Fakulta Matematiky, Fyziky a Informatiky
Univerzity Komenského
v Bratislave

Diplomová Práca

Bratislava 2010

Bc. Marian Šimko

UNIVERZITA KOMENSKÉHO V BRATISLAVE
Fakulta matematiky, fyziky a informatiky

Stochastické Úlohy Optimálneho Riadenia

Diplomová práca

Študijný odbor: 9.1.9 Aplikovaná matematika
Študijný program: Ekonomická a finančná matematika

Vedúci diplomovej práce: Doc. RNDr. Margaréta Halická, CSc.

Bratislava 2010

Bc. Marian Šimko

Čestné prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím odbornej literatúry a ďalších zdrojov, ktoré uvádzam v zozname použitej literatúre.

Bratislava 04.08.2010 . . . .

Vlastnoručný podpis

Poďakovanie

Ďakujem svojej vedúcej diplomovej práce, doc. RNDr. Margaréte Halickej, CSc. za podnetné pripomienky k práci a odborné vedenie, za trpezlivosť, venovaný čas a podržanie v neľahkej situácii. Vďaka patrí aj mojej rodine a blízkym, za neustálu podporu a povzbudenia. Zároveň ďakujem Pánu Bohu za posilnenie a pomoc, najmä v rozhodujúcich okamihoch.

Autor

Abstrakt

V diplomovej práci sa zaoberáme stochastickými úlohami optimálneho riadenia. V stručnosti približujeme teóriu, ktorej súčasťou je aj rovnica dynamického programovania. Uvádzame jej algoritmus a možnosti jej využitia pri riešení stochastických úloh optimálneho riadenia. V troch kapitolách sa venujeme úlohám z oblasti ekonómie a financií. Postupne riešime spravovanie hotovosti v bankomate, maximalizovanie zisku obchodníka, až v poslednej kapitole, úlohu o hazardnom hráčovi. Každú z úloh sme sformulovali a spísali motiváciu k jej zostaveniu. Následne uvádzame výsledky riešenia v programe Matlab. Na záver práce pripájame prílohu, kde je možné nájsť príslušné kódy programov.

Kľúčové slová

Diskrétne optimálne riadenie, stochastické úlohy, rovnica dynamického programovania, spravovanie hotovosti v bankomate

Abstract

Diploma thesis deals with the stochastic optimal control problems. Shortly, approaching the theory, which includes the equation of dynamic programming. Here's the algorithm and the possibility of its use in solving the stochastic optimal control problems. In three chapters we discuss problems of economics and finance. In sequence we solve cash management in ATMs, maximize the profit of the seller, until the last chapter on the role of gambler. Each of these problems we have formulated and described motivation for the compilation. Subsequently, we present results of solution in the program Matlab. Finally, align ourselves work attachment, where you can find corresponding program codes.

Key words

Discrete optimal control, stochastic problems, dynamic programming equation, cash management in ATM

Obsah

Úvod

V diplomovej práci stručne priblížime teóriu optimálneho riadenia, konkrétne stochastické dynamické programovanie. Zaoberáme sa diskrétnymi stochastickými úlohami. Cieľom bolo nájsť, prípadne zostaviť problémy z oblasti ekonómie a financií, sformulovať ich ako úlohy optimálneho riadenia a ukázať, ako možno tieto úlohy riešiť pomocou rovnice dynamického programovania (RDP). Spomíname zároveň algoritmus pre RDP, pre názornejšie predstavenie RDP ako vhodného nástroja na riešenie takýchto úloh. Sformulované úlohy sme naprogramovali, poslúžil nám matematický softvér Matlab. Pri práci sme využili odbornú literatúru a materiály, pomocou nám bol aj *Help*, ktorý je súčasťou samotného Matlabu.

Postupne sme sa zaoberali troma úlohami, ktoré sú spracované v 2. až 4. kapitole. Ťažiskom diplomovej práce je druhá kapitola, kde sa venujeme spravovaniu hotovosti v bankomate. Inšpiráciou nám bol článok [1], v ktorom sa hlbšie zaoberajú touto problematikou. Nami sformulovaná úloha je zaujímavá neštandardnými ohraničeniami, kedy v nich vystupuje aj náhodná premenná. Venujeme sa autonómnej aj neautonómnej verzii, na záver kapitoly uvádzame prípad, kedy neuspokojený dopyt po výberoch vedie k strate pre banku. Obsahom ďalšej kapitoly je úloha o maximalizovaní zisku obchodníka. V tejto kapitole už nevyžadujeme uspokojený dopyt, ako je tomu v základnej formulácii úlohy o bankomate, pričom táto skutočnosť nevedie k strate. V poslednej kapitole sa venujeme modifikáciám úlohy o hazardnom hráčovi. Postupne sme riešili prípady, kedy je stávka celočíselná, pre dve rôzne účelové funkcie a kedy máme možnosť vsadiť pomerovú časť kapitálu. Na tomto mieste sme využili diskretizáciu stavovej, respektíve riadiacej premennej. Konečným cieľom riešenia týchto úloh je určenie optimálnej spätnej väzby. Pri riešení úloh sme v práci volili jednoduché vstupy. Výsledky sme komentovali, interpretovali, ponúkli aj grafické riešenie.

Úlohy, ktoré sú obsahom práce, sú jednoduchými modelmi situácií, s ktorými sa môžeme stretnúť v praxi. Cieľom teda nebolo riešiť konkrétny problém aj numericky, skôr iba poskytnúť predstavu o možnostiach riešenia v zmysle teórie stochastického optimálneho riadenia. V práci sa vyskytuje pomerne veľké množstvo obrázkov a tabuliek, majú slúžiť najmä k lepšiemu porozumeniu riešení a k vyvodeniu základných vzťahov, myšlienok pre danú úlohu.

Kapitola 1

Teoretický úvod

Na začiatok uvedieme základné pojmy, s ktorými operujeme v diplomovej práci. Zároveň stručne priblížime problematiku stochastického optimálneho riadenia. Hlavným vodítkom pre spísanie teórie nám bola literatúra [2]. Zaoberáme sa diskrétnymi úlohami optimálneho riadenia s pevným časom. Podrobnejšie spracovanú teóriu možno nájsť okrem knihy [2] aj v literatúre [3] - [6].

V druhej podkapitole spravíme základný popis algoritmu pre rovnicu dynamického programovania a programov vytvorených v Matlabe. Príslušné kódy uvádzame v prílohe. Informácie umožňujúce zvládnuť elementárnu ako aj pokročilejšiu prácu v Matlabe možno nájsť napríklad v [7] - [15].

1.1 Stochastické úlohy optimálneho riadenia

Predpokladajme, že máme systém, ktorého správanie budeme riadiť v priebehu T etáp. Stav systému na začiatku i-tej etapy, $i = 0, \ldots, T-1$, je opísaný stavovou premennou x_i. Správanie systému riadime pomocou riadiacej premennej $u_i \in U_i$, ktorá je vstupom do systému. V našom prípade pôsobia na systém aj náhodné vonkajšie vplyvy, ktorých hodnoty vopred presne nepoznáme. Z toho plynie názov, stochastické úlohy. Tieto vplyvy zachytáva náhodná premenná $z \in Z_i$. Množiny U_i a Z_i sú množinami povolených hodnôt pre spomenuté premenné.

Hodnotami x_i, u_i a realizáciami náhodnej premennej z_i je jednoznačne určená hodnota $x_{i+1} = f(x_i, u_i, z_i)$, kde f_i je daná funkcia. Výnos v i-tej etape je určený hodnotou $f_i^0(x_i, u_i, z_i)$, kde f_i^0 je daná funkcia. Predpokladajme, že na počiatku je hodnota stavovej premennej x_0 rovná zadanej hodnote a. Pracujeme bez reálnych ohraničení ($x_i \in X_i$) na stavovú premennú, rovnako predpokladáme, že aj koncový stav x_T je vždy voľný. To znamená, že pre každé $x_i \in X_i$, $u_i \in U_i$ a $z \in Z_i$ je $x_{i+1} = f(x_i, u_i, z_i) \in X_{i+1}$.

Úlohou je určiť v každej etape hodnotu riadiacej premennej u_i tak, aby boli splnené všetky podmienky a aby očakávaný súčet výnosov za všetky etapy bol maximálny. Poznamenajme, že na to, aby sme vedeli počítať strednú hodnotu a riešiť tak úlohu, potrebujeme poznať pravdepodobnostné rozdelenie premennej z_i.

Štandardná stochastická úloha optimálneho riadenia:

$$\begin{aligned}
\text{maximalizovať} \quad & E\left[\sum_{i=0}^{T-1} f_i^0(x_i, u_i, z_i)\right] && (1.1)\\
\text{pri podmienkach} \quad & x_{i+1} = f_i(x_i, u_i, z_i), \quad i = 0, \dots, T-1, && (1.2)\\
& x_0 = a, && (1.3)\\
& u_i \in U_i, \quad i = 0, \dots, T-1, && (1.4)\\
& z_i \in Z_i, \quad i = 0, \dots, T-1. && (1.5)
\end{aligned}$$

Maximum hľadáme vzhľadom na premenné u_i a x_i, avšak stavová premenná je určená premennou u_i a realizáciou náhodnej premennej z_i, ako riešenie (1.2) a (1.3). Riadením budeme nazývať postupnosť $\mathcal{U} = \{u_0, \dots, u_{T-1}\}$ spĺňajúcu $u_i \in U_i$ pre každé $i = 0, \dots, T-1$. Označme X_i množinu všetkých možných realizácií hodnôt odozvy v čase i.

Potrebné je ešte určiť, akým spôsobom budeme voliť riadenie. Rozumné je voliť ho ako postupnosť spätných väzieb, v každej etape $u_i = v_i(x_i)$, kde v_i je funkcia priraďujúca každej hodnote stavu hodnotu optimálneho riadenia. Názov spätná väzba je odôvodnený tým, že funkcia v_i určuje, ako musí vstup reagovať na jeho okamžitý stav, aby ho udržal v optimálnom režime. Takéto riadenie sa v teórii stochastického dynamického programovania nazýva stratégia $\mathcal{V} = \{v_0, \ldots, v_{T-1}\}$. V našom prípade je každá stratégia prípustná, keďže nemáme ohraničenia na riadiacu premennú.

Označme si postupnosť realizácií náhodných premenných $\mathcal{Z} = \{z_0, \ldots, z_{T-1}\}$. V tomto prípade je $\mathcal{Z}$ vlastne viacrozmerná náhodná premenná. Je zrejmé, že pre danú stratégiu $\mathcal{V}$ a danú realizáciu náhodnej premennej $\mathcal{Z}$ je jednoznačne určená hodnota účelovej funkcie

$$J(\mathcal{V}, \mathcal{Z}) := \sum_{i=0}^{T-1} f_i^0(x_i, v_i(x_i), z_i), \tag{1.6}$$

$$\text{kde} \quad x_{i+1} = f_i(x_i, v_i(x), z_i), \quad i = 0, \ldots, T-1, \tag{1.7}$$

$$x_0 = a. \tag{1.8}$$

Optimálnu stratégiu definujeme ako takú prípustnú stratégiu, ktorá maximalizuje strednú hodnotu $J(\mathcal{V}, \mathcal{Z})$ v triede všetkých prípustných stratégií. Označme $\mathcal{S}$ triedu prípustných stratégií, potom

$$\max_{\mathcal{V} \in \mathcal{S}} E\left[J(\mathcal{V}, \mathcal{Z})\right]. \tag{1.9}$$

1.1.1 Rovnica dynamického programovania

Rovnica dynamického programovania je vhodný nástroj na riešenie aj stochastických úloh optimálneho riadenia.

Označme pre každé $j \in \{0, \ldots, T-1\}$ a $x \in X_j$: $\mathcal{Z}_j = \{z_j, \ldots, z_{T-1}\}, \mathcal{V}_j = \{v_j, \ldots, v_{T-1}\}$ a tiež

$$J_j(x, \mathcal{V}_j, \mathcal{Z}_j) := \sum_{i=j}^{T-1} f_i^0(x_i, v_i(x_i), z_i), \tag{1.10}$$

$$\text{kde} \quad x_{i+1} = f_i(x_i, v_i(x_i), z_i) \quad i = j, \ldots, T-1, \tag{1.11}$$

$$x_j = x. \tag{1.12}$$

Potom pod úlohou $\mathcal{D}_j(x)$ budeme rozumieť úlohu

$$\text{maximalizovať} \quad E\left[J_j(x, \mathcal{V}_j, \mathcal{Z}_j)\right]. \tag{1.13}$$

V tejto chvíli je užitočné sformulovať nasledovné dva predpoklady, ktoré ďalej využívame. Predpoklad 1.1 potrebujeme, aby sme mohli definovať hodnotovú funkciu.

Predpoklad 1.1. *[2] Pre každé $j \in [0, T-1]$ existuje stratégia $\mathcal{V}_j$ optimálna pre každú úlohu $D_j(x)$, kde $x \in X_j$. To znamená, existuje $\hat{\mathcal{V}}_j$ také, že*

$$\max_{\mathcal{V}_j} E_{j,T-1}\left[J_j(x, \mathcal{V}_j, \mathcal{Z}_j)\right] = E_{j,T-1}\left[J_j(x, \hat{\mathcal{V}}_j, \mathcal{Z}_j)\right] \tag{1.14}$$

pre každé $x \in X_j$.

Predpoklad 1.2. *[2] Množiny Z_i sú konečné alebo aspoň spočítateľné, teda $\mathcal{Z}_j$ je diskrétnou viacrozmernou náhodnou premennou.*

Definícia 1.1. *[2]Hodnotová funkcia.*
Pre každé $j \in [0, T-1]$ definujeme $V_j : X_j \to \mathbb{R}$ predpisom

$$V_j(x) = \max_{\mathcal{V}_j} E_{j,T-1}\left[J_j(x, \mathcal{V}_j, Z_j)\right]. \tag{1.15}$$

Funkciu V_j nazveme hodnotovou funkciou pre systém úloh $D_j = \{D_j(x) : x \in X_j\}$ a postupnosť funkcií $V = \{V_0, \ldots, V_{T-1}\}$ hodnotovou funkciou pre systém úloh $\mathcal{D}$.

Veta 1.1. *[2]Rovnica dynamického programovania.*
Nech úloha (1.9) spĺňa predpoklady 1.1 a 1.2.

(i) Ak $\hat{\mathcal{V}} = \{\hat{v}_0, \ldots, \hat{v}_{T-1}\}$ je optimálna stratégia a $V = \{V_0, \ldots, V_{T-1}\}$ hodnotová funkcia, potom funkcie V_j, $\hat{v}_j$, $j = 0, \ldots, T-1$ spĺňajú pre každé $j = 0, \ldots, T-1$ a x rovnicu dynamického programovania:

$$V_j(x) = \max_{u \in U_i} E_j\left[f_j^0(x, u, z_j) + V_{j+1}(f_j(x, u, z_j))\right] \tag{1.16}$$

$$= E_j\left[f_j^0(x, \hat{v}_j(x), z_j) + V_{j+1}(f_j(x, \hat{v}_j(x), z_j))\right], \tag{1.17}$$

$$V_T(x) = 0, \textit{pre každé } x \tag{1.18}$$

(ii) Naopak, ak pre každé $j = 0, \ldots, T-1$ a x funkcie $\hat{v}_j$ a V_j spĺňajú (1.17) a (1.18) potom $V = \{V_0, \ldots, V_{T-1}\}$ je hodnotová funkcia a $\hat{\mathcal{V}} = \{\hat{v}_0, \ldots, \hat{v}_{T-1}\}$ je optimálna stratégia.

Poznamenajme, že pre optimálne úlohy v neštandardnom tvare sa rovnica dynamického programovania mierne zmení. V práci sa zaoberáme aj úlohami, kedy je účelová funkcia v Mayerovom tvare[1], to znamená iba funkcia koncového stavu. Označme ju φ, teda $J(\mathcal{V}, \mathcal{Z}) = E\left[\varphi(x)\right]$. Ak máme na riadiacu premennú ďalšie ohraničenia, zmení sa aj množina prípustných hodnôt riadení, označme ju $\Gamma(x)$. Potom

$$\begin{aligned} V_j(x) &= \max_{u \in \Gamma_j(x)} E[V_{j+1} f_j(x, u, z_j)], \quad j = 0, \ldots, T-1, & (1.19) \\ V_T(x) &= \varphi(x_T). & (1.20) \end{aligned}$$

Rovnica dynamického programovania ponúka istý rekurzívny vzťah na riešenie úloh optimálneho riadenia, ako je to možné vidieť v (1.16). Konečným cieľom riešenia úlohy optimálneho riadenia je určenie optimálnej spätnej väzby.

1.2 Algoritmus

Na tomto mieste uvedieme a popíšeme všeobecný algoritmus rovnice dynamického programovania pre stochastické úlohy optimálneho riadenia. Zároveň sa stručne oboznámime so základnými princípmi našich programov a s označením premenných, ktoré v nich vystupujú.

Už vyššie sme pracovali s pojmom stredná hodnota, v tejto časti bude užitočným si ju zadefinovať. Strednú hodnotu niekedy nazývame aj očakávanou hodnotou náhodnej premennej. Pripomeňme, že sa zaoberáme diskrétnymi úlohami.

[1] Štandardná úloha optimálneho riadenia má Lagrangeov tvar účelovej funkcie.

Definícia 1.2. *Stredná hodnota $E(Y)$ diskrétnej náhodnej premennej Y, sa rovná súčtu súčinov hodnôt tejto premennej a pravdepodobností ich realizácie:*

$$E(Y) = \sum_{i=1}^{n} y_i P(Y = y_i). \tag{1.21}$$

Prejdime k spoločným označeniam pre premenné a parametre v programoch:
T - označuje počet etáp, dĺžku obdobia alebo počet kôl hry,
x - stavová premenná,
u - riadiaca premenná,
z - náhodná premenná vystupujúca v systéme, v našom prípade objekt, kde $z.h$ sú možné hodnoty a $z.p$ pravdepodobnosti realizácie týchto hodnôt,
f - hodnota stavovej rovnice,
$f0$ - hodnota účelovej funkcie,
V - matica hodnôt hodnotovej funkcie,
v - matica hodnôt optimálnej spätnej väzby,
$xmin, xmax$ - minimálna, (max.) hodnota, ktorú môže dosiahnuť stavová premenná,
$umin, umax$ - maximálna hodnota riadiacej premennej.

Poznamenajme, že hoci matica V má rozmery $(T + 1) \times (xmax + 1)$ a matica v rozmery $(T) \times (xmax + 1)$, uchovávajú informácie pre všetky $t = 0, \ldots, T$ a $x = 0, \ldots, xmax$, resp. $t = 0, \ldots, T - 1$ a $x = 0, \ldots, xmax$.

Všeobecný algoritmus pre rovnicu dynamického programovania.

```
dané vstupné parametre:
T, xmax, xmin, umax, z.h, z.p, ...

nie je nutné matice vytvoriť takýmto spôsobom, ale ozrejmíme si ich rozmery
V=zeros(T+1,xmax+1); v=zeros(T,xmax+1);

V(:,:)=-inf; v prípade hľadania maximálnej hodnoty
```

```
V(:,:)=inf; v prípade hľadania minimálnej hodnoty
V(T+1,:)=0; voľný koniec, prípadne funkcia koncového stavu

for i=T:(-1):1
    for x=0:xmax
        for u=0:umax
            f = f(x,u,z.h); % stavová rovnica
            f0 = f0(x,u,z.h); % účelová funkcia
            if do úvahy prichádzajú prípadné ohraničenia na stavovú premennú

                hľadám maximum / minimum, vtedy "<"
                if (f0 + V(i+1,f+1))*z.p' > V(i,x+1)
                    V(i,x+1) = (f0 + V(i+1,f+1))*z.p';
                    v(i,x+1) = u;
                end

            end
        end
    end
end

% optimálna hodnota
opt_hodnota=V(1,x0+1)
```

Ako vidíme, pri hľadaní riešenia začíname od času T a postupujeme smerom k času 0. Využijeme skutočnosť, že poznáme hodnoty $V_T(x)$. Postupne pre každú možnú hodnotu stavovej premennej x a riadiacej premennej u počítame stavovú rovnicu f a účelovú funkciu $f0$. Poznamenajme, že f a prípadne aj $f0$ sú riadkové vektory, kde zložkami sú hodnoty vypočítané pre jednotlivé realizácie $z.h$ náhodnej premennej z. Ďalej sme pri samotnej RDP. Vynásobením vektorom pravdepodobností $z.p$ dostávame podľa definície 1.2 strednú hodnotu. Tú porovnáme s už existujúcim záznamom v matici, výsledky sa sčítajú atď. Vždy si uložíme do matice v aj hodnotu príslušnej optimálnej spätnej väzby. Nakoniec dostávame hľadanú očakávanú optimálnu hodnotu optimálnu spätnú väzbu pre všetky možné prípady.

Poznamenajme, že programy sú pre každú úlohu zostavené všeobecne a je možné jednoducho meniť vstupné parametre. Čo sa týka pravdepodobnostného rozdelenia náhodných premenných, v úlohách uvažujeme najmä rovnomerné rozdelenie. Postačuje uviesť interval hodnôt, pravdepodobnosti sa dopočítajú automaticky. V niektorých prípadoch sme pracovali aj s približne diskretizovaným normálnym rozdelením. Využili sme vstavanú funkciu v Matlabe *normpdf*. Jej vstupmi sú vektor hodnôt, stredná hodnota μ a štandardná odchýlka σ. Následne ešte zabezpečíme, aby súčet pravdepodobností bol rovný 1.

```
z.h=[]; z.p=normpdf(z.h,,); z.p=z.p/sum(z.p)*1;
```

Normal probability density function (normpdf)

Y = normpdf(X,mu,sigma)

$$y = f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{1.22}$$

Na vykresľovanie obrázkov používame vytvorené funkcie (miestami modifikované, hlavne, čo sa týka mierky pre osi, prípadne rozsahu). Ich zdrojový kód v Matlabe nájdeme v prílohe F a G.

V poslednej úlohe v diplomovej práci sa stretneme s diskretizáciou stavovej, resp. riadiacej premennej.

Kapitola 2

Spravovanie hotovosti v bankomate

Spravovanie hotovosti je jednou z hlavných úloh banky. V tejto časti našej práce sa budeme zaoberať úlohou spravovania peňazí v bankomatoch. Inšpiráciou pre zostavenie úlohy bol článok [1]. Zostavili sme zjednodušený model situácie, naprogramovali úlohu a komentovali výsledky.

Priblížme si najskôr situáciu. V tomto prípade musia banky držať v bankomate určitý objem peňazí. Tento obnos hotovosti závisí od budúceho dopytu, ktorý nie je presne známy - je náhodný. Jeho hodnoty a pravdepodobnostné rozdelenie sa však dá približne určiť z vypozorovaných historických dát. Samotný objem dopytu je samozrejme špecifický pre jednotlivé bankomaty. Závisí napríklad od polohy daného bankomatu. S najväčšou pravdepodobnosťou sa mení aj v závislosti na jednotlivých dňoch v roku. Ak je dopyt väčší ako množstvo peňazí v bankomate (vzniká neuspokojený dopyt), môže byť banka sankciovaná. V lepšom prípade iba stráca dobré meno a dôveru zákazníka. Naopak v prípade držania výrazne väčšieho objemu peňazí bude banka opäť strácať. Dôvodom je cena stratenej príležitosti. Výhodnejšie by bolo pre ňu prostriedky investovať, prípadne umiestniť prebytočnú hotovosť na medzibankovom trhu - prinášali by jej príjem z úrokov. Spomenuté skutočnosti predstavujú pre banku isté náklady. Navyše s dovozom a dopĺňaním peňazí do bankomatov sú taktiež spojené určité náklady. V článku [1] sa uvádzajú dva prístupy, ktoré sa v praxi využívajú. Prvým je, že banky platia význačný fixný poplatok za doplnenie bankomatu, ktorý nezávisí od objemu peňazí (napr. 50 €), a extra poplatky (tie sú menšie) za objem peňazí (napr. 0,59 € za každých 10 000 €). V druhom prípade platia nižší fixný poplatok (napr. 20 €) a vyššie poplatky za určitý obnos peňazí (napr. 30 € za každých 10 000 €). Do bankomatu sa vkladajú peniaze v balíkoch o určitej sume. Jeden taký balík má napr. hodnotu 10 000 €. Banka sa rozhoduje, koľko balíkov má doplniť v daný čas. V práci nebudeme uvažovať rôzne nominálne hodnoty bankoviek, zaoberať sa budeme iba samotným objemom hotovosti.

Cieľom každej banky je rozhodnúť, aké množstvo peňazí je optimálne držať v bankomate. Teda koľko, a v ktorý deň je potrebné peniaze doplniť tak, aby bol podľa možnosti pokrytý dopyt a celkové priemerné resp. očakávané náklady boli minimálne.

2.1 Autonómna verzia úlohy

V našej práci sa zaoberáme stochastickými úlohami. Tu dopredu nepoznáme dopyt po peniazoch, presný objem výberov z bankomatu. Poznáme však jeho pravdepodobnostné rozdelenie. To znamená, rozsah hodnôt a pravdepodobnosti ich realizácie. Výbery nás zaujímajú pre jednotlivé dni za plánovacie obdobie. Na začiatok predpokladajme, že všetky dni sú povahovo rovnaké. Z hľadiska objemu výberov teda nerozlišujeme napr. medzi pracovným dňom a víkendom. Rovnako ostatné vstupné parametre sú v čase konštantné. Stavovou premennou je objem peňazí v bankomate, ktorý sa mení v závislosti na dopĺňaniach a výberoch. Riadením je počet balíkov peňazí, ktoré necháme vložiť do bankomatu. Doplniť môžeme až po kapacitu bankomatu a vždy len po celých balíkoch. Samotné doplnenie prebehne vždy na začiatku dňa, ešte pred prvými výbermi. Riadenie v každej etape budeme voliť ako optimálnu spätnú väzbu na stav hotovosti v bankomate.

Vzhľadom na to, že systém je vystavený náhodným vplyvom, pracujeme s úlohami s voľným koncom. Nezaujíma nás, koľko ostane peňazí uložených v bankomate po sledovanom období.

Označme

T	–	obdobie (predstavuje počet dní, mesiacov, ...),
x_i	–	objem peňazí v bankomate na začiatku i-teho dňa,
u_i	–	počet balíkov, ktoré do bankomatu doplníme na začiatku i-teho dňa,
z_i	–	dopyt po výberoch, peniazoch v i-ty deň, náhodná premenná,
v	–	hodnota jedného balíka peňazí, $v > 0$,
D	–	minimálny objem peňazí, ktorý musíme mať - napr. dané zákonom, $D \geq 0$,
H	–	maximálny objem peňazí, ktorý môžeme mať - technické možnosti, $H > 0$,
c	–	strata na jednotku prebytočne držanej hotovosti v bankomate na deň, $c \geq 0$,
k	–	fixný poplatok za naplnenie bankomatu, $k \geq 0$,
k_v	–	poplatok za každý vložený balík, hodnoty v jednotiek, $k_v \geq 0$,
a	–	počiatočné naplnenie bankomatu, $a \geq 0$.

Jednotlivé hodnoty môžeme napríklad uvažovať v tisícoch eur.

2.1.1 Formulácia úlohy

Teraz sformulujeme príklad ako úlohu optimálneho riadenia:

$$\min_u \quad E\left[\sum_{i=0}^{T-1} cx_i + k\mathrm{sgn}(u_i) + k_v v u_i\right] \tag{2.1}$$

$$\text{pri podm.} \quad x_{i+1} = x_i + vu_i - z_i, \quad i = 0, \ldots, T-1, \tag{2.2}$$

$$x_0 = a, \tag{2.3}$$

$$x_i + vu_i - z_i \geq D, \tag{2.4}$$

$$x_i + vu_i \leq H, \tag{2.5}$$

$$u_i \in \mathbb{N}_0. \tag{2.6}$$

Všimnime si, že úloha (2.1)-(2.6) nie je v štandardnom tvare kvôli zmiešaným ohraničeniam na stavovú a riadiacu premennú. V ohraničení (2.4) navyše vystupuje aj náhodná premenná z. Podmienkou, aby sme vedeli riešiť takúto úlohu je, že z nadobúda iba hodnoty z ohraničenej množiny. Potom môžeme ohraničenie preformulovať ako $x_i + vu_i - \max\{z\} \geq D$. Spomenuté zmiešané ohraničenia (2.4) a (2.5) zabezpečujú, aby bol uspokojený dopyt a zároveň nebola prekročená kapacita bankomatu pri dopĺňaní.

Pripomeňme, že stavová aj riadiaca premenná nadobúdajú celočíselné hodnoty. Podobne dopyt je vyjadrený v celých číslach.

2.1.2 Príklad s konkrétnymi hodnotami

Vyššie sme sformulovali všeobecne zadanú úlohu. V tejto časti zvolíme konkrétne vstupné parametre tak, aby sme úlohu vedeli aj analyticky vyriešiť. Hodnoty by sme mohli uvažovať v tisícoch, my však budeme pre jednoduchosť pracovať v jednotkách eur. Uvažujme nasledovné hodnoty. Majme krátke obdobie $T = 3$ a rovnomerne rozdelený dopyt z na intervale $[1, 3]$, v čase nemenný. Začíname s prázdnym bankomatom, teda $a = 0$, dopĺňanie prebieha v 'balíkoch' hodnoty $v = 1$ €, obmedzenia na objem peňazí v bankomate sú $D = 0$ a $H = 10$. Nepripúšťame teda neuspokojený dopyt, no vyberať je možné až na dno a horná kapacita bankomatu je 10 €. Jednotlivé náklady zvoľme, $c = 0,03$, $k = 0,2$ a $k_v = 0,05$. Inými slovami, každé prebytočné 1 € uložené cez noc v bankomate prináša stratu $0,03$ €, za doplnenie ako činnosť zaplatíme 0,2 € a za každé jedno vložené 1 € navyše manipulačný poplatok $0,05$ €. Riadeniami sú počty 'balíkov' $u \in \{0, \ldots, 10\}$ (musia byť zároveň splnené ohraničenia na stav).

Poznamenajme ešte, že zvolené hodnoty neodrážajú presne realitu. Najmä parameter c je vysoký v pomere k ostatným vstupom. V tomto prípade však ide hlavne o demonštráciu riešenia a v následnej analýze o zachytenie základných vzťahov.

Sformulovaná úloha s konkrétnymi vstupmi:

$$\begin{aligned}
&\min_{u} && E\left[\sum_{i=0}^{2} 0,03x_i + 0,2\mathrm{sgn}(u_i) + 0,05u_i\right] && (2.7)\\
&\text{pri podm.} && x_{i+1} = x_i + u_i - z_i, \quad i = 0,1,2, && (2.8)\\
& && x_0 = 0, && (2.9)\\
& && x_i + u_i \geq 3, && (2.10)\\
& && x_i + u_i \leq 10, && (2.11)\\
& && u_i \in \mathbb{N}_0, && (2.12)
\end{aligned}$$

kde

$$z_i = \begin{cases} 1 & \text{s pravdepodobnosťou } \frac{1}{3}, \\ 2 & \text{s pravdepodobnosťou } \frac{1}{3}, \\ 3 & \text{s pravdepodobnosťou } \frac{1}{3}. \end{cases} \qquad (2.13)$$

Poznáme maximálnu hodnotu, ktorú môže nadobúdať premenná z, preto uvádzame už preformulované ohraničenie (2.10).

2.1.3 Riešenie pomocou rovnice dynamického programovania

Teraz ilustrujeme výpočet pomocou rovnice dynamického programovania, definovanej (1.1). Pripomeňme, že dopyt je v každej etape rovnako rozdelený, preto pre náhodnú premennú z nebudeme písať dolný index. Nezabúdajme na to, že v každom okamihu musíme uspokojiť dopyt a zároveň pri dopĺňaní na začiatku dňa nesmieme prekročiť kapacitu bankomatu. Prípustnými hodnotami riadenia sú teda v každom čase $\Gamma(x) = \{u \in \mathbb{N}_0 | x + u \leq 10, x + u \geq 3\}$. Budeme optimalizovať vzhľadom na tieto riadenia. Počítame strednú hodnotu.

$$\begin{aligned}
V_3(x) &= 0\\
V_2(x) &= \min_{u\in\Gamma(x)} E\left[f_2^0(x,u,z) + V_3(f(x,u,z))\right] =\\
&= \min_{u\in\Gamma(x)} (0,03x + 0,2\mathrm{sgn}(u) + 0,05u)
\end{aligned}$$

Priebeh riešenia nás zvádza zvoliť v tomto kroku $\hat{u} = 0$ pre $\forall x$. Nesmieme však zabudnúť na podmienku (2.10), ktorá zabezpečuje, aby bol dopyt vždy uspokojený. Spravme teda diskusiu vzhľadom na x. Optimálne riadenie alebo lepšie optimálna spätná väzba v závislosti na stave x bude nasledovná:

$$v_2(x) = \begin{cases} 3, & \text{pre } x = 0, \\ 2, & \text{pre } x = 1, \\ 1, & \text{pre } x = 2, \\ 0, & \text{pre } x \geq 3 \end{cases}$$

a hodnotová funkcia potom

$$V_2(x) = \begin{cases} 0,35, & \text{pre } x = 0, \\ 0,33, & \text{pre } x = 1, \\ 0,31, & \text{pre } x = 2, \\ 0,03x, & \text{pre } x \geq 3. \end{cases}$$

Pokračujeme pre $t = 1$:

$$V_1(x) \quad = \quad \min_{u \in \Gamma(x)} E\left[0,03x + 0,2\text{sgn}(u) + 0,05u + V_2(x + u - z)\right].$$

Podobne, ako v predchádzajúcom prípade, aj teraz spravíme diskusiu pre stavy $x \leq 2$. Pre $x \geq 3$ je opäť optimálna spätná väzba rovná nule. Situácia je teraz mierne zložitejšia, počítajme.

$\underline{x = 0:}$

$$\begin{aligned} V_1(0) &= \min_{u \in \Gamma(x)} E\left[0,2\text{sgn}(u) + 0,05u + V_2(u - z)\right] \\ u &= 3: \quad 0,2 + 0,15 + \frac{1}{3}V_2(3-1) + \frac{1}{3}V_2(3-2) + \frac{1}{3}V_2(3-3) = 0,68, \\ u &= 4: \quad 0,2 + 0,20 + \frac{1}{3}V_2(3) + \frac{1}{3}V_2(2) + \frac{1}{3}V_2(1) = 0,643, \\ u &= 5: \quad 0,2 + 0,25 + \frac{1}{3}V_2(4) + \frac{1}{3}V_2(3) + \frac{1}{3}V_2(2) = 0,623, \\ u &= 6: \quad 0,2 + 0,30 + \frac{1}{3}V_2(5) + \frac{1}{3}V_2(4) + \frac{1}{3}V_2(3) = 0,62, \leftarrow \\ u &= 7: \quad 0,2 + 0,35 + \frac{1}{3}V_2(6) + \frac{1}{3}V_2(5) + \frac{1}{3}V_2(4) = 0,7. \end{aligned}$$

$\underline{x = 1\text{:}}$

$$\begin{aligned}
V_1(1) &= \min_{u \in \Gamma(x)} E\left[0,03 + 0,2\text{sgn}(u) + 0,05u + V_2(1 + u - z)\right] \\
u &= 2: \quad 0,03 + 0,2 + 0,1 + \frac{1}{3}V_2(2) + \frac{1}{3}V_2(1) + \frac{1}{3}V_2(0) = 0,66, \\
u &= 3: \quad 0,38 + \frac{1}{3}V_2(3) + \frac{1}{3}V_2(2) + \frac{1}{3}V_2(1) = 0,623, \\
u &= 4: \quad 0,43 + \frac{1}{3}V_2(4) + \frac{1}{3}V_2(3) + \frac{1}{3}V_2(2) = 0,603, \\
u &= 5: \quad 0,48 + \frac{1}{3}V_2(5) + \frac{1}{3}V_2(4) + \frac{1}{3}V_2(3) = 0,60, \leftarrow \\
u &= 6: \quad 0,53 + \frac{1}{3}V_2(6) + \frac{1}{3}V_2(5) + \frac{1}{3}V_2(4) = 0,68.
\end{aligned}$$

$\underline{x = 2\text{:}}$

$$\begin{aligned}
V_1(2) &= \min_{u \in \Gamma(x)} E\left[0,06 + 0,2\text{sgn}(u) + 0,05u + V_2(2 + u - z)\right] \\
u &= 1: \quad 0,06 + 0,2 + 0,05 + \frac{1}{3}V_2(2) + \frac{1}{3}V_2(1) + \frac{1}{3}V_2(0) = 0,64, \\
u &= 2: \quad 0,36 + \frac{1}{3}V_2(3) + \frac{1}{3}V_2(2) + \frac{1}{3}V_2(1) = 0,603, \\
u &= 3: \quad 0,41 + \frac{1}{3}V_2(4) + \frac{1}{3}V_2(3) + \frac{1}{3}V_2(2) = 0,583, \\
u &= 4: \quad 0,46 + \frac{1}{3}V_2(5) + \frac{1}{3}V_2(4) + \frac{1}{3}V_2(3) = 0,58, \leftarrow \\
u &= 5: \quad 0,51 + \frac{1}{3}V_2(6) + \frac{1}{3}V_2(5) + \frac{1}{3}V_2(4) = 0,66.
\end{aligned}$$

Optimálna spätná väzba na jednotlivé stavy x vyzerá nasledovne:

$$v_1(x) = \begin{cases} 6, & \text{pre } x = 0, \\ 5, & \text{pre } x = 1, \\ 4, & \text{pre } x = 2, \\ 0, & \text{pre } x \geq 3. \end{cases}$$

Zároveň jednoducho dopočítame hodnotovú funkciu:

$$V_1(x) = \begin{cases} 0,620, & \text{pre } x = 0, \\ 0,600, & \text{pre } x = 1, \\ 0,580, & \text{pre } x = 2, \\ 0,420, & \text{pre } x = 3, \\ 0,363, & \text{pre } x = 4, \\ 0,323, & \text{pre } x = 5, \\ 0,06x - 6, & \text{pre } x \geq 6. \end{cases}$$

Nakoniec pre $t = 0$:

$$V_0(x) \quad = \quad \min_{u \in \Gamma(x)} E\left[0,03x + 0,2\mathrm{sgn}(u) + 0,05u + V_1(x + u - z)\right].$$

Zopakovali by sme analogicky vyššie spomenuté výpočty, až by sme dospeli ku konečnému riešeniu. Optimálna spätná väzba:

$$v_0(x) = \begin{cases} 6, & \text{pre } x = 0, \\ 5, & \text{pre } x = 1, \\ 4, & \text{pre } x = 2, \\ 0, & \text{pre } x \geq 3. \end{cases}$$

Hodnotová funkcia:

$$V_0(x) = \begin{cases} 0,869, & \text{pre } x = 0, \\ 0,849, & \text{pre } x = 1, \\ 0,829, & \text{pre } x = 2, \\ 0,690, & \text{pre } x = 3, \\ 0,653, & \text{pre } x = 4, \\ 0,604, & \text{pre } x = 5, \\ 0,549, & \text{pre } x = 6, \\ 0,539, & \text{pre } x = 7, \\ 0,568, & \text{pre } x = 8, \\ 0,630, & \text{pre } x = 9, \\ 0,720, & \text{pre } x = 9. \end{cases}$$

Analogicky je možné riešiť úlohu aj pre dopyt s iným pravdepodobnostným rozdelením. Výhoda rovnomerného rozdelenia na tomto mieste spočíva najmä v jednoduchosti počítania.

V našom prípade sme začali s prázdnym bankomatom, t.j. $x_0 = 0$. Celkové minimálne očakávané náklady na prevádzku nášho bankomatu za obdobie $T = 3$ dni pri určenom dopyte dostávame ako $V_0(0)$, čo predstavuje $0,869$ €.

2.1.4 Riešenie v programe Matlab

Na tomto mieste uvedieme výsledky riešenia vyššie zadanej úlohy v programe Matlab. Označenie premenných ostáva, iba v je v programe nahradené zápisom $value$, hlavne kvôli lepšej orientácii. Parameter a nespomíname, priamo využívame parameter x_0. Samotné z je v tomto prípade vlastne 3-rozmerný vektor, pričom každá jeho zložka sa realizuje s pravdepodobnosťou 1/3. Tieto hodnoty vyjadrujú $z.h$, resp. $z.p$. Parameter $umax$ zvolíme rovný hodnote $xmax$. Všeobecne je však rozumné ho voliť v programe ako $umax = fix(xmax/value)$. Sú tak pokryté všetky možnosti, pre výšku doplnenej hotovosti. Zdrojový kód uvádzame v prílohe A.

Pripomeňme, že celý program je spravený všeobecne a je možné jednoducho meniť vstupné parametre. Pre rovnomerné rozdelenie postačuje uviesť interval hodnôt, pravdepodobnosti sa dopočítajú.

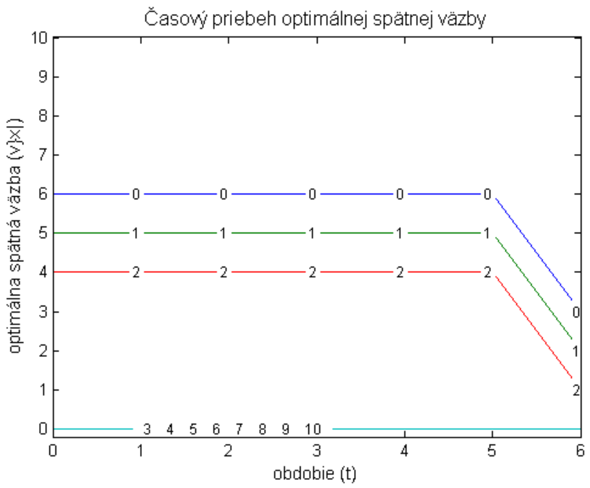
Nakoniec teda priblížime a popíšeme výstupy z nášho programu. Tabuľka (2.1) uvádza hodnoty optimálnej spätnej väzby pre stav x v čase t. Hoci sme vyššie uvažovali $T = 3$, výsledky teraz prezentujeme pre dlhšie obdobie, $T = 7$, aby bolo možné vypozorovať niektoré ďalšie skutočnosti.

Výsledky vidíme aj na obrázku (2.1) a na obrázku (2.2). Reprezentujú dva pohľady. V prvom prípade krivky predstavujú hladiny stavov, v druhom hladiny väzieb. Hoci je

Tabuľka 2.1: Hodnoty $v(t, x)$ optimálnej spätnej väzby, konštantne rozdelený dopyt, dĺžka obdobia $T = 7$

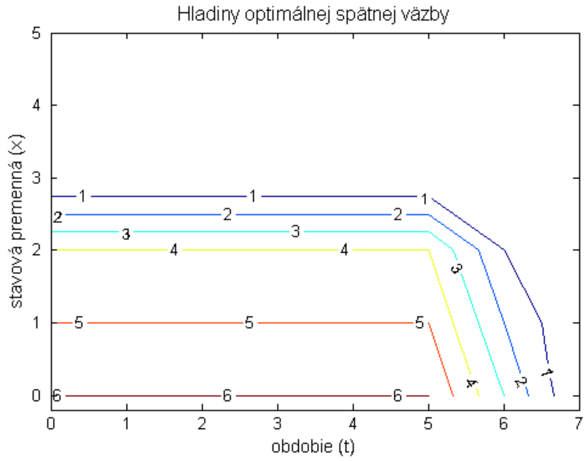
t \x	0	1	2	3	4	5	6	7	8	9	10
0	6	5	4	0	0	0	0	0	0	0	0
1	6	5	4	0	0	0	0	0	0	0	0
2	6	5	4	0	0	0	0	0	0	0	0
3	6	5	4	0	0	0	0	0	0	0	0
4	6	5	4	0	0	0	0	0	0	0	0
5	6	5	4	0	0	0	0	0	0	0	0
6	3	2	1	0	0	0	0	0	0	0	0

spätná väzba definovaná iba po čas $T-1$, obrázok (2.2) zachytáva jej priebeh ku koncu obdobia v prípade menších časových jednotiek. Zároveň na tomto obrázku vidíme aj väzby pre neceločíselné stavy, tie sa však v čase $t = 6$ realizujú pre celé x. Nedajme sa pomýliť neceločíselnými hodnotami.



Obr. 2.1: Časový priebeh optimálnej spätnej väzby, dĺžka obdobia $T = 7$

Pozrime sa bližšie, čo nám hovoria výsledky. Ako je vidieť, pre $x \geq 3$ nepotrebu-



Obr. 2.2: Hladiny optimálnej spätnej väzby, dĺžka obdobia $T = 7$

jeme dopĺňať, pretože maximálny dopyt je rovný 3. Optimálne spätné väzby sa v čase nemenia (keďže sa nemenia ani parametre), rozdiel je až ku koncu obdobia. Vtedy dopĺňame len toľko, aby sme s istotou uspokojili dopyt. Nevytvárame si zásobu na ďalší deň, už ju nevyužijeme. Predtým sa nám oplatilo naraz doplniť vždy viac, vzhľadom na fixné náklady spojené s doplnením.
V prípade autonómnej úlohy by situácia vyzerala rovnako aj pre dlhšie časové obdobie.

Tabuľka (2.2) zobrazuje hodnoty hodnotovej funkcie pre všetky prípady času t a stavu x. Posledný riadok je nulový z toho dôvodu, že akýkoľvek objem peňazí v bankomate na konci sledovaného obdobia už nemá pre nás žiadnu hodnotu. Náklady narastajú s dĺžkou obdobia a so znižujúcim sa stavom. Zaujímavé je si však všimnúť, že od istej výšky stavu v každom čase náklady narastajú s rastúcim x. Je to spôsobené stratou, ktorú generuje držanie prebytočnej hotovosti viac dní. Kumulatívna strata v týchto prípadoch je vyššia, ako neskoršie doplnenie predmetnej hotovosti. Poznamenajme, že riešenie nášho príkladu, kedy $T = 3$, teraz zodpovedá situácii $t = 4$, $x = 0$, resp. $V(5, 1)$.

Tabuľka 2.2: Hodnotová funkcia, konštantne rozdelený dopyt, dĺžka obdobia $T = 7$

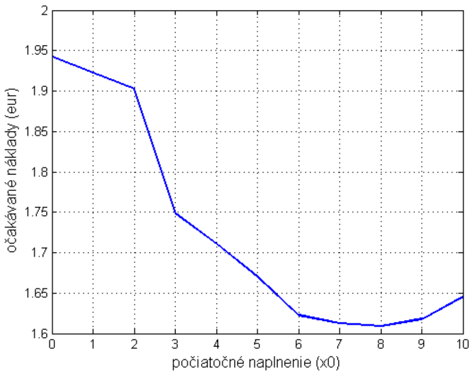
t \x	0	1	2	3	4	5	6	7	8	9	10
0	1,942	1,922	1,902	1,748	1,711	1,67	1,622	1,613	1,609	1,618	1,646
1	1,678	1,658	1,638	1,477	1,445	1,405	1,358	1,344	1,341	1,352	1,382
2	1,407	1,387	1,367	1,219	1,179	1,136	1,087	1,081	1,079	1,085	1,113
3	1,149	1,129	1,109	0,939	0,909	0,874	0,829	0,812	0,804	0,822	0,879
4	0,869	0,849	0,829	0,69	0,653	0,604	0,549	0,539	0,568	0,63	0,72
5	0,62	0,6	0,58	0,42	0,363	0,323	0,3	0,36	0,42	0,48	0,54
6	0,35	0,33	0,31	0,09	0,12	0,15	0,18	0,21	0,24	0,27	0,3
7	0	0	0	0	0	0	0	0	0	0	0

2.1.5 Analýza citlivosti

Podrobme náš príklad analýze citlivosti na vybrané vstupné parametre. Pre $T = 7$.

Počiatočné naplnenie

Obrázok (2.3) zobrazuje výšku minimálnych očakávaných nákladov na prevádzku bankomatu v závislosti na jeho počiatočnom naplnení. Ako vidíme, pre $x_0 \geq 8$ sa prejavuje efekt spomenutý vyššie - strata spojená s držaním prebytočnej hotovosti prevýši cenu jej prípadného doplnenia. Pre vyššie hodnoty parametra c by sa to prejavilo už od nižších hodnôt počiatočného stavu x_0.



Obr. 2.3: Minimálne očakávané náklady v závislosti na počiatočnom naplnení x_0, dĺžka obdobia $T = 7$

Cena za držanie prebytočnej hotovosti

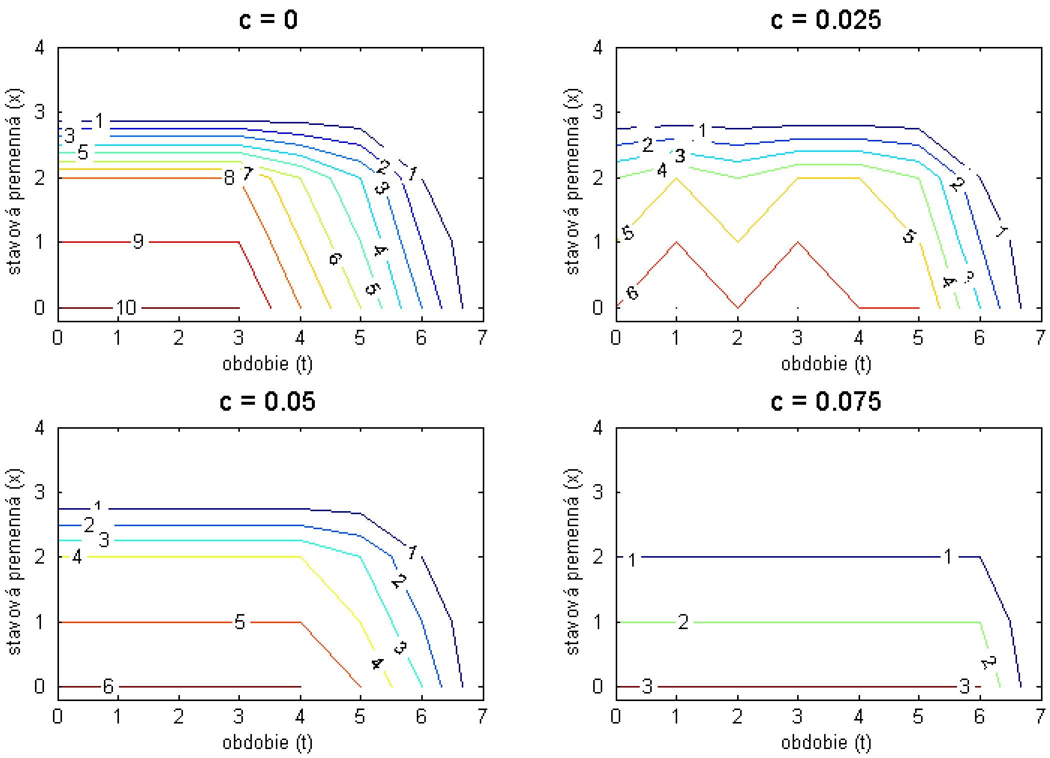
Uvedieme výsledky pre rôzne hodnoty parametra c. Všimnime si, ako sa menia optimálne spätné väzby - obrázok (2.4). Tabuľka (2.3) zachytáva iba nenulové väzby. Ak je strata spojená s držaním prebytočnej hotovosti nulová, dopĺňame vždy maximálne prípustné množstvo. Až ku koncu obdobia menej. Postačí toľko, aby sme pokryli očakávaný dopyt v ostávajúcom čase. Ušetríme tak na fixných nákladoch na doplnenie. S rastúcim c sa dopĺňané množstvo znižuje, až od istej hranice budeme chcieť úplne minimalizovať prebytočnú hotovosť - doplníme iba množstvo, ktoré s istotou uspokojí dopyt pre každý deň.

Rastúce očakávané náklady s rastúcim c.

c	0	0,025	0,05	0,075
$V_0(0)$	1,213	1,855	2,279	2,6

Tabuľka 2.3: Hodnoty $v(t,x)$ optimálnej spätnej väzby v závislosti na c, $T=7$

	$c=0$			$c=0,025$			$c=0,05$			$c=0,075$		
t \x	0	1	2	0	1	2	0	1	2	0	1	2
0	10	9	8	6	5	4	6	5	4	3	2	1
1	10	9	8	7	6	5	6	5	4	3	2	1
2	10	9	8	6	5	4	6	5	4	3	2	1
3	10	9	8	7	6	5	6	5	4	3	2	1
4	8	7	6	6	5	5	6	5	4	3	2	1
5	6	5	4	6	5	4	5	4	3	3	2	1
6	3	2	1	3	2	1	3	2	1	3	2	1



Obr. 2.4: Hodnoty optimálnej spätnej väzby v závislosti na voľbe parametra c, ktorý predstavuje stratu v dôsledku držania prebytočnej hotovosti, dĺžka obdobia $T=7$

Cena za doplnenie

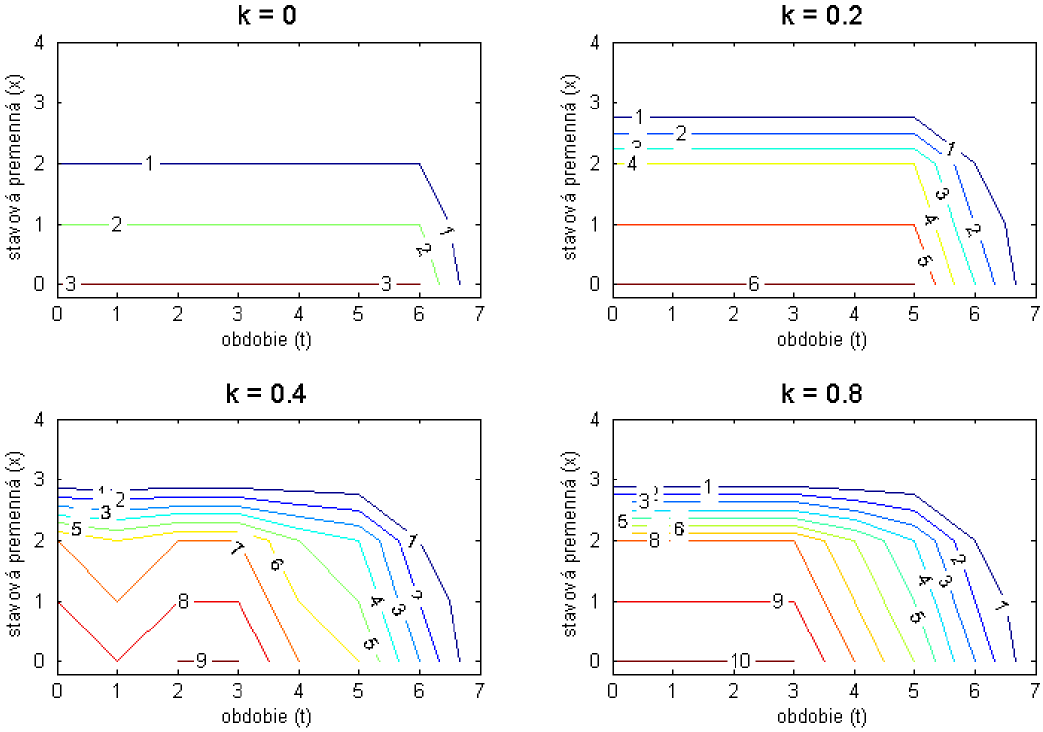
Výsledky pre rôzne hodnoty parametra k. Všimnime si, ako sa menia optimálne spätné väzby - obrázok (2.5). Tabuľka (2.4) zachytáva nenulové väzby. Ak sú fixné náklady na doplnenie peňazí nulové, doplníme vždy iba minimálne množstvo zaručujúce uspokojenie dopytu. Postupne, s rastúcim k, stúpajú aj dopĺňané množstvá. Keď už to niečo stojí, oplatí sa naraz vložiť viac peňazí do bankomatu. Avšak iba po určitú hranicu, aby nebola cena stratenej príležitosti privysoká. S blížiacim sa koncom obdobia dopĺňaný objem opäť klesá. Nepotrebujeme sa zásobiť, keď už tento dodatočný obnos nevyužijeme.

Rastúce očakávané náklady s rastúcim k.

k	0	0,2	0,4	0,8
V(0)	0,93	1,942	2,479	3,331

Tabuľka 2.4: Hodnoty $v(t,x)$ optimálnej spätnej väzby v závislosti na k, $T=7$

	$k=0$			$k=0,02$			$k=0,04$			$k=0,08$		
t \x	0	1	2	0	1	2	0	1	2	0	1	2
0	3	2	1	6	5	4	9	8	7	10	9	8
1	3	2	1	6	5	4	8	7	6	10	9	8
2	3	2	1	6	5	4	9	8	7	10	9	8
3	3	2	1	6	5	4	9	8	7	10	9	8
4	3	2	1	6	5	4	7	6	5	8	7	6
5	3	2	1	6	5	4	6	5	4	6	5	4
6	3	2	1	3	2	1	3	2	1	3	2	1



Obr. 2.5: Hodnoty optimálnej spätnej väzby v závislosti na voľbe parametra k, ktorý určuje fixné náklady na doplnenie bankomatu, dĺžka obdobia $T = 7$

2.2 Neautonómna verzia úlohy

Doteraz sme pracovali s autonómnou úlohou. To znamená, že parametre sa nemenili v čase. Rovnako dopyt po peniazoch bol každý deň rovnako rozdelený. V praxi to však nie je zvyčajne pravdou. Pozrime sa na rozšírenie úlohy, ktorou sa zaoberáme.

Predpokladajme, že cez víkend je dopyt vo všeobecnosti vyšší. Náš bankomat je napríklad umiestnený v nejakom nákupnom centre, kde je cez voľné dni na konci týždňa väčší pohyb zákazníkov. Navyše predpokladajme, že ľudia preferujú platiť hotovosťou a preto vyžívajú možnosť vyberať si peniaze z bankomatu.

Majme teda rovnomerne rozdelený dopyt na intervale $[1, 3]$ počas pracovných dní a na intervale $[2, 5]$ cez víkend. Poznamenajme, že ako začiatok týždňa uvažujeme pondelok. Nedeľa je potom 7.deň. Zároveň celé naše pozorovanie začína v pondelok, teda $t = 0 \rightarrow$ po ráno, $t = 1 \rightarrow$ ut ráno, atď. Dopĺňanie prebieha vždy na začiatku dňa.

Ako je potrebné upraviť náš kód v Matlabe? Hneď na začiatok prvého *for* cyklu vložíme nasledovné podmienky, ktoré budú rozlišovať rozdelenie dopytu pre jednotlivé dni. Celý kód je obsahom prílohy A.

```
switch mod(i,7)
    case 1
        z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
    case 2
        z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
    case 3
        z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
    case 4
        z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
    case 5
        z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
    case 6
        z.h=[2:5]; z.p=1/length(z.h)*ones(1,length(z.h));
    case 0
        z.h=[2:5]; z.p=1/length(z.h)*ones(1,length(z.h));
    otherwise
        error('This is impossible')
end
```

Pripomeňme, že $z.h$ predstavuje jednotlivé hodnoty, ktoré môže nadobúdať naša náhodná premenná z a $z.p$ zase pravdepodobnosť ich realizácie. Opäť sme pracovali iba s rovnomerným rozdelením, túto skutočnosť je však možné jednoducho meniť. Navyše je týmto prístupom možné meniť aj ďalšie parametre, s čím aj ďalej budeme pracovať, no príslušný kód už nebudeme explicitne spomínať.

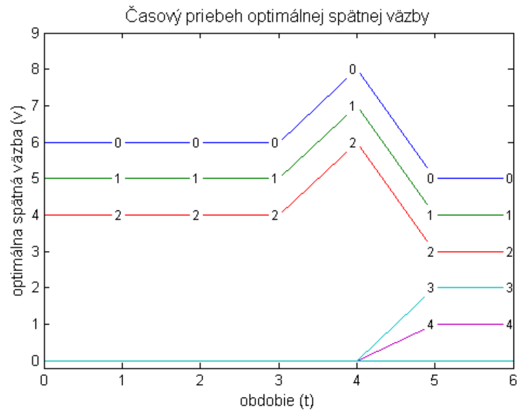
Zvýšený dopyt cez víkend sa zákonite prejaví vo vyšších hodnotách optimálnej spätnej väzby v týchto dňoch. Všimnime si, že už deň pred víkendom dochádza k vyššiemu dopĺňaniu hotovosti. Numerické výsledky nájdeme v tabuľke (2.5) - porovnajme s (2.1). Rovnako ich zobrazujú obrázky (2.6) a (2.7). Z tabuľky (2.6) vyčítame nárast hodnôt oproti (2.2).

Tabuľka 2.5: Hodnoty $v(t, x)$ optimálnej spätnej väzby, rôzne rozdelený dopyt, T=7

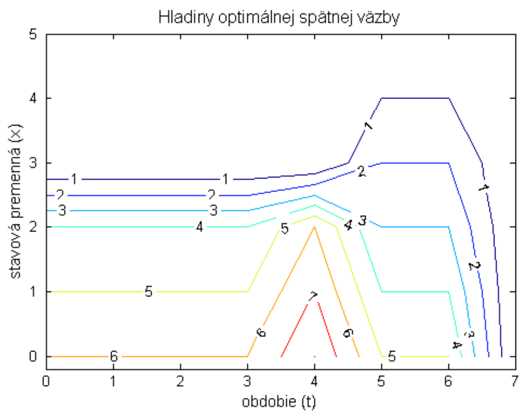
t \x	0	1	2	3	4	5	6	7	8	9	10
0	6	5	4	0	0	0	0	0	0	0	0
1	6	5	4	0	0	0	0	0	0	0	0
2	6	5	4	0	0	0	0	0	0	0	0
3	6	5	4	0	0	0	0	0	0	0	0
4	8	7	6	0	0	0	0	0	0	0	0
5	5	4	3	2	1	0	0	0	0	0	0
6	5	4	3	2	1	0	0	0	0	0	0

Tabuľka 2.6: Hodnotová funkcia, rôzne rozdelený dopyt, T=7

t \x	0	1	2	3	4	5	6	7	8	9	10
0	2,232	2,212	2,192	2,045	2,004	1,961	1,912	1,907	1,904	1,91	1,937
1	1,975	1,955	1,935	1,762	1,735	1,7	1,655	1,636	1,629	1,645	1,688
2	1,692	1,672	1,652	1,52	1,478	1,427	1,372	1,368	1,384	1,411	1,44
3	1,45	1,43	1,41	1,233	1,189	1,154	1,13	1,148	1,147	1,124	1,125
4	1,163	1,143	1,123	0,94	0,95	0,96	0,903	0,857	0,803	0,814	0,817
5	0,87	0,85	0,83	0,81	0,79	0,57	0,58	0,54	0,513	0,498	0,495
6	0,45	0,43	0,41	0,39	0,37	0,15	0,18	0,21	0,24	0,27	0,3
7	0	0	0	0	0	0	0	0	0	0	0



Obr. 2.6: Časový priebeh optimálnej spätnej väzby, rôzne rozdelený dopyt $T = 7$

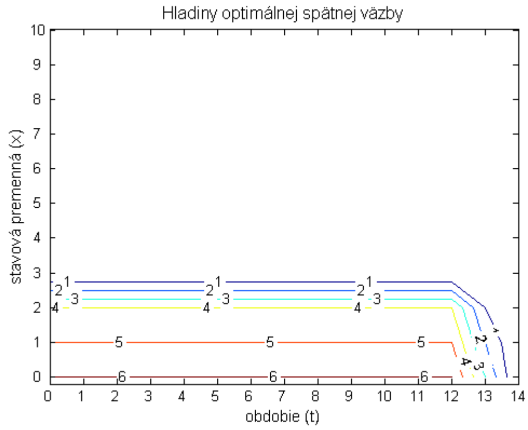


Obr. 2.7: Hladiny optimálnej spätnej väzby, rôzne rozdelený dopyt $T = 7$

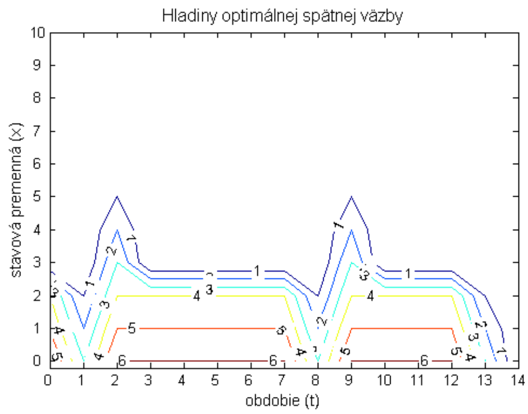
Podobné úvahy nás vedú k zamysleniu, ako by sa menilo riešenie pri zmene ďalších parametrov. Napríklad môžeme predpokladať, že v nedeľu budú samotné fixné náklady na naplnenie výrazne vyššie. Súvisí to napríklad so zvýšenými nákladmi na mzdy pre pracovníkov zabezpečujúcich dopĺňanie.

Nasledovné obrázky ukazujú výsledky pre rôzne prípady nekonštantných fixných nákladov za doplnenie bankomatu. Všetky obrázky sú pre obdobie $T = 14$ dní, aby bolo možné zmeny lepšie pozorovať. Obrázok 2.8 je výsledok pre konštantné fixné náklady. Dajme tomu, že každú stredu by bolo dopĺňanie zadarmo, zmenu zachytáva obrázok

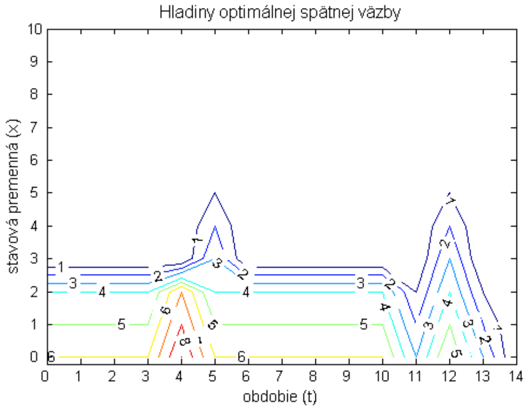
2.9. Nakoniec, ako sme už vyššie spomenuli, je tu prípad drahšej nedele. Obrázok 2.10 zobrazuje optimálne spätné väzby, ak predpokladáme, že každý 7.deň zaplatíme za doplnenie štyrikrát viac ($k = 0,8$).



Obr. 2.8: Hladiny optimálnej spätnej väzby, $T = 14$



Obr. 2.9: Hladiny optimálnej spätnej väzby, $T = 14$, každú stredu je dopĺňanie zadarmo



Obr. 2.10: Hladiny optimálnej spätnej väzby, v nedeľu je doplnenie 4x drahšie, teda 0,8

2.3 Verzia úlohy s pokutou

V tejto časti našu úlohu o spravovaní hotovosti v bankomate rozšírime o ďalšiu úvahu. Doteraz sme striktne odmietali neuspokojený dopyt. Dalo by sa povedať, že sme boli ochotní zabezpečiť prípadnú požadovanú hotovosť za každú cenu. Tu však budeme uvažovať aj možnosť, že dopyt po výberoch z bankomatu presiahne jeho naplnenosť. Aj táto skutočnosť povedie k určitej strate. V konečnom dôsledku však môže byť nižšia ako náklady na doplnenie alebo držanie tejto dodatočnej hotovosti. Mohli by sme sa na situáciu pozrieť aj tak, že práve náklady na mimoriadne doplnenie budú tou "stratou". Avšak úvahy týmto smerom majú zmysel iba vtedy, ak by tieto náklady boli nižšie ako štandardné. Potom by však už nebol celkom na mieste výraz mimoriadne, kde prirodzene predpokladáme, že označuje vyššie náklady.

Budeme uvažovať 3 rôzne prístupy, ako definovať stratu v prípade neuspokojeného dopytu. Konštantnú pokutu v prípade vzniku deficitu, ďalej v závislosti na výške deficitu až napokon exponenciálnu.

Zmena teda nastane vo formulácii účelovej funkcie, v ktorej sa objaví funkcia straty $s(x_i, u_i, z_i, D, S)$, kde S je parameter $S \geq 0$. Ostatné označenie platí ako v časti 2.1.

2.3.1 Formulácia úlohy

Teraz môžeme sformulovať nasledovnú úlohu optimálneho riadenia, so stratou:

$$\min_u \quad E\left[\sum_{i=0}^{T-1} cx_i + k\text{sgn}(u_i) + k_v v u_i + \text{s}(x_i, u_i, z_i, D, S)\right] \tag{2.14}$$

$$\text{pri podm.} \quad x_{i+1} = \max\{x_i + vu_i - z_i, 0\}, \quad i = 0, \dots, T-1, \tag{2.15}$$

$$x_0 = a, \tag{2.16}$$

$$x_i + vu_i \leq H, \tag{2.17}$$

$$u_i \in \mathbb{N}_0. \tag{2.18}$$

Stratu $\text{s}(x_i, u_i, z_i, D, S)$ môžeme uvažovať ako jednu z troch možností

$$S|\min\{(x_i + vu_i - z_i) - D, 0\}|, \tag{2.19}$$

$$S\text{sgn}(|\min\{(x_i + vu_i - z_i) - D, 0\}|), \tag{2.20}$$

$$S^{|\min\{(x_i + vu_i - z_i) - D, 0\}|}, \tag{2.21}$$

v poslednom prípade vyžadujeme $S > 1$.

Zápis stavovej rovnice 2.15 zaručuje nezáporný zostatok hotovosti v bankomate.

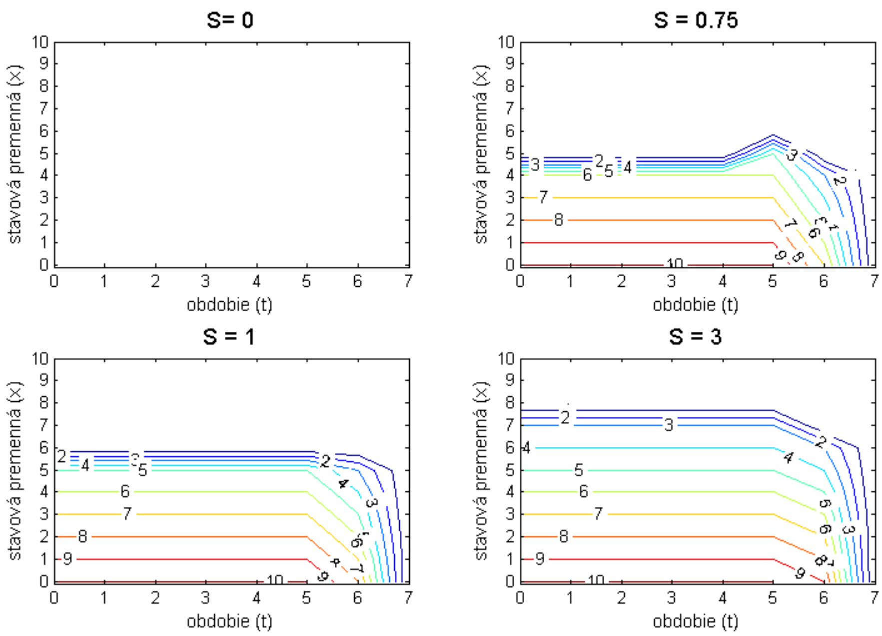
2.3.2 Riešenie v programe Matlab

Vychádzame z kódu A, kde sme implementovali vyššie popísané zmeny. Preformulovali sme teda stavovú rovnicu, ako aj účelovú funkciu. Uvádzame všetky tri typy, hoci grafické výsledky spomenieme iba pre jeden prípad straty. Rozmer matice V hodnotovej funkcie aj matice v hodnôt optimálnej spätnej väzby sme zvýšili o maximálnu hodnotu realizácie náhodného dopytu, čo využijeme pri hľadaní riešenia. Výsledné riešenie však prezentujeme iba pre hodnoty stavov $x \geq 0$.

Hodnoty vstupných parametrov ostávajú rovnaké z časti 2.1.2, parameter straty zvolíme $S = 0,5$ eur. To znamená, že v prípade neuspokojeného dopytu sa naše náklady na prevádzku bankomatu navyšujú o túto sumu. Za každý jeden prípad deficitu. Budeme uvažovať funkciu straty v tvare (2.19). Sumu S teda zaplatíme bez ohľadu na výšku neuspokojeného dopytu. Zdrojový kód uvádzame v prílohe B.

Dopyt budeme uvažovať "približne normálne"rozdelený na intervale $[0, 10]$, so strednou hodnotou $\mu = 5$ a štandardnou odchýlkou $\sigma = 2$. Rozdelenie, ktoré je v niektorých prípadoch dobrou diskretizáciou normálneho rozdelenia, sme vysvetlili v prvej kapitole, v časti 1.2. Podotýkame, že pravdepodobnosti sú určené iba pre celé čísla zo spomínaného rozsahu.

Na obrázku (2.11) máme možnosť vidieť hodnoty optimálnej spätnej väzby pre rôzne hodnoty parametra straty S. Všimnime si, že ak by neuspokojenie dopytu so sebou nenieslo žiadne negatíva, najvýhodnejšie bude pre mňa nič nedopĺňať. Moje náklady by boli rovné nule. S rastúcou výškou prípadnej straty sa vždy viac snažím zabezpečiť dostatočnú hotovosť pre potenciálne výbery.



Obr. 2.11: Hodnoty optimálnej spätnej väzby v závislosti na voľbe parametra S, prípad konštantnej výšky pokuty v prípade vzniku deficitu, dĺžka obdobia $T = 7$, "približne normálne"rozdelenie

Kapitola 3

Maximalizovanie zisku obchodníka

V tejto kapitole sa pozrieme na ďalšiu úlohu. Sformulujeme ju, naprogramujeme a následne vyriešime, pričom výsledky budeme aj krátko komentovať.

Majme predajňu alebo maloobchodného predajcu, ktorý predáva istý typ výrobku. Tento výrobok nakupuje od veľkoobchodného predajcu, prípadne priamo od výrobcu. Dopyt po jeho výrobku je náhodný, avšak známe je jeho pravdepodobnostné rozdelenie. Každý deň prichádzajú do predajne zákazníci, ktorí nechcú odchádzať sklamaní. Inými slovami, náš predavač sa snaží mať pre zákazníka žiadaný tovar. Na druhej strane, nie je ochotný uspokojiť tento dopyt za každú cenu. Predpokladajme, že v takom prípade neplynú pre predajcu žiadne negatíva (napríklad strata dobrého imidžu). Jeho úlohou teda bude na začiatku každého dňa skontrolovať stav zásob na predajni a objednať potrebné množstvo výrobkov. Kapacita jeho skladu je obmedzená, preto aj výška samotných objednávok bude limitovaná. Predpokladajme, že výrobky kupuje za nákupnú cenu a predáva za vyššiu, predajnú cenu. Výrobca navyše posiela výrobky v prepravkách, ktoré majú tiež svoju kapacitu. Okrem nákupnej ceny bude teda obchodník platiť aj za počet prepraviek, ktoré mu prídu. Ďalšie náklady má aj so skladovaním nepredaných výrobkov. Ako asi každý dobrý predajca, aj ten náš bude chcieť maximalizovať svoj očakávaný zisk. Dosiahne to vhodnou stratégiou, resp. objednávkami.

3.1 Formulácia úlohy

Uvažujme prípad, kedy sa jednotlivé parametre nebudú v čase meniť. Tiež pravdepodobnostné rozdelenie dopytu bude rovnaké po celé obdobie. Stavová premenná bude vyjadrovať počet výrobkov na predajni, hodnotami riadenia bude počet objednaných výrobkov od výrobcu. Ten určí aj počet potrebných prepraviek. Predpokladajme, že samotné dodanie trvá veľmi krátko. To znamená, že všetko prebehne každé ráno ešte skôr, ako vstúpi do predajne prvý zákazník. Ďalej predpokladáme, že tovar nestarne a ani nijako inak nestráca na hodnote. Výrobky sa predávajú za svoju stanovenú cenu bez ohľadu na to, ako dlho ich držím u seba na predajni (v sklade). Stavová a aj riadiaca premenná má prirodzene celočíselný charakter.

Vzhľadom na to, že systém je vystavený náhodným vplyvom, pracujeme s úlohami s voľným koncom. Nezaujíma nás, koľko ostane na predajni výrobkov po skončení sledovaného obdobia.

Označme

T – obdobie (dni),

x_i – množstvo výrobkov v sklade (predajni) na začiatku i-teho dňa,

u_i – množstvo výrobkov, ktoré objednám a je mi dodané na začiatku i-teho dňa,

z_i – dopyt po výrobku v i-ty deň, náhodná premenná,

p – predajná cena výrobku, $p > 0$,

n – nákupná cena výrobku, $0 \leq n < p$,

d – náklady na jednu prepravku, $d \geq 0$,

D – kapacita jednej prepravky, $D > 0$,

s – náklady na skladovanú jednotku výrobku, $s \geq 0$,

K – kapacita skladu, $K > 0$,

a – počiatočné zásoby, $a \geq 0$.

Pripomeňme, že stavová aj riadiaca premenná nadobúdajú celočíselné hodnoty. Podobne dopyt je vyjadrený v celých číslach.

Dostávame nasledovnú úlohu optimálneho riadenia:

$$\max_u \quad E\left[\sum_{i=0}^{T-1} p\min\{x_i+u_i, z_i\} - nu - d\lceil u_i/D\rceil - sx_i\right] \tag{3.1}$$

$$\text{pri podm.} \quad x_{i+1} = \max\{x_i+u_i-z_i, 0\}, \quad i=0,\ldots,T-1, \tag{3.2}$$

$$x_0 = a, \tag{3.3}$$

$$x_i + u_i \le K, \tag{3.4}$$

$$u_i \in \mathbb{N}_0. \tag{3.5}$$

Poznamenajme, že výraz $\lceil u_i/D\rceil$ v účelovej funkcii značí "hornú celú časť". Určuje nám, koľko prepraviek je potrebné použiť na objednané množstvo výrobkov. Zároveň to znamená, že náklady na prepravku nie sú závislé na jej naplnení. Nezáleží teda na tom, či je naplnená D výrobkami alebo je v nej prevážaný iba jeden. V oboch prípadoch to pre nás predstavuje náklady d na každú z nich.

Všimnime si, ako je sformulovaná stavová rovnica (3.2). Funkcia max{} nám zaručuje nezápornosť zásob, ale nevylučuje možnosť neuspokojeného dopytu, čo je v súlade s naším zadaním úlohy. Podmienka (3.4) nám hovorí, že môžem objednať iba toľko výrobkov, aby spolu s výrobkami, ktoré už mám na predajni neprekračovali možnosti nášho skladu.

Iný zápis

Účelovú funkciu a stavovú rovnicu by sme mohli napísať aj nasledovným spôsobom:

$$f^0(x_i,u_i,z_i) := \begin{cases} pz_i - d\lceil u_i/D\rceil - sx_i, & \text{ak} \quad x_i+u_i \ge z_i, \\ p(x_i+u_i) - d\lceil u_i/D\rceil - sx_i, & \text{ak} \quad x_i+u_i < z_i \end{cases} \tag{3.6}$$

a

$$f(x_i,u_i,z_i) := \begin{cases} x_i+u_i-z_i, & \text{ak} \quad x_i+u_i \ge z_i, \\ 0, & \text{ak} \quad x_i+u_i < z_i. \end{cases} \tag{3.7}$$

3.1.1 Príklad s konkrétnymi hodnotami

V tejto časti sa budeme zaoberať príkladom s konkrétnymi vstupmi. Majme obdobie $T = 7$ dní, pre ktoré budeme voliť alebo plánovať objednávky výrobkov. Dopyt bude "približne normálne"rozdelený na intervale $[0, 10]$, so strednou hodnotou $\mu = 5$ a štandardnou odchýlkou $\sigma = 2$. Vymyslené rozdelenie sme vysvetlili v prvej kapitole, v časti 1.2. Podotýkame, že pravdepodobnosti sú určené iba pre celé čísla zo spomínaného rozsahu. Predpokladajme, že nakupujeme za cenu $n = 1,5$ eur a predávame za $p = 2,5$ eur za každý výrobok. Uvažujme prepravky s kapacitou $D = 5$ výrobkov a nákladmi $d = 0,5$ eur na jednu. Skladovacie náklady nech sú $s = 0,2$ eur. Najviac môžeme mať na predajni (v pridruženom sklade) $K = 10$ výrobkov a na začiatku nemáme k dispozícii žiadne zásoby, t.j. $a = 0$. Hodnotami riadenia budú také objednané počty, aby bola splnená podmienka (3.4).

Pripomeňme, že uvedené vstupy majú iba približne reálnu výpovednú hodnotu. Sú vymyslené za účelom riešenia úlohy a ozrejmenia niektorých súvislostí a vzťahov, na ktoré nemajú vplyv. Úlohu na tomto mieste nebudeme riešiť analyticky. Príklad takéhoto riešenia si môže čitateľ pozrieť v časti 2.1.3. Sústredíme sa iba na komentovanie výstupov z nášho programu.

Sformulovaná úloha:

$$\max_u \quad E\left[\sum_{i=0}^{6} 2,5\min\{x_i + u_i, z_i\} - 1,5u - 0,5\lceil u_i/5\rceil - 0,2x_i\right] \tag{3.8}$$

$$\text{pri podm.} \quad x_{i+1} = \max\{x_i + u_i - z_i, 0\}, \quad i = 0, \dots, 6, \tag{3.9}$$

$$x_0 = 0, \tag{3.10}$$

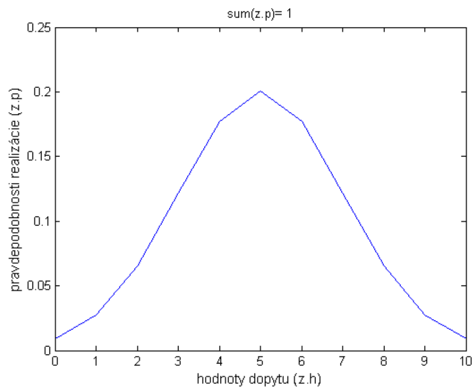
$$x_i + u_i \leq 10, \tag{3.11}$$

$$u_i \in \mathbb{N}_0. \tag{3.12}$$

3.1.2 Riešenie v programe Matlab

Pozrime sa, ako vyzerá riešenie naprogramovanej zadanej úlohy. Označenie parametrov ostáva rovnaké, iba namiesto a budeme pracovať priamo s x_0. Pre určenie pravdepodobností približne diskretizovaného normálneho rozdelenia si pomôžeme príkazom *normpdf* v Matlabe. Jeho vstupmi sú vektor hodnôt, stredná hodnota μ a štandardná odchýlka σ. Následne ešte zabezpečíme, aby súčet pravdepodobností bol rovný 1. Viac je možné sa dočítať v časti 1.2, v teoretickom úvode. Zdrojový kód nájdeme v prílohe C.

Na obrázku (3.1) môžeme vidieť pravdepodobnosti $z.p$ pre $z.h \in \{0, 1, 2, \ldots, 10\}$ realizácie hodnôt náhodnej premennej z.



Obr. 3.1: Približne diskretizované normálne rozdelenie, interval [0,10], $\mu = 5$ a $\sigma = 2$

Hodnoty optimálnej spätnej väzby vidíme v tabuľke (3.1). Všimnime si, že nie sme pripravení uspokojiť možný dopyt (môže byť až 10 v každom čase). Pre nás je optimálne mať k dispozícii väčšinou 8 kusov tovaru. To znamená, že očakávaný zisk je nižší, ako náklady na zabezpečenie týchto dodatočných výrobkov. Súvisí to s nízkou pravdepodobnosťou pre hodnoty dopytu 9 a 10 (podobne 0 a 1). Všimnime si, že pri hodnotách stavu 1 a 2 sa prejaví kapacita prepravky, ktorá v našom prípade pojme 5 výrobkov. Neoplatí sa mi načínať ďalšiu, kvôli jednému, resp. dvom výrobkom. Pri troch už áno, čo je vidieť z hodnôt pre stav $x = 0$. Vzhľadom na nemenné parametre, sa ani hodnoty optimálnej spätnej väzby zo začiatku nemenia. Ku koncu sa prejaví fakt, že sa mi neoplatí budovať zásoby a preto klesá počet objednaných výrobkov. Rovnako to môžeme pozorovať aj z obrázku (3.2) a (3.3).

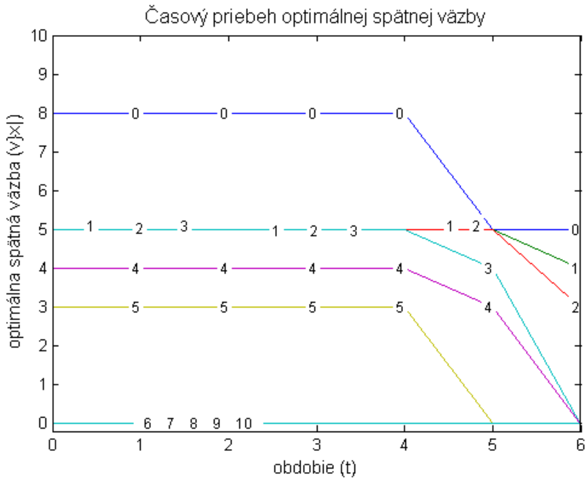
Tabuľka 3.1: Hodnoty $v(t, x)$ optimálnej spätnej väzby, obdobie T=7

t \x	0	1	2	3	4	5	6	7	8	9	10
0	8	5	5	5	4	3	0	0	0	0	0
1	8	5	5	5	4	3	0	0	0	0	0
2	8	5	5	5	4	3	0	0	0	0	0
3	8	5	5	5	4	3	0	0	0	0	0
4	8	5	5	5	4	3	0	0	0	0	0
5	5	5	5	4	3	0	0	0	0	0	0
6	5	4	3	0	0	0	0	0	0	0	0

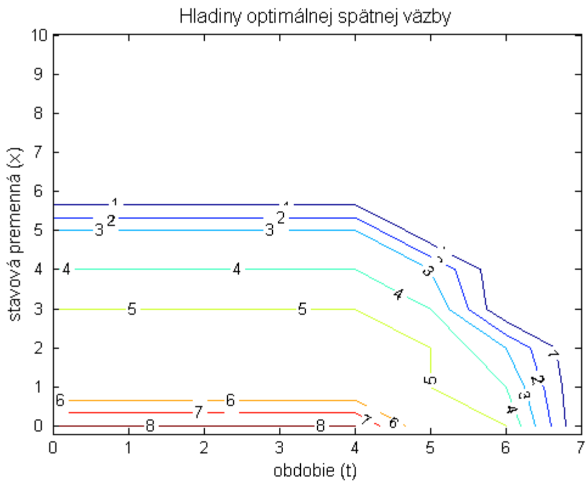
Hodnoty hodnotovej funkcie zobrazuje tabuľka (3.2). V predposlednom riadku, pre vyššie hodnoty stavovej premennej je vidieť, že hodnoty začínajú klesať. Je to spôsobené tým, že je malá pravdepodobnosť, že sa mi podarí predať toľko výrobkov. Očakávaný zisk začína byť menší ako náklady spojené s ich skladovaním. Pre vyššiu hodnotu parametra s by to bolo výraznejšie, aj pre iné časy.

Tabuľka 3.2: Hodnotová funkcia, obdobie T=7

t \x	0	1	2	3	4	5	6	7	8	9	10
0	25,18	26,79	28,26	29,58	30,88	32,18	33,79	35,26	36,58	37,81	39,02
1	21,37	22,98	24,45	25,77	27,07	28,37	29,98	31,45	32,77	34,01	35,21
2	17,56	19,17	20,64	21,96	23,26	24,56	26,17	27,64	28,96	30,2	31,41
3	13,75	15,37	16,84	18,15	19,45	20,75	22,37	23,84	25,15	26,39	27,6
4	9,942	11,57	13,03	14,34	15,64	16,94	18,57	20,03	21,34	22,57	23,77
5	6,179	7,79	9,199	10,5	11,8	13,18	14,79	16,2	17,44	18,51	19,37
6	2,579	3,879	5,179	6,535	8,279	9,579	10,38	10,74	10,79	10,68	10,5
7	0	0	0	0	0	0	0	0	0	0	0



Obr. 3.2: Časový priebeh optimálnej spätnej väzby pre jednotlivé stavy x, $T = 7$



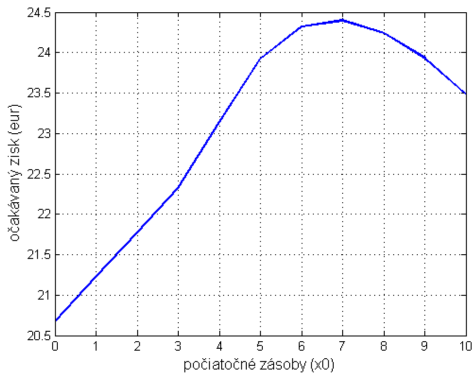
Obr. 3.3: Hladiny optimálnej spätnej väzby, $T = 7$

3.1.3 Analýza citlivosti

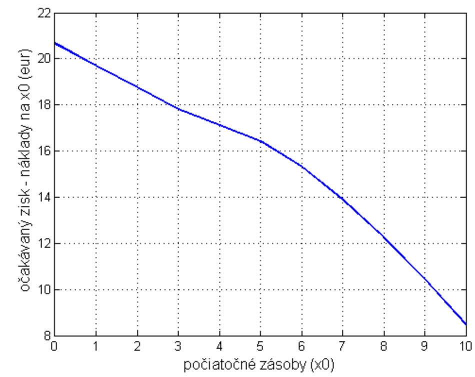
V tejto časti budeme skúmať riešenie úlohy pre rôzne hodnoty vybraných parametrov. Uvedieme iba grafické riešenie a vplyv na očakávaný zisk. Vychádzame z vyššie sformulovanej úlohy a zadaných vstupov. Aby sa prejavili efekty, na ktoré chceme poukázať, uvažujeme v niektorých prípadoch zmenené vstupy, čo spomíname v texte. Viaceré obrázky zobrazujú aj hodnoty väzby pre neceločíselné stavy, ale tým sa nedajme mýliť.

Počiatočné zásoby

Na začiatok sa pozrime, ako vyzerá maximálny očakávaný zisk v závislosti na počiatočných zásobách. Predpokladajme, že náklady na skladovanie sú v tomto prípade vyššie, $s = 0,95$ eur. Na obrázku (3.4) si môžeme všimnúť, že pre $x_0 \geq 7$ začína zisk klesať. Je to spôsobené tým, že náklady na skladovanie prebytočných výrobkov prevýšia náklady na ich objednanie a kúpu v čase potreby. Čím by boli náklady na skladovanie s vyššie, tým skôr by sa tento efekt prejavil. Pre zaujímavosť, obrázok (3.5) nám zobrazuje situáciu, kedy by sme za počiatočné zásoby platili. To znamená, každý výrobok by nás stál jeho nákupnú cenu n, čo odpočítame od očakávaného zisku.



Obr. 3.4: Maximálny očakávaný zisk v závislosti na x_0, $T = 7$



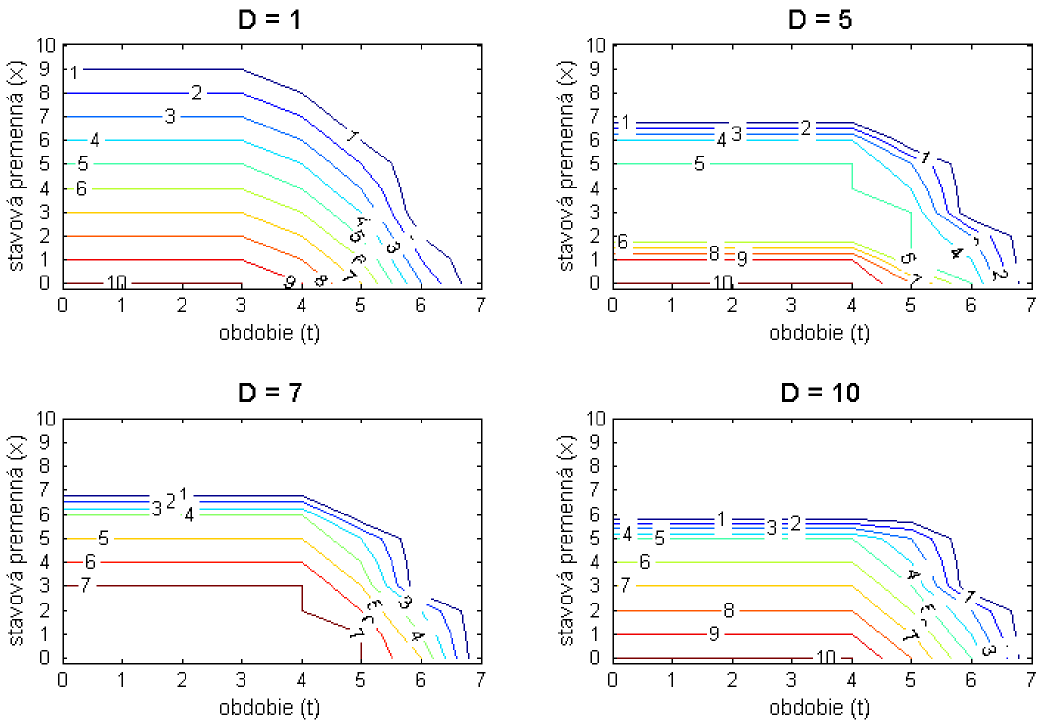
Obr. 3.5: Max. očakávaný zisk v závislosti na x_0, odpočítaná nákupná cena x_0, $T = 7$

Kapacita prepravky

V tomto prípade budeme predpokladať, že za skladovanie neplatíme, to znamená $s = 0$. Na obrázku (3.6) máme možnosť vidieť, ako sa menia hodnoty optimálnej spätnej väzby s hodnotami kapacity prepravky. Výhodnejšia je pre nás prirodzene prepravka s vyššou kapacitou, kedy ich na objednané množstvo potrebujeme menej. Prináša to menšie náklady, čo zachytáva aj tabuľka nižšie. Všimnime si, že ak objednávame, vždy na plný sklad. Je to v dôsledku nulových nákladov na skladovanie.

Rastúci očakávaný zisk s rastúcim parametrom D.

D	1	5	7	10
V(0)	15,94	29,03	29,67	30,1



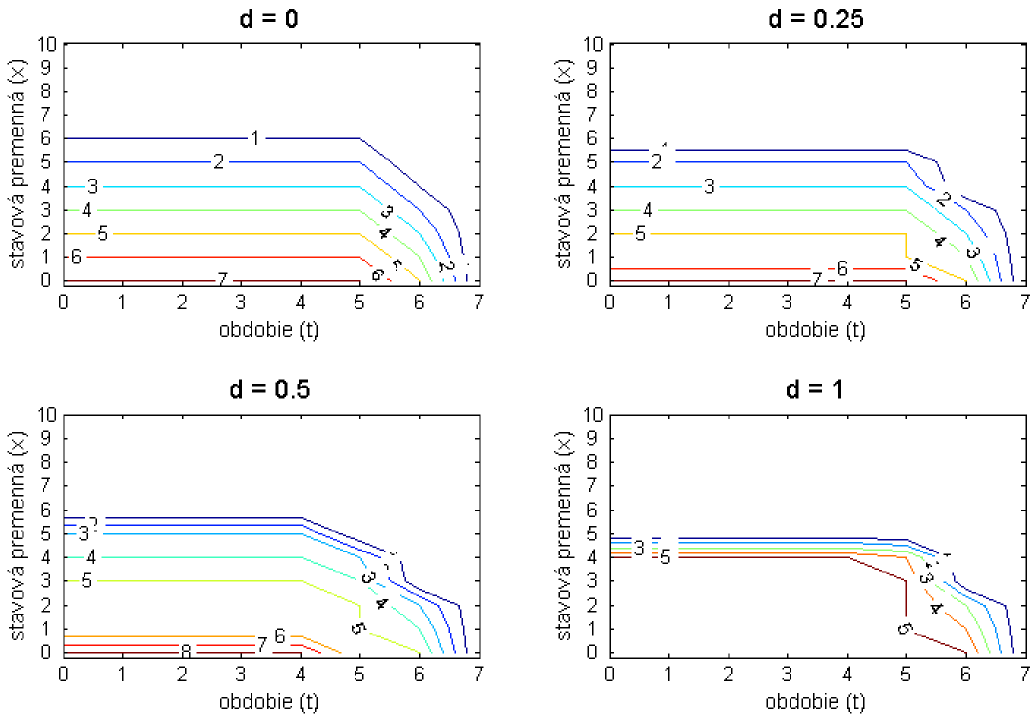
Obr. 3.6: Hodnoty optimálnej spätnej väzby v závislosti na voľbe parametra D, ktorý predstavuje kapacitu jednej prepravky, dĺžka obdobia $T = 7$

Cena za prepravku

Teraz sa budeme zaoberať vplyvom ceny d za jednu prepravku na hodnoty optimálnej spätnej väzby. Výsledky pre zvolené hodnoty d zobrazuje obrázok (3.7). Čím je prepravka drahšia, tým viac ju budem chcieť využiť.

Klesajúci očakávaný zisk s rastúcim parametrom d.

d	0	0,25	0,5	1
V(0)	29,6	27,3	25,18	21,75



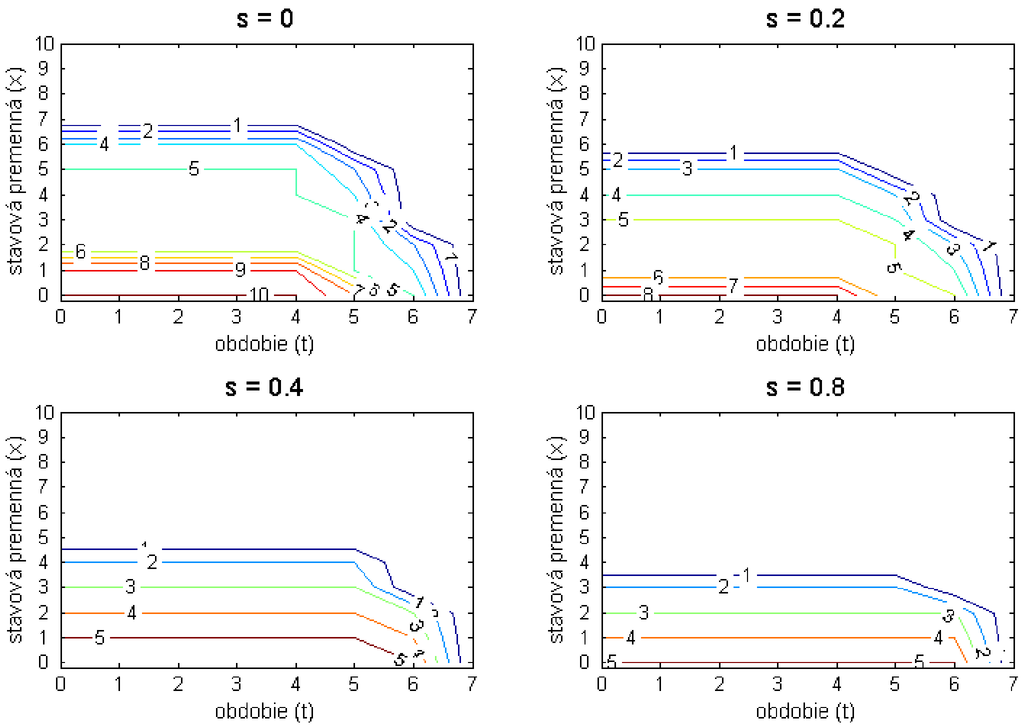
Obr. 3.7: Hodnoty optimálnej spätnej väzby v závislosti na voľbe parametra d, ktorý hovorí o cene za jednu prepravku, dĺžka obdobia $T = 7$

Cena za skladovanie

Nakoniec sa pozrieme na hodnoty optimálnej spätnej väzby pre rôznu výšku nákladov na skladovanie výrobkov. S ich nárastom klesá objednávané množstvo, čo je v súlade s prirodzenou úvahou. Situáciu zachytáva obrázok (3.8). Všimnime si, že pokým na začiatku plním sklad až po jeho kapacitu ($K = 10$), v poslednom prípade sa mi oplatí mať pripravených k predaju najviac 5 výrobkov.

Klesajúci očakávaný zisk s rastúcim parametrom s.

s	0	0,2	0,4	0,8
V(0)	29,03	25,18	23,57	21,35



Obr. 3.8: Hodnoty optimálnej spätnej väzby v závislosti na voľbe parametra s, ktorý predstavuje náklady na skladovanie jedného výrobku, dĺžka obdobia $T = 7$

Kapitola 4

Úloha o hazardnom hráčovi

V poslednej kapitole sa pozrieme na tri modifikácie úlohy o hazardnom hráčovi. Zadania čerpáme z knihy [2]. Poznamenajme, že v týchto úlohách bude T na rozdiel od predchádzajúcich kapitol predstavovať počet kôl alebo samotných stávok.

4.1 Stávkami sú žetóny

Najprv sa budeme zaoberať prípadom, kedy stávkami budú žetóny. Náš hráč sa účastní hry s nasledovnými pravidlami. Na začiatku má k dispozícii určitý počet žetónov. Hra prebieha v niekoľkých kolách, pričom v každom z nich sa rozhoduje, koľko žetónov vsadí. Môže z nich vsadiť ľubovoľný počet, pričom s pravdepodobnosťou P rovnaký počet získa alebo s pravdepodobnosťou $(1 - P)$ stratí všetko, čo vsadil. Pre výhru v celej hre je potrebné, aby mal hráč na konci vopred určený počet žetónov. Vtedy obdrží finančnú odmenu. Snahou hráča bude získať túto odmenu. Dalo by sa povedať, že sa bude snažiť maximalizovať očakávaný počet žetónov na konci hry, no len po istú hranicu. Touto hranicou je už spomenutý počet žetónov postačujúci na výhru.

Označme

T – počet kôl hry,

x_i – počet žetónov na začiatku i-tej hry,

u_i – počet žetónov, ktoré v i-tej hre vsadíme,

z_i – výhra, resp. prehra v i-tom kole, náhodná premenná,

C – počet žetónov potrebných na konci hry na víťazstvo, $C > 0$,

p – peňažná odmena pre hráča, ak na konci spĺňa podmienky na výhru v hre, $p > 0$,

h – funkcia výhry, podmienka na výhru celej hry,

a – počet žetónov k dispozícii na začiatku hry, $a \geq 0$.

4.1.1 Formulácia úlohy

Teraz môžeme sformulovať stochastickú úlohu optimálneho riadenia:

$$\begin{aligned} \max_{u} \quad & E[p\mathrm{h}(x_T - C)] && (4.1) \\ \text{pri podm.} \quad & x_{i+1} = x_i + z_i u_i, \quad i = 0, \ldots, T-1, && (4.2) \\ & x_0 = a, && (4.3) \\ & 0 \leq u_i \leq x_i, && (4.4) \\ & u_i \in \mathbb{N}, && (4.5) \end{aligned}$$

kde

$$\mathrm{h}(x) = \begin{cases} 1, & \text{ak } x \geq 0, \\ 0, & \text{ak } x < 0. \end{cases} \tag{4.6}$$

Táto úloha sa líši od štandardnej typom účelovej funkcie (4.1) a zmiešanými ohraničeniami (4.4) na stavovú a riadiacu premennú. Účelová funkcia je Mayerovho tvaru, to znamená, funkcia iba koncového stavu. Ohraničenie (4.4) zaručuje, že nemôžeme vsadiť viac žetónov, ako máme práve k dispozícii.

Tieto zmeny spôsobia, že rovnica dynamického programovania má pre túto úlohu tvar

$$V_j(x) = \max_{0 \le u \le x} E[V_{j+1}(x + z_j u)], \quad j = 0, \ldots, T-1, \tag{4.7}$$

$$V_T(x) = p\mathrm{h}(x - C). \tag{4.8}$$

4.1.2 Príklad s konkrétnymi hodnotami

Vyriešime vyššie sformulovanú úlohu. Budeme hrať $T = 3$ kolá, začíname s 3 žetónmi a pravdepodobnosť výhry v každom kole je 2/3 (resp. prehry 1/3). Na výhru v celej hre nám postačuje mať $C = 5$ žetónov. Zaujíma nás, koľko žetónov je optimálne vsadiť v jednotlivých kolách. Poznamenajme, že riadiaca aj stavová premenná majú celočíselný charakter.

Potom úloha vyzerá nasledovne:

$$\max_{u} \quad E[\mathrm{h}(x_3 - 5)] \tag{4.9}$$

$$\text{pri podm.} \quad x_{i+1} = x_i + z_i u_i, \quad i = 0, 1, 2, \tag{4.10}$$

$$x_0 = 3, \tag{4.11}$$

$$0 \le u_i \le x_i, \tag{4.12}$$

$$u_i \in \mathbb{N}, \tag{4.13}$$

kde

$$\mathrm{h}(x) = \begin{cases} 1, & \text{ak } x \ge 0, \\ 0, & \text{ak } x < 0 \end{cases} \tag{4.14}$$

a

$$z_i = \begin{cases} 1 & \text{s pravdepodobnosťou } \frac{2}{3}, \\ -1 & \text{s pravdepodobnosťou } \frac{1}{3}. \end{cases} \tag{4.15}$$

4.1.3 Riešenie v programe Matlab

Úlohu sme naprogramovali, na tomto mieste uvádzame výsledky riešenia. Úloha je zaujímavá tým, že v niektorých kolách existuje viacero optimálnych riešení. Náš program zachytáva všetky možnosti do objektu $vsetky(,).riesenia$, pre každé kolo a možný stav. Matica v optimálnych spätných väzieb bude zahŕňať vždy najmenšie z riešení. Ak by bol napríklad za každý vsadený žetón manipulačný poplatok, bola by jediným riešením práve táto matica. Všimnime si, že nám stačí uvažovať $xmax = C$, pretože akonáhle dosiahneme tento počet žetónov, už nebudeme mať dôvod ďalej stávkovať. Zdrojový kód nájdeme v prílohe D.

Tabuľka (4.1) zobrazuje spomínané najnižšie z hodnôt optimálnej spätnej väzby. Tento výsledok máme možnosť vidieť aj na obrázku (4.1).

Tabuľka 4.1: Hodnoty $v(t, x)$ optimálnej spätnej väzby, $T = 3$ kolá

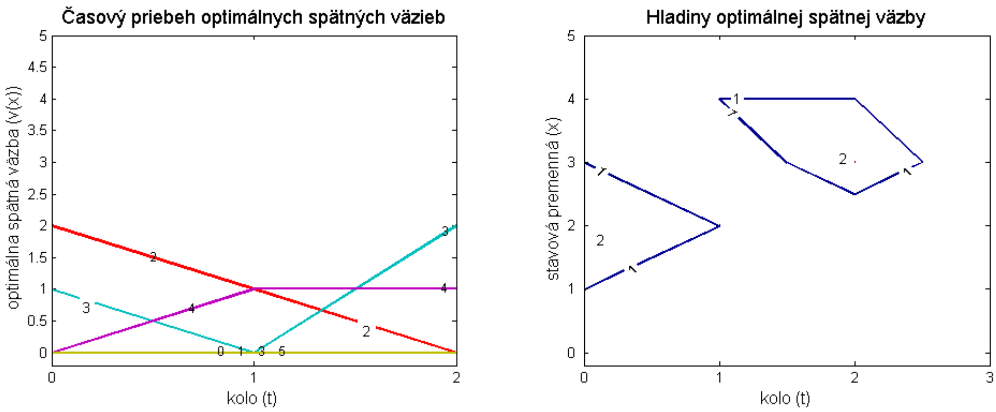
t \x	0	1	2	3	4	5
0	0	1	2	1	0	0
1	0	0	1	0	1	0
2	0	0	0	2	1	0

Tabuľka (4.2) zobrazuje hodnoty hodnotovej funkcie. Prvý riadok nám v tomto prípade ($p = 1$) predstavuje pravdepodobnosť výhry v samotnej hre, pre rôzny počet žetónov, ktoré máme na začiatku hry k dispozícii. Rovnako to vidíme na obrázku (4.2).

Tabuľka 4.2: Hodnotová funkcia, $T = 3$ kolá

t \x	0	1	2	3	4	5
0	0	0,296	0,593	0,741	0,889	1
1	0	0	0,444	0,667	0,889	1
2	0	0	0	0,667	0,667	1
3	0	0	0	0	0	1

Počet všetkých riešení nájdeme v tabuľke (4.3), obsahom tabuľky (4.4) sú potom všetky tieto riešenia. Ako vidíme, existuje mnoho optimálnych stratégií.



Obr. 4.1: Časový priebeh a hladiny optimálnej spätnej väzby, $T = 3$ kolá

Tabuľka 4.3: Počet riešení - hodnôt $v(t, x)$ optimálnych spätných väzieb, $T = 3$ kolá

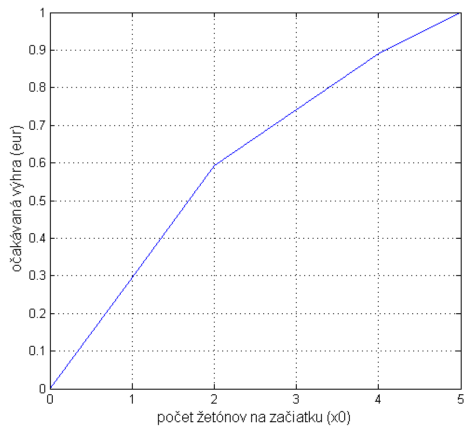
t \x	0	1	2	3	4	5
0	1	1	1	1	2	1
1	1	2	2	3	1	1
2	1	2	3	2	4	1

Tabuľka 4.4: Všetky riešenia - hodnoty optimálnej spätnej väzby, $T = 3$ kolá

t \x	0	1	2	3	4	5
0	0	1	2	1	0; 1	0
1	0	0; 1	1; 2	0; 2; 3	1	0
2	0	0; 1	0; 1; 2	2; 3	1; 2; 3; 4	0

4.2 Stávkami sú celé eurá

Tentokrát budú predmetom stávok peniaze. Predpokladajme, že hráč bude hrať s eurami a staviť môže iba celé eurá. Náš hráč sa účastní hry, kde na začiatku dostane k dispozícii istý obnos peňazí. Hra prebieha v niekoľkých kolách, pričom v každom z nich sa rozhoduje, koľko eur vsadí. Nemôže staviť viac, ako má, pričom s pravdepo-



Obr. 4.2: Očakávaná výhra vzhľadom na počiatočný počet žetónov x_0, $T = 3$ kolá

dobnosťou P rovnakú sumu získa alebo s pravdepodobnosťou $(1 - P)$ stratí všetko, čo stavil. Snahou hráča bude maximalizovať očakávaný počet eur na konci hry.

Označme

T	–	počet kôl hry,
x_i	–	počet eur na začiatku i-tej hry,
u_i	–	počet eur, ktoré v i-tej hre vsadíme,
z_i	–	výhra, resp. prehra v i-tom kole, náhodná premenná,
a	–	eurá k dispozícii na začiatku hry, $a \geq 0$.

4.2.1 Formulácia úlohy

Ide vlastne o podobnú úlohu, ako v predchádzajúcej časti. Zmena nastane v účelovej funkcii, teraz nemáme ďalšie podmienky na hru. Riadiaca aj stavová premenná majú opäť celočíselný charakter.

Dostávame teda úlohu

$$\max_u \quad E[x_T] \tag{4.16}$$

$$\text{pri podm.} \quad x_{i+1} = x_i + z_i u_i, \quad i = 0, \ldots, T-1, \tag{4.17}$$

$$x_0 = a, \tag{4.18}$$

$$0 \leq u_i \leq x_i, \tag{4.19}$$

$$u_i \in \mathbb{N}. \tag{4.20}$$

Rovnica dynamického programovania má pre túto úlohu tvar

$$V_j(x) = \max_{0 \leq u \leq x} E[V_{j+1}(x + z_j u)], \quad j = 0, \ldots, T-1, \tag{4.21}$$

$$V_T(x) = x. \tag{4.22}$$

4.2.2 Riešenie v programe Matlab

Rovnako, ako v predchádzajúcej verzii úlohy o hazardnom hráčovi, aj teraz zvolíme $T = 3$ kolá a pravdepodobnosť výhry v každom kole danú vzťahom 4.15. Na začiatku hry však budeme začínať s $x_0 = 1$ eurom. V zdrojovom kóde, uvedenom v prílohe D, zmeníme spomínanú účelovú funkciu koncového stavu a taktiež parameter $xmax$. Aby sme pokryli všetky možné hodnoty stavovej premennej, zvolíme ho $xmax = 2^T x0$. Viac v tejto hre za daný počet kôl nemôžeme dosiahnuť.

Tabuľka (4.5) zobrazuje najnižšie z hodnôt optimálnej spätnej väzby. Tento výsledok máme možnosť vidieť aj na obrázku (4.3).

Tabuľka (4.6) zobrazuje hodnoty hodnotovej funkcie. Prvý riadok nám predstavuje očakávaný zisk na konci hry, pri zadanom počte eur. Podobne je to vlastne v každom čase. Údaj v tabuľke uvádza, aký je môj očakávaný zisk s daným počtom eur do konca hry.

Počet všetkých riešení nájdeme v tabuľke (4.7), obsahom tabuľky (4.8) sú potom všetky tieto riešenia. Ako vidíme, existuje viac optimálnych stratégií.

Tabuľka 4.5: Hodnoty $v(t, x)$ optimálnej spätnej väzby, $T = 3$ kolá

t \x	0	1	2	3	4	5	6	7	8
0	0	1	0	1	2	1	0	1	0
1	0	1	2	1	0	1	2	1	0
2	0	1	2	3	4	3	2	1	0

Tabuľka 4.6: Hodnotová funkcia, $T = 3$ kolá

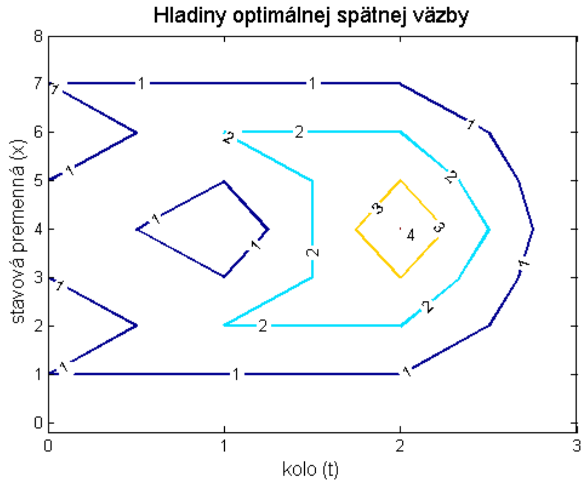
t \x	0	1	2	3	4	5	6	7	8
0	0	2,370	3,556	4,741	5,926	6,519	7,111	7,704	8
1	0	1,778	3,556	4,444	5,333	6,222	7,111	7,556	8
2	0	1,333	2,667	4,000	5,333	6,000	6,667	7,333	8
3	0	1	2	3	4	5	6	7	8

Tabuľka 4.7: Počet riešení - hodnôt $v(t, x)$ optimálnych spätných väzieb, $T = 3$ kolá

t \x	0	1	2	3	4	5	6	7	8
0	1	1	3	2	1	2	2	1	1
1	1	1	1	2	4	3	1	1	1
2	1	1	1	1	1	1	1	1	1

Tabuľka 4.8: Všetky riešenia - hodnoty optimálnej spätnej väzby, $T = 3$ kolá

t \x	0	1	2	3	4	5	6	7	8
0	0	1	0; 1; 2	1; 3	2	1; 3	0; 2	1	0
1	0	1	2	1; 2	0; 1; 3; 4	1; 2; 3	2	1	0
2	0	1	2	3	4	3	2	1	0



Obr. 4.3: Hladiny optimálnej spätnej väzby, dĺžka obdobia $T = 3$

4.3 Stávkami sú eurá

Nakoniec uvedieme tretiu verziu úlohy o hazardnom hráčovi. Stávkami sú opäť peniaze. Predpokladajme, že hráč bude hrať s eurami, no staviť nemusí nutne iba celé eurá. Náš hráč sa účastní hry s nasledovnými pravidlami. Na začiatku dostane k dispozícii objem peňazí. Hra prebieha v niekoľkých kolách, pričom v každom z nich sa rozhoduje, akú časť eur, ktoré má, vsadí. Môže staviť ľubovoľný obnos, pričom s pravdepodobnosťou P rovnakú sumu získa alebo s pravdepodobnosťou $(1-P)$ stratí všetko, čo stavil. Snahou hráča bude maximalizovať očakávaný počet eur na konci hry.

Označme

T – počet kôl hry,

x_i – počet eur na začiatku i-tej hry,

u_i – pomerná časť kapitálu, ktorú v i-tej hre hráč vsadí,

z_i – výhra, resp. prehra v i-tom kole, náhodná premenná,

a – eurá k dispozícii na začiatku hry, $a \geq 0$.

4.3.1 Formulácia úlohy

Poznamenajme, že v tejto úlohe má stavová ako aj riadiaca premenná kontinuálny charakter. Preto budeme pri jej riešení voliť ich diskretizáciu.

Teraz môžeme sformulovať nasledovnú úlohu optimálneho riadenia:

$$\begin{aligned} \max_{u} \quad & E[x_T] & (4.23)\\ \text{pri podm.} \quad & x_{i+1} = x_i + z_i u_i x_i, \quad i = 0, \dots, T-1, & (4.24)\\ & x_0 = a, & (4.25)\\ & u_i \in [0,1]. & (4.26) \end{aligned}$$

Rovnica dynamického programovania má pre túto úlohu tvar

$$\begin{aligned} V_j(x) &= \max_{u \in [0,1]} E[V_{j+1}(x + z_j u x)], \quad j = 0, \dots, T-1, & (4.27)\\ V_k(x) &= x. & (4.28) \end{aligned}$$

4.3.2 Príklad s konkrétnymi hodnotami

Majme hru, ktorú budeme hrať $T = 2$ kolá, na začiatku dostaneme k dispozícii $x_0 = 1$ euro. Volíme preto taký nízky počet kôl, aby boli prezentované výsledky ľahšie čitateľné. V každom kole môžeme vyhrať s pravdepodobnosťou 3/4 (prehrať s pravdep. 1/4). Naším cieľom bude v každom kole staviť takú časť eur, čo máme, aby bola očakávaná suma na konci hry maximálna.

Dostávame teda úlohu:

$$\begin{aligned} \max_{u} \quad & E[x_2] & (4.29)\\ \text{pri podm.} \quad & x_{i+1} = x_i + z_i u_i x_i, \quad i = 0, 1, & (4.30)\\ & x_0 = 1, & (4.31)\\ & u_i \in [0,1], & (4.32) \end{aligned}$$

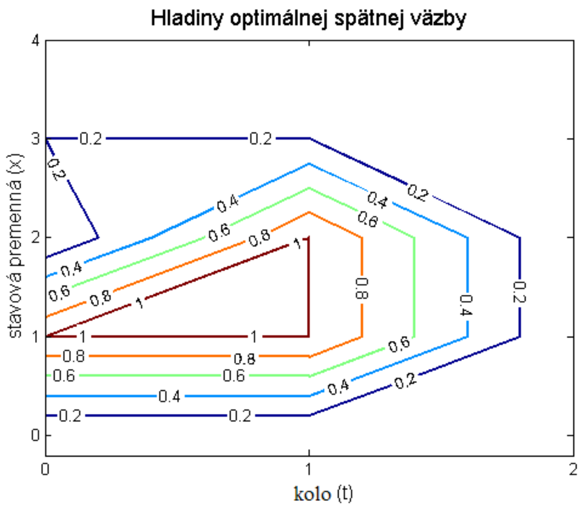
kde

$$z_i = \begin{cases} 1 & \text{s pravdepodobnosťou } \frac{3}{4}, \\ -1 & \text{s pravdepodobnosťou } \frac{1}{4}. \end{cases} \qquad (4.33)$$

4.3.3 Riešenie v programe Matlab

Ako už sme vyššie spomenuli, stavová premenná x a aj riadiaca premenná u má kontinuálny charakter. Budeme preto voliť stupeň ich diskretizácie. Poslúži nám na to parameter N, resp. M. Aby sme mohli jednoduchšie pracovať s výsledkami, vytvorili sme premenné, ktoré budú obsahovať výsledky iba pre celočíselné stavy. Označili sme ich na začiatku písmenom "C". Zdrojový kód programu uvádzame v prílohe E.

Hladiny optimálnej spätnej väzby pre $N = 100$ a $M = 5$ môžeme vidieť na obrázku (4.4). Získavame teda návod, ako stávkovať v jednotlivých kolách pri danom počte eur, aby sme sa na konci dopracovali k maximálnemu očakávanému zisku.



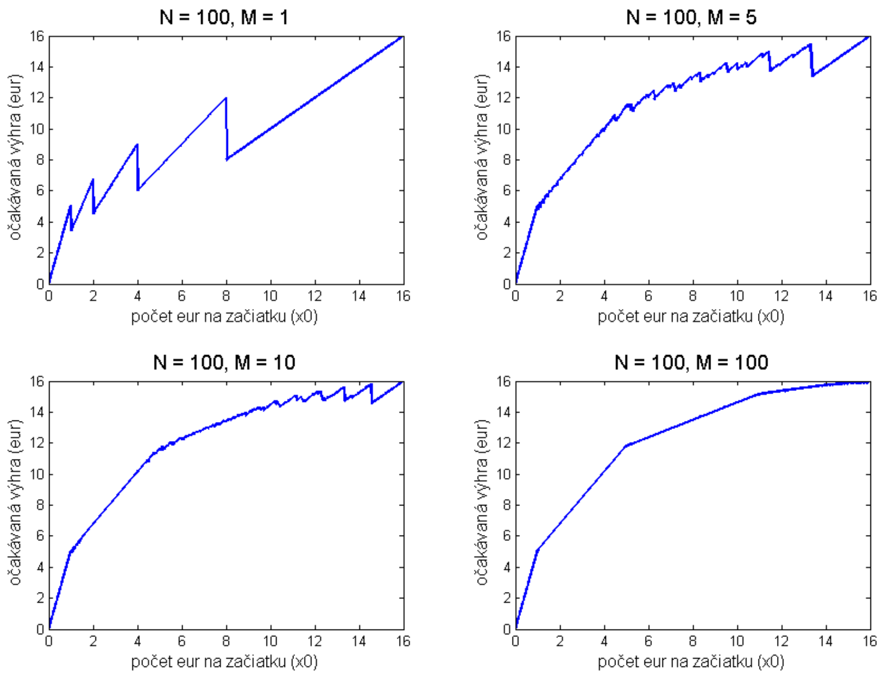
Obr. 4.4: Hladiny optimálnej spätnej väzby, $T = 2$ kolá

Tabuľka (4.9) zobrazuje hodnoty hodnotovej funkcie pre celočíselné stavy x. V našom prípade, kedy začíname s jedným eurom a pri zadanej pravdepodobnosti, je náš maximálny očakávaný zisk rovný 2,25 eur. Z obrázka (4.4) vyčítame, že na začiatku je optimálne vsadiť všetko.

Tabuľka 4.9: Hodnotová funkcia, $T = 2$ kolá

V(t,x)	0	1	2	3	4
0	0	2,25	3	3,48	4
1	0	1,50	3	3,30	4
2	0	1	2	3	4

Pre zaujímavosť uvádzame, ako sa bude meniť očakávaná výhra v závislosti na počiatočnom kapitále, pre zvolené rôzne stupne diskretizácie M riadiacej premennej u a pevný stupeň diskretizácie $N = 100$ pre stavovú premennú x. Zvolený počet kôl hry je v tomto prípade $T = 4$ kolá. Výsledok zobrazuje obrázok (4.5). Vidieť postupné zhladenie riešenia.



Obr. 4.5: Očakávaná výhra v závislosti na x0, stupni diskretizácie M riadiacej premennej u, $T = 4$ kolá

Záver

V diplomovej práci sme v stručnosti spracovali teóriu stochastického optimálneho riadenia. Predstavili sme rovnicu dynamického programovania, pričom sme uviedli aj všeobecný algoritmus na riešenie úloh pomocou nej. Snažili sme sa priblížiť problematiku čitateľovi a zároveň vzbudiť u neho hlbší záujem o túto oblasť.

V troch kapitolách sme sa venovali zostaveným úlohám. Prvou a zároveň ústrednou úlohou tejto diplomovej práce bola úloha o spravovaní hotovosti v bankomate. Inšpirovali sme sa článkom [1]. Zostavili sme model, popísali ho a pre jednoducho zadané vstupy sme úlohu aj analyticky vyriešili. Následne sme ju naprogramovali a komentovali výstupy nášho programu. Venovali sme sa autonómnej, ako i neautonómnej verzii, kedy sa v čase niektoré parametre menili. Nakoniec sme uviedli rozšírenie tejto úlohy. Oproti prvotnej formulácii sme teraz uvažovali možnosť neuspokojenia dopytu, s čím bola spojená strata, ktorá sa objavila v účelovej funkcii. Úloha ponúka ešte bohaté možnosti pre ďalšie rozšírenia a úvahy.

Podobne sme pristúpili k ďalším dvom úlohám, maximalizovaniu zisku obchodníka a úlohe o hazardnom hráčovi. Spísali sme motiváciu k ich zostaveniu, sformulovali ich a následne aj naprogramovali. Dosiahnuté výsledky a riešenia sme komentovali.

Programy sú zostavené všeobecne, je jednoduché meniť vstupné parametre. Zaujímavým by mohlo byť viacej sa zaoberať rozdelením náhodnej premennej, porovnávať výsledky pre rôzne rozdelenia. To však nebolo náplňou tejto práce.

Prínos práce vidíme v poukázaní na využitie rovnice dynamického programovania, ponúknutie určitého návodu ako s ňou pracovať. Súčasťou práce sú aj samotné kódy programov v Matlabe, ktoré pripájame v prílohe. Môžu rovnako slúžiť záujemcom ako motivácia pre ďalšie štúdium, ak by sa chceli vydať týmto smerom.

Literatúra

[1] Castro, J., *A stochastic programming approach to cash management in banking*, EJOR, 2009.
web: www-eio.upc.es/~jcastro/publications/reports/dr2004-14.pdf

[2] Halická, M., Brunovský, P., Jurča, P., *Optimálne riadenie - Viacetapové rozhodovacie procesy v ekonómii a financiách*, Epos, 2009.

[3] Bertsekas, Dimitri P., *Dynamic Programming and Optimal Control*, (Third Edition), Athena Scientific, 2005.

[4] Bertsekas, Dimitri P., Shreve, Steven E., *Stochastic Optimal Control*, Athena Scientific, 1996.

[5] Birge, J., Louveaux, F., *Introduction to Stochastic Programming*, Springer, 2001.

[6] Sethi, S.P., Thompson, G.L., *Optimal Control Theory, Application to Management Science and Economics*, Springer, 2006.

[7] Žáková, K., *Základy práce v Matlabe*, STU 2006.

[8] Register, Andy H., *A Guide to MATLAB, Object-Oriented Programming*, SciTech Publishing Inc., 2007.

[9] Hunt, Brian R., Lipsman, Ronald L., Rosenberg, Jonathan M., *A Guide to MATLAB*, Cambridge University Press, 2001.

[10] Wilson, Howard B., Turcotte, Louis H., Halpern, David, *Advanced Mathematics and Mechanics Applications Using*, Chapman & Hall/CRC Press, 2003.

[11] Anderson, Patrick L., *Business Economics and Finance with MATLAB, GIS, and Simulation Models*, Chapman & Hall/CRC Press, 2005.

[12] Otto, S.R., Denier, J.P., *An Introduction to Programming and Numerical Methods in Mathlab*, Springer, 2005.

[13] Martinez, Wendy L., Martinez, Angel R., *Exploratory Data Analysis with MATLAB*, Chapman & Hall/CRC Press, 2003.

[14] Yang, W.Y., Cao, W., Chung, T.S., Morris, J., *Applied Numerical Methods using MATLAB*, Wiley-Interscience, 2005.

[15] *MATLAB® 7 Graphics*
web: www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/graphg.pdf

Kapitola

Príloha

Zdrojové kódy programov v Matlabe.

A Bankomat

```
% T - casove obdobie (dni)
% c - naklady na 1 Euro ulozene cez noc v bankomate
% k - fixny poplatok za 1 naplnenie
% kv - poplatok za vlozenie balika penazi v hodnote 'value' eur
% value - hodnota balika penazi na doplnanie
% x0 - pociatocny stav
% z - dopyt po vyberoch, peniazoch, nahodna premenna
%       z.h - hodnoty
%       z.p - pravdepodobnosti

T=7; c=0.03; k=0.2; kv=0.05; value=1; x0=0;
% rovnomerne rozdelenie na intervale [1,3]
z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
% z.h=[]; z.p=normpdf(z.h,,); z.p=z.p/sum(z.p)*1; % "normalne" mu=, sigma=

xmax=10; % technicka horna hranica objemu penazi v bankomate
xmin=0; % napr. zakonom stanovena dolna hranica objemu penazi v bankomate
umin=0; umax=fix(xmax/value); % pripustne hodnoty riadenia, naplnenie bankomatu

% vytvorim matice
% V - hodnotova funkcia, v - optimalna spatna vazba
V=zeros(T+1,xmax+1); v=zeros(T,xmax+1);
V(:,:)=inf; V(T+1,:)=0;

for i=T:(-1):1
```

```
%    switch mod(i,7)
%        case 1
%            z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
%        case 2
%            z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
%        case 3
%            z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
%        case 4
%            z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
%        case 5
%            z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
%        case 6
%            z.h=[2:5]; z.p=1/length(z.h)*ones(1,length(z.h));
%        case 0
%            z.h=[2:5]; z.p=1/length(z.h)*ones(1,length(z.h));
%        otherwise
%            error('This is impossible')
%    end
    for x=0:xmax
        for u=0:umax
            f = x + value*u - z.h;              % stavova rovnica
            f0 = c*x + k*sign(u) + kv*value*u;  % ucelova funkcia
            if (f>=xmin) & (x+value*u<=xmax)
                if (f0 + V(i+1,f+1))*z.p' < V(i,x+1) % hladam minimum
                    V(i,x+1) = (f0 + V(i+1,f+1))*z.p';
                    v(i,x+1) = u;
                end
            end
        end
    end
end

% vycislena optimalna hodnota
opt_hodnota=V(1,x0+1)

% vykreslenie
% casovy priebeh optimalnej spatnej vazby
PriebehOSV(T,xmax,v,1,1,T)
% hladiny optimalnej vazby
KresliHladiny(T,xmax,v,1)
```

B Bankomat, verzia s pokutou

```
% T - casove obdobie (dni)
% c - naklady na 1 euro ulozene cez noc v bankomate
% k - fixny poplatok za 1 naplnenie
% kv - poplatok za vlozenie balika penazi v hodnote 'value' eur
% value - hodnota balika penazi na doplnanie
% x0 - pociatocny stav
% S - strata suvisiaca s neuspokojenym dopytom
% z - dopyt po vyberoch, peniazoch, nahodna premenna
%       z.h - hodnoty
%       z.p - pravdepodobnosti

T=7; c=0.03; k=0.2; kv=0.05; value=1; x0=0; S=0.5; pp=0;
% rovnomerne rozdelenie na intervale [1, 3]
z.h=[1:3]; z.p=1/length(z.h)*ones(1,length(z.h));
% z.h=[]; z.p=normpdf(z.h,,); z.p=z.p/sum(z.p)*1; % "normalne" mu=, sigma=

xmax=5; % technicka horna hranica objemu penazi v bankomate
xmin=0; % napr. zakonom stanovena dolna hranica objemu penazi v bankomate
umin=0; umax=fix(xmax/value); % pripustne hodnoty riadenia, naplnenie bankomatu

% vytvorim matice
% V - hodnotova funkcia, v - optimalna spatna vazba
V=zeros(T+1,xmax+1+max(z.h)); v=zeros(T,xmax+1+max(z.h));
V(:,:)=inf; V(T+1,:)=0;

for i=T:(-1):1
    for x=-max(z.h):1:xmax
        for u=0:umax
            f = max(x,0) + value*u - z.h; % stavova rovnica
%             ucelova funkcia s roznymi typmi sankcii
f0 = c*max(x,0)+k*sign(u)+kv*value*u + S*abs(sign(min(max(x,0)+value*u-z.h-D,0)));
% f0 = c*max(x,0)+k*sign(u)+kv*value*u + S*abs(min(max(x,0)+value*u-z.h-D,0));
% f0 = c*max(x,0)+k*sign(u)+kv*value*u + S.^abs(min(max(x,0)+value*u-z.h-D,0));
            if (x+value*u)<=xmax
            if (f0 + V(i+1,f+1+max(z.h)))*z.p' < V(i,x+1+max(z.h)) % hladam minimum
                        V(i,x+1+max(z.h)) = (f0 + V(i+1,f+1+max(z.h)))*z.p';
                        v(i,x+1+max(z.h)) = u;
            end
            end
        end
    end
```

```
end

% vycislena optimalna hodnota
opt_hodnota=V(1,x0+1+max(z.h));
```

C Obchod

```
% T - casove obdobie (dni)
% p - predajna cena vyrobku
% n - nakupna cena vyrobku
% d - naklady za jednu prepravku
% D - kapacita jednej prepravky
% K - kapacita skladu
% x0 - pociatocne zasoby
% z - dopyt po vyrobku, nahodna premenna
%       z.h - hodnoty
%       z.p - pravdepodobnosti

T=7; p=2.5; n=1.5; D=5; d=0.5; s=0.2; x0=0;
% z.h=[1:10]; z.p=1/length(z.h)*ones(1,length(z.h)); % rovnomerne
z.h=[0:10]; z.p=normpdf(z.h,5,2); z.p=z.p/sum(z.p)*1; % "normalne" mu=5, sigma=2

xmin=0; xmax=10; % kapacita skladu
umin=0; umax=xmax; % pripustne hodnoty riadenia, pocet objednanych vyrobkov

% vytvorim matice
% V - hodnotova funkcia, v - optimalna spatna vazba
V=zeros(T+1,xmax+1); v=zeros(T,xmax+1);
V(:,:)=-inf; V(T+1,:)=0;

for i=T:(-1):1
    for x=0:xmax
        for u=0:umax
            f = max(x+u-z.h, xmin);                          % stavova rovnica
            f0 = p*min(x+u, z.h) - n*u - d*ceil(u/D) - s*x;  % ucelova funkcia
            if (x+u<=xmax)
                if (f0 + V(i+1,f+1))*z.p' > V(i,x+1)         % hladam maximum
                    V(i,x+1) = (f0 + V(i+1,f+1))*z.p';
                    v(i,x+1) = u;
                end
            end
        end
    end
```

```
end

% vycislena optimalna hodnota
opt_hodnota=V(1,x0+1)
```

D Hazardný hráč, verzia 1

```
% T - pocet kol hry, casove obdobie
% p - penazna odmena za vyhru
% C - kolko potrebujem mat na konci zetonov, aby som vyhral
% u - riadenie, kolko stavim, najviac x
% x0 - pociatocny kapital
% z - oznacuje vyhru/prehru a ich pravdepodobnosti, nahodna premenna
%       z.h - hodnoty
%       z.p - pravdepodobnosti

T=3; p=1; C=5; z.h=[1,-1]; z.p=[2/3,1/3]; x0=3;
xmax=C; % celkom mi staci pocitat po tuto hranicu
% xmax=2^T*x0; % celkom mi staci pocitat po tuto hranicu

% vytvorim matice
% V - hodnotova funkcia, v - optimalna spatna vazba
V=zeros(T+1,xmax+1); v=zeros(T,xmax+1);
V(:,:)=-inf;
for x=0:xmax
    V(T+1,x+1)=p*sign(max(x-(C-1),0));
%     V(T+1,x+1)=x;
end

for i=T:(-1):1
    for x=0:xmax
        vsetky(i,x+1).riesenia=[];  % zaznamenava vsetky hodnoty OSV
        for u=0:x
            f0=0;                               % ucelova rovnica
            f = min(max(x + z.h*u,0), xmax);    % stavova rovnica
            if (f0+V(i+1,f+1))*z.p' >= V(i,x+1)
                if (f0+V(i+1,f+1))*z.p' > V(i,x+1)  % hladam maximum
                    V(i,x+1) = (f0+V(i+1,f+1))*z.p';
                    v(i,x+1) = u;
                    vsetky(i,x+1).riesenia=u; d=1;
                else
                    vsetky(i,x+1).riesenia=[vsetky(i,x+1).riesenia u]; d=d+1;
                end
```

```
            end
        end
        pr(i,x+1)=d; % matica, udava pocet rieseni
    end
end
pr

% pre zadany cas(I) a stav(J) urci vsetky hodnoty optimalnej spatnej vazby
I=2; J=4;
vsetky(I+1,J+1).riesenia

% vycislena optimalna hodnota
opt_hodnota=V(1,x0+1)
```

E Hazardný hráč, verzia 2

```
% T - pocet kol hry, casove obdobie
% u - riadenie, aku cast kapitalu stavim
% x0 - pociatocny kapital
% z - oznacuje vyhru/prehru a ich pravdepodobnosti, nahodna premenna
%       z.h - hodnoty
%       z.p - pravdepodobnosti
% N - nami volena jemnost diskretizacie pre stav x -> 1/N
% M - nami volena jemnost diskretizacie pre riadenie u -> 1/M
% d - premenna, v ktorej zaznamenavam pocet rieseni
% premenne zacinajuce pismenom "C" uvadzaju vysledky pre celociselne
% pripady stavov

T=3; z.h=[1,-1]; z.p=[3/4,1/4]; x0=1; N=10; M=5; d=0;
xmax=2^T*x0; % max objem kapitalu pocas hry, viac za T kol nedosiahnem
xmaxn=xmax*N;
umin=0; umax=1;

% vytvorim matice
V=zeros(T+1,xmaxn+1); v=zeros(T,xmaxn+1);
V(:,:)=-inf;
for x=0:xmaxn
%     V(T+1,x+1)=sqrt(x/N);
    V(T+1,x+1)=x/N;
end

for i=T:(-1):1
    for x=0:xmaxn
```

```
        vsetky(i,x+1).riesenia=[];  % zaznamenava vsetky hodnoty OSV
        for u=0:(1/M):1
            f0 = 0;                                 % ucelova rovnica
            f = x+u*x*z.h;                          % stavova rovnica
            if f<=xmaxn
                f = round(f);
                if (f0+V(i+1,f+1))*z.p' >= V(i,x+1)
                if (f0+V(i+1,f+1))*z.p' > V(i,x+1)  % hladam maximum
                    V(i,x+1) = (f0+V(i+1,f+1))*z.p';
                    v(i,x+1) = u;
                    vsetky(i,x+1).riesenia=u; d=1;
                else
                    vsetky(i,x+1).riesenia=[vsetky(i,x+1).riesenia u]; d=d+1;
                end
            end
            end
        end
        pr(i,x+1)=d; % matica, udava pocet rieseni
    end
end
pr;

% vycislena optimalna hodnota
opt_hodnota=V(1,x0*N+1)
% udaje pre celociselne stavy
Cv=[]; CV=[]; Cpr=[];
for x=0:xmaxn
    if mod(x,N)==0
        Cv=[Cv v(:,x+1)];
        CV=[CV V(:,x+1)];
        Cpr=[Cpr pr(:,x+1)];
        for i=1:T
            Cvsetky(i,x/N+1).riesenia=vsetky(i,x+1).riesenia;
        end
    end
end
% pre zadany cas(I) a stav(J) urci hodnoty optimalnej spatnej vazby
% I=1; J=1;
% vsetky(I+1,J+1).riesenia;
% Cvsetky(I+1,J+1).riesenia;

% vykreslenie optimalnej spatnej vazby
KresliHladiny(T,xmax,Cv,1)
```

```
% ocakavana vyhra vzhladom na x0
figure
plot(0:1/N:xmax,V(1,:),'LineWidth',1.5)
xlabel('počet eur na začiatku (x0)','FontSize',12)
ylabel('očakávaná výhra (eur)','FontSize',12)
% grid
```

F Vykreslenie1

```
function PriebehOSV(T,xmax,v,f,p,d)
% casovy priebeh optimalnych spatnych vazieb pre jednotlive stavy
if f==1
%   ak je parameter f==1, spravi novy obrazok
    figure;
end
nulove=[];
% plot(0:T-1,v,'LineWidth',1)
% parameter d oznacuje, kolko poslednych dni obdobia vykresli
plot(T-d:T-1,v(T-d+1:T,:),'LineWidth',1)
for j=T-d+1:1:T-1
    for i=0:xmax
%       parameter p mi povie, ako 'husto' ktore pozicie vypise text s
%       prislusnou hodnotou stavu
        if v(j+1,i+1)~=0 & mod(j,p)==0
            h=text(j,v(j+1,i+1),num2str(i));
            set(h,'FontSize',9,'BackgroundColor',[1 1 1])
        end
        if v(j+1,i+1)==0 & j==1
            nulove=[nulove '  ' num2str(i) '  '];
        end
    end
end
text(p-0.05,0,nulove,'FontSize',9,'BackgroundColor',[1 1 1])
set(gca, 'XTick', 0:1:T-1); set(gca,'XTickLabel',[0:1:T-1]);
xlabel('obdobie (t)','FontSize',11)
ylabel('optimálna spätná väzba (v}x|)','FontSize',11)
title('Časový priebeh optimálnych spätných väzieb','FontSize',12)
% grid
axis([T-d T-1 -0.1 xmax])
```

G Vykreslenie2

```
function KresliHladiny(T,xmax,v,f)
% vykresli hladiny optimalnych vazieb pre kazdy stav a cas
% az po cas T, hoci v tom uz nie su optimalne spatne vazby definovane
v=[v; zeros(1,xmax+1)];
if f==1
%   ak je parameter f==1, spravi novy obrazok
    figure;
end
[C,h]=contour(v', 1:max(max(v)), 'LineWidth',1);
h = clabel(C,h);
set(h,'FontSize',10,'BackgroundColor',[1 1 1],'EdgeColor',[1 1 1])
set(gca, 'XTick', 1:1:T+1); set(gca,'XTickLabel',[0:1:T]);
set(gca, 'YTick', 0:1:xmax+1); set(gca,'YTickLabel',[-1:1:xmax]);
xlabel('obdobie (t)','FontSize',11)
ylabel('stavová premenná (x)','FontSize',11)
title('Hladiny optimálnych spätných väzieb','FontSize',12)
% grid
axis([1 T+1 0.9 xmax+1])
```