude $3f_t^0\left(\mathbf{y_t^{(1)}} + \mathbf{u_t^{(1)}}\right)$.

Algoritmus 4 pracuje nasledovne: Pre terajší stav $y_t^{(2)}$ a cieľovú úroveň zaplnenia m_t^1 inicializujeme aktuálnu váhu batoha $s = y_t^{(2)}$. Množina $\mathcal{G}_t$ obsahuje všetky predmety, ktoré sa ešte v batohu nenachádzajú, a množina $\mathcal{F}_t$ obsahuje tie z nich, ktoré je možné pridať bez prekročenia hranice m_t^1.

V každom kroku Algoritmus 4 vyberie predmet i^* z množiny $\mathcal{F}_t$, ktorý dostaneme vyriešením úlohy

$$i^* = \arg\max_{i \in \mathcal{F}_t} 3f_t^0\left(\mathbf{y_t^{(1)}} + \mathbf{u_t^{(1)}}\right) + \mathbb{E}\left[\bar{V}_{t+3}(\mathbf{x_{t+3}})\right]. \tag{27}$$

Po pridaní predmetu sa aktualizuje váha s, zmenší sa množina dostupných predmetov a nanovo sa vytvorí množina $\mathcal{F}_t$. Tento proces sa opakuje, kým je možné pridať ďalší predmet bez prekročenia kapacitného obmedzenia.

Výsledkom Algoritmu je konkrétna stratégia $\mathbf{u_t}$, ktorá zodpovedá uvažovanému typu vývoja systému.

Algoritmus 4 Výpočet $\mathbf{u_t}$

for $y_t^{(2)} \in \{0, 1, \ldots, c_t - 1\}$ a $m_t^1 \in \{y_t^{(2)}, y_t^{(2)} + 1, \ldots, c_t\}$ **do**
  Inicializujeme $s = y_t^{(2)}$, $u_{t,i} = 0 \; \forall i = 1, \ldots, n$, $\mathcal{G}_t = \{i \in \{1, \ldots, n\} \mid y_{t,i}^{(1)} = 0\}$ a $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m_t^1\}$
  while $\mathcal{F}_t \neq \emptyset$ **do**
    $i^* = \arg\max_{i \in \mathcal{F}_t} \left[3f^0(\mathbf{y_t} + \mathbf{u_t}) + \mathbb{E}[\bar{V}_{t+3}(x_{t+3})]\right]$
    $u_{t,i^*} = 1$
    $s = s + w_{i^*}$
    $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$
    $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m_t^1\}$
  end while
  return $\mathbf{u_t}$
end for

Analogickým spôsobom Algoritmy 5, 6 a 7 rozširujú riešenie získané v Algoritme 4 na ďalšie časové periódy. Schematický popis Algoritmov 5, 6 a 7 je obsahom Prílohy A, na tomto mieste stručne vysvetlíme ich princíp. Na rozdiel od Algoritmu 4, ktorý konštruuje riadiacu premennú $\mathbf{u_t}$ pre dané m_t^1, tieto algoritmy už nevychádzajú zo stavu $\mathbf{y_t^{(1)}}$, ale nadväzujú na riešenie získané v Algoritme 4.

Konkrétne, Algoritmus 5 pre zvolenú dvojicu (m_t^1, m_{t+1}^1) vychádza zo stavu $\mathbf{y_{t+1}^{(1)}}$, ktorý je výsledkom Algoritmu 4 a rozširuje ho o predmety, ktoré sa pridávajú až v čase $t + 1$.

Algoritmus 6 pre zvolenú dvojicu (m_t^1, m_{t+2}^1) taktiež vychádza zo stavu $\mathbf{y_{t+1}^{(1)}}$, ktorý je výsledkom Algoritmu 4 a uvažuje situáciu, keď sa nové predmety pridávajú až v čase $t + 2$. Napokon Algoritmus 7 reprezentuje najvšeobecnejší prípad, v ktorom sa riešenie konštruuje pre všetky tri časové kroky t, $t + 1$ a $t + 2$, pričom vychádza zo stavu $\mathbf{y_{t+2}^{(1)}}$, získaného v Algoritme 5.

Vo všetkých prípadoch algoritmy postupne konštruujú riadiace premenné $\mathbf{u_t}$, $\mathbf{u_{t+1}}$ a $\mathbf{u_{t+2}}$ tak, že v každom kroku berú do úvahy už existujúce riešenie a pridávajú ďalšie predmety do batoha v danom čase.

Po vyhodnotení všetkých „rozumných" kombinácií $(m_t^1, m_{t+1}^1, m_{t+2}^1)$ vyberieme riešenie, ktoré maximalizuje hodnotu výrazu (26). Z takto získanej trojice stratégií vykonáme iba stratégiu $\mathbf{u_t}$ a následne celý postup opakujeme pre ďalší časový krok.

3 Numerická analýza problému

V tejto kapitole predstavíme výsledky numerickej analýzy inkrementálneho problému batoha. Porovnáme efektívnosť troch stratégií. Stratégiu získanú pomocou Algoritmu predstaveného v časti 2.6 budeme označovať ako *Regresná stratégia*. Stratégiu získanú pomocou MPC algoritmu z časti 2.8 nazveme *MPC stratégia*. Ako benchmark použijeme tzv. *Naivnú stratégiu*. Pri tejto stratégii sa v každom čase t do batoha vkladajú najhodnotnejšie dostupné predmety, ktorých súčet váh nepresiahne kapacitu batoha v čase t.

Každý numerický experiment pozostáva z nasledujúcich krokov:

1. Najprv vygenerujeme vstupné parametre úlohy: počet predmetov, ich váhy a hodnoty a dĺžku časového horizontu T. Generovanie prebieha s pevne nastaveným generátorom náhodných čísel (t.j. používame set seed), čím zabezpečujeme reprodukovateľnosť výsledkov.

2. Následne pristúpime k odhadu hodnotovej funkcie, t.j. určeniu parametrov θ, ktoré slúžia na jej aproximáciu. Tento odhad realizujeme iba raz pre danú množinu predmetov.

3. Vygenerujeme trajektórie kapacity batoha c_t, ktoré reprezentujú možný vývoj kapacity v čase.

4. Na základe vygenerovaných trajektórií určujeme aproximačné riadenie pre jednotlivé algoritmy. V prípade regresnej stratégie pritom využívame už odhadnuté parametre θ.

5. V poslednom kroku vyhodnocujeme kvalitu získaných riešení, najmä z hľadiska dosiahnutého zisku, a porovnávame výkonnosť jednotlivých prístupov.

3.1 Ilustračný príklad

Správnosť a funkčnosť navrhnutých heuristických postupov najprv overíme na ilustračnom príklade. Pre jednoduchosť uvažujme deväť predmetov a osem časových periód, t.j. $T = 7$. Parametre predmetov sme sa snažili zvoliť tak, aby bolo možné jednoznačne overiť optimálnosť získaného riešenia. Váhy a hodnoty predmetov ako aj kapacitu batoha v jednotlivých časových periódach uvádzame v Tabuľke 4.

Kapacita batoha c_t v čase t bude v tomto ilustračnom príklade vopred deterministicky určená, pričom platí $c_t < c_{t+1}$.

i	1	2	3	4	5	6	7	8	9
w_i	1	1	1	3	3	4	5	20	21
p_i	1	1	2	6	10	10	14	60	84

Tabuľka 4: Váhy a hodnoty predmetov v ilustračnom príklade.

Všetky uvedené algoritmy sme implementovali v programovacom jazyku Python. Pri implementácii sme využili knižnice `NumPy` na numerické výpočty a `Matplotlib` na vizualizáciu výsledkov. Pre urýchlenie výpočtov sme použili knižnicu `Numba`.

V Tabuľke 5 uvádzame nájdené stratégie pre všetky tri typy predstavených algoritmov. Symbolom $\emptyset$ označíme stav, keď algoritmus nepridal žiadny predmet do batoha v čase t.

t	0	1	2	3	4	5	6	Celkový zisk
c_t	20	21	25	28	30	33	37	-
Regresná str.	$\emptyset$	$\{9\}$	$\{3,5\}$	$\{4\}$	$\emptyset$	$\{1,6\}$	$\emptyset$	610
MPC str.	$\emptyset$	$\{9\}$	$\{3,5\}$	$\{4\}$	$\emptyset$	$\{7\}$	$\{6\}$	626
Naivná str.	$\{8\}$	$\{3\}$	$\{1,5\}$	$\{4\}$	$\{2\}$	$\{6\}$	$\emptyset$	535

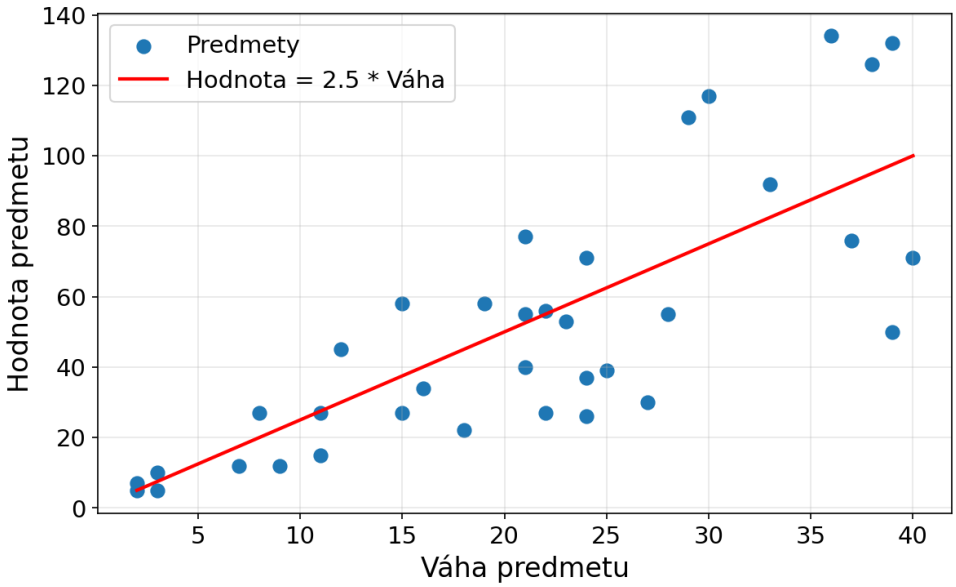
Tabuľka 5: Ilustračný príklad: Porovnanie Regresnej, MPC a Naivnej stratégie.

Na výsledkoch v Tabuľke 5 vidíme, že najväčší celkový zisk sa dosiahol pri MPC stratégii vkladania predmetov. Dosiahnutý zisk mal hodnotu 626 p.j.. Rozdiel v Regresnej a MPC stratégii je v čase $t = 5$ a $t = 6$, kde tieto algoritmy vložili iné predmety do batoha. Táto odchýlka viedla k poklesu celkového zisku na hodnotu 610 p.j. v prípade Regresnej stratégie. Naivná stratégia viedla k celkovému zisku iba 535 p.j., čo je výrazne menej než v prípade oboch predchádzajúcich metód.

Ľahko overíme, že v tomto ilustračnom príklade MPC stratégia je identická s presným optimálnym riadením. Z jedného experimentu síce nemôžeme vysloviť jednoznačný záver ohľadom schopnosti predstavených algoritmov nájsť optimálne riešenie, ale poskytuje nám to aspoň prvotnú predstavu o funkčnosti predstavených algoritmov a presnosti nájdeného riešenia. Testovaniu sa budeme podrobnejšie venovať v ďalšej časti.

3.2 Porovnanie algoritmov

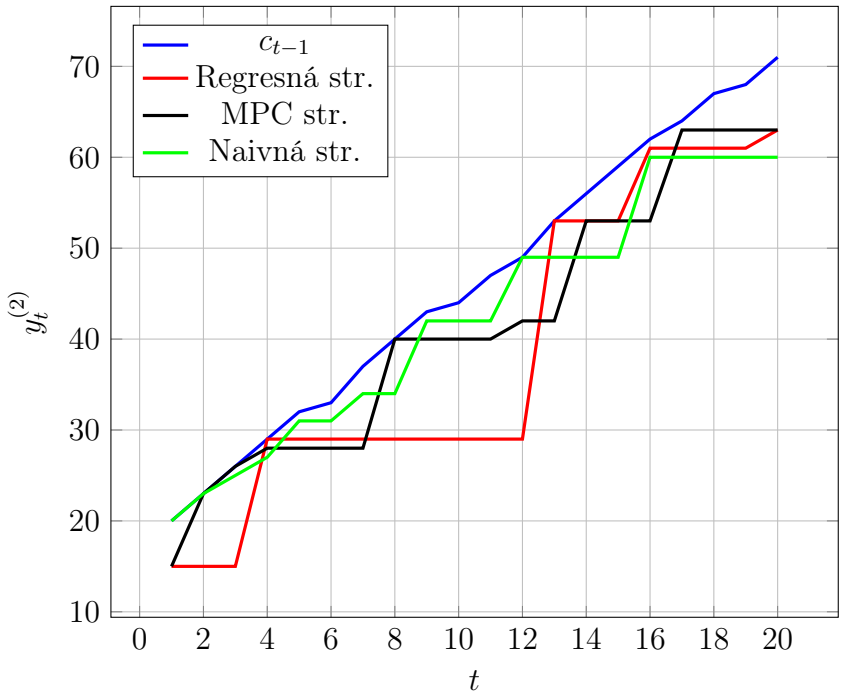
V tejto časti predstavíme výsledky rozsiahlejšieho numerického testovania predstavených algoritmov na riešenie úlohy inkrementálneho batoha. V nasledujúcich experimentoch počet predmetov generujeme z diskrétneho intervalu $[30, 50]$. Následne vygenerujeme 100 rôznych dvojíc parametrov predmetov, pričom váhy predmetov generujeme z diskrétneho intervalu $[1, 40]$, a hodnotu predmetu stanovíme na základe vygenerovaného pomeru $\frac{p_i}{w_i}$ z intervalu $[1, 4]$. Príklad jednej takto vygenerovanej vzorky dát pre dvojice $[w_i, p_i]$ pre 36 predmetov sme graficky znázornili na Obr. 1.



Obr. 1: Vzťah medzi váhou a hodnotou predmetov v jednej zo simulácií.

Počiatočnú veľkosť batoha sme nastavili na hodnotu $c_0 = 20$. Tejto hodnote sme prispôsobili nastavenie náhodnej premennej z_t: predpokladali sme, že náhodná premenná nadobúda hodnoty $\{1, 2, 3, 4\}$ s pravdepodobnosťami $\mathbf{q_t} = (0.2, 0.2, 0.5, 0.1)$. Kvôli zjednodušeniu predpokladáme, že z_t nadobúda uvedené hodnoty pre každé $t = 0, 1, \ldots, T-1$. Trajektóriu kapacity $\{c_t\}_{t=1}^{T-1}$ sme následne generovali v súlade so stavovou rovnicou (14).

Počiatočnú veľkosť batoha sme nastavili na hodnotu $c_0 = 20$. Tejto hodnote sme prispôsobili nastavenie náhodnej premennej z_t: predpokladáme, že náhodná premenná nadobúda hodnoty {1,2,3,4} s pravdepodobnosťami $\mathbf{q_t} = (0.2, 0.2, 0.5, 0.1)$. Kvôli zjednodušeniu predpokladáme, že z_t nadobúda uvedené hodnoty pre každé $t = 0, 1, \ldots, T-1$. Dĺžku časového horizontu sme nastavili na $T = 20$.



Obr. 2: Vývoj hodnôt c_{t-1} a $y_t^{(2)}$ v čase v jednej zo simulácii.

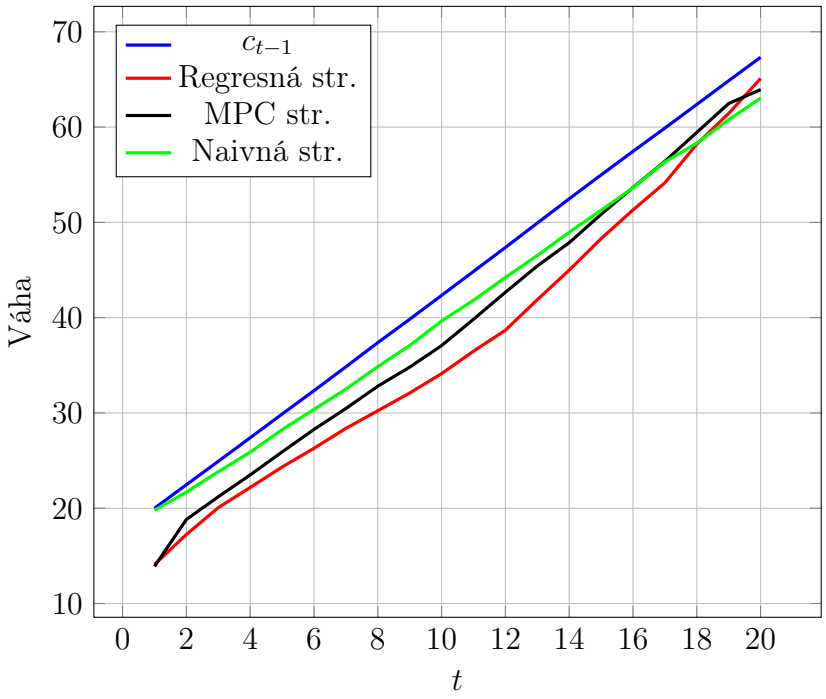
Pre každú sadu vygenerovaných predmetov sme vygenerovali päť rôznych trajektórií vývoja kapacity c_t, takže sme celkovo získali 500 rôznych kombinácií predmetov a vývojov kapacity.

V jednotlivých simuláciách sme postupne aplikovali tri aproximačné riadenia. V každej simulácii sme sledovali priebežné zaplnenie batoha a dosiahnutý zisk.

Na ilustráciu fungovania jednotlivých stratégií sme na Obr. 2 vykreslili jeden konkrétny priebeh vývoja kapacity a zaplnenia batoha. V tomto ilustračnom príklade uvažujeme predmety, ktorých charakteristiky sme znázornili na Obr. 1.

Na Obr. 2 môžeme pozorovať, že pri Regresnej stratégii medzi časmi $t = 2$ a $t = 8$ nie sú do batoha pridané žiadne predmety. Kvôli takémuto rozhodnutiu síce dočasne zaostáva za MPC stratégiou v priebežnej hodnote dosiahnutého zisku, ale súčasne si vytvára priestor na neskoršie vloženie ťažších, ale hodnotnejších predmetov. Naopak, Naivná stratégia na začiatku sledovaného obdobia rýchlo zapĺňa batoh, ale v neskorších časoch už nemá k dispozícii vhodné predmety na vloženie, keďže zostávajúce predmety majú príliš veľkú váhu.

Na základe výsledkov realizovaných simulácií sme stanovili priemernú mieru zaplnenia kapacity batoha, ktorej grafický priebeh je znázornený na Obr. 3. Z výsledkov vyplýva, že

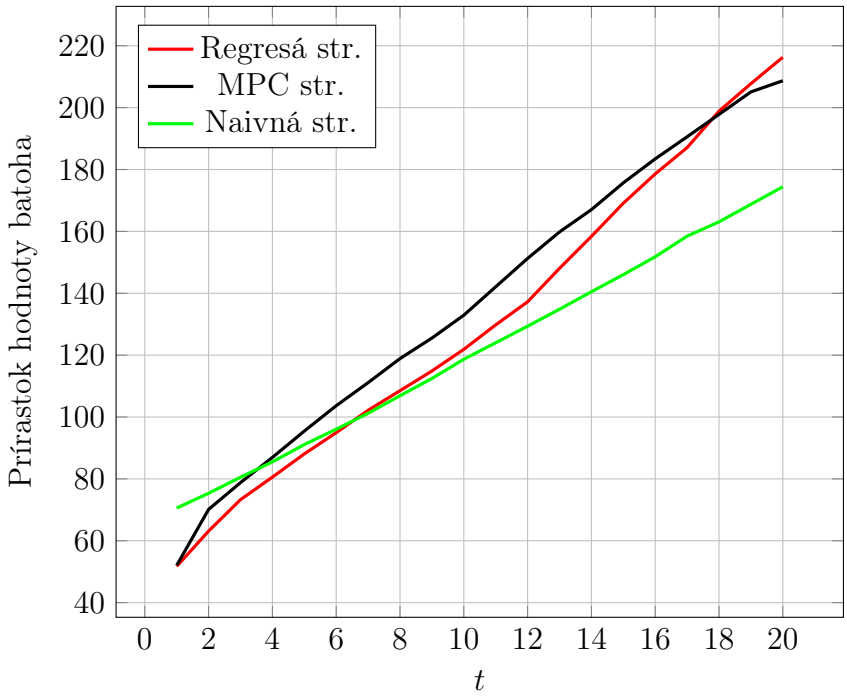


Obr. 3: Priemerný vývoj hodnôt c_{t-1} a $y_t^{(1)}$ v čase.

v priebehu väčšiny sledovaného obdobia dosahuje najnižšie priemerné zaplnenie Regresná stratégia, nasledovaná MPC stratégiou.

Najvyššiu mieru využitia kapacity spočiatku vykazuje Naivná stratégia. V záverečnej fáze časového horizontu však dochádza k signifikantnej zmene trendu: najvyššie zaplnenie batoha začína dosahovať práve Regresná stratégia. Tento jav potvrdzuje predpoklad o limitoch Naivnej stratégie, ktorá pracuje na princípe greedy (chamtivého) výberu. Keďže táto stratégia nepracuje s predikciou a neumožňuje strategicky „šetriť“ kapacitu pre hodnotnejšie položky, v neskorších krokoch už nedisponuje vhodným portfóliom predmetov. Menšie predmety, ktoré by mohli efektívne vyplniť zvyškovú kapacitu, boli totiž vyčerpané v počiatočných fázach, zatiaľ čo sofistikovanejšie prístupy (MPC a Regresná stratégia) si ich dokázali ponechať pre neskoršie využitie.

Z hľadiska hodnotenia kvality stratégie je dôležitejšie sledovať vývoj priemerného zisku než samotné zaplnenie batoha. Na Obr. 4 sme graficky znázornili prírastky hodnoty batoha, t.j. hodnotu $\sum_{i=1}^{n} p_i u_{t,i}$ v jednotlivých časových periódach, čím získame možnosť nahliadnuť do jednotlivých aproximačných riadení podrobnejšie než len porovnaním kumulatívnej výslednej hodnoty. Vidíme, že počas prvých období dosahuje najväčšie prírastky hodnoty batoha stratégia získaná pomocou Naivného algoritmu. Avšak počnúc $t = 3$



Obr. 4: Priemerné prírastky hodnoty batoha v čase.

sa poradie mení. MPC stratégia si udržiava najväčší prírastok hodnoty batoha počas väčšiny časového horizontu. Toto pozorovanie zrejme súvisí s väčším priemerným zaplnením batoha ako v prípade Regresnej stratégie. Od času $t = 17$ však dosahuje najväčší prírastok hodnoty batoha Regresná stratégia, ktorá vďaka opatrnejšiemu postupu dokáže neskôr vložiť do batoha hodnotnejšie predmety.

3.3 Porovnanie úspešnosti stratégií

V tejto časti porovnávame dosiahnutý zisk pri jednotlivých stratégiách a analyzujeme ich úspešnosť. Keďže nájsť optimálnu stratégiu nie je možné kvôli problému dimenzionality, presnosť riešenia stanovíme vzhľadom ku tzv. *Referenčnej stratégii*. Túto stratégiu získame riešením modifikácie pôvodného modelu, v ktorom je možné predmety do batoha nielen vkladať, ale aj vyberať. V takomto prípade vieme nájsť optimálnu stratégiu opakovaným riešením úlohy o batohu z časti 2.1 s aktuálnou hodnotou kapacity batoha.

Zisk, dosiahnutý pri Referenčnej stratégii, poskytuje horné ohraničenie dosiahnuteľného zisku pre problém inkrementálneho batoha. Porovnaním ostatných stratégií s Referenčnou stratégiou získame dolné ohraničenie ich percentuálnej presnosti.

Z Tabuľky 6 vidíme, že najväčší priemerný zisk dosahuje Referenčná stratégia, nasle-

duje MPC stratégia s 85.4% presnosťou, Regresná stratégia s 82.0% presnosťou a najnižší zisk dosahuje Naivná stratégia s 75.3% presnosťou. Na prvý pohľad by sa mohlo zdať, že MPC stratégia je vždy lepšia než Regresná stratégia. Kvôli dôkladnejšiemu porovnaniu sme analyzovali podiel prípadov, v ktorých jednotlivé stratégie viedli k najväčšiemu zisku. Výsledky numerických experimentov demonštrujú, že MPC stratégia bola najlepšia v 63.2% prípadov, Regresná stratégia v 35.4%, v 1.2% prípadov bola najlepšia Naivná stratégia a v 0.2% prípadov bola remíza medzi viacerými stratégiami. Tento experiment poukazuje na skutočnosť, že MPC stratégia neposkytuje vždy najlepšiu stratégiu a výsledok môže závisieť aj od konkrétnych predmetov a vývoja kapacity c_t.

Stratégia	**Priemerný zisk**	**Podiel víťazstiev (%)**	**Presnosť (%)**
Referenčná stratégia	3228.008	-	100.0
Regresná stratégia	2648.522	35.4	82.0
MPC stratégia	2756.984	63.2	85.4
Naivná stratégia	2429.592	1.2	75.3

Tabuľka 6: Porovnanie zisku a úspešnosti jednotlivých stratégií.

Z hľadiska výpočtovej náročnosti je časovo najnáročnejší výpočet koeficientov $\theta_{t,i}$ pre všetky $t = 0, 1, \ldots, T-1$ a $i = 1, 2, \ldots, n$. Dĺžka výpočtu závisí od počtu predmetov, pričom sa pohybuje v intervale 60 až 150 sekúnd.

Algoritmus hľadania Regresnej stratégie z časti 2.7 trvá približne 0.02 sekúnd pre jednu vzorku predmetov a jednu trajektóriu vývoja kapacity c_t. Hľadanie MPC stratégie z časti 2.8 trvá približne 0.08 sekúnd pre jednu vzorku predmetov a jednu trajektóriu vývoja kapacity c_t. Výpočet Naivnej a Referenčnej stratégie pre jednu vzorku predmetov a jednu trajektóriu vývoja kapacity c_t trval menej ako 0.01 sekúnd. Všetky uvedené časy boli namerané na zariadení Dell Latitude 7490 s procesorom Intel Core i5-8350U (4 jadrá) a 16 GB RAM.

4 Obchodovanie s batériou

V tejto kapitole sa budeme zaoberať problémom hľadania optimálnej stratégie obchodovania s batériou na vnútrodennom trhu s elektrickou energiou. Nadviažeme na diplomovú prácu [5]. Našim cieľom je predstaviť alternatívne metódy riešenia daného problému, ktoré vychádzajú z aproximačných techník predstavených v Kapitole 1.

Pod pojmom *batéria* budeme rozumieť veľkokapacitné úložisko elektrickej energie. Batérie umožňujú uskladniť elektrickú energiu, čím prispievajú k zvyšovaniu flexibility systému, stabilizácii elektrickej siete a efektívnejšiemu využívaniu obnoviteľných zdrojov energie. S pomocou batérie môžeme akumulovať elektrickú energiu v čase nízkeho dopytu alebo nízkych cien a následne ju dodať do siete v prípade zvýšeného dopytu alebo vyšších cien.

Batérie sa čoraz častejšie využívajú aj ako aktívny obchodný nástroj. Hlavným cieľom je maximalizácia ekonomického zisku na základe cenových rozdielov v čase. Tento prístup je obzvlášť významný na vnútrodenných trhoch, ktoré umožňujú reagovať na krátkodobé zmeny v ponuke a dopyte po elektrickej energii.

Vnútrodenný trh s elektrinou umožňuje obchodovanie elektrickej energie až do jednej hodiny pred samotnou dodávkou. Obchodovanie je realizované v diskrétnych časových intervaloch s dĺžkou 15 minút, ktoré predstavujú štandardné časové rozlíšenie európskych vnútrodenných trhov s elektrinou.

V ďalšej časti predstavíme matematickú formuláciu uvedeného problému. Našou snahou je pri modelovaní zohľadniť všetky technické obmedzenia batérie, akými sú kapacitné limity, účinnosť nabíjania a vybíjania a maximálny počet cyklov. Súčasne do modelu zahrnieme trhové obmedzenia vyplývajúce z dostupnosti nákupu a predaja elektrickej energie v jednotlivých časových okamihoch. Cieľom optimalizačnej úlohy je maximalizácia zisku z obchodovania pri dodržaní všetkých uvedených obmedzení.

4.1 Formulácia problému obchodovania s batériou

V tejto časti predstavíme matematickú formuláciu problému obchodovania s elektrickou energiou pomocou batérie. Analogicky ako v časti 2.4, aj problém obchodovania s batériou sformulujeme ako úlohu diskrétneho dynamického programovania.

Uvažujme časový horizont rozdelený do diskrétnych intervalov $t \in \{0, 1, \ldots, T-1\}$. Predpokladajme, že v každom čase t môže obchodník realizovať nákup, resp. predaj elektrickej energie pre m rôznych časových periód dňa.

Rozhodnutie o nákupe a predaji elektriny v čase t pre periódu i reprezentujeme dvoma riadiacimi premennými. Premenná $u_{t,i} \in [0, 2c], \quad t = 0, \ldots, T-1, \ i = 0, \ldots, m-1$, označuje množstvo nakúpenej elektriny v čase t pre periódu i. Premenná $v_{t,i} \in [0, 2c], \quad t = 0, \ldots, T-1, \ i = 0, \ldots, m-1$, označuje množstvo predanej elektriny v čase t pre periódu i. Pre lepšiu prehľadnosť budeme riadiace premenné v čase t zapisovať aj vo vektorovom tvare $\mathbf{u_t} = (u_{t,0}, \ldots, u_{t,m-1})^\top, \ \mathbf{v_t} = (v_{t,0}, \ldots, v_{t,m-1})^\top$.

Parameter c označuje maximálne množstvo elektriny, ktorú môžeme nakúpiť alebo v ľubovoľnom čase t a pre ľubovoľnú periódu i. Tento parameter je konštantný, t.j. nezávisí od času t ani od periódy i.

Na popísanie aktuálneho stavu batérie v čase t definujeme sedem stavových premenných. Prvou stavovou premennou je množstvo zobchodovanej elektriny do času t na periódu i, ktoré označíme ako $y^{(1)}_{t,i}$. Dynamiku vývoja premennej $y^{(1)}_{t,i}$ popíšeme stavovou rovnicou

$$\begin{aligned} y^{(1)}_{t,i+1} &= y^{(1)}_{t,i} + u_{t,i} - v_{t,i}, \quad t = 0, \ldots, T-1, \qquad i = 0, \ldots, m-1, \\ \left| y^{(1)}_{t,i} \right| &\leq c, \qquad\qquad\qquad t = 0, \ldots, T-1, \qquad i = 0, \ldots, m-1, \\ y^{(1)}_{0,i} &= 0, \qquad\qquad\qquad\qquad\qquad\quad i = 0, \ldots, m-1. \end{aligned} \tag{28}$$

Horné ohraničenie v (28) zabezpečuje, že na žiadnu periódu i nie je nakúpená alebo predaná elektrina vo väčšom objeme ako je maximálna veľkosť jedného obchodu c. Počiatočná podmienka stanovuje, že na začiatku uvažovaného nebola žiadna elektrická energia zobchodovaná.

Druhou stavovou premennou je množstvo elektriny dostupné v batérii v čase t pre periódu i, ktoré označíme ako $y^{(2)}_{t,i}$. Dynamiku tejto stavovej premennej popíšeme rovnicou

$$\begin{aligned} y^{(2)}_{t,i+1} &= y^{(2)}_{t,i} + y^{(1)}_{t,i} + u_{t,i} - v_{t,i}, \qquad t = 0, \ldots, T-1, \ i = 0, \ldots, m-1, \\ L_y \leq y^{(2)}_{t,i} &\leq U_y, \qquad\qquad\qquad\quad t = 0, \ldots, T-1, \ i = 0, \ldots, m-1, \\ y^{(2)}_{t,0} &= a_0, \qquad\qquad\qquad\qquad\quad t = 0, \ldots, T-1, \\ y^{(2)}_{t,m} &= a_m, \qquad\qquad\qquad\qquad\quad t = 0, \ldots, T-1. \end{aligned} \tag{29}$$

Podmienky na stavovú premennú $y^{(2)}_{t,i}$ uvedené v (29) zabezpečujú, že množstvo elektriny v

batérii ostáva v rámci jej kapacitných obmedzení. Počiatočná podmienka určuje množstvo elektriny v batérii na začiatku dňa a koncová podmienka určuje množstvo elektriny v batérii na konci dňa.

Premenná $y_{t,i}^{(1)}$ je potrebná na sledovanie čistej pozície v čase t pre danú periódu i. Aj v prípade, že stav batérie $y_{t,i}^{(2)}$ umožňuje väčšie nabitie alebo vybitie, počas jednej periódy i nie je možné nakúpiť ani predať viac ako c. Parameter c udáva koľko elektrickej energie vie batéria dodať, resp. odobrať z elektrickej siete počas jednej periódy i.

Treťou stavovou premennou je kumulatívna spotreba batérie, ktorú označíme ako $y_{t,i}^{(3)}$. Táto premenná zachytáva opotrebenie batérie spôsobené nabíjaním a vybíjaním. Jej dynamika je daná vzťahom

$$\begin{aligned} y_{t,i+1}^{(3)} &= y_{t,i}^{(3)} + \left| y_{t,i}^{(1)} + u_{t,i} - v_{t,i} \right|, && t = 0, \ldots, T-1,\ i = 0, \ldots, m-1, \\ y_{t,m}^{(3)} &\leq 2g(U_y - L_y), && t = 0, \ldots, T-1, \\ y_{t,0}^{(3)} &= 0, && t = 0, \ldots, T-1. \end{aligned} \tag{30}$$

Horné ohraničenie v (30) zabezpečuje, že spotreba batérie neprekročí povolený denný počet cyklov batérie. Počiatočná podmienka stanovuje, že na začiatku nebola žiadna elektrická energia zobchodovaná.

Ďalšou stavovou premennou je cena pri predaji, resp. pri nákupe, elektriny v čase t pre periódu i, označíme $p_{t,i}^{bid}$, resp. $p_{t,i}^{ask}$. Vývoj cien modelujeme ako stochastický proces, pričom dynamiku ceny opisujeme funkciou f, závislou od aktuálnej ceny a náhodnej veličiny. Pre obe premenné $p_{t,i}^{bid}$ a $p_{t,i}^{ask}$ uvažujeme rovnakú funkciu f. Dynamika vývoja cien je daná vzťahmi

$$\begin{aligned} p_{t+1,i}^{bid} &= f(p_{t,i}^{bid}, z_{t,i}), \quad t = 0, \ldots, T-1, \qquad i = 0, \ldots, m-1, \\ p_{t+1,i}^{ask} &= f(p_{t,i}^{ask}, z_{t,i}), \quad t = 0, \ldots, T-1, \qquad i = 0, \ldots, m-1, \end{aligned} \tag{31}$$

kde $z_{t,i}$ je náhodná veličina reprezentujúca neistotu vývoja cien. Modelovaniu cien sa podrobnejšie budeme venovať v časti 4.5.

Poslednými dvoma stavovými premennými sú $p_{t,i}^{(1)}$ a $p_{t,i}^{(2)}$, ktoré označujú poslednú cenu do času t, za ktorú bola elektrina pre danú periódu i nakúpená, resp. predaná. Tieto stavové premenné zavádzame z dôvodu zohľadnenia účinnosti nabíjania a vybíjania batérie. Pri prenose elektriny do alebo zo siete dochádza k stratám, t.j. množstvo energie dodané do batérie nie je rovnaké ako množstvo energie odobraté zo siete a naopak.

Tieto premenné sú určené vzťahmi

$$
\begin{aligned}
p^{(1)}_{t+1,i} &= \begin{cases} p^{(1)}_{t,i}, & \text{ak } y^{(1)}_{t,i} > 0, \\ p^{ask}_{t,i}, & \text{ak } u_{t,i} > 0, \\ 0, & \text{inak,} \end{cases} \qquad t = 0, \ldots, T-1, \qquad i = 0, \ldots, m-1, \\
p^{(2)}_{t+1,i} &= \begin{cases} p^{(2)}_{t,i}, & \text{ak } y^{(1)}_{t,i} < 0, \\ p^{bid}_{t,i}, & \text{ak } v_{t,i} > 0, \\ 0, & \text{inak,} \end{cases} \qquad t = 0, \ldots, T-1, \qquad i = 0, \ldots, m-1, \\
p^{(1)}_{0,i} &= 0, \qquad i = 0, \ldots, m-1, \\
p^{(2)}_{0,i} &= 0, \qquad i = 0, \ldots, m-1.
\end{aligned}
\tag{32}
$$

Stavové premenné budeme tiež zapisovať aj vo vektorovom tvare $\mathbf{y^{(1)}_t} = (y^{(1)}_{t,0}, \ldots, y^{(1)}_{t,m-1})^\top$, $\mathbf{y^{(2)}_t} = (y^{(2)}_{t,0}, \ldots, y^{(2)}_{t,m-1})^\top$, $\mathbf{y^{(3)}_t} = (y^{(3)}_{t,0}, \ldots, y^{(3)}_{t,m-1})^\top$, $\mathbf{p^{bid}_t} = (p^{bid}_{t,0}, \ldots, p^{bid}_{t,m-1})^\top$, $\mathbf{p^{ask}_t} = (p^{ask}_{t,0}, \ldots, p^{ask}_{t,m-1})^\top$, $\mathbf{p^{(1)}_t} = (p^{(1)}_{t,0}, \ldots, p^{(1)}_{t,m-1})^\top$ a $\mathbf{p^{(2)}_t} = (p^{(2)}_{t,0}, \ldots, p^{(2)}_{t,m-1})^\top$.

Cieľom je maximalizovať celkový zisk z obchodovania počas uvažovaného časového horizontu, čo reprezentujeme účelovou funkciou v tvare

$$
\max \sum_{t=0}^{T-1} \sum_{i=0}^{m-1} \left(p^{bid}_{t,i} v_{t,i} - p^{ask}_{t,i} u_{t,i} \right) + K(\mathbf{y^{(2)}_T}, \mathbf{p^{(1)}_T}, \mathbf{p^{(2)}_T}), \tag{33}
$$

kde funkcia $K(\mathbf{y^{(2)}_T}, \mathbf{p^{(1)}_T}, \mathbf{p^{(2)}_T})$ kvantifikuje stratu, ku ktorej dochádza pri nabíjaní a vybíjaní batérie. Podrobná formulácia presného tvaru účelovej funkcie, ako aj formulácia celého modelu je pomerne rozsiahla, preto ju pre zachovanie prehľadnosti uvádzame v prílohe A.

4.2 Problém dimenzionality pri obchodovaní s batériou

Ako sme uviedli v časti 1.2, nájdenie optimálnej stratégie obchodovania vyžaduje riešiť príslušnú RDP. V modeli predstavenom v časti 4.1 máme až sedem stavových premenných, preto pre zvýšenie prehľadnosti zavedieme označenie $\mathbf{x_t} = (\mathbf{y^{(1)}_t}, \mathbf{y^{(2)}_t}, \mathbf{y^{(3)}_t}, \mathbf{p^{bid}_t}, \mathbf{p^{ask}_t}, \mathbf{p^{(1)}_t}, \mathbf{p^{(2)}_t})$. Príslušnú RDP potom vyjadríme v nasledujúcom tvare

$$V_t(\mathbf{x_t}) = \max_{\mathbf{u_t}\in\mathcal{U}(\mathbf{x_t}),\mathbf{v_t}\in\mathcal{V}(\mathbf{x_t})} \left[f_t^0(\mathbf{x_t},\mathbf{u_t},\mathbf{v_t}) + \mathbb{E}\left[V_{t+1}(\mathbf{x_{t+1}})\right]\right], t = 0,\ldots,T-1, \tag{34}$$

$$V_T(\mathbf{x_T}) = \begin{cases} 0, & \text{pre } (\mathbf{x_T}) \in \mathcal{X}_T, \\ -\infty, & \text{pre } (\mathbf{x_T}) \notin \mathcal{X}_T, \end{cases} \tag{35}$$

kde $f_t^0(\mathbf{x_t},\mathbf{u_t},\mathbf{v_t}) = \sum_{i=0}^{m-1}\left(p_{t,i}^{bid}v_{t,i} - p_{t,i}^{ask}u_{t,i}\right)$ a $\mathcal{X}_T$ je množina prípustných stavov systému v čase T.

Nájdenie presného riešenia je vzhľadom na rozmer úlohy výpočtovo náročné. Výpočet hodnotovej funkcie $V_t(\mathbf{x_t})$ vyžaduje určiť jej hodnotu pre každý prípustný stav $\mathbf{x_t}$. Celkový počet stavov je daný súčinom počtu možných hodnôt jednotlivých zložiek stavového vektora $\mathbf{x_t}$. Výpočtová náročnosť pri obchodovaní s batériou závisí od diskretizácií cien $\mathbf{p_t^{bid}}, \mathbf{p_t^{ask}}, \mathbf{p_t^{(1)}}$ a $\mathbf{p_t^{(2)}}$. Ak by sme ceny diskretizovali na 300 hodnôt, rozmer stavu cien by bol 300^4, t.j. približne 8.1 miliardy. Avšak ani takáto diskretizácia nemusí byť dostatočná, nakoľko na vnútrodennom trhu rozdiel medzi najnižšou a najvyššou zobchodovanou cenou v niektoré dni presahuje hodnotu 300€ a navyše sa obchoduje s presnosťou na jeden cent.

4.3 Lineárny odhad hodnotovej funkcie

Vzhľadom na veľkosť stavového priestoru pri obchodovaní s batériou nie je možné nájsť optimálne riadenie priamym spätným riešením RDP (34) v reálnom výpočtovom čase. Podobne ako v časti 2.6, preto využijeme aproximáciu hodnotovej funkcie pomocou lineárnej kombinácie bázických funkcií.

Základný postup ostáva rovnaký ako pri inkrementálnom probléme batoha. Hodnotovú funkciu $V_t(\mathbf{x_t})$ nepočítame presne pre všetky možné stavy $\mathbf{x_t}$, ale nahradíme ju aproximáciou $\bar{V}_t(\mathbf{x_t})$. V každom čase t najprv vypočítame hodnoty $\tilde{V}_t(\mathbf{x_t})$ pre vhodne zvolené stavy $\mathbf{x_t}$ a následne na základe dvojíc $[\mathbf{x_t}, \tilde{V}_t(\mathbf{x_t})]$ odhadneme koeficienty θ_t v lineárnom modeli hodnotovej funkcie.

Rozdiel oproti inkrementálnemu problému batoha spočíva najmä v štruktúre stavovej premennej. Ako sme už spomenuli, stavová premenná $\mathbf{x_t}$ má sedem zložiek. Ťažisko generovania stavovej premennej ešte spočíva v „rozumnej" voľbe zložiek $\mathbf{y_t^{(1)}}$, $\mathbf{y_t^{(2)}}$, $\mathbf{y_t^{(3)}}$, $\mathbf{p_t^{(1)}}$ a $\mathbf{p_t^{(2)}}$. Generovanie cien $\mathbf{p_t^{bid}}$ a $\mathbf{p_t^{ask}}$ podrobne popíšeme v časti 4.5.

Generovanie vektorov $\mathbf{y_t^{(1)}}, \mathbf{y_t^{(2)}}, \mathbf{y_t^{(3)}}, \mathbf{p_t^{(1)}}$ a $\mathbf{p_t^{(2)}}$ realizujeme pomocou *Benchmarkovej stratégie*. Táto stratégia v každom čase t zvolí takú kombináciu riadiacich premenných $\mathbf{u_t}$ a $\mathbf{v_t}$, ktorá maximalizuje zisk v čase t, t.j.

$$\max_{\mathbf{u_t}\in\mathcal{U}(\mathbf{x_t}),\mathbf{v_t}\in\mathcal{V}(\mathbf{x_t})} \sum_{i=0}^{m-1} \left(p_{t,i}^{bid} v_{t,i} - p_{t,i}^{ask} u_{t,i} \right). \tag{36}$$

Pri generovaní hodnôt stavových premenných začneme s počiatočnými podmienkami v tvare $y_{t-1,i}^{(1)} = 0$, $y_{t-1,i}^{(2)} = 0$, $y_{t-1,i}^{(3)} = 0$, $p_{t-1,i}^{(1)} = 0$ a $p_{t-1,i}^{(2)} = 0$ pre všetky $i = 0, \ldots, m-1$. Aplikáciou Benchmarkovej stratégie v čase $t-1$ získame riadiace premenné $\mathbf{u_{t-1}}$ a $\mathbf{v_{t-1}}$, na základe ktorých následne pomocou stavových rovníc 28, 29, 30 a 32 určíme hodnoty $\mathbf{y_t^{(1)}}, \mathbf{y_t^{(2)}}, \mathbf{y_t^{(3)}}, \mathbf{p_t^{(1)}}$ a $\mathbf{p_t^{(2)}}$.

Výpočet hodnôt $\tilde{V}_t(x_t)$ budeme, analogicky ako pri riešení RDP, vykonávať spätne od konca časového horizontu. Je potrebné riešiť úlohu

$$\begin{aligned} \tilde{V}_t(\mathbf{x_t}) \;&=\; \max_{\mathbf{u_t}\in\mathcal{U}(\mathbf{x_t}),\mathbf{v_t}\in\mathcal{V}(\mathbf{x_t})} \left[f_t^0(\mathbf{x_t}, \mathbf{u_t}, \mathbf{v_t}) + \mathbb{E}\left[\bar{V}_{t+1}(\mathbf{x_{t+1}}) \right] \right], \\ V_T(\mathbf{x_T}) &= 0, \text{ pre } \mathbf{x_T} \in \mathcal{X}_T. \end{aligned} \tag{37}$$

V modeli predpokladáme, že v prípade, ak je elektrická energia obchodovateľná, je možné nakúpiť alebo predať elektrinu v objeme aspoň $2c$, čo predstavuje horné ohraničenie riadiacich premenných $\mathbf{u_t}$ a $\mathbf{v_t}$.

V prvom kroku si uvedomme, aké hodnoty môžu nadobúdať riadiace premenné $u_{t,i}$ a $v_{t,i}$. Na začiatku máme prázdnu batériu, $y_{0,0} = 0$. Ak realizujeme nákup, tak kúpime elektrinu v maximálnej kapacite, ktorá je prípustná, tj. $u_{0,0} = c$. Ak by sme nakúpili menej, tj. $u_{0,0} < c$, to by znamenalo, že sme ušetrili nejakú kapacitu, ktorú by sme mohli speňažiť za výhodnejšiu cenu. Potom by ale bolo lepšie zvoliť $u_{0,0} = 0$ a počkať s nákupom až na tú výhodnejšiu cenu. To znamená, že ak realizujeme nákup, tak musí jeho veľkosť byť c. Pozrime sa teraz na situáciu, v ktorej $y_{t,i} = -c$, t.j. do času t sme realizovali predaj na periódu i. V čase t môžeme uzavrieť túto pozíciu, vtedy nakúpime $u_{t,i} = c$. Avšak môžeme chcieť nakúpiť na túto periódu. V takom prípade by zvolíme $u_{t,i} = 2c$. Analogicky postupujeme aj pri predaji, a preto stačí uvažovať nasledujúce hodnoty riadiacich premenných $u_{t,i} \in \{0, c, 2c\}$, resp. $v_{t,i} \in \{0, c, 2c\}$.

Po zostavení dvojíc $[\mathbf{x_t}, \tilde{V}_t(\mathbf{x_t})]$ môžeme pristúpiť k aproximácii hodnotovej funkcie $\bar{V}_t(\mathbf{x_t})$. Odhadneme ju v tvare (8). Pre obchodovanie s batériou sme navrhli nasledujúce tri vysvetľujúce bázické funkcie:

1. Konštantná funkcia $\phi_{t,1} = 1$, ktorá umožňuje zachytiť „základnú" úroveň hodnotovej funkcie.

2. Funkcia $\phi_{t,2} = y_{t,m}^{(3)}$, ktorá vyjadruje kumulatívnu spotrebu elektriny.

3. Funkcia $\phi_{t,3} = d_t$, ktorá vyjadruje rozdiel medzi zobchodovanými cenami a priemernými cenami na hodnotu riešenia.

Parametrom μ_i označíme priemernú úroveň, ku ktorej sa cena $p_{t,i}$ vracia. Tento parameter uvažujeme ako priemer za obdobie jedného mesiaca. Potom si vieme vyjadriť

$$d_t = \sum_{i=0}^{m-1} y_{t,i}^{(1)}(p_{t,i}^{(1)} + p_{t,i}^{(2)} - \mu_i). \tag{38}$$

Na nasledujúcom príklade demonštrujeme dôležitosť parametra d_t. Predstavme si, že sme v čase $t = 10$ a máme kapacitu už len na jeden nákup a jeden predaj. Ďalej predpokladajme, že ceny $p_{t,i}$, za ktoré je možné elektrinu nakúpiť alebo predať, a priemerné ceny μ_i v i-tej perióde majú hodnotu ako v Tabuľke 7. Uvažujme dva scenáre: scenár A, v ktorom predáme elektrinu na periódu $i = 90$ a nakúpime na periódu $i = 91$, a scenár B, v ktorom predáme elektrinu na periódu $i = 92$ a nakúpime na periódu $i = 93$.

Čas	Perióda	μ_i	$p_{t,i}$
10	90	60	80
10	91	80	60
10	92	80	80
10	93	60	60

Tabuľka 7: Možnosti nákupu a predaja pre jednotlivé periódy

Pri oboch scenároch je okamžitý zisk 20. Rozdiel medzi scenármi spočíva v tom, že v scenári A sa aktuálne ceny $p_{t,i}$ líšia od priemerných cien μ_i, zatiaľ čo v scenári B sa aktuálne ceny rovnajú priemerným cenám.

Nech v čase $t = 20$ sa ceny z oboch scenárov dostanú na ich priemernú cenu pre periódu i, čo je znázornené v Tabuľke 8.

Čas	Perióda	μ_i	$p_{t,i}$
20	90	60	60
20	91	80	80
20	92	80	80
20	93	60	60

Tabuľka 8: Možnosti nákupu a predaja pre jednotlivé periódy v čase $t = 20$

V scenári B sa nič nezmení a náš zisk bude stále 20. V scenári A zakúpenú elektrinu na periódu 91 predáme a predanú elektrinu na periódu 90 kúpime späť. Z toho budeme mať zisk 40 a ďalších 20 získame z nákupu elektriny na 90. periódu a predaj elektriny na periódu 91.

V prvom scenári je $d_{t+1} = 40$, zatiaľ čo v druhom scenári je $d_{t+1} = 0$, takže vidíme, že vyššia hodnota parametra d_t môže mať vplyv na vyšší zisk v budúcnosti.

Rovnako ako v časti 2.6 chceme overiť, že koeficienty $\theta_{t,i}$ sú štatisticky významné na hladine významnosti 5 %. Na tento účel sme pre čas $t = T-1$ vykonali t-test významnosti parametrov v lineárnom regresnom modeli.

Výsledky t-testu pre čas $t = T - 1$ sú uvedené v Tabuľke 9. Z Tabuľky 9 je zrejmé, že štatisticky nevýznamný sa ukázal byť koeficient $\theta_{T-1,3}$, ktorý zodpovedá parametru d_t (rozdiel medzi zobdchodovanými a priemernými cenami). Naopak, ostatné koeficienty sú štatisticky významné.

Premenná	Hodnota	p-value
$\theta_{T-1,1}$	17.547057	$< 10^{-6}$
$\theta_{T-1,2}$	-1.694648	3.04×10^{-5}
$\theta_{T-1,3}$	0.021092	0.1412

Tabuľka 9: Štatistická významnosť parametrov

Pri aproximácii hodnotovej funkcie $\bar{V}_t(x_t)$ sme sa rozhodli použiť aj bázickú funkciu $\phi_{t,3}$, ktorej koeficient sa ukázal ako štatisticky nevýznamný. Dôvodom je skutočnosť, že máme len tri bázické funkcie a v skorších časoch by mohla zachytávať očakávanú budúcu hodnotu, ktorá nie je plne vysvetlená ostatnými bázickými funkciami.

4.4 Parametre modelu

Do modelu predstaveného v časti 4.1 vstupuje viacero parametrov, ktorých prehľad spolu s ich významom uvádzame v Tabuľke 10. V poslednom stĺpci Tabuľky 10 uvádzame hodnoty parametrov, s ktorými realizujeme numerickú analýzu.

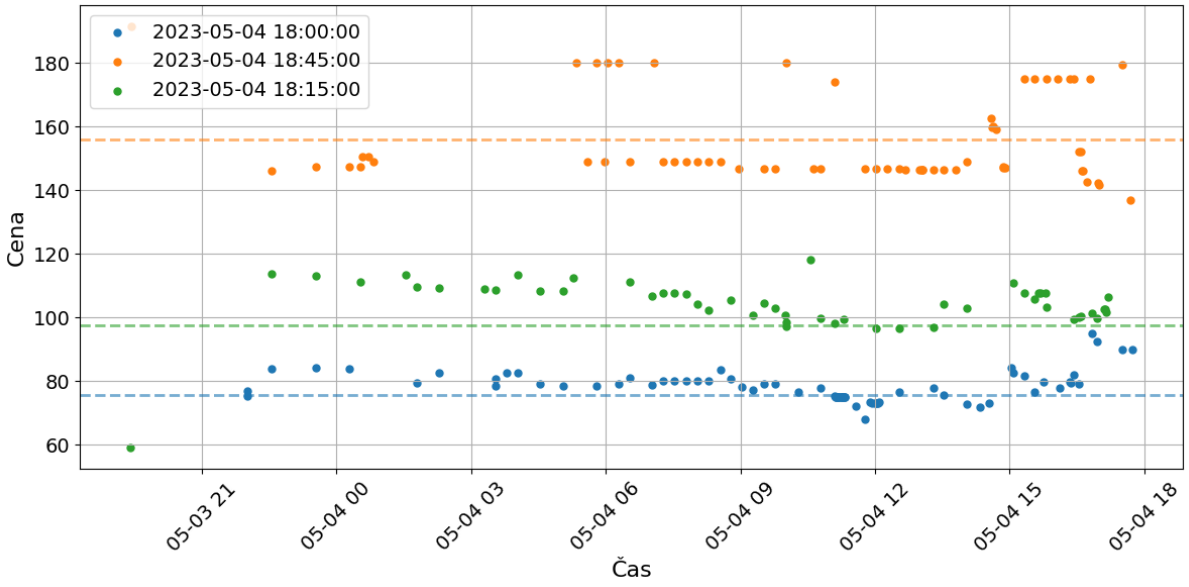
Pri numerickej analýze uvažujeme batériu s kapacitou 1 MW, z čoho vyplývajú hranice $L_x = 0$ MW a $U_x = 1$ MW. Maximálna veľkosť jedného obchodu c zodpovedá maximálnemu výkonu batérie pri 15-minútovej perióde. Koeficienty účinnosti α a β boli zvolené na základe typických hodnôt pre súčasné batérie. Parameter g, ktorý obmedzuje počet cyklov nabitia a vybitia, bol zvolený s cieľom zabrániť degradácii batérie pri nadmernom využívaní.

Parametre	Popis	Hodnota
a_0	Počiatočné množstvo elektriny dostupné v batérii	0.5 MW
a_m	Konečné množstvo elektriny dostupné v batérii	0.5 MW
L_y	Dolná hranica množstva elektriny dostupnej v batérii	0 MW
U_y	Horná hranica množstva elektriny dostupnej v batérii	1 MW
$r^a_{t,i}$	Indikátor dostupnosti nákupu elektriny	$\{0, 1\}$
$w_{t,i}$	Status batérie	$\{0, 1\}$
$r^b_{t,i}$	Indikátor dostupnosti predaja elektriny	$\{0, 1\}$
c	Maximálna povolená veľkosť jedného obchodu	$\frac{U_x - L_x}{4}$ MW
g	Maximálny povolený počet cyklov nabitia a vybitia za deň	2
α	Koeficient účinnosti nabíjania batérie	0.92
β	Koeficient účinnosti vybíjania batérie	0.92
T	Dĺžka obchodovacieho obdobia (t)	
m	Počet periód (i)	96

Tabuľka 10: Vstupné parametre modelu obachodovania s batériou.

4.5 Modelovanie cien

Ceny elektriny na vnútrodennom trhu sa pre jednotlivé periódy i menia v priebehu dňa. Na Obr. 5 sme na ilustráciu znázornili priebeh zobchodovaných cien pre periódy $i = 73$ (18:00-18:15), 74 (18:15-18:30) a 76 (18:45-19:00) počas dňa 4.5.2023.



Obr. 5: Vývoj cien počas dňa 4.5.2023 pre periódy $i = 73, 74$ a 76.

Vidíme, že ceny majú tendenciu vracať sa k svojej priemernej úrovni. Na modelovanie cien $\mathbf{p_t^{bid}}$ a $\mathbf{p_t^{ask}}$ preto využijeme *mean reversion proces* v tvare

$$dp_{t,i} = \gamma(\mu_i - p_{t,i})\, dt + \sigma\, dW_t. \tag{39}$$

Parameter μ_i predstavuje priemernú úroveň, ku ktorej sa cena $p_{t,i}$ vracia. Parameter γ vyjadruje rýchlosť návratu k tejto úrovni a parameter σ určuje volatilitu ceny. Člen W_t označuje Wienerov proces (štandardný Brownov pohyb), pričom jeho prírastok dW_t reprezentuje náhodnú zložku vývoja ceny. Model (39) predstavuje jeden zo základných prístupov používaných na modelovanie vývoja cien elektrickej energie, viď napr. [3].

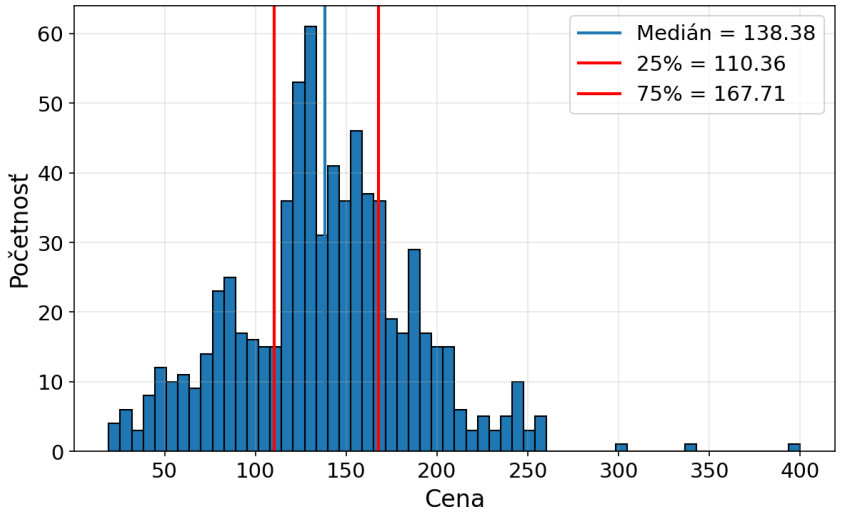
Nakoľko v ďalšej práci uvažujeme diskrétny časový model, spojitý proces (39) aproximujeme jeho diskrétnou verziou nasledovne

$$p_{t+1,i} = p_{t,i} + \gamma(\mu_i - p_{t,i})\Delta t + \sigma\sqrt{\Delta t}\,\varepsilon_t, \tag{40}$$

kde ε_t je náhodná veličina so štandardným normálnym rozdelením $\mathcal{N}(0, 1)$.

Parametre γ, μ_i a σ boli odhadnuté z historických dát na HUPX vnútrodennom trhu z obdobia 1.1.– 31.5.2023, pričom dáta boli poskytnuté spoločnosťou Slovenské elektrárne. Ceny elektrickej energie sa vyznačujú výrazným výskytom odľahlých hodnôt, čo dokumentuje Obr. 6. Parameter μ_i sme preto určili ako medián zobchodovaných cien za máj 2023 pre danú periódu i. Samostatný odhad parametrov γ a σ pre každú periódu i je problematický, pretože pri periódach s nízkym počtom pozorovaní by takéto odhady mohli byť

nepresné a nestabilné. Z tohto dôvodu sme tieto parametre uvažovali ako konštantné pre všetky periódy.



Obr. 6: Histogram početnosti zobchodovaných cien pre periódu $i = 33$ v máji 2023

Samotnému odhadu parametrov γ a σ predchádzala filtrácia odľahlých hodnôt. Tento krok zabezpečuje nielen vyššiu presnosť štatistického odhadu, ale aj realistickejší charakter následne generovaných cien. Z pôvodného súboru dát sme vylúčili spodný a horný kvartil, teda 25% najnižších a 25% najvyšších cien. Na ilustráciu sme pre periódu $i = 33$ na Obr. 6 znázornili histogram cien spolu s cenovým intervalom, ktorý bol uvažovaný pri odhade parametrov.

Parametre γ a σ sme odhadli z dvojíc cien, ktoré boli pre periódu i najbližšie k sebe. Keďže dva po sebe idúce obchody pre nejakú periódu i môžu byť hodiny vzdialené, tak sme sa rozhodli odhadnúť parametre γ a σ len na obchodoch, ktoré boli maximálne 60 minút vzdialené od seba.

Ďalšia situácia, ktorá môže nastať pri obchodovaní s elektrinou je, že sa cena pre nejakú periódu i za krátky čas výrazne zmení. Takéto zmeny by sa síce dali modelovať, ale pre zjednodušenie sme sa pri modelovaní cien, rozhodli takéto skoky nebrať do úvahy. To je dôvod, prečo sme pri odhade parametrov γ a σ vylúčili 5% dvojíc, ktorých pomer v absolútnej hodnote bol najvyšší. Na takto upravených dvojiciach sme následne pomocou lineárnej regresie odhadli parametre γ a σ.

Pri analýze historických cien a objednávok, napríklad zo dňa 4.5.2023, si môžeme všimnúť určitú zotrvačnosť obchodovateľnosti jednotlivých periód. Ak sa pre i-tu periódu

objaví objednávka na nákup alebo predaj, často zostáva v knihe objednávok určitý čas, kým je realizovaná alebo zrušená. Naopak, ak v čase t pre i-tu periódu neexistuje žiadna objednávka, pravdepodobnosť, že sa v nasledujúcom čase $t+1$ objaví nová objednávka, býva relatívne nízka.

Z historických dát preto odhadneme dve podmienené pravdepodobnosti: pravdepodobnosť, že cena bude v čase t obchodovateľná za predpokladu, že bola obchodovateľná v čase $t-1$, a pravdepodobnosť, že cena bude v čase t obchodovateľná za predpokladu, že v čase $t-1$ obchodovateľná nebola. Následne vygenerované ceny $\mathbf{p_t^{bid}}$ a $\mathbf{p_t^{ask}}$ upravíme tak, aby ich obchodovateľnosť zodpovedala týmto odhadnutým pravdepodobnostiam.

4.6 Výpočet riadiacich premenných

Po vypočítaní parametrov $\theta_{t,i}$ pre všetky $t = 0, 1, \ldots, T-1$ a $i = 0, 1, \ldots, m-1$ môžeme pristúpiť k samotnému hľadaniu $\mathbf{u_t}$ a $\mathbf{v_t}$. V každej časovej perióde t chceme pre aktuálny stav $\mathbf{x_t}$ určiť riadiace premenné $\mathbf{u_t}$ a $\mathbf{v_t}$. To znamená vyriešiť úlohu

$$\begin{aligned} &\arg \max_{\mathbf{u_t} \in \mathcal{U}(\mathbf{x_t}), \mathbf{v_t} \in \mathcal{V}(\mathbf{x_t})} \left[f_t^0(\mathbf{x_t}, \mathbf{u_t}, \mathbf{v_t}) + \mathbb{E}\left[\bar{V}_{t+1}(\mathbf{x_{t+1}}) \right] \right], \\ &V_T(\mathbf{x_T}) = 0. \end{aligned} \tag{41}$$

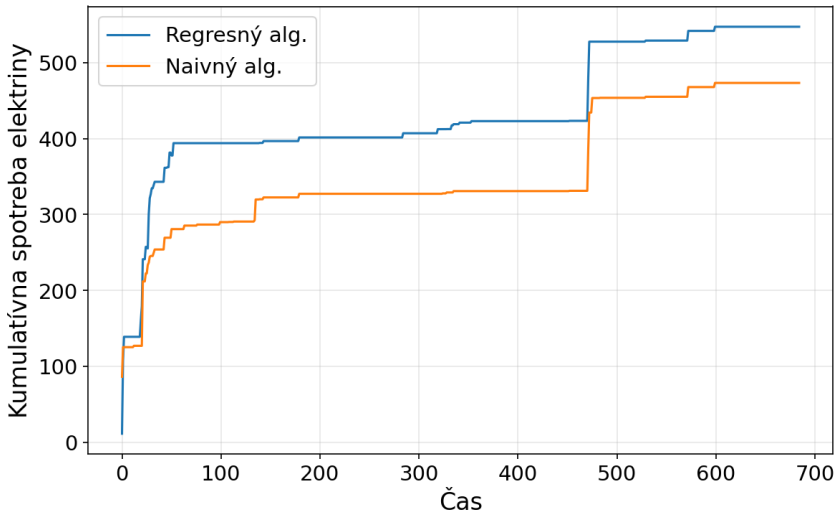
Vďaka zvolenej forme bázických funkcií vieme vyhodnotiť aj člen $\mathbb{E}\left[\bar{V}_{t+1}(\mathbf{x_{t+1}})\right]$. V dôsledku toho nie je potrebné navrhovať samostatný aproximačný algoritmus, ako v časti 2.7 pri inkrementálnom probléme batoha, ale riadiace premenné $\mathbf{u_t}$ a $\mathbf{v_t}$ je možné získať priamym výpočtom 41.

4.7 Výsledky

V tejto časti prezentujeme výsledky numerického testovania predstaveného algoritmu na riešenie úlohy obchodovania s batériou. Na základe odhadnutých parametrov vstupujúcich do modelovania cien z časti 4.5 sme následne vygenerovali 50 trajektórií cien $\mathbf{p_t^{bid}}$ a $\mathbf{p_t^{ask}}$.

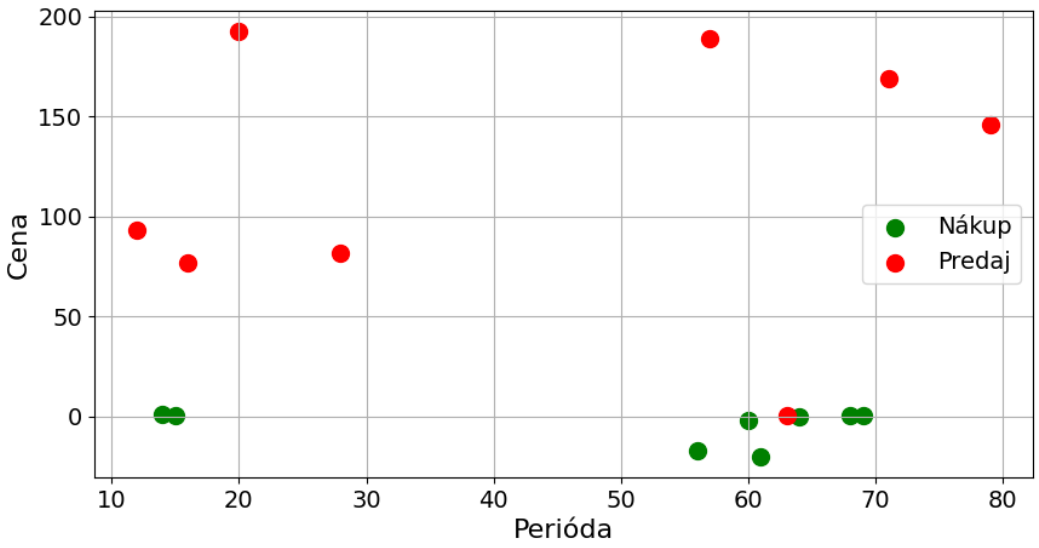
Pre všetky trajektórie cien sme vypočítali hodnotu parametrov $\theta_{t,i}$ pre všetky $t = 0, 1, \ldots, T-1$ a $i = 0, 1, \ldots, m-1$. Stratégiu získanú riešením úlohy (41) budeme označovať ako *Regresnú obchodovaciu stratégiu*.

Porovnanie algoritmov sme najprv realizovali na jednom konkrétnom dni, konkrétne 4.6.2023. Vývoj zisku počas tohto dňa je znázornený na Obr. 7.



Obr. 7: Vývoj zisku počas dňa

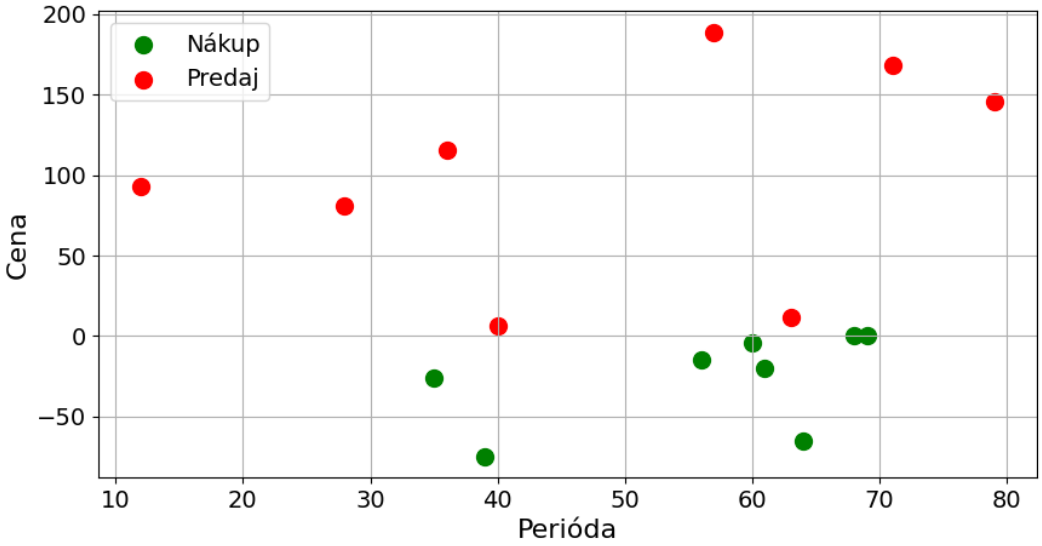
Okrem vývoja zisku, na Obr. 8 sledujeme $\mathbf{p_T^{(1)}}$ a $\mathbf{p_T^{(2)}}$, čo označuje poslednú cenu zobchodovanej elektriny do času T, pre Benchmarkovú stratégiu. Môžeme si všimnúť, že nákupy sú väčšinou za nízke ceny okolo 0€ alebo aj za záporné ceny. Predaje sú za vysoké ceny, okrem periódy $i = 63$.



Obr. 8: Cena zobchodovanej elektriny do času T pre Benchmarkovú stratégiu

Na Obr. 9 sledujeme $\mathbf{p_T^{(1)}}$ a $\mathbf{p_T^{(2)}}$, pre Regresnú obchodovaciu stratégiu. Väčšina nákupov a predajov je rovnakých pre obe stratégie. Ako vidíme, k rozdielu dochádza napríklad pri predaji, ktorý v prípade Regresnej obchodovacej stratégie nenastal v perióde $i = 20$,

ale až v perióde $i = 40$. Pri nákupoch pozorujeme, že v prípade Regresnej obchodovacej stratégie sa nákup realizoval v perióde $i = 39$, zatiaľ čo pri Naivnej stratégii už v perióde $i = 15$. Zaujímavá situácia nastala v perióde $i = 64$, kde pri oboch stratégiách došlo k nákupu, ale v prípade Regresnej obchodovacej stratégie sa to podarilo za nižšiu cenu.



Obr. 9: Cena zobchodovanej elektriny do času T pre Regresnú obchodovaciu stratégiu

Na záver sme porovnali efektívnosť Benchmarkovej stratégie a Regresnej obchodovacej stratégie počas celého júna 2023. V Tab. 11 uvádzame priemerné denné zisky a podiel dní, v ktorých jednotlivé stratégie viedli k najväčšiemu zisku. Môžeme si všimnúť, že Regresnej obchodovacej stratégii sa darilo lepšie ako Benchmarkovej stratégii, pričom jej priemerný zisk bol väčší o 20.1%. Regresná obchodovacia stratégia dosiahla väčší zisk v 96.7% dní v danom mesiaci a vo zvyšných 3.3% dosiahla väčší zisk Benchmarková stratégia. Ako sme na začiatku kapitoly 4 spomínali, nadviazali sme na diplomovú prácu [5]. V nej finálny model dosiahol počas júna 2023 priemerný zisk 637.12€, tj. Regresná obchodovacia stratégia dosiahla o 12.9% väčší priemerný zisk.

Stratégia	**Priemerný zisk**	**Podiel víťazstiev (%)**
Regresná obchodovacia stratégia	718.99€	96.7
Benchmarková stratégia	598.76€	3.3

Tabuľka 11: Porovnanie zisku a úspešnosti jednotlivých stratégií.

Záver

V tejto diplomovej práci sme sa zamerali na skúmanie a aplikáciu aproximačných techník pri riešení úloh stochastického optimálneho riadenia. Hlavnou motiváciou pre využitie týchto metód je tzv. problém dimenzionality, ktorý spôsobuje, že priame riešenie pomocou RDP je výpočtovo aj pamäťovo náročné, respektíve v reálnom čase nerealizovateľné. V komplexných praktických úlohách s rozsiahlym stavovým priestorom, viacerými riadiacimi premennými a stochastickým vývojom systému predstavujú aproximačné techniky efektívny nástroj na získanie kvalitnej a robustnej stratégie riadenia. V práci sme sa bližšie venovali dvom vybraným prístupom: aproximácii hodnotovej funkcie pomocou lineárnej kombinácie bázických funkcií a metóde MPC.

Efektívnosť vybraných techník sme testovali na dvoch vybraných problémoch. Prvou skúmanou úlohou bol inkrementálny problém batoha. Tento problém predstavoval vhodný modelový príklad: pri relatívne jednoduchom zadaní viedol k rýchlemu rastu stavového aj riadiaceho priestoru. Pre túto úlohu sme navrhli aproximačný algoritmus založený na regresnom odhade hodnotovej funkcie. Získanú regresnú stratégiu sme porovnali so stratégiou nájdenou pomocou MPC prístupu. Ako benchmark sme použili Naivnú stratégiu, pri ktorej sa do batoha vkladá vždy najhodnotnejší predmet, pridaním ktorého sa neprekročí aktuálna kapacita batoha. Numerické experimenty ukázali, že aproximačné metódy dokážu významne prekonať Naivnú stratégiu. Pri porovnaní na väčšom súbore simulácií dosiahla najvyšší priemerný zisk MPC stratégia, konkrétne 2756.984 p.j., zatiaľ čo Regresná stratégia dosiahla priemerný zisk 2648.522 p.j. a Naivná stratégia 2429.592 p.j. MPC stratégia vykázala najväčší zisk v 63.2% prípadov, Regresná stratégia v 35.4% prípadov a Naivná stratégia iba v 1.2% prípadov.

Z výsledkov zároveň vyplynulo, že najvyššie okamžité zaplnenie kapacity nemusí viesť k najlepšiemu celkovému výsledku. Naivná stratégia síce často rýchlo zapĺňala batoh v počiatočných periódach, no tým si znižovala možnosti budúceho rozhodovania. Regresná stratégia a MPC stratégia dokázali lepšie zohľadniť budúci vývoj systému a v niektorých prípadoch si ponechať kapacitu na neskoršie vloženie hodnotnejších predmetov. Tento výsledok potvrdzuje význam počítania s budúcimi výnosmi pri sekvenčných rozhodovacích úlohách.

Druhý skúmaný problém bol inšpirovaný problémami súčasnej praxe v oblasti ener-

getiky. Narastajúci podiel obnoviteľných zdrojov pri výrobe elektrickej energie so sebou prináša zvýšený dopyt po technológiách na uskladnenie elektriny. Potreba efektívne reagovať na výkyvy v sieti podnietila nielen rozvoj batériových úložísk , ale vynútila si otvorenie tzv. vnútrodenného trhu, na ktorom je obchodovanie s elektrickou energiou možné uskutočniť ešte hodinu pred dodaním elektrickej energie. V tejto diplomovej práci sme nadviazali na prácu [5]. Problém obchodovania s batériou na vnútrodennom trhu sme sformulovali ako problém stochastického optimálneho riadenia, v ktorom rozhodnutia o nákupe a predaji elektriny závisia od aktuálneho stavu batérie, technických obmedzení a vývoja cien elektriny. Vzhľadom na vysoký rozmer stavového priestoru, najmä v dôsledku cenových premenných a obchodovania v 96 pätnásťminútových intervaloch, nebolo možné pristúpiť k presnému riešeniu pomocou dynamického programovania. Preto sme pri riešení tejto úlohy skúmali použitie aproximačných techník. Získanú Regresnú obchodovaciu stratégiu sme porovnali s Benchmarkovou stratégiou, pri ktorej sa obchodovacia stratégia riadi len okamžitým ziskom.

Numerické výsledky pre obchodovanie s batériou ukázali, že navrhnutá Regresná obchodovacia stratégia dosiahla lepšie výsledky ako Benchmarková stratégia. Na testovaných dátach (júna 2023) sa pri použití Regresnej obchodovacej stratégie dosiahol priemerný denný zisk 718.99€, zatiaľ čo Benchmarková stratégia dosiahla zisk len 598.76€. Regresná obchodovacia stratégia tak dosiahla približne o 20.1% vyšší priemerný zisk a bola úspešnejšia v 96.7% sledovaných dní. Zároveň prekonala aj referenčný model z diplomovej práce [5], ktorého priemerný zisk počas rovnakého testovacieho obdobia bol 637.12€.

Uvedené výsledky potvrdzujú, že aproximačné metódy predstavujú vhodný nástroj na riešenie rozsiahlych stochastických úloh optimálneho riadenia. V oboch skúmaných aplikáciách sa ukázalo, že stratégie zohľadňujúce budúcu hodnotu rozhodnutí dosahujú lepšie výsledky ako jednoduché naivné stratégie. Práca tak potvrdila, že aj relatívne jednoduché aproximačné modely hodnotovej funkcie môžu viesť k prakticky použiteľným stratégiám, ktoré zlepšujú kvalitu rozhodovania pri zachovaní prijateľnej výpočtovej náročnosti. Práca však súčasne demonštruje, že implementácia teoretických konceptov do praxe si vyžaduje prekonávanie špecifických metodických výziev vzhľadom na špecifiká konkrétnej úlohy.

Medzi obmedzenia práce patrí skutočnosť, že použité aproximačné metódy neposkytujú všeobecnú záruku optimálnosti a ich kvalita závisí od voľby bázických funkcií, ge-

nerovania stavov a nastavenia modelových parametrov. Priestor na ďalší výskum preto vidíme najmä v rozšírení množiny bázických funkcií, použití nelineárnych aproximačných metód, podrobnejšom modelovaní cien elektriny a testovaní stratégií na väčšom množstve historických dát.

Literatúra

[1] Esra Bas. A capital budgeting problem for preventing workplace mobbing by using analytic hierarchy process and fuzzy 0–1 bidimensional knapsack model. *Expert Systems with Applications*, 38(10):12415–12422, 2011.

[2] Daniel Bienstock, Jay Sethuraman, and Chun Ye. Approximation algorithms for the incremental knapsack problem via disjunctive programming. *arXiv preprint arXiv:1311.4563*, 2013.

[3] Alvaro Cartea and Marcelo G Figueroa. Pricing in electricity markets: a mean reverting jump diffusion model with seasonality. *Applied Mathematical Finance*, 12(4):313–335, 2005.

[4] Yuri Faenza, Danny Segev, and Lingyi Zhang. Approximation algorithms for the generalized incremental knapsack problem. *Mathematical Programming*, 198(1):27–83, 2023.

[5] Matej Farský. Optimálne riadenie flexibility na krátkodobom trhu s elektrickou energiou. Diplomová práca, Univerzita Komenského v Bratislave, Bratislava, 2025. Dostupné online.

[6] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[7] Tobias Geyer and Daniel E Quevedo. Multistep finite control set model predictive control for power electronics. *IEEE Transactions on power electronics*, 29(12):6836–6846, 2014.

[8] Margaréta Halická, Pavel Brunovskỳ, and Pavol Jurča. Optimálne riadenie. 2009.

[9] Ursula Porod. Dynamics of markov chains for undergraduates. *Preprint: https://www. math. northwestern. edu/documents/book-markov-chains. pdf*, 2021.

[10] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

[11] Jacques Richalet, André Rault, Jean-Louis Testud, and Jean Papon. Model predictive heuristic control. *Automatica (journal of IFAC)*, 14(5):413–428, 1978.

A Príloha A

Formulácia modelu obchodovania s batériou

$$\max \sum_{t=0}^{T-1} \sum_{i=0}^{m-1} p_{t,i}^{bid}\, v_{t,i} - p_{t,i}^{ask}\, u_{t,i} + K(\mathbf{y_T^{(2)}}, \mathbf{p_T^{(1)}}, \mathbf{p_T^{(2)}})$$

$$y_{t,i+1}^{(1)} = y_{t,i}^{(1)} + u_{t,i} - v_{t,i}$$

$$y_{t,i+1}^{(2)} = y_{t,i}^{(2)} + y_{t,i}^{(1)} + u_{t,i} - v_{t,i}$$

$$y_{t,i+1}^{(3)} = y_{t,i}^{(3)} + \left| y_{t,i}^{(1)} + u_{t,i} - v_{t,i} \right|$$

$$\left| y_{t,i}^{(1)} \right| \leq c \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$y_{0,i}^{(1)} = 0 \qquad \text{pre} \quad i = 0, \ldots, m-1$$

$$L_y \leq y_{t,i}^{(2)} \leq U_y \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$y_{t,m}^{(3)} \leq 2g(U_y - L_y) \qquad \text{pre} \quad t = 0, \ldots, T-1$$

$$y_{t,0}^{(2)} = a_0,\ y_{t,m}^{(2)} = a_m,\ y_{t,0}^{(3)} = 0 \qquad \text{pre} \quad t = 0, \ldots, T-1$$

$$p_{t+1,i}^{bid} = f(p_{t,i}^{bid}, z_{t,i}) \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$p_{t+1,i}^{ask} = f(p_{t,i}^{ask}, z_{t,i}) \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$p_{t+1,i}^{(1)} = \begin{cases} p_{t,i}^{(1)}, & \text{ak } y_{t,i}^{(1)} > 0, \\ p_{t,i}^{ask}, & \text{ak } u_{t,i} > 0, \\ 0, & \text{inak}, \end{cases} \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$p_{t+1,i}^{(2)} = \begin{cases} p_{t,i}^{(2)}, & \text{ak } y_{t,i}^{(1)} < 0, \\ p_{t,i}^{bid}, & \text{ak } v_{t,i} > 0, \\ 0, & \text{inak}, \end{cases} \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$p_{0,i}^{(1)} = 0,\ p_{0,i}^{(2)} = 0 \qquad \text{pre} \quad i = 0, \ldots, m-1$$

$$u_{t,i} \in [0, 2c] \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$v_{t,i} \in [0, 2c] \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$u_{t,i} \leq 2c \cdot r_{t,i}^{a} w_{t,i} \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$v_{t,i} \leq 2c \cdot r_{t,i}^{b} (1 - w_{t,i}) \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1$$

$$w_{t,i} \in \{0, 1\} \qquad \text{pre} \quad t = 0, \ldots, T-1, \quad i = 0, \ldots, m-1,$$

kde $K(\mathbf{y_T^{(2)}}, \mathbf{p_T^{(1)}}, \mathbf{p_T^{(2)}}) = -5y_{T,m}^{(2)} + \sum_{i=0}^{m-1} -p_{T,i}^{(1)} \left(\frac{1}{\alpha} - 1\right) - p_{T,i}^{(2)}(1 - \beta)$.

Algoritmy pri MPC modeli

V Algoritme 5 budeme brať do úvahy situáciu, keď $y^{(2)}_{t+1} \neq y^{(2)}_{t+2} = y^{(2)}_{t+3}$. Výplata $f^0(\mathbf{y^{(1)}_{t+1}} + \mathbf{u_{t+1}}) = f^0(\mathbf{y^{(1)}_{t+2}} + \mathbf{u_{t+2}})$, čo znamená, že ich súčet si vieme prepísať na $2f(\mathbf{y^{(1)}_{t+1}} + \mathbf{u_{t+1}})$.

Algoritmus 5 Výpočet $\mathbf{u_{t+1}}$

1: **for** $y^{(1)}_t \in \{0, 1, \ldots, c_t - 1\}$, $m^1_t \in \{y^{(2)}_t, y^{(2)}_t + 1, \ldots, c_t\}$ a $m^1_{t+1} \in \{c_t + 1, c_t + 2, \ldots, c_{t+1}\}$ **do**

2: Inicializujeme $s = m^1_t$, $u_{t+1,i} = 0$, $\forall i = 1, \ldots, n$,
$\mathcal{G}_t = \{i \in \{1, \ldots, n\} \mid y^{(1)}_{t+1,i} = 0\}$ a $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m^1_{t+1}\}$

3: **while** $\mathcal{F}_t \neq \emptyset$ **do**

4: $i^* = \arg\max_{i \in \mathcal{F}_t} \left[2f^0(\mathbf{y^{(1)}_{t+1}} + \mathbf{u_{t+1}}) + \mathbb{E}[\bar{V}_{t+3}(x_{t+3})]\right]$

5: $u_{t+1,i^*} = 1$

6: $s = s + w_{i^*}$

7: $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$

8: $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m^1_{t+1}\}$

9: **end while**

10: **return** $\mathbf{u_{t+1,i}}$

11: **end for**

V Algoritme 6 budeme brať do úvahy situáciu, keď $y^{(2)}_{t+1} = y^{(2)}_{t+2} \neq y^{(2)}_{t+3}$.

Algoritmus 6 Výpočet $\mathbf{u_{t+2}}$

for $y_t^{(1)} \in \{0, 1, \ldots, c_t - 1\}$, $m_t^1 \in \{y_t^{(2)}, y_t^{(2)} + 1, \ldots, c_t\}$ a $m_{t+2}^1 \in \{c_{t+1} + 1, c_{t+1} + 2, \ldots, c_{t+2}\}$ **do**

  Inicializujeme $s = m_t^1$, $u_{t+2,i} = 0$, $\forall i = 1, \ldots, n$,

  $\mathcal{G}_t = \{i \in \{1, \ldots, n\} \mid y_{t+2,i}^{(1)} = 0\}$ a $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m_{t+2}^1\}$

  while $\mathcal{F}_t \neq \emptyset$ **do**

    $i^* = \arg\max_{i \in \mathcal{F}_t} \left[f^0(\mathbf{y_{t+2}^{(1)}} + \mathbf{u_{t+2}}) + \mathbb{E}[\bar{V}_{t+3}(x_{t+3})] \right]$

    $u_{t+2,i^*} = 1$

    $s = s + w_{i^*}$

    $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$

    $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m_{t+2}^1\}$

  end while

  return $\mathbf{u_{t+2}}$

end for

V Algoritme 7 budeme brať do úvahy situáciu, keď $y_{t+1}^{(2)} \neq y_{t+2}^{(2)} \neq y_{t+3}^{(2)}$.

Algoritmus 7 Výpočet $\mathbf{u_{t+2}}$

for $y_t^{(1)} \in \{0, 1, \ldots, c_t - 1\}$, $m_t^1 \in \{y_t^{(2)}, y_t^{(2)} + 1, \ldots, c_t\}$, $m_{t+1}^1 \in \{c_t + 1, c_t + 2, \ldots, c_{t+1}\}$ a $m_{t+2}^1 \in \{c_{t+1} + 1, c_{t+1} + 2, \ldots, c_{t+2}\}$ **do**

  Inicializujeme $s = m_{t+1}^1$, $u_{t+2,i} = 0$, $\forall i = 1, \ldots, n$,

  $\mathcal{G}_t = \{i \in \{1, \ldots, n\} \mid y_{t+2,i}^{(1)} = 0\}$ a $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m_{t+2}^1\}$

  while $\mathcal{F}_t \neq \emptyset$ **do**

    $i^* = \arg\max_{i \in \mathcal{F}_t} \left[f^0(\mathbf{y_{t+2}^{(1)}} + \mathbf{u_{t+2}}) + \mathbb{E}[\bar{V}_{t+3}(x_{t+3})] \right]$

    $u_{t+2,i^*} = 1$

    $s = s + w_{i^*}$

    $\mathcal{G}_t = \mathcal{G}_t \setminus \{i^*\}$

    $\mathcal{F}_t = \{i \in \mathcal{G}_t \mid s + w_i \leq m_{t+2}^1\}$

  end while

  return $\mathbf{u_{t+2,i}}$

end for