Optimálne riadenie

Viacetapové rozhodovacie procesy v ekonómii a financiách

Margaréta Halická
Pavel Brunovský
Pavol Jurča

EPOS

Bratislava 2009

Kniha predstavuje komplexný výklad teórie optimálneho rozhodovania vo viacerých časových alebo iných etapách. Ukazuje možnosti využitia teórie na formuláciu a riešenie úloh z oblasti ekonómie, financií a manažmentu. Od čitateľa sa predpokladá znalosť vysokoškolskej matematiky na úrovni matematických, ekonomických, technických či manažérskych odborov. Môže slúžiť aj ako učebný text v spomínaných odboroch.

S prispením grantu Nadácie Tatra banky pre podporu vzdelávania vydalo Nakladateľstvo EPOS v roku 2009.

Recenzenti: Mgr. Soňa Kilianová, PhD., RNDr. Juraj Zeman, CSc.
Jazyková úprava: Mária Goceliaková
Náklad 500 kusov, 1. vydanie, počet strán 200.

Obsah

Predhovor **1**

1 Úvod do diskrétnej teórie optimálneho riadenia **6**

1.1 Príklady 6
1.2 Štandardný tvar úlohy optimálneho riadenia 9
1.3 Ďalšie príklady 14
1.4 Základné problémy diskrétnej teórie 18
1.5 Úlohy na riešenie 22

2 Dynamické programovanie pre diskrétne úlohy **27**

2.1 Úloha s pevným časom 28
2.1.1 Vnorenie úlohy a hodnotová funkcia 28
2.1.2 Rovnica dynamického programovania ako nutná a postačujúca podmienka optimality . . 31
2.1.3 Princíp optimality a optimálna spätná väzba . 35
2.1.4 Poznámky k využitiu rovnice dynamického programovania 39
2.1.5 Riešenie úloh 42
2.1.6 Úlohy na riešenie 52
2.2 Úloha s nekonečným horizontom 57

2.2.1 Autonómna úloha 58
2.2.2 Úloha s voľným časom 59
2.2.3 Úloha s nekonečným horizontom 60
2.2.4 Metódy riešenia 61
2.2.5 Úlohy na riešenie 66
2.3 Lineárno-kvadratická úloha 68
2.3.1 Riccatiho diferenčná rovnica pre úlohu s pevným časom 68
2.3.2 Riccatiho rovnica pre úlohu s nekonečným časovým horizontom 71
2.3.3 Úlohy na riešenie 74
2.4 Iné optimalizačné úlohy 76
2.4.1 Úlohy optimálneho riadenia s ohraničeniami na stavovú premennú 76
2.4.2 Úlohy optimálneho riadenia s inými typmi účelových funkcií 79
2.4.3 Úlohy z iných oblastí optimalizácie 82
2.4.4 Úlohy na riešenie 88
2.5 Stochastické úlohy 88
2.5.1 Formulácia stochastickej úlohy optimálneho riadenia 89
2.5.2 Rovnica dynamického programovania 95
2.5.3 Dôkaz Vety 2.7 98
2.5.4 Riešenie úloh 103
2.5.5 Úlohy na riešenie 118

3 Princíp maxima pre diskrétne úlohy 123
3.1 Označenia, symbolika a formulácia úlohy 124
3.2 Nutné podmienky optimality 128
3.2.1 Odvodenie a formulácia nutných podmienok . . 129

3.2.2 Poznámky 131
3.2.3 Iné odvodenie nutných podmienok 134
3.3 Princíp maxima 138
3.4 Nutné podmienky pre úlohu bez ohraničení 142
3.4.1 Formulácia a odvodenie nutných podmienok . . 142
3.4.2 Súvis rovnice dynamického programovania a nutných podmienok optimality 145
3.4.3 Ekonomická interpretácia adjungovanej premennej 148
3.4.4 Eulerova rovnica 149
3.5 Úloha s nekonečným horizontom 152
3.6 Riešenie úloh 156
3.7 Úlohy na riešenie 177

4 Dodatky **180**
4.1 Princíp maxima v nelineárnom programovaní 180
4.1.1 Štandardná úloha nelineárneho programovania 180
4.1.2 Nutná podmienka optimality pre úlohu nelineárneho programovania 181
4.1.3 Princíp maxima v nelineárnom programovaní . 184
4.2 Obálková veta 186
4.3 Diferenčné rovnice a ich systémy 187
4.4 Elementy z teórie pravdepodobnosti 190

Literatúra **195**

Použité označenia **197**

Register **198**

Predhovor

Lístok unášaný Dunajom nevie, v ktorom ramene jeho delty sa ocitne. A nemôže to ani ovplyvniť. Rybka môže. *Volí* si rameno, podľa toho, kde očakáva viac potravy alebo lepšie podmienky pre svoje potomstvo. Hľadá najlepšiu z možností, *optimalizuje.* Živé tvory sa od mŕtvej hmoty odlišujú okrem iného aj tým, že sa vedia rozhodovať. Čím vyšší je ich vývojový stupeň, tým zložitejších rozhodnutí sú schopné.

Človek ako tvor na najvyššom vývojovom stupni je v rozhodovaní najďalej. Od ostatnej prírody ho odlišuje jeho tvorivosť – vie sa rozhodovať aj v dilemách, s ktorými sa nikdy nestretol. Niekedy mu na to stačí „zdravý sedliacky rozum“, intuícia. Inokedy je rozhodovanie zložitejšie a nezaobíde sa bez dlhšieho usudzovania, tvorby modelov danej situácie a bez výpočtov. Tu nachádza svoje uplatnenie matematika. Otázkam rozhodovania a optimalizácie sa venuje jej časť, *operačná analýza.* Patria k nej napríklad lineárne, nelineárne a celočíselné programovanie, sieťová optimalizácia a mnohé iné.

Osobitné miesto v operačnej analýze majú úlohy, v ktorých nestačí jedno rozhodnutie, ale treba urobiť rad rozhodnutí vo viacerých na seba nadväzujúcich *etapách.* Rozhodnutie v jednej etape pritom ovplyvňuje nielen výsledok v neskoršej etape ale aj možnosti ďalšej voľby. Napríklad racionálny a perspektívne rozmýšľajúci jedinec, ktorý má k dispozícii väčšiu finančnú čiastku, nechá časť z nej zhodnotiť, aby z nej mal úžitok neskôr. Dobrý príklad nájdeme v Starom zákone: Jozef prikázal odkladať úrodu v siedmich úrodných rokoch, aby vytvoril zásoby na neúrodné roky. Analógia s dôchodkovým zabezpečením na starobu je očividná.

Oblasť operačnej analýzy, ktorá sa úlohám viacetapového rozhodovania venuje, vznikla v druhej polovici minulého storočia. Najčastejšie sa nazýva *teóriou optimálneho riadenia*. Termínom *riadenie* (v angličtine *control*) sa totiž zvykne nazývať veličina, ktorá je nástrojom rozhodovania. Okrem toho sa však používajú alternatívne pomenovania ako *teória optimálnych procesov*, *dynamická optimalizácia* alebo *dynamické programovanie*. Posledný názov však nie je celkom priliehavý, pretože dynamické programovanie predstavuje iba jeden z prístupov k problematike.

Počiatočný podnet na vznik teórie dali rozvoj automatizovaného riadenia a kozmonautiky v bývalom Sovietskom zväze a v USA. To je aj príčina, prečo sa historicky skôr vyvíjala jej *spojitá* verzia, v ktorej sa rozhodnutia robia nepretržite v čase. Kľúčovým výsledkom sa stal *Pontrjaginov princíp maxima* (L. S. Pontrjagin), ktorý vznikol v ZSSR. Paralelne s Pontrjaginovou teóriou sa v USA rozvíjal alternatívny prístup k riešeniu úloh optimálneho riadenia, motivovaný skôr ekonomickými úlohami. Spája sa najmä s menom R. Bellmana. Na rozdiel od Pontrjaginovej spojitej teórie sa venuje predovšetkým rozhodovaniu v oddelených *diskrétnych* časových okamihoch, etapách. Vychádza z jednoduchého princípu optimality a vedie k rekurzívnej metóde výpočtu *optimálneho riadenia*, Bellmanom nazvanej metódou *dynamického programovania*. Časom sa vyjasnil súvis dynamického programovania s Pontrjaginovým princípom maxima, resp. variačným počtom.

Teória optimálneho riadenia nachádzala sprvu uplatnenie najmä v technických odboroch ako kozmonautika, chemické inžinierstvo, elektrotechnika, robotika. V posledných desaťročiach však nachádza stále viac aplikácií v ekonomických vedách a výpočtových financiách, napríklad v makroekonomickej teórii rastu, v mikroekonomickej teórii firmy a spotrebiteľa alebo v manažmente investičných portfólií. Pre sporiteľa v druhom dôchodkovom pilieri môže byť zaujímavá úloha optimálneho výberu fondu. Tieto oblasti aplikácií spätne kládli požiadavky na rozvoj špecifických oblastí teórie a dávali impulzy na jej ďalší vývoj.

O teórii optimálneho riadenia jestvuje v svetovej literatúre rad kníh rozličnej povahy. Niektoré sa venujú rigoróznemu výkladu teórie, iné aplikáciám v niektorej oblasti. Od vydania jedinej pôvodnej knihy na Slovensku [5] uplynulo bezmála 30 rokov a neveľmi vydarený preklad [12] je ešte starší. V češtine podľa našich informácií vyšla iba krátka úvodná brožúrka [20].

Naša kniha charakterom vychádza z [5]. Kým [5] nachádzala motivácie a aplikácie skôr v technických odboroch, v tomto texte kladieme väčší dôraz na aplikácie v ekonomickej teórii a finančníctve. Rozsah riešených i neriešených úloh s aplikačnou zložkou je oproti [5] oveľa vyšší. Ťažko by sme aj vo svetovej literatúre hľadali dielo, v ktorom by boli teoretická a aplikačná také vyvážené. Zaoberáme sa pritom výlučne *diskrétnymi* úlohami optimálneho riadenia. Sú to úlohy, v ktorých sa rozhodnutia robia v oddelených časových okamihoch, etapách. Takéto úlohy vznikajú jednak modelovaním časovo prirodzene diskrétnych javov, jednak diskretizáciou spojitých úloh. Spojitú teóriu máme v úmysle spracovať v budúcnosti v osobitnej knihe.

Čo sú diskrétne úlohy optimálneho riadenia sa dozvieme v prvej kapitole. Druhá kapitola je venovaná dynamickému programovaniu, ktoré je v prípade diskrétnych úloh účinným nástrojom na numerické riešenie úloh. Jedna jej podkapitola sa venuje stochastickému dynamickému programovaniu. Tretia kapitola sa zaoberá princípom maxima, prevzatým zo spojitej teórie. Napriek tomu, že v diskrétnej teórii platí iba s určitými obmedzeniami, predstavuje užitočný prostriedok kvalitatívnej analýzy. Posledná kapitola obsahuje dodatky, v ktorých sú zhrnuté poznatky z nelineárneho programovania, diferenčných rovníc a teórie pravdepodobnosti potrebné v druhej a tretej kapitole.

Výklad je určený predovšetkým čitateľovi, ktorý sa chce do hĺbky oboznámiť so základmi diskrétnej teórie optimálneho riadenia. V porovnaní s mnohými inými textami, kde sa viaceré pojmy a princípy vykladajú viac na intuitívnej úrovni, táto kniha sa snaží o matematickú presnosť a všetky hlavné tvrdenia sú dostatočne matematicky zdôvodnené. Text je členený tak, že umožňuje čitateľovi vynechať niektoré dôkazy, resp. obmedziť sa len na tvrdenia pre jednoduchšie situácie. Od čitateľa sa vyžaduje znalosť základov matematickej analýzy funkcií viacerých premenných. Výhodou je, ak čitateľ má základné poznatky z teórie nelineárneho programovania a teórie pravdepodobnosti. Nie je to však podmienkou, pretože všetky nevyhnutné pojmy a tvrdenia z týchto oblastí sú uvedené v dodatkoch.

V knihe sme sa snažili ilustrovať teóriu a použitie v nej vyvinutých nástrojov na širokom spektre konkrétnych úloh s pôvodom v ekonomickej, manažérskej alebo finančnej praxi. Príkladom takýchto úloh je napr. úloha o optimálnej spotrebe, úloha o optimálnom rozdelení prostriedkov medzi rôzne investičné projekty, úloha o optimálnom investovaní do fondov II. piliera dôchodkového sporenia alebo rôzne typy úloh o optimálnej obnove zariadení. Úlohy sú zväčša zjednodušené do takej miery, aby ich riešenie nebolo priveľmi časovo náročné. Pomáha to lepšiemu pochopeniu teórie. Problémy z praxe sa často nedajú takto jednoducho riešiť najmä ak v sebe zahŕňajú nejakú neurčitosť. Preto sa v knihe uvádzajú aj ukážky, ako preberanú teóriu aplikovať na vytvorenie programov v Matlabe vhodných aj na riešenie rozsiahlejších úloh. Popri týchto úlohách obsahuje kniha aj úlohy, ktoré dávajú návody na rozšírenie preberanej teórie.

Ďalšie príklady majú za cieľ naznačiť rôznorodosť problémov, ktoré je možné riešiť metódami knihy. Umožní to čitateľovi získať cit, na aké typy problémov možno v praxi túto teóriu aplikovať. Zároveň sa v prvej kapitole pomerne detailne zaoberáme tým, ako úlohy z praxe správne formulovať v tvare, ktorý nám umožní vyriešiť ich pomocou preberanej teórie.

Časť príkladov je vyriešená úplne, riešenie iných sme nechali ako cvičenie čitateľovi. Niektoré úlohy sa vyskytujú vo viacerých modifikáciach v rôznych častiach textu. Umožňuje to porovnať navzájom rozličné metódy teórie. Príkladom je úloha o optimálnej spotrebe. Venuje sa jej pomerne veľa priestoru aj preto, lebo je základom súčasných dynamických makroekonomických modelov správania sa ekonomiky.

Na záver sa chceme poďakovať všetkým, ktorí prispeli k vzniku knihy v jej konečnej podobe. Predovšetkým ďakujeme odborným lektorom Mgr. Soni Kilianovej, PhD. a RNDr. Jurajovi Zemanovi, CSc. ako aj jazykovej lektorke Márii Goceliakovej za pozorné prečítanie rukopisu. Ich cenné pripomienky viedli k odstráneniu mnohých nedostatkov a chýb v texte. Ďalej ďakujeme viacerým študentom magisterského programu Ekonomická a finančná matematika, ktorí svojimi otázkami a pripomienkami prispeli k čitateľnosti a zrozumiteľnosti výkladu. Úprimné poďakovanie patrí doc. Milanovi Hamalovi, CSc. za jeho poznámky ku časti o nelineárnom programovaní a prof. Dr. Mikulášovi Luptáčikovi za povzbudenie a podnety k napísaniu úvodu. V neposlednom rade ďakujeme nášmu kolegovi doc. RNDr. Danielovi Ševčovičovi, CSc. za všestrannú pomoc a podporu ako aj za cenné rady k technike písania knihy.

Autori

Kapitola 1

Úvod do diskrétnej teórie optimálneho riadenia

V tejto kapitole vymedzíme predmet diskrétnej teórie optimálneho riadenia. Najskôr uvedieme niekoľko konkrétnych úloh, ktoré nám pomôžu orientovať sa vo všeobecnej formulácii štandardnej úlohy optimálneho riadenia.

1.1 Príklady

Príklad 1.1. Optimálne rozdeľovanie zdrojov. Investor má k dispozícii kapitál o počiatočnej veľkosti $a > 0$ a dve možnosti jeho investovania: do ropných vrtov a do nákupu nehnuteľností. Na začiatku každého roka sa má rozhodnúť, ako na daný rok rozdelí kapitál, ktorý má práve k dispozícii, medzi uvedené dva spôsoby investovania. Kapitál veľkosti y, investovaný do ropných vrtov, mu prinesie za rok zisk gy, kde $g > 0$; ak ho investuje do nehnuteľností, získa hy, kde $h > 0$. Zisk už ďalej neinvestuje. Kapitál sa za rok amortizuje, a to koeficientom $0 \leq b < 1$ pre investície do ropných vrtov a koeficientom $0 \leq c < 1$ pre investície do nehnuteľností. Súčet týchto dvoch amortizovaných kapitálov na konci roku tvorí kapitál k dispozícii na investovanie v nasledujúcom roku. Cieľom investorovho rozhodovania je maximalizovať celkový zisk za plánovacie obdobie k rokov.

Označme $i = 0, \dots, k-1$ poradové číslo roka. Ďalej označme x_i veľkosť kapitálu, ktorý je k dispozícii na prerozdeľovanie na začiatku i-teho roku. Nech $u_i \in [0,1]$ je pomer veľkosti kapitálu určeného na investovanie do ropných vrtov k jeho celkovému objemu x_i. To znamená, že z hodnoty x_i určenej na prerozdeľovanie v i-tom roku bude časť $u_i x_i$ investovaná do ropných vrtov a zvyšok, t.j. $(1-u_i)x_i$, do nákupu nehnuteľností. Toto rozhodnutie prinesie investorovi na konci roka zisk o veľkosti $gu_i x_i + h(1-u_i)x_i$ a k dispozícii na investovanie v ďalšom, t.j. v $(i+1)$-om roku, bude mať kapitál o veľkosti $x_{i+1} = bu_i x_i + c(1-u_i)x_i$. Cieľom investorovho rozhodovania je

$$\begin{aligned}
&\text{maximalizovať} && \sum_{i=0}^{k-1} [gu_i x_i + h(1-u_i)x_i] && (1.1)\\
&\text{pri podmienkach} && x_{i+1} = bu_i x_i + c(1-u_i)x_i,\ i = 0, \dots, k-1, && (1.2)\\
& && x_0 = a, && (1.3)\\
& && u_i \in [0,1], \quad i = 0, \dots, k-1, && (1.4)
\end{aligned}$$

kde a, b, c, g, h sú dané konštanty a maximum hľadáme vzhľadom na premenné $u_i,\ i = 0, \dots, k-1$ a $x_i,\ i = 0, \dots, k$.

Dostali sme úlohu matematického programovania s určitou špeciálnou štruktúrou, ktorá je charakteristická pre tzv. viacetapové rozhodovacie procesy. Celý proces rozhodovania (u nás prerozdeľovania a investovania kapitálu) je rozdelený na určité *etapy* (u nás roky), ktoré číslujeme pomocou $i = 0, \dots, k-1$. Stav procesu na začiatku i-tej etapy je opísaný *stavovou premennou* x_i (kapitál určený na prerozdeľovanie) a do procesu možno zasahovať zvonku rozhodnutiami u_i (určujúcimi rozdelenia kapitálu medzi dva spôsoby investovania), ktoré nazývame *riadiacou premennou.* Hodnoty stavovej a riadiacej premennej v i-tej etape určujú výnos i-tej etapy (i-ty sčítanec v účelovej funkcii (1.1)) a hodnotu stavovej premennej na začiatku nasledujúcej $(i+1)$-ej etapy (pravá strana diferenčnej rovnice (1.2)). Poznamenajme, že rozhodnutie, ktoré maximalizuje celkový zisk, nemusí maximalizovať čiastkové zisky jednotlivých etáp.

Rozdelenie optimalizačných premenných na stavové a riadiace premenné, určitá separovateľnosť účelovej funkcie, ako aj výskyt ohraničujúcich podmienok v tvare diferenčnej rovnice sa budú vyskytovať aj v nasledujúcej úlohe.

Príklad 1.2. Optimálna spotreba. Predpokladajme, že vlastníme kapitál, ktorého veľkosť na začiatku i-teho mesiaca označíme x_i. Kapitál x_i nám prinesie za i-ty mesiac úrok veľkosti rx_i. Na konci každého mesiaca sa môžeme rozhodnúť, aké množstvo kapitálu spotrebujeme. Spotreba vo výške u_i nám prinesie (diskontovaný) úžitok veľkosti $(\frac{1}{1+\delta})^i \ln u_i$, kde $\delta > 0$ je mierou netrpezlivosti pri spotrebovávaní.[1] Predpokladáme, že na začiatku procesu máme k dispozícii množstvo kapitálu $a > 0$. Úlohou je určiť spotrebu v každom mesiaci tak, aby hodnota držaného kapitálu na konci procesu trvajúceho k mesiacov nadobudla predpísanú hodnotu $b \geq 0$ a aby celkový diskontovaný úžitok zo spotreby za k mesiacov bol maximálny.

Dostávame úlohu

$$\text{maximalizovať} \quad \sum_{i=0}^{k-1} \left(\frac{1}{1+\delta}\right)^i \ln u_i \tag{1.5}$$

$$\text{pri podmienkach} \quad x_{i+1} = (1+r)x_i - u_i, \ i = 0, \dots, k-1, \tag{1.6}$$

$$x_0 = a, \tag{1.7}$$

$$x_k = b, \tag{1.8}$$

kde a, b, δ, r, sú dané konštanty a maximum hľadáme vzhľadom na premenné u_i, $i = 0, \dots, k-1$ a x_i, $i = 0, \dots, k$.

[1] Voľbou rastúcej konkávnej funkcie $\ln u_i$ ako miery úžitku vyjadrujeme to, že jednotkový nárast spotreby predstavuje pre spotrebiteľa menší nárast úžitku pri väčších hodnotách spotreby ako pri menších. Diskontom zase vyjadrujeme, že spotrebiteľ uprednostňuje okamžitú spotrebu pred odloženou.

Aj v tomto prípade možno úlohu charakterizovať ako k-*etapový rozhodovací proces*, kde etapami sú jednotlivé mesiace. Stavovou premennou je x_i; riadiacou premennou je u_i, ktorá je vstupom do procesu – voľbou u_i ovplyvňujeme správanie procesu. Správanie sa procesu pri prechode z i-tej etapy do $(i+1)$-ej opisuje *diferenčná rovnica* (1.6), ktorej pravá strana opäť závisí iba od x_i a u_i – hodnoty stavovej a riadiacej premennej v i-tej etape. *Účelová funkcia* sčituje diskontovaný úžitok za jednotlivé etapy, ktorý v každej etape i závisí iba od x_i a u_i.

Podobne ako v predchádzajúcom príklade sa účelová funkcia (1.5) vyznačuje určitou separovateľnosťou a dynamika systému je opísaná diferenčnou rovnicou (1.6), doplnenou o počiatočnú podmienku (1.7). Na rozdiel od Príkladu 1.1 máme pre stavovú premennú aj koncovú podmienku (1.8) a nemáme žiadne ohraničenie na riadiacu premennú typu (1.4).

1.2 Štandardný tvar úlohy optimálneho riadenia

V tejto podkapitole sformulujeme všeobecnú optimalizačnú úlohu, ktorá zahŕňa Príklady 1.1 a 1.2.

Predpokladajme, že máme objekt (systém), ktorého správanie budeme riadiť v priebehu k etáp. Stav systému na začiatku i-tej etapy, $i = 0, \dots, k-1$, je opísaný stavovou premennou $x_i \in X_i$. Správanie objektu v i-tej etape ovplyvňujeme (riadime) pomocou riadiacej premennej $u_i \in U_i$, ktorá je vstupom do systému. Tu X_i je daná množina povolených stavov a U_i je množina povolených hodnôt riadiacej premennej v i-tej etape. Hodnotou x_i a u_i je jednoznačne určená hodnota $x_{i+1} := f_i(x_i, u_i)$, kde f_i je daná funkcia. Výnos v i-tej etape je určený hodnotou $f_i^0(x_i, u_i)$, kde f_i^0 je daná funkcia. Predpokladajme, že na počiatku je hodnota stavovej premennej x_0 rovná zadanej hodnote a a budeme žiadať, aby x_k, hodnota stavovej premennej na konci procesu, bola z danej množiny cieľových stavov C. Úlohou je určiť v každej etape hodnotu riadiacej premennej u_i tak, aby boli splnené všetky podmienky a aby súčet výnosov za všetky etapy bol maximálny.

Úlohu možno zapísať nasledovne:

$$\begin{aligned}
\text{maximalizovať} \quad & \sum_{i=0}^{k-1} f_i^0(x_i, u_i) && (1.9)\\
\text{pri podmienkach} \quad & x_{i+1} = f_i(x_i, u_i), \; i = 0, \dots, k-1, && (1.10)\\
& x_0 = a, && (1.11)\\
& x_k \in C, && (1.12)\\
& u_i \in U_i, \quad i = 0, \dots, k-1, && (1.13)\\
& x_i \in X_i, \quad i = 0, \dots, k-1. && (1.14)
\end{aligned}$$

Maximum hľadáme vzhľadom na premenné u_i, $i = 0, \dots, k-1$ a x_i, $i = 0, \dots, k$. Pri takto sformulovanej úlohe majú premenné x_i a u_i navonok rovnocenné postavenie. Všimnime si však, že voľbou riadiacich premenných u_i sú premenné x_i jednoznačne určené ako riešenia diferenčnej rovnice (1.10) s počiatočnou podmienkou (1.11). Táto skutočnosť sa odráža v chápaní úlohy (1.9)–(1.14) ako úlohy optimálneho riadenia, v následnej terminológii ako aj v použitých metódach na riešenie úloh.

Poďme teraz sformulovať úlohu (1.9)–(1.14) ako úlohu optimálneho riadenia. K tomu budeme potrebovať definovať pojmy riadenia, odozvy na riadenie, prípustného riadenia a optimálneho riadenia.

Riadením budeme nazývať postupnosť hodnôt riadiacich premenných $\mathcal{U} = \{u_0, \dots, u_{k-1}\}$ spĺňajúcu $u_i \in U_i$ pre každé $i = 0, \dots, k-1$. *Odozvou* na riadenie $\mathcal{U}$ pri zadanom počiatočnom stave (1.11) rozumieme postupnosť $\mathcal{X} = \{x_0, \dots, x_k\}$, kde jednotlivé $x_i = x_i(\mathcal{U})$ sú riešením *stavovej rovnice* (1.10) pri danom $\mathcal{U}$ s *počiatočnou podmienkou* (1.11). Je zrejmé, že odozva na určité riadenie môže, ale nemusí spĺňať ohraničenia (1.12) a (1.14). Ak ich spĺňa, tak príslušné riadenie nazývame prípustným. To znamená, že *prípustné riadenie* a jeho odozva spĺňajú všetky ohraničenia úlohy (1.9)–(1.14). Triedu prípustných riadení budeme označovať $\mathcal{P}$. Hodnotu účelovej funkcie v (1.9) teraz môžeme chápať ako hodnotu závislú na voľbe riadenia $\mathcal{U}$, a teda ju budeme označovať

$$J(\mathcal{U}) := \sum_{i=0}^{k-1} f_i^0(x_i(\mathcal{U}), u_i).$$

Úlohou optimálneho riadenia je spomedzi všetkých prípustných riadení $\mathcal{U}$ nájsť také, v ktorom účelová funkcia $J(\mathcal{U})$ nadobúda maximálnu hodnotu. Toto riadenie nazývame *optimálnym.* Úlohu formálne zapíšeme nasledovne:

$$\max_{\mathcal{U}\in\mathcal{P}} J(\mathcal{U}).$$

Poznámka 1.1. Do štandardného tvaru možno sformulovať aj úlohy, v ktorých sa hľadá *minimum* tak, že sa ako účelová funkcia vezme záporne vzatá maximalizovaná funkcia.

Poznámka 1.2. Premenná $i = 0, \ldots, k-1$, ktorá určuje poradové číslo etapy, má spravidla charakter času, preto ju aj tak budeme nazývať. Od času závisia nielen hodnoty riadiacich a stavových premenných, ale závisieť môžu aj dáta úlohy: funkcie f_i, f_i^0 a množiny U_i a X_i. Takéto úlohy nazývame *neautonómne.* Ak tieto dáta nezávisia od i, úloha sa nazýva *autonómna.* Rozlíšiť autonómnu úlohu od neautonómnej bude dôležité najmä v nasledujúcej kapitole. Všimnime si, že Príklad 1.1 viedol na autonómnu úlohu, zatiaľ čo Príklad 1.2 na neautonómnu, pretože $f_i^0(x_i, u_i) = (1+\delta)^{-i} \ln u_i$ závisí od i.

Poznámka 1.3. Niekedy sa v zadaní úlohy nevyskytuje ohraničenie (1.13) resp. (1.14). Vtedy hovoríme o *úlohe bez ohraničení na riadenie*, resp. o *úlohe bez ohraničení na stav.* Riadiace, resp. stavové premenné, môžu vtedy nadobúdať ľubovoľné hodnoty z prirodzeného definičného oboru funkcií f_i a f_i^0. Príklad 1.1 viedol na úlohu s ohraničením na riadenie (viď (1.4)) a je bez ohraničení na stav. Príklad 1.2

viedol na úlohu bez ohraničení na riadenie či stav. Prirodzený definičný obor funkcie ln vystupujúcej v (1.5) špecifikuje obor riadiacej premennej na reálne kladné $u_i > 0$. Obidva motivujúce príklady majú funkcie f_i a f_i^0 definované na reálnych priestoroch resp. ich kladných, či nezáporných podmnožinách, to znamená, že majú kontinuálny charakter. Budeme sa však stretávať aj s úlohami, kde u_i a x_i budú mať, buď vďaka ohraničeniam (1.13), (1.14), alebo vďaka prirodzenému definičnému oboru funkcií f_i a f_i^0, diskrétny charakter.

Poznámka 1.4. Ohraničenie (1.12) je ohraničením na koncový stav. V prípade, že úloha nemá takéto ohraničenie, to znamená, že nijako nie sú obmedzené možné hodnoty x_k, hovoríme o *úlohe s voľným koncom*. V prípade, že množina C pozostáva z jediného bodu, hovoríme o *úlohe s pevným koncom*. Vo všetkých ostatných prípadoch hovoríme o *úlohe s čiastočne pevným koncom*. Zrejme Príklad 1.1 viedol na úlohu s voľným koncom a Príklad 1.2 na úlohu s pevným koncom.

Poznámka 1.5. Vo všetkých motivačných príkladoch uvedených v tejto kapitole majú riadiace a stavové premenné iba jednu zložku, sú *jednorozmerné*. Vo všeobecnosti však môžu byť riadiace a/alebo stavové premenné aj *viacrozmerné*. Ku úlohe s dvojrozmerným stavom vedie napríklad Úloha 1.4 a s viacrozmernými úlohami sa stretneme aj v nasledujúcich kapitolách.

Poznámka 1.6. V štandardnej úlohe maximalizujeme súčet výnosov za jednotlivé etapy, čo vedie k účelovej funkcii tvaru (1.9). Úloha optimálneho riadenia s takouto účelovou funkciou sa v literatúre nazýva *Lagrangeovou*. Niekedy však viacetapové rozhodovacie procesy vedú k potrebe maximalizovať aj nejakú funkciu koncového stavu. Vtedy má účelová funkcia tvar $\sum_{i=0}^{k-1} f_i^0(x_i, u_i) + \phi(x_k)$, resp. len $\phi(x_k)$, kde ϕ je daná funkcia. V prvom prípade hovoríme o úlohe v *Bolzovom* tvare, v druhom o úlohe v *Mayerovom* tvare.

Poznámka 1.7. Veľakrát, najmä v ekonomických úlohách, je výnos $f_i(x_i, u_i)$ za i-tu etapu v tvare $f_i(x_i, u_i) = \beta^i F_i(x_i, u_i)$, kde $F_i(x_i, u_i)$

predstavuje hodnotu výnosu v čase i-tej etapy a $\beta \in (0,1)$ je koeficient, ktorý prepočítava (diskontuje) túto hodnotu do hodnoty na začiatku celého procesu, t.j. do času $i = 0$. Faktor β môže odrážať stratu hodnoty výnosu v dôsledku inflácie, možnosť alternatívne ho zhodnotiť ako vklad s úrokovou mierou $r = \frac{1}{\beta} - 1$ za i-tu etapu (viď Úloha 1.17), ako aj sklon uprednostňovať skoršiu spotrebu, ak $\beta = \frac{1}{1+\delta}$, ako to bolo v prípade úlohy o optimálnej spotrebe v Príklade 1.2.Úlohu, v ktorej je účelová funkcia v špeciálnom tvare $J = \sum_{i=0}^{k-1} \beta^i F_i(x_i, u_i)$, nazývame *úlohou s diskontným faktorom.* Pretože hodnota takto vyjadrenej účelovej funkcie meria súčasnú hodnotu budúcich výnosov, nazýva sa niekedy aj *účelovou funkciou súčasnej hodnoty.*

Je zrejmé, že úlohy s diskontáciou sú v zmysle Poznámky 1.2 neautonómne. Ak však diskontný faktor je jediným zdrojom neautonómnosti v úlohe, tak takéto úlohy nazývame *autonómne s diskontným faktorom.* Takéto pomenovanie má svoje opodstatnenie, pretože pre autonómne úlohy s diskontným faktorom dostaneme rovnako silné výsledky ako pre autonómne úlohy. Autonómnou úlohou s diskontáciou je úloha z Príkladu 1.2.

Poznámka 1.8. V štandardnej úlohe (1.9)–(1.14) je k, ktoré označuje počet etáp, pevne dané. Preto túto úlohu nazývame *úlohou s pevným časom.* Možno však formulovať aj také úlohy optimálneho riadenia, v ktorých sa maximum hľadá aj vzhľadom na k. Takéto úlohy nazývame *úlohami s voľným časom.* V úlohách s voľným časom je riadením každá konečná postupnosť riadiacich premenných ľubovoľnej dĺžky. V ekonomických aplikáciách sa často vyskytujú *úlohy s nekonečným časovým horizontom,* kde riadeniami sú nekonečné postupnosti. S úlohami s voľným časom a s nekonečným horizontom sa stretneme neskôr v ďalšej kapitole. Oba Príklady 1.1 a 1.2 sú úlohami s pevným časom.

Na záver tejto podkapitoly uvedieme ešte poznámku o označovaní a zápise úloh.

Poznámka 1.9. Často bude pre nás výhodné diskrétnu množinu časových okamžikov $\{j, \dots, k\}$ nazývať intervalom a označovať $[j, k]$. Pri zápise úloh budeme zase niekedy používať skrátenú formu, kde slová *maximalizovať*, resp. *minimalizovať* nahradíme tvarom *max*, resp. *min* a slová *pri podmienkach* úplne vynecháme. Podmienky budú oddelené od účelovej funkcie buď dvojbodkou, alebo budú uvedené v novom riadku pod účelovou funkciou.

1.3 Ďalšie príklady

Uveďme ešte niekoľko príkladov, ktoré budú poukazovať na rozmanitosť úloh vedúcich k diskrétnym úlohám optimálneho riadenia. Začneme jednou jednoduchou úlohou, kde riadiaca a stavová premenná budú nadobúdať iba hodnoty z konečných množín.

Príklad 1.3. Odvoz kontajnerov. Firma dostáva železnicou kontajnery. Má auto, ktorým odvezie naraz najviac 2 kontajnery, a môžu ho na stanicu poslať denne najviac jedenkrát. Na nasledujúci týždeň má avizované, že dostane v jednotlivé dni 1, 3, 1, 2, 1 kontajnerov. Jedna cesta na stanicu ju stojí 70 eur, penále za prestoj kontajnera na deň ju stoja 50 eur. Z minulého týždňa nezostali nijaké kontajnery a na konci týždňa majú byť všetky kontajnery odvezené. Treba určiť, v ktoré dni má firma poslať auto na stanicu, aby náklady za prevoz a penále boli minimálne.

V tomto prípade etapou budú jednotlivé dni, ktoré budeme číslovať pomocou $i = 0, \dots, 4$. Označíme
x_i - počet neodvezených kontajnerov z predchádzajúceho dňa,
a_i - počet dovezených kontajnerov v i-tom dni,
u_i - počet kontajnerov, ktoré sa rozhodneme odviesť v i-tom dni.

Pretože za deň možno odviesť najviac 2 kontajnery, tak $u_i \in \{0, 1, 2\}$. Je zrejmé, že rozhodnutie $u_i = 0$ znamená neposielať auto na stanicu, a teda nulové náklady na odvoz, rozhodnutie $u_i = 1$ alebo $u_i = 2$ znamená vyslať auto na stanicu s nákladmi na odvoz rovnými

70. Odvoz kontajnerov v i-tom dni realizujeme až potom, keď sa na stanici nachádza, okrem zostatku x_i z predchádzajúceho dňa, aj všetkých a_i avizovaných kontajnerov. Po odvoze zostane na konci i-teho dňa na stanici $x_{i+1} := x_i + a_i - u_i$ kontajnerov. Náklady za tento deň budú jednak penále $50(a_i + x_i - u_i)$ za neodvezené kontajnery a náklady $70\mathrm{sgn}(u_i)$ za auto. Tu $\mathrm{sgn}(u)$ je funkcia, ktorá kladným hodnotám u priraďuje 1 a nule nulu. Dostali sme úlohu

$$\begin{aligned}
&\text{minimalizovať} && \sum_{i=0}^{4}[70\mathrm{sgn}(u_i) + 50(a_i + x_i - u_i)] && (1.15)\\
&\text{pri podmienkach} && x_{i+1} = x_i + a_i - u_i, \quad i = 0, \ldots, 4, && (1.16)\\
& && x_0 = 0, \quad x_5 = 0, && (1.17)\\
& && u_i \in \{0, 1, 2\}, && (1.18)\\
& && x_i \geq 0, \quad i = 0, \ldots, 4, && (1.19)
\end{aligned}$$

kde ohraničenie $x_i \geq 0$ vyjadruje skutočnosť, že v jednotlivé dni nemožno odviesť viac kontajnerov, ako sa na stanici nachádza a podmienka $x_5 = 0$, že na konci posledného dňa budú všetky kontajnery odvezené.

Túto úlohu možno charakterizovať ako neautonómnu úlohu optimálneho riadenia s pevným časom, pevným koncom, ohraničeniami na stav a riadenie. Všimnime si, že z podmienok úlohy vyplýva, že stavová premenná môže nadobúdať iba nezáporné celé čísla, ktoré sú zhora ohraničené počtom všetkých dovezených kontajnerov.

Ďalšie dva príklady budú zaujímavé najmä tým, že etapou nebude časové obdobie (rok, mesiac, deň), ako to bolo doteraz. Prvý z príkladov bude (podobne ako Príklad 1.1) opäť spadať do širokej triedy úloh o rozdeľovaní zdrojov. Druhý predstavuje veľmi jednoduchý variant rozsiahlej triedy *úloh optimálnej obnovy.* Bude poučný aj tým, že hodnoty riadenia nebudú mať číselný, ale logický charakter.

Príklad 1.4. Optimálna alokácia prostriedkov. K dispozícii je k investičných plánov, ako možno využiť prostriedky vo výške S.

Každý z nich prinesie daný výnos $h_i(u_i)$, $i = 0, \ldots, k-1$, kde u_i je objem investovaných prostriedkov. Treba určiť, ako rozdeliť prostriedky medzi jednotlivé investičné plány tak, aby bol dosiahnutý maximálny celkový výnos, t.j. treba

$$\text{maximalizovať} \sum_{i=0}^{k-1} h_i(u_i) \quad \text{pri podmienke} \sum_{i=0}^{k-1} u_i \leq S.$$

Všimnime si, že zatiaľ čo v predchádzajúcich úlohách bol proces jasne rozdelený na časové etapy, v tejto úlohe tomu tak nie je. Ak teda máme uspieť pri formulácii tejto úlohy ako úlohy optimálneho riadenia, musíme takéto rozdelenie procesu vniesť do úlohy umelo. Urobíme to tak, že jednorazový proces rozdelenia prostriedkov do k plánov zmeníme na postupný, etapovitý proces investovania najprv do prvého plánu, potom do druhého, atď.

Pred investičným rozhodnutím v i-tej etape, $i = 0, \ldots, k-1$, vezmeme do úvahy množstvo prostriedkov x_i, ktoré ešte máme k dispozícii. Rozhodnutie spočíva vo voľbe výšky u_i prostriedkov, ktoré budeme investovať. Zostane nám $x_{i+1} = x_i - u_i$ prostriedkov na investovanie do ďalších plánov. Pritom zrejme $x_0 = S$ vyjadruje celkovú sumu prostriedkov a podmienka $x_k \geq 0$ zabezpečí, aby sme v jednotlivých etapách neinvestovali viacej, ako máme k dispozícii.

Dostali sme úlohu optimálneho riadenia

$$\text{maximalizovať} \quad \sum_{i=0}^{k-1} h_i(u_i) \tag{1.20}$$

$$\text{pri podmienkach} \quad x_{i+1} = x_i - u_i, \quad i = 0, \ldots, k-1, \tag{1.21}$$

$$x_0 = S, \tag{1.22}$$

$$u_i \geq 0, \tag{1.23}$$

$$x_k \geq 0. \tag{1.24}$$

Úlohu možno charakterizovať ako neautonómnu úlohu s pevným časom, s ohraničením na riadenie, bez ohraničenia na stav, s čiastočne pevným koncom, kde $C = \{x : \ x \geq 0\}$.

Príklad 1.5. Optimálna údržba stroja. Stroj na lisovanie šťavy z ríbezlí sa zanáša. Podľa toho, koľko bolo na ňom vylisované, sa jeho výkon znižuje tak, že po každom kilograme vylisovaných ríbezlí sa čas lisovania nasledujúceho kilogramu zvýši o 20 minút. Vyčistenie stroja trvá 30 minút. Ríbezle sú balené po 1 kg, ktorý po načatí treba ihneď spracovať. Celkove treba spracovať k kg ríbezlí. Treba rozhodnúť, či a po ktorom kilograme stroj čistiť, aby sa žiadané množstvo spracovalo za najkratší čas. Predpokladajme, že na začiatku máme stroj, na ktorom bolo od posledného čistenia vylisovaných $a \geq 0$ kilogramov ríbezlí a na konci necháme stroj nevyčistený.

Zavedieme nasledovné označenia:
i – poradové číslo kilogramu, ktoré má byť spracované, $i = 0, \ldots, k-1$,
x_i – počet spracovaných kilogramov od ostatného čistenia pred lisovaním i-teho kila,
$u_i = \check{C}$, ak stroj čistíme pred lisovaním i-teho kilogramu,
$u_i = N$, ak stroj nečistíme pred lisovaním i-teho kilogramu.

V tomto prípade je i-tou etapou časť procesu začínajúca po skončení lisovania $(i-1)$-ho kila a končiaca vylisovaním i-teho kila. Opíšeme teraz časovú postupnosť jednotlivých úkonov i-tej etape. Najprv máme informáciu x_i o stupni znečistenia stroja. Vzápätí zvolíme hodnotu premennej u_i, t.j. rozhodneme sa, či stroj čistíme ($u_i = \check{C}$) alebo nečistíme ($u_i = N$). Ak sme zvolili $u_i = \check{C}$, tak stroj vyčistíme. Potom nasleduje lisovanie i-teho kilogramu a nakoniec si poznačíme, že na stroji bolo od posledného čistenia vylisovaných x_{i+1} kilogramov ríbezlí, kde $x_{i+1} = x_i + 1$, ak sme nečistili a $x_{i+1} = 1$, ak sme čistili. Zaznamenáme tiež časové straty v minútach, ktoré sú 30, ak sme čistili a $20x_i$, ak sme nečistili. Zrejme $x_0 = a$. Našou úlohou je nájsť také hodnoty u_i, aby sme minimalizovali celkové časové straty pri lisovaní ríbezlí.

Dostali sme úlohu optimálneho riadenia

$$\text{minimalizovať} \quad \sum_{i=0}^{k-1} f^0(x_i, u_i),$$

$$\text{pri podmienkach}\quad x_{i+1} = \begin{cases} x_i + 1, & \text{ak } u_i = N, \\ 1, & \text{ak } u_i = \check{C}, \end{cases} \qquad i = 0, \ldots, k-1,$$
$$x_0 = a.$$

kde
$$f^0(x,u) = \begin{cases} 30, & \text{ak } u = \check{C}, \\ 20x, & \text{ak } u = N. \end{cases}$$

Úlohu charakterizujeme ako autonómnu úlohu s pevným časom, voľným koncom, bez ohraničení na riadenie a stavovú premennú. Riadiaca premenná má logický charakter a stavová nadobúda hodnoty z množiny prirodzených čísel.

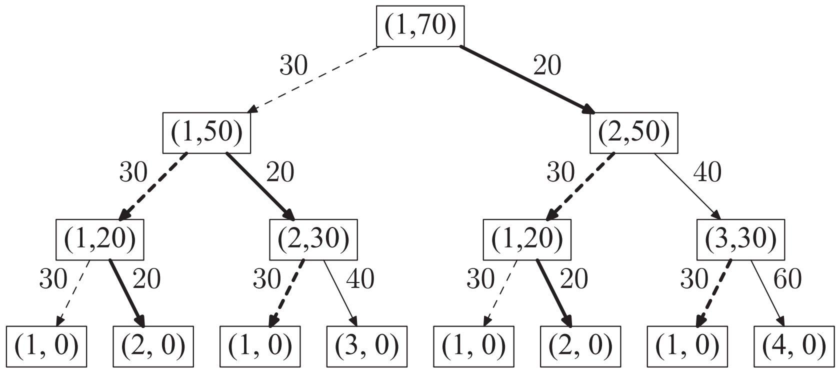
1.4 Základné problémy diskrétnej teórie optimálneho riadenia

Samozrejme, prvoradým cieľom teórie optimálneho riadenia je vyvinúť účinné metódy, ktorými by bolo možné nájsť riešenia úloh – optimálne riadenia. Čo však rozumieť pod metódou riešenia trochu spresníme. Ako uvidíme, často sa ani pomerne jednoduché príklady nedajú riešiť v uzavretom tvare. Existencia vysoko výkonných počítačov zvádza k predstave, že diskretizáciou všetkých kontinuálnych veličín sa dá prejsť na úlohu maximalizácie funkcie konečného počtu hodnôt, ktorú možno riešiť prebratím všetkých možností. Ukážeme si, že aj táto cesta je pri trocha zložitejších problémoch neschodná, a to pre veľký rozsah úlohy, ktorá takto vznikne. Ťažkosti, ktoré môžu vzniknúť a možné východiská, ktoré sa nám ponúkajú, ilustrujeme na úlohe o optimálnej údržbe stroja (Príklad 1.5).

Príklad 1.6. Optimálna údržba stroja z Príkladu 1.5. Táto úloha je prirodzene diskrétna, riadenie $\{u_i\}_{i=0}^{k-1}$ má logický charakter, kde $u_i \in \{\check{C}, N\}$, a teda celkový počet prípustných riadení pre túto úlohu je 2^k. Metóda prebratia všetkých možností by teda mala exponenciálnu zložitosť, pretože by vyžadovala výpočet účelovej funkcie v každom prípustnom riadení. To pri veľkých hodnotách k môže viesť k úlohe nezvládnuteľnej ani súčasnými vysoko výkonnými počítačmi. Preto je zrejmé, že sa treba snažiť o nejaký iný, dômyselnejší postup. Pokúsme sa ho teraz navrhnúť.

Vyjdime najprv z postupu prehľadávania všetkých možností a všimnime si, že v každom kroku tohto algoritmu treba znova zopakovať celý výpočet účelovej funkcie, a to aj v prípade, ak sa zmení postupnosť $\{u_i\}_{i=0}^{k-1}$ napr. len v poslednom prvku u_{k-1}. Tento nedostatok možno odstrániť tým, že účelovú funkciu budeme počítať postupne „odzadu“. Napr. pri porovnávaní riadení $\{\check{C},\ldots,\check{C},\check{C}\}$ a $\{\check{C},\ldots,\check{C},N\}$ môžeme využiť, že pre obe riadenia bude stav stroja x_{k-1} rovnaký ($x_{k-1}=1$) a v tomto stave je optimálne stroj nečistiť. Stačí si teda zapamätať, že pokiaľ je stroj pred lisovaním posledného kila v stave $x_{k-1}=1$, najmenšie možné časové straty pri lisovaní posledného kilogramu sú 20 min., a to v prípade, ak stroj nebudeme čistiť, teda ak $u_{k-1}=N$. Analogicky možno zistiť, že ak by bol stroj pred lisovaním posledného kila v stave $x_{k-1}=2$, optimálne je stroj vyčistiť ($u_{k-1}=\check{C}$) a najmenšie možné straty sú 30 min. Rovnako možno určiť najmenšie možné časové straty (označme ich V_{k-1}) pri lisovaní posledného kilogramu a optimálnu hodnotu u_{k-1} pre všetky ostatné hodnoty x_{k-1}. Podobne možno postupovať ďalej: ak je napr. stroj v stave $x_{k-2}=1$, pri čistení budú najmenšie možné straty pri vylisovaní posledných 2 kg ríbezlí 30+20 min., pretože čistením sa stroj dostane do stavu $x_{k-1}=1$, pre ktorý sú najmenšie možné straty 20 min. Ak stroj nevyčistíme, najmenšie straty pri lisovaní posledných 2 kg ríbezlí budú 20+30 min. V tomto prípade sú teda optimálne riadenia $u_{k-2}=\check{C}$ aj $u_{k-2}=N$.

Riešenie úlohy pre $x_0=1$ a $k=3$ s využitím uvedeného postupu je zobrazené na Obrázku 1.1. Jednotlivé uzly zobrazujú stav stroja (t.j. hodnotu x_i) a najmenšie straty pri lisovaní zvyšných kilogramov ríbezlí. V každom uzle sa môžeme rozhodnúť, či budeme stroj čistiť (zobrazené prerušovanou šípkou) alebo nie (zobrazené plnou šípkou). Číslo pri šípke udáva dodatočný čas potrebný na vylisovanie nasledujúceho kilogramu ríbezlí. Hrubou šípkou sú znázornené optimálne rozhodnutia. V takto navrhnutom algoritme treba urobiť rozhodnutie v každom uzle, pričom uzlov je $2^k - 1$. Zložitosť algoritmu teda zostáva exponenciálna, hoci výpočet účelovej funkcie je jednoduchší.

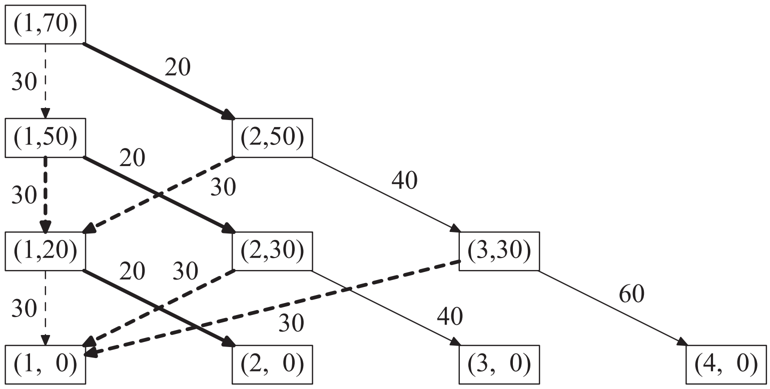


Obr. 1.1: K riešeniu príkladu o optimálnej údržbe stroja, exponenciálny algoritmus

Všimnime si však, že v predposlednom riadku sa dvakrát vyskytuje rozhodnutie pre stav $x_{k-1} = 1$, ktoré je prirodzene v oboch prípadoch rovnaké. Ak sa totiž stroj nachádza pred vylisovaním i-teho kilogramu v stave x_i, ďalšie rozhodovanie bude nezávislé na tom, ako sa do tohto stavu dostal. Ako ukážeme neskôr, táto „markovovská" vlastnosť úlohy bude kľúčovou pre riešenie úloh pomocou dynamického programovania, ktoré bude vysvetlené v nasledujúcej kapitole. V našom prípade to znamená, že všetky uzly s rovnakým stavom v jednom riadku môžeme zlúčiť do jedného. Riešenie úlohy potom zobrazuje schéma uvedená na Obrázku 1.2. Hodnoty x_i však môžu byť iba z množiny $\{1, \ldots, i+1\}$. Navzájom rôznych uzlov, v ktorých treba urobiť rozhodnutie, je teda $\frac{(k+1)k}{2}$ a navrhnutý algoritmus má polynomiálnu zložitosť. Pre veľkú hodnotu k to môže predstavovať výraznú úsporu.

Poznamenajme, že pre úlohy, kde majú premenné x_i a u_i kontinuálny charakter, je pôvodne navrhovaný algoritmus pomocou prehľadávania všetkých možností ekvivalentný hľadaniu extrému funkcie k premenných, zatiaľ čo nový algoritmus hľadá extrém v každom kroku iba podľa jednej premennej, pričom využíva vzťah (viď Úloha 1.2)

$$\max_{x_1,x_2}[g(x_1) + h(x_1, x_2)] = \max_{x_1}[g(x_1) + \max_{x_2} h(x_1, x_2)]. \tag{1.25}$$



Obr. 1.2: K riešeniu príkladu o optimálnej údržbe lisu ríbezlí, polynomiálny algoritmus

Snahou diskrétnej teórie optimálneho riadenia je vytvoriť metódy, ktoré by (podobne ako v príklade vyššie) využívali špeciálnu štruktúru úloh optimálneho riadenia a umožnili efektívne riešiť úlohy.

To však nie je všetko. Ďalším cieľom teórie je skúmať vlastnosti optimálnych riadení. Pod tým treba rozumieť, že chceme niečo vyvodiť o optimálnych riadeniach priamo z charakteru úlohy bez toho, že by sme poznali ich numerické hodnoty. S podobnou situáciou sa stretávame napríklad v teórii stability, kde vieme rozhodnúť o stabilite diferenciálnej rovnice bez toho, že by sme poznali jej riešenia. Je to dôležité jednak preto, že presné optimálne riadenie zriedkakedy vieme nájsť, jednak nám takáto informácia niekedy môže radikálne obmedziť množinu kandidátov na optimálne riadenie.

Čitateľovi, ktorý pozná teóriu nelineárneho programovania, môže prísť na um analógia s Kuhn-Tuckerovými podmienkami. Táto analógia je odôvodnená, nutné podmienky optimality, ktoré odvodíme v tretej kapitole, Kuhn-Tuckerovým podmienkam skutočne zodpovedajú, využívajú však špecifikum úloh optimálneho riadenia.

1.5 Úlohy na riešenie

Úloha 1.1. Vyriešte ľubovoľným spôsobom úlohu z Príkladu 1.1 o optimálnom rozdeľovaní zdrojov pre prípad $k = 2$; $b = 0{,}7$; $c = 0{,}4$; $g = 2$; $h = 2{,}5$ a odôvodnite správnosť riešenia.

Úloha 1.2. Za predpokladu, že všetky maximá v (1.25) existujú, dokážte rovnosť (1.25).

Úloha 1.3. Ukážte, že úlohu optimálneho riadenia v Mayerovom tvare možno previesť na úlohu v Lagrangeovom tvare a naopak. Návod: pri dôkaze opačnej implikácie zaveďte novú stavovú premennú.

Úloha 1.4. Majme úlohu optimálneho riadenia (1.9)–(1.14) s dodatočnou podmienkou

$$\sum_{i=0}^{k-1} g_i(x_i, u_i) = 0.$$

Vzhľadom na túto dodatočnú podmienku úloha nie je formulovaná v štandardnom tvare. Zavedením novej stavovej premennej preformulujte úlohu do štandardného tvaru. (Návod: pozri Príklad 1.4 o optimálnej alokácii prostriedkov.)

Úloha 1.5. Optimálne gazdovanie na ranči. Jednooký Bill má na ranči Hrdzavá podkova čriedu dlhorohého texaského dobytka s a kusmi. Ak na výročný trh v Santa Fé donesie y kusov dobytka, dostane za ne $\phi(y)$ dolárov; zvyšok sa mu za rok rozmnoží $b > 1$ krát.

Bill chce na ranči gazdovať 10 rokov a samozrejme chce za týchto 10 rokov dosiahnuť najväčší možný zisk.

Označte x_i stav čriedy pred trhom v i-tom roku a sformulujte tento problém ako úlohu optimálneho riadenia v dvoch verziách: v jednej označte $u_i \in [0,1]$ pomernú časť čriedy, ktorú v i-tom roku privedie Bill na trh; v druhej verzii označte u_i počet kusov dobytka, ktoré privedie na trh v i-tom roku. Zdôvodnite výhodnosť jednej či druhej formulácie pre túto situáciu.

Úloha 1.6. Porovnajte obidve formulácie problému v predchádzajúcej úlohe s Príkladom 1.2. Naformulujte problém z Príkladu 1.2 do úlohy, v ktorej riadenie bude tiež vyjadrovať pomernú časť kapitálu určenú na spotrebu. Charakterizujte takto sformulovanú úlohu riadenia.

Úloha 1.7. Dopravný dispečer. Sformulujte nasledovný problém ako úlohu optimálneho riadenia. Skúste odhadnúť jeho riešenie.
Dopravný dispečer vie, že nasledujúci (5 dňový) týždeň mu príde 10, 3, 7, 2, 5 vagónov, ktoré musí do konca týždňa vyložiť. Na vykladanie má k dispozícii čatu vykladačov, ktorí vyložia jeden vozeň za dve hodiny. Ak pracujú viac ako 8 hodín, musí im za každú hodinu nadčasov vyplatiť 15 eur. Za nevyložený vozeň platí penále 20 eur za deň. Treba rozhodnúť, ktorý deň koľko vozňov majú robotníci vyložiť, aby náklady na dodatočné mzdy a penále boli minimálne. Predpokladáme, že robotníci pracujú len celé násobky dvoch hodín a že z jedného dňa na druhý nezostávajú čiastočne vyložené vagóny.

Úloha 1.8. Preteky Formuly 1. V pretekoch Formuly 1 klesá priemerný čas prejdenia jedného kola v závislosti od opotrebenia pneumatík. Predpokladajme, že dodatočný čas v sekundách je možné vyjadriť vzťahom $0,001n^2$, kde n je počet celých kôl od ostatnej výmeny pneumatík. Výmena trvá 10 s. Kedy sa oplatí meniť pneumatiky? Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Úloha 1.9. Optimalizácia hotovosti. Prvá východoslovenská banka, a.s. s centrálou v Prešove chce zabezpečiť dostatok hotovosti vo svojej pobočke v Snine. Budúci týždeň má dohodnuté objemy výberov (v tis. eur) uvedené v Tabuľke 1.1.

Tabuľka 1.1: Dohodnuté objemy výberov hotovosti (v tis. eur)

Deň	Po	Ut	St	Št	Pi
Výber	200	250	130	420	310

Hotovosť sa dováža z centrály skoro ráno. Jeden dovoz hotovosti stojí 100 eur bez ohľadu na množstvo dovezených penazí. S ohľadom na bezpečnosť však nie je možné prepravovať naraz viac ako 1 mil. eur. Držanie prebytočnej hotovosti v pobočke z jedného dňa na druhý vedie k strate, pretože ju nemožno uložiť na medzibankovom trhu. Predpokladajme, že daná strata predstavuje 0,02% za deň. V ktoré dni treba zabezpečiť dovoz? Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Úloha 1.10. Vykladanie vagónov. Dopravnému dispečerovi príde zajtra ráno 10 vagónov suroviny, ktorú treba spracovať do 3 dní. Ak sa rozhodne, že sa surovina bude vykladať rýchlosťou 3 vagóny za deň alebo nižšou, pri vykladaní budú 5% straty. Ak bude rýchlosť vykladania 4 vagóny za deň, straty budú 10%, a ak 5 vagónov za deň, tak 15% (viac sa nedá). Práca však súri, lebo surovina sa kazí: prvú noc sa pokazí 10% nevyloženej suroviny a druhú noc 20%. Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Úloha 1.11. Optimálna výmena auta. Nech $\psi(x)$ je hodnota istého typu auta starého x rokov a nech $\varphi(x)$ sú ročné náklady na údržbu auta x rokov starého. Za predpokladu, že sa budú vyrábať stále tie isté autá, ich cena a náklady na ich údržbu sa nebudú s časom meniť, treba určiť, kedy kupovať nové auto, aby celkové náklady naň za daný počet rokov k boli minimálne. Ďalej predpokladáme, že predaj auta sa uskutočňuje vždy na začiatku roku, nikdy nevlastníme auto staršie ako 5 rokov a že v k-tom roku auto predáme. Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Úloha 1.12. Optimálny rozvrh generálnych opráv. Cena generálnej opravy (GO) stroja je 1 000 eur. Používaním sa kvalita stroja znižuje a vznikajú straty, pričom výšku ročnej straty možno odhadnúť ako $200(n+1)$ eur, kde n je počet celých rokov od ostatnej GO. Kedy treba dať stroj do GO, ak ho chceme využívať 5 rokov a naposledy bol v GO včera? Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Úloha 1.13. Kombajnista. Kombajnista má tri dni na to, aby pokosil pole o rozlohe 10 hektárov. Za deň je schopný pokosiť rozlohu 5 hektárov, ale s rýchlosťou kosby vzrastajú straty a to takto:
ak za deň pokosí 3 hektáre alebo menej, strata je zanedbateľná,
ak za deň pokosí 4 hektáre, strata je 5 percent,
ak za deň pokosí 5 hektárov, strata je 15 percent.
Straty však vzrastajú aj preto, že obilie dozrieva a vysýpa sa z klasov, a to takto:
z obilia pokoseného 1. deň je strata 10 percent,
z obilia pokoseného 2. deň je strata 15 percent,
z obilia pokoseného 3. deň je strata 20 percent.
Za predpokladu, že sa uvedené dva druhy strát každý deň sčítavajú, treba určiť, koľko má ktorý deň kombajnista pokosiť, aby celková strata bola minimálna. Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Úloha 1.14. Optimálny rozvrh objednávok pomarančov. Obchod s ovocím má objednať pomaranče na mesiace september, október a november. Nákupná a predajná cena sú uvedené v Tabuľke 1.2. V jednotlivých mesiacoch očakáva odbyt 2000 kg, 3000 kg, 4000 kg. V príručnom sklade môže uložiť najviac 3000 kg. Ak chce uložiť viac, musí si prenajať sklad, za ktorý zaplatí 1000 eur za mesiac, pričom maximálna kapacita skladu je 4000 kg. Na začiatku septembra má zásobu 1000 kg. Na konci novembra majú byť sklady prázdne. Treba rozhodnúť, koľko pomarančov objednať na jednotlivé mesiace, aby jeho celkový zisk bol maximálny. Pomaranče sa objednávajú a dodávajú vždy na začiatku mesiaca a len v celých tisícoch kg. Sformulujte v štandardnom tvare diskrétnej úlohy optimálneho riadenia.

Tabuľka 1.2: Nákupná a predajná cena pomarančov

	September	Október	November
Nákupná cena (euro/kg)	0,5	1	1,5
Predajná cena (euro/kg)	1	1,5	2

Úloha 1.15. Sformulujte ako úlohu optimálneho riadenia modifikáciu úlohy z Príkladu 1.5, kde na začiatku celého procesu bude k dispozícii čistý stroj a na konci ho treba vyčistiť.

Úloha 1.16. Optimálna alokácia času medzi prácu a vzdelávanie. Študent vysokej školy sa každý semester môže rozhodnúť, ako rozdelí svoj čas medzi zvyšovanie svojho vzdelania (napr. návštevou cvičení) a zárobkovú činnosť. Podobná dilema pokračuje aj po ukončení štúdia. Dosiahnuté vzdelanie zvyšuje jeho mzdu v budúcnosti, v čase sa však znižuje konštantným tempom (proces samozabúdania). Cieľom je maximalizovať celkový (diskontovaný) zárobok za celé plánovacie obdobie. Sformulujte túto úlohu ako úlohu optimálneho riadenia. Pokúste sa bližšie opísať charakter použitých funkcií (rastúca, lineárna, konvexná) tak, aby čo najlepšie zachytili uvedený problém.

Úloha 1.17. V i-tom období ($i > 0$) dostaneme dôchodok veľkosti D jednotiek. Aká je jeho hodnota v súčasnosti (nulté obdobie), ak predpokladáme, že počas jedného obdobia sa jednej jednotke pripisuje úrok veľkosti r? Výsledok tejto úlohy bol využitý v Poznámke 1.7.

Kapitola 2

Dynamické programovanie pre diskrétne úlohy

Life must be lived forward
and understood backwards.
Soren Kierkegaard.

V tejto kapitole sa budeme venovať prvému z prístupov k úlohám optimálneho riadenia, spomínaných v úvode, ktorý sa nazýva *dynamickým programovaním*. Jeho podstata je v tom, že sa úloha zaradí do systému úloh a jej riešenie dostaneme postupným riešením úloh z tohoto systému. Pri riešení nasledujúcej úlohy sa vždy využije riešenie predchádzajúcej. Podobný postup sme použili v Príklade 1.6 pri riešení úlohy o optimálnej údržbe stroja z Príkladu 1.5. Výsledkom je rekurentný vzťah nazývaný *Bellmanova rovnica dynamického programovania*. Jej využitím môžeme oproti pasívnemu prehľadávaniu všetkých možností dospieť k podstatným úsporám pri výpočte optimálneho riadenia.

Rovnicu dynamického programovania odvodíme najprv pre úlohu s pevným časom. V ďalších podkapitolách potom rozšírime naše poznatky o tejto rovnici na iné typy úloh. V poslednej podkapitole budeme hovoriť o rovnici dynamického programovania v prípade procesov vystavených náhodným vplyvom. Takéto úlohy budeme nazývať

Dostaneme tak úlohu

$$
\begin{aligned}
\text{maximalizovať} \quad & \sum_{i=0}^{2} \ln u_i \\
\text{pri podmienkach} \quad & x_{i+1} = 1{,}5x_i - u_i, \quad i = 0, 1, 2, \\
& x_0 = 3, \quad x_3 = 2, \\
& u_i \geq 0, \quad x_i \in N.
\end{aligned}
$$

Riešenie je uvedené v Tabuľke 2.3.

Tabuľka 2.3: Riešenie Príkladu 2.6.

i	x	u	f_i^0	f_i	$f_i^0 + V_{i+1}(f_i)$	V_i	v_i
3						$\boxed{0}$	
2	2	1	ln 1	2	0+0=0	0	1
	$\boxed{3}$	$\boxed{2{,}5}$	ln 2,5	$\boxed{2}$	0+ln 2,5=ln 2,5	ln 2,5	$\boxed{2{,}5}$
	4	4	ln 4	2	0+ln 4=ln 4	ln 4	4
1	2	1	ln 1	2	0+0=0	0	1
		0,5	ln 0,5	4	$\ln 0,5 + \ln 4 = \ln 2$		
	$\boxed{3}$	$\boxed{1{,}5}$	ln 1,5	$\boxed{3}$	$\ln 1{,}5 + \ln 2{,}5 = \ln 3{,}75$	ln 3,75	$\boxed{1{,}5}$
		2,5	ln 2,5	2	$\ln 2{,}5 + 0 = \ln 2{,}5$		
		1	ln 1	5	$0 + \ln 5,5 = \ln 5,5$		
	4	2	ln 2	4	$\ln 2 + \ln 4 = \ln 8$	ln 8	2
		3	ln 3	3	$\ln 3 + \ln 2{,}5 = \ln 7{,}5$		
		4	ln 4	2	$\ln 4 + 0 = \ln 4$		
		0,5	ln 0,5	4	$\ln 0,5 + \ln 8 = \ln 4$		
0	$\boxed{3}$	$\boxed{1{,}5}$	ln 1,5	$\boxed{3}$	$\ln 1{,}5 + \ln 3{,}75 = \ln 5{,}625$	$\boxed{\ln 5{,}625}$	$\boxed{1{,}5}$
		2,5	ln 2,5	2	$\ln 2{,}5 + 0 = \ln 2{,}5$		

Z jej hodnôt vyčítame, že optimálna hodnota účelovej funkcie je $V_0(3) = \ln 5{,}625$, optimálne riadenie je $\mathcal{U} = \{1{,}5; 1{,}5; 2{,}5\}$ a jeho odozva je $\mathcal{X} = \{3, 3, 3, 2\}$.

Príklad 2.7. Optimálna spotreba (riešenie pomocou počítača). Tentoraz budeme postupovať tak, že spojité veličiny (stavovú a riadiacu premennú) nahradíme diskrétnymi pomocou ekvidistantného delenia. V tomto prípade sa môže stať, že potrebujeme poznať hodnotu V_{i+1} v takom bode x, v ktorom nebola vyčíslená. Vtedy vyberieme ku x najbližšiu hodnotu $\tilde{x}$ a nahradíme $V_{i+1}(x) \sim V_{i+1}(\tilde{x})$. Úlohu budeme riešiť pre prípad $r = 0{,}2$, $k = 21$, $\delta = 0$, $x_1 = 3$ a $x_{21} = 0$. Dostaneme tak úlohu

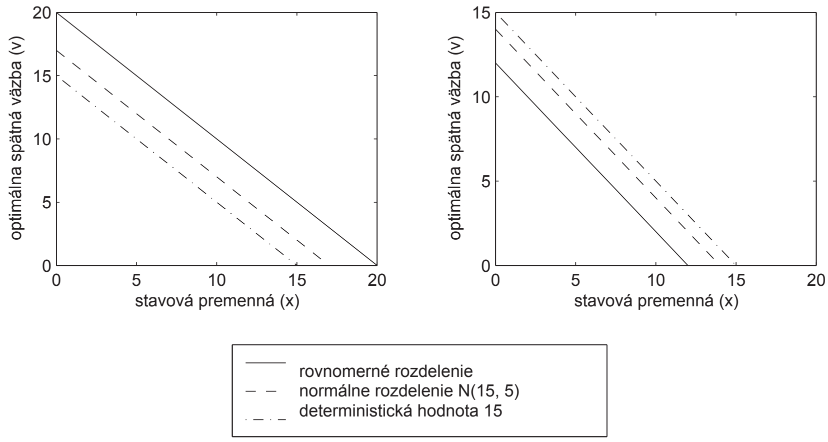
$$
\begin{aligned}
\text{maximalizovať} \quad & \sum_{i=1}^{20} \ln u_i \\
\text{pri podmienkach} \quad & x_{i+1} = 1,2x_i - u_i, \quad i = 1, \ldots, 20, \\
& x_1 = 3, \\
& x_{21} = 0.
\end{aligned}
$$

Na riešenie využijeme program MATLAB.[1]

```
k=20; N=50; x_steps=N; u_steps=N; x_max=25; u_max=7;
x=0:(x_max/x_steps):x_max; u=0:(u_max/u_steps):u_max;
V=zeros(k+1,x_steps+1); v=zeros(k+1,x_steps+1);
vopt=zeros(k+1,1);

%výpočet hodnotovej funkcie a optimálnej spätnej vazby
V(:,:)=-inf;
V(k+1,1)=0; %prípustné je len x_k=0
for i=k:-1:1
    for j=1:(x_steps+1)
        for m=1:(u_steps+1)
            F=1.2*x(j)-u(m);
            if (F>=0) & (F<=x_max)
               F=round(F*x_steps/x_max)+1;
               if log(u(m))+V(i+1,F)>V(i,j)
```

[1] Tu uvedený program bol napísaný pre MATLAB 6.5.



Obr. 2.6: Optimálna spätná väzba pre obdobie $i = 10$ pre rôzne rozdelenia náhodnej premennej z_iv prípade $s = 0{,}1$ (vľavo) a $s = 0{,}3$ (vpravo).

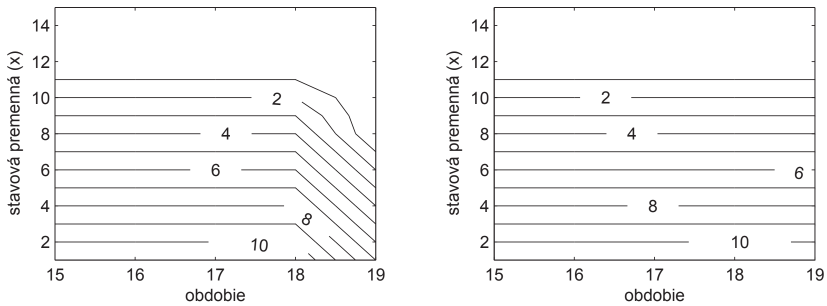
Ako vidieť na uvedenom obrázku, zmena závisí od parametra s (strata v dôsledku predaja až na nasledujúci deň). V prípade, že $z_i = 15$, obchodník objedná vždy toľko zväzkov reďkoviek, aby ich mal aj spolu so zväzkami z predchádzajúceho dňa 15. Ak je s malé, počet objednaných zväzkov reďkoviek sa zvyšuje so štandardnou odchýlkou rozdelenia náhodnej premennej z_i. Dôvodom je, že sa zvyšuje pravdepodobnosť vyššieho dopytu, pričom straty v prípade nízkeho dopytu sú relatívne malé. Naopak, ak je s väčšie, počet objednaných zväzkov s rastúcou štandardnou odchýlkou klesá.

Zmena rozdelenia dopytu sa v programe prejaví predovšetkým v premennej z, ktorú treba generovať pomocou inverznej kumulatívnej distribučnej funkcie aplikovanej na rovnomerne rozdelený vektor. V tomto prípade však už hodnoty nebudú celočíselné, preto pri výpočte novej hodnoty stavovej premennej zaokrúhľovať alebo interpolovať. Z dôvodu dosiahnutia porovnateľnej presnosti ako v prípade s rovnomerným rozdelením preto treba zjemniť delenie uvažovaných hodnôt stavovej a riadiacej premennej.

Pokles optimálnej spätnej väzby v čase $i = 18$ a $i = 19$, ktorý možno pozorovať na Obrázku 2.5, je spôsobený tým, že reďkovky, ktoré zostanú po ukončení všetkých dní predaja, už neprinesú žiadne príjmy. Môžeme však úlohu zmeniť tak, že budeme predpokladať, že tieto zväzky možno odpredať za zostatkovú hodnotu 0,15 eur za zväzok. Účelová funkcia bude mať tvar

$$J = E\left(\sum_{i=0}^{19} 0{,}7 \min(x_i + u_i; z_i) - 0{,}5u_i - 0{,}1x_i + 0{,}15x_{20}\right).$$

Ako vidieť na Obrázku 2.7, pri takto zmenenej účelovej funkcii je pokles objednávok v posledných dňoch menej výrazný. Uvedené riešenie získame tak, že príkaz `V(k+1,:)=0;` nahradíme príkazom `V(k+1,:)=0:0.15:(0.15*xmax);`.



Obr. 2.7: Optimálna spätná väzba pre obdobia 15 až 19 a $s = 0{,}3$ v prípade, že výkupná cena nepredaných zväzkov je nulová (vľavo), resp. je rovná 0,15 eur (vpravo).

Možným rozšírením uvedenej úlohy by bolo aj uvažovať rôzne rozdelený dopyt v jednotlivých dňoch. V takom prípade by bolo možné očakávať, že optimálna spätná väzba nebude pre danú hod-

Kapitola 3

Princíp maxima pre diskrétne úlohy

The farther backward you can look,
the farther forward you can see.
Winston Churchill.

V tejto kapitole sa budeme venovať alternatívnemu prístupu k riešeniu úloh optimálneho riadenia. Budeme mu hovoriť *variačný*, a to z niekoľkých dôvodov.

Základným prvkom tohto prístupu je test optimality riadenia, ktorým sa overuje, či „malé variácie" testovaného riadenia nevedú k nárastu účelovej funkcie, čo je samozrejme nutnou podmienkou optimality. Čaro tohto prístupu je v tom, že pri pozornom vyčerpaní všetkých možností, dostaneme formálne dostatok podmienok na určenie riadení v tom zmysle, že podmienok je toľko, koľko je hľadaných optimalizačných parametrov.

Prototypom takto získaných podmienok optimality sú optimalizačné podmienky prvého rádu, ako napríklad nulovosť parciálnych derivácií funkcie viacerých premenných v úlohe na voľný extrém, Langrangeove podmienky na viazaný extrém alebo Kuhn-Tuckerove podmienky v nelineárnom programovaní. V nekonečnorozmernom prípade im zodpovedajú podmienky variačného počtu, z ktorého sme prevzali pomenovanie.

Spomínané analógie naznačujú, že prístup využíva jemnejší aparát matematickej analýzy ako doteraz. Napríklad je potrebné, aby funkcie, vystupujúce v úlohách, bolo možné diferencovať. Preto je nevyhnutné, aby boli definované na otvorených podmnožinách vektorových priestorov. Z toho dôvodu sa obmedzíme na úlohy, v ktorých sú riadiace aj stavové premenné z Euklidovských priestorov.

Široká trieda úloh optimálneho riadenia s konečným počtom časových krokov sa takto redukuje na úlohy nelineárneho programovania. Nutné podmienky pre ne sú potom vlastne iba prepisom Kuhn-Tuckerových podmienok do tvaru odrážajúceho špecifiká rekurentného charakteru úlohy optimálneho riadenia.

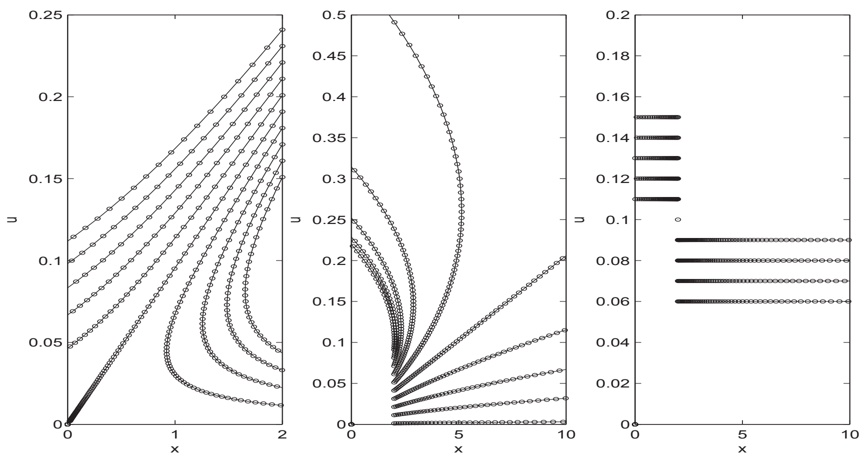
Prečo teda hovoríme o princípe maxima? Dôvod je historický. Diskrétnej úlohe optimálneho riadenia časovo predchádzala jej spojitá verzia. V nej bol ako základná variačná nutná podmienka optimality odvodený*Pontrjaginov princíp maxima*. Pre diskrétne úlohy boli z neho prevzaté niektoré prvky, ako je *Hamiltonova funkcia* alebo *adjungované premenné*. Tieto pojmy sa ukazujú ako užitočné a názorné, a to napriek tomu, že úplná analógia Pontrjaginovho princípu maxima zo spojitej teórie v diskrétnej teórii neplatí. Dôvodom je, že spojitý čas umožňuje variácie, ktoré sú „malé“ v tom, že sa síce od testovaného riadenia môžu hoci aj veľmi odchýliť, ale trvajú krátko. To pri diskrétnom čase nie je možné.

Na prvý pohľad nevidno, že by variačný prístup súvisel s dynamickým programovaním, ktorým sme sa doteraz zaoberali. Ukážeme si však, že súvis tu je a viaceré objekty a pojmy z jednej teórie majú prirodzenú interpretáciu v druhej z nich.

3.1 Označenia, symbolika a formulácia úlohy

Z dôvodov, uvedených v úvode ku tejto kapitole, budeme odteraz pracovať v reálnom euklidovskom priestore: n-rozmerný priestor budeme označovať $\mathbb{R}^n$. Jeho prvky, n-rozmerné vektory, budeme chápať ako *ceste*, čo je priamka daná rovnicou $u = \delta(1+r)/(1+\delta)x$. To vysvetľuje existenciu optimálneho riadenia iba pre $b = 0$.

V prípade $r = \delta$, má úloha nekonečne veľa rovnovážnych stavov, ktoré sú nestabilné. Rovnovážnym stavom je každá dvojica $\hat{x}, \hat{u}$ taká, že $\hat{u} = r\hat{x}$. Trajektórie ležia na priamkach rovnobežných s osou x (viď Obr. 3.1 vpravo). To znamená, že riešenia, ktoré sme našli vyššie pre $a = b$ zodpovedajú rovnovážnym stavom systému. Všetky riešenia pre iné dvojice počiatočných a koncových bodov divergujú.



Obr. 3.1: Trajektórie odozvy a riadenia v Príklade 3.8 pre $x_0 = 2$, pre rôzne počiatočné hodnoty u_0 a pre $r < \delta$ (vľavo), $r > \delta$ (v strede) a $r = \delta$ (vpravo).

Nasledujúci príklad je zjednodušenou deterministickou verziou úlohy o reálnych obchodných cykloch (real business cycle, RBC). Modelujú sa ňou národohospodárske odozvy na technologické a iné šoky. RBC modely sú východiskom pre DGSE (dynamic general stochastic equilibrium) modely, ktoré sú v súčasnosti populárnym nástrojom ekonomickej analýzy a prognózovania.