Soubor řešených příkladů do předmětu Optimalizace

Ondřej Kolesík

Bakalářská práce
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Ondřej Kolesík**
Osobní číslo:	**A15051**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**Prezenční**
Téma práce:	**Soubor řešených příkladů do předmětu Optimalizace**
Téma práce anglicky:	**A Set of Demonstrative Examples for the Optimisation Course**

Zásady pro vypracování

1. Seznamte se obsahem předmětu Optimalizace a vypracujte jeho sylabus.
2. Pro vybrané kapitoly obecně definujte úlohy a popište metody jejich řešení.
3. Navrhněte vhodné úlohy s důrazem na jejich atraktivitu a praktickou aplikovatelnost a vybranými metodami je vyřešte.
4. K vybraným kapitolám navrhněte sadu neřešených úloh.
5. Získané výsledky vhodným způsobem prezentujte a zpřístupněte.

Rozsah bakalářské práce:
Rozsah příloh:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. PEKAŘ, Libor. *Optimalizace* [online]. Elektronický interaktivní učební text FAI UTB ve Zlíně. Zlín, 2014 [cit. 2019-11-15], 181 s. Dostupné z: https://moodle.utb.cz/mod/resource/view.php?id=205912
2. MAŇAS, Miroslav. *Optimalizační metody*. 1. vyd. Praha: SNTL, 1979, 257 s.
3. TEMLÍK, Petr. *Optimalizace* [online]. Zlín, 2005 [cit. 2019-11-15]. Dostupné z: https://stag.utb.cz. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Roman Prokop.
4. VENKATARAMAN, P. *Applied Optimization with Matlab Programming*. New York: John Wiley & Sons, 2002, 398 s. ISBN 0471349585
5. VÍTEČKOVÁ, Miluše, JEDLIČKA, David. *Statická optimalizace systémů* [online]. VŠB-TU v Ostravě, 2003 [cit. 2019-11-15]. Dostupné z: http://books.fs.vsb.cz/StatickaOptimalizace/index.htm.

Vedoucí bakalářské práce: **doc. Ing. Libor Pekař, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **20. prosince 2019**
Termín odevzdání bakalářské práce: **15. května 2020**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Ve Zlíně dne 20. prosince 2019

Jméno, příjmení: Ondřej Kolesík
Název bakalářské práce: Soubor řešených příkladů do předmětu Optimalizace
Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 11.8.2020

Ondřej Kolesík, V.R.
podpis diplomanta

ABSTRAKT

Cílem této práce bylo přispět pedagogickému výkladu a snaha pomoct studentům pochopit danou látku. V první části práce jsou představeny jednotlivé druhy optimalizačních metod, a jejich obecní postup řešení. Druhá část práce se zabývá praktickými úkoly a jejich řešení. Praktické úlohy jsou zároveň uvedené v prezentaci.

Klíčová slova: extrém funkce, optimalizace, iterace, komparace, gradient, simplex, lineární programování, dynamické programování

ABSTRACT

The aim of this thesis was to contribute to the pedagogical interpretation and help students to understand the given topic. In the first part, we are introduced to each type of optimizations methods and their general solution procedure. The second part deals with practical tasks and their solutions. Practical tasks are also stated in the presentation.

Keywords: extreme functions, optimization, iteration, comparison, gradient, simplex, linear programming, dynamic programming

OBSAH

ÚVOD

Optimalizační metody jsou matematické disciplíny, u kterých se snažíme nalézt takovou hodnotu proměnných, pro které účelová funkce nabývá svou maximální nebo minimální hodnotu. S těmito metodami se velmi často setkáváme i v praxi a pomáhají nám získat v různých odvětvích nejlepší možné řešení. Díky optimalizaci můžeme například zajistit maximální zisk výroby, najít ideální cestu nebo nejlepší využití energie. Jedná se tedy o velice silný nástroj, díky němuž lze zefektivnit nebo zkvalitnit mnoho případů lidské činnosti, je tedy nutné se jí patřičně věnovat a správně pochopit její využití. Optimalizační metody můžeme rozdělit do několika kategorií podle toho, co se snažíme zjistit a jakým způsobem se k výsledku dostaneme. Mezi tyto metody patří analytické, iterační, grafické nebo speciální metody. Analytické metody využívají diferenciální či variační počty, a to buď v klasické, nebo neklasické formě. Iterační metody používají informace z předchozích výpočtů ke zlepšení řešení. Pod pojmem speciální metody rozumíme lineární a dynamické programování, které velice často využíváme v každodenním životě. Tyto příklady různých metod jsou pouhým zlomkem optimalizačních metod, proto je nesmírně nutné danou problematiku patřičně pochopit, aby bylo možné zjistit ideální způsob řešení daných úkolů. Práce samotná se snaží podpořit výuku Optimalizace a pomoct studentům pochopit základní principy a postupy. V teoretické části práce je ukázáno několik vybraných metod optimalizace a je zde popsán způsob jejich řešení. Praktická část práce se zabývá konkrétními příklady a jejich řešení, nacházíme zde ale i několik neřešených příkladů k procvičování. Jednotlivé příklady jsou zároveň přístupné v přiložené prezentaci.

I. TEORETICKÁ ČÁST

1 SYLABUS

1.1 Cíl předmětu

Cílem předmětu je vyhledávání extrémů funkcí při různých omezeních na definiční obor. V tomto smyslu se rozlišuje volný extrém funkce, klasický vázaný extrém (omezení rovnostmi) a neklasický vázaný extrém (omezení nerovnostmi). Z hlediska metod se jedná o metody analytické, iterační a speciální metody, které v sobě zahrnují operační analýzy. Na cvičeních je kladen důraz na optimalizační úkoly, které mají technologický nebo ekonomický kontext.

1.2 Kapitoly

1. Řešení jednorozměrného extrému analytickou metodou
2. Řešen vícerozměrného extrému analytickou metodou
    a. Vícerozměrný volný extrém
    b. Vícerozměrný klasický vázaný extrém
    c. Vícerozměrný neklasický vázaný extrém
3. Komparativní iterační metody
    a. Jednorozměrné komparativní iterační metody
    b. Vícerozměrné komparativní iterační metody
4. Gradientní iterační metody
    a. Jednorozměrné gradientní iterační metody
    b. Vícerozměrné gradientní iterační metody
5. Iterační metody s omezením, metody náhodného vyhledávání
    a. Metoda projekce gradientu
    b. Využití penalizační funkce
    c. Využití bariérové funkce
    d. Metody náhodného vyhledávání

6. Lineární programování
    a. Lineární programování
    b. Metoda simplexové tabulky
    c. Primární a duální model lineárního programování
    d. Celočíselné lineární programování
7. Dynamické programování
    a. Úloha dynamického programování
    b. Tabulková forma dynamického programování
    c. Síťová forma dynamického programování

2 ŘEŠENÍ JEDNOROZMĚRNÉHO EXTRÉMU ANALYTICKOU METODOU

2.1 Jednorozměrný volný extrém

V této kapitole budeme pomocí analytické metody hledat volný extrém. Pod pojmem volný extrém rozumíme lokální maximum nebo minimum, které se nachází v libovolném bodě definičního oboru funkce.

Při řešení těchto úkolů budeme pracovat s reálnou funkcí $f(x) \in R$, která obsahuje jednu reálnou proměnnou $x \in R$. Zadání této úlohy lze vyjádřit vztahem:

$$\text{extrém } f(x) \in R, x \in R$$

Pokud má funkce pouze jediný extrém, nazýváme ji unimodální. V případě, že funkce má více extrémů, nazýváme ji multimodální. [5]

2.1.1 Postup řešení

Metody analytického vyhledávání extrémů jsou založené na výpočtu derivací, proto jsou vhodné hlavně pro analyticky zadané funkce.

Pro řešení těchto úloh budeme používat tyto čtyři věty:

Věta 1.

Je-li $f(x)$, v bodě x_0 diferencovatelný a je v něm lokální maximum nebo minimum, pak je $f'(x_0) = 0$, tudíž v bodě x_0 se nachází stacionární bod. [1]

Věta 2.

Je-li x_0 stacionárním bodem $f(x)$ a $f'(x_0)$ je v x_0 rostoucí, má funkce $f(x)$ v x_0 ostré lokální minimum. Pokud je klesající má funkce $f(x)$ v x_0 ostré lokální maximum.

Rostoucí nebo klesající trend $f'(x_0)$ lze často určit podle znaménka druhé derivace:

- Je-li $f'(x_0) = 0$ a $f''(x_0) > 0$, je $f'(x)$ v x_0 rostoucí.
- Je-li $f'(x_0) = 0$ a $f''(x_0) < 0$, je $f'(x)$ v x_0 klesající.
- Je-li $f'(x_0) = 0$ a $f''(x_0) = 0$, je nutné v tomto případě vypočítat vyšší derivaci. [1]

Věta 3

Jestliže pro $f(x)$ v x_0 nastane situace, kdy je diferencovatelná, až do řádu n a

$$f(x_0) = f'(x_0) = f''(x_0) = \cdots = f^{(n-1)}(x_0) = 0 \text{ a } f^{(n)}(x_0) \neq 0 \text{, pak:}$$

- Je-li n sudé a $f^{(n)}(x_0) > 0$, je v x_0 ostré lokální minimum $f(x)$.
- Je-li n sudé a $f^{(n)}(x_0) < 0$, je v x_0 ostré lokální maximum $f(x)$.
- Je-li n liché, je v x_0 inflexní bod $f(x)$.

Pokud nastane situace, kdy první nebo druhá derivace neexistuje, postupuje se podle čtvrté věty. [1]

Věta 4

Je-li $f'(x_0) = 0$ nebo neexistuje, a $f''(x_0)$ neexistuje, pak:

- Pokud je $f'(x_0) > 0$ na levém okolí x_0 a $f'(x_0) < 0$ na pravém okolí x_0, je v x_0 ostré lokální maximum $f(x)$.
- Pokud je $f'(x_0) < 0$ na levém okolí x_0 a $f'(x_0) > 0$ na pravém okolí x_0, je v x_0 ostré lokální minimum $f(x)$. [1]

3 ŘEŠENÍ VÍCEROZMĚRNÉHO EXTRÉMU ANALYTICKOU METODOU

3.1 Vícerozměrný volný extrém

Ve většině optimalizačních úloh se vyskytuje více jak jeden optimální parametr, takové úkoly jsou označované, jako vícerozměrné úlohy, kde vícero zadaných parametrů tvoří vektor. U řešení vícerozměrných úloh se na definičním oboru funkcí můžeme setkat s omezení nebo s podmínkami. Podle nich určujeme, jestli je extrém volný nebo vázaný. Úlohy volného extrému rozumíme takové, u kterých je definiční obor R^n.

3.1.1 Postup řešení

Stejně jako u úlohy jednorozměrného volného extrému je i zde úkolem nalézt lokální nebo globální extrém. S tím rozdílem, že extrém účelové funkce $f(x)$ je zde chápán, jako extrém $f(x) \in R, x \in R^n$.

Při řešení úlohy nejdříve zjistíme gradient funkce. Ten vypočítáme tak, že funkci $f(x)$, která je na intervalu $I \subseteq R^n$, parciálně derivujeme.[7]

$$grad\ f(x) = \nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n}\right]^T$$

Po výpočtu gradientu, vypočítáme stacionární bod. Pro jeho vypočet platí podmínka: $\nabla f(x) = 0 = [0,0,\dots,0]^T$.

Nyní vypočítáme Hessovu matici. Ta zobecňuje druhou derivaci a je tedy maticí druhých parciálních derivací $f(x)$. Hessova matice $H(f(x))$ je vždy čtvercová symetrická matice. [9]

$$H\big(f(x)\big) = \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Pro zjištění extrému se musí určit, zda je Hessova matice pozitivně nebo negativně definitivní. Pokud je matice $H(f(x_0))$ pozitivně definitivní, znamená to, že v bodě x_0 se nachází lokální minimum, pokud je negativní, nalézá se v něm lokální maximum. Definitivnost matice lze určit pomocí Silvestrova kritéria. To nám říká, že pokud jsou hlavní subdeterminanty matice kladné, matice je pozitivně definitivní. Pokud střídají podmínku, ale zároveň je první subdeterminant záporný, matice je negativně definitivní. [3]

$\begin{pmatrix} 3 & 1 \\ 3 & 2 \end{pmatrix}$ První subdeterminant je 3, druhy je 3. Matice je tedy pozitivně definitivní.

$\begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$ První subdeterminant je -2, druhy je 3. Matice je negativně definitivní.

$\begin{pmatrix} 1 & 3 \\ 3 & 2 \end{pmatrix}$ První subdeterminant je 1, druhy je -7. Matice není ani pozitivně definitivní ani negativně definitivní. [11]

3.2 Vícerozměrný klasický vázaný extrém

Pokud úloha má vazební podmínku ve tvaru rovnosti, označuje se za úlohu klasického vázaného extrému.

3.2.1 Postup řešení

Úkolem je nalézt extrém funkce $f(x)$, u kterého platí omezení (vázaná podmínka) ve tvaru $g_j(x_1, \dots, x_n) = 0;\ j = 1, \dots, m;\ m < n$. Funkce je v tomto případě ve tvaru: $f(x) \in R$, $x \in R^n$. Pro nalezení bodů podezřelých z extrému, použijeme větu o Lagrangeových multiplikátorech.

Nejprve vytvoříme Lagrangeovu funkci (lagrangián), ta je definovaná jako:

$$\Phi(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)$$

Poté zjistíme gradient této funkce pomocí derivace jednotlivých proměnných. Výsledné rovnice položíme rovno nule. [6]

$$\nabla\Phi(x_0, \lambda) = \nabla f(x) + \sum_{j=1}^{m} \lambda_j \nabla g_j(x_0) = 0$$

Z rovnic následně vypočítáme nulové body, které následně vložíme do Hessovy matice, z níž poté vypočítáme extrémy funkce. [1]

3.3 Vícerozměrný neklasický vázaný extrém

Stejně jako v předchozím případě, i zde úkoly vychází z Lagrangeovy funkce, s tím rozdílem, že úkoly jsou zde omezeny podmínkou ve tvaru nerovnosti. Při řešení těchto úloh je nutné počítat s výskytem sedlového bodu, ten je možné vyřešit pomocí Kuhn-Tuckerovy věty.

3.3.1 Postup řešení

Kuhn-Tuckerova věta nám říká, pokud existuje $x_0 \geq 0$ a $\lambda_0 \geq 0$, pro které platí $x \geq 0, \lambda \geq 0$, potom Lagrangeova funkce $\Phi(x, \lambda)$ bude ve tvaru $\Phi(x_0, \lambda) \leq \Phi(x_0, \lambda_0) \leq \Phi(x, \lambda_0)$. Optimální řešení x_0 takové úlohy je tedy minimum Lagrangeovy funkce $\Phi(x, \lambda)$, ve směru x a maximum ve směru λ, v tomto místě se nachází sedlový bod. [6]

Kuhn-Tuckerova věta se při počítání analytických úloh používá většinou v upravené formě. Pokud máme reálné funkce $f(x), g_1(x), \dots g_m(x)$, které mají spojité parciální derivace. Kuhn-Tuckerova věta bude ekvivalentní následujícím podmínkám:

$$\nabla_{\mathrm{x}} \Phi(x_0, \lambda_0) \geq 0$$

$$\nabla_{\mathrm{y}} \Phi(x_0, \lambda_0) \leq 0$$

$$\mathrm{x}_{0\mathrm{i}} \frac{\partial \Phi(x_0, \lambda_0)}{\partial \mathrm{x}_\mathrm{i}} = 0, i = 1, \dots, n$$

$$\lambda_{0j} \frac{\partial \Phi(x_0, \lambda_0)}{\partial \lambda_j} = 0, j = 1, \dots, n$$

$$x_{0i}, \lambda_{0j} \geq 0$$

Pokud nastane situace, že v zadání úlohy se nám nevyskytuje nutná podmínka $x_i \geq 0, i = 1, \dots, n$, je možné použít jiný postup výpočtu.

Pokud máme reálné funkce $f(x), g_1(x), \dots g_m(x)$, které mají spojité parciální derivace a bodem lokálního maxima funkce $f(x)$ je x_0, existuje v úloze neklasického vázaného extrému takové $\lambda_0 \geq 0$, že můžeme úlohu změnit na klasický vázaný extrém. To provedeme nahrazením všech omezujících nerovností $g_j(x) \leq 0$ na rovnosti $g_j(x) = 0$. [4]

4 KOMPARATIVNÍ ITERAČNÍ METODY

4.1 Jednorozměrný komparativní iterační metody

Iterační úlohy jsou takové úlohy, při kterých se využívají předchozí výpočty diskrétních hodnot účelové funkce a jejich porovnání. Pokud je nově vypočítaná hodnota lepší jak předchozí, ponecháváme si novější hodnotu, v opačném případě zůstává stará hodnota. Tímto způsobem postupně zpřesňujeme konečný výsledek. U komparativních metod není potřeba výpočet derivací, ale pouze hodnot účelové funkce. Mezi jednorozměrné komparativní iterační metody patří Fibonacciho metoda nebo metoda zlatého řezu. U jednorozměrného případu lze princip vyjádřit vztahem:

$$x_{(k+1)} = x_{(k)} + \Delta_{(k)}$$

$$\lim_{k \to \infty} x_{(k)} = x_{Opt}$$

Kde $\left\{x_{(k)}\right\}_{k=1}^{\infty}$ je posloupnost odhadů řešení optimalizační úlohy, $\Delta_{(k)}$ je přírůstek odhadu v iteračním kroku k a x_{Opt} je optimální řešení. [1]

4.1.1 Postup řešení

4.1.1.1 Fibonacciho metoda

Při řešení touto metodou, budeme mít funkci $f(x)$, která je omezena na intervalu $\langle a, b \rangle$. Při řešení této úlohy budeme postupně zmenšovat interval ve kterém předpokládáme výskyt extrému.

Fibonacciho posloupnost lze vyjádřit podle vztahu $\{F_k\}_{k=1}^{N}$ [7], nebo:

$$F_{(k)} = F_{(k-1)} + F_{(k-2)}, \text{ kde } F_{(0)} = F_{(1)} = 1.$$

Prvních několik členů posloupnosti potom vypadá následovně:

$F = \{1,1,2,3,5,8,13,21,34,55, \dots\}$.

Pokud nemáme určený počet kroků N, je nutné ho vypočítat podle vztahu:

$F_N \geq \frac{2}{\varepsilon}(b-a)$, kde F_N je Ntý prvek Fibonacciho posloupnosti, ε je přesnost.

Vnitřní body intervalu $\langle a_{(i)}, b_{(i)}\rangle$ se počítají podle vztahu:

$$x_{1,(i)} = b_{(i)} - \frac{F_{N-(i)}}{F_{N-(i+1)}}\left|b_{(i)} - a_{(i)}\right|$$

$$x_{2,(i)} = a_{(i)} + \frac{F_{N-(i)}}{F_{N-(i+1)}}\left|b_{(i)} - a_{(i)}\right|$$

Následně se vypočítá funkční hodnota $f\left(x_{1,(i)}\right), f\left(x_{2,(i)}\right)$ a provedeme jejich porovnání. Pokud hledáme minimum funkce, ohraničíme bod s nižší funkční hodnotou a vznikne nový interval. Pokud pro hledání minima platí $f\left(x_{1,(i)}\right) < f\left(x_{2,(i)}\right)$, pak

$a_{(i+1)} = a_{(i)}; b_{(i+1)} = x_{2,(i)}$.

Tento postup pokračuje do té doby, dokud počet iterací i nebude roven počtu kroků N.

4.1.1.2 Metoda zlatého řezu

Zlatým řezem je nazývána hodnota limitu podílu dvou po sobě jdoucích členů Fibonacciho posloupnosti $\lim_{i\to\infty}\frac{F_i}{F_{i+1}}$, proto je metoda zlatého řezu hodně podobná výše zmíněné metodě. I zde jsou vypočítáme v každém kroku dva body, z nichž jeden se stane v následující iteraci bodem hraničním a druhý zůstane ohraničený.

Hlavní rozdíl oproti Fibonacciho metodě je ten, že na místo poměru $\frac{F_{N-i}}{F_{N-i+1}}$ používáme hodnotu $\frac{\sqrt{5}-1}{2} \cong 0{,}618$. [7]

4.2 Vícerozměrný komparativní iterační metody

Nyní se budeme zabývat komparativními iteračními metodami, které se vyskytují ve vícerozměrném prostoru. Mezi tyto metody můžeme zařadit metodu mapování kriteriální plochy, Boxovu-Wilsonovu metodu, metodu cyklické záměny parametrů nebo simplexové metody. V této kapitole se budeme zabývat posledními dvěma zmíněními, tedy simplexovými metodami a metodou cyklické záměny parametru. Pro pochopení simplexových metod si nejdříve vysvětlíme, co znamená slovo simplex.

Pod pojmem simplex rozumíme nejmenší konvexní útvar v daném prostoru o $n+1$ vrcholech. Z dané definice tedy plyne, že ve dvou-rozměrném prostoru, bude simplex představovat trojúhelník. Pokud se jedná o pravidelný simplex, jedná se o útvar, který má všechny svoje vrcholy od sebe stejně vzdálené.[3]

4.2.1 Postup řešení

4.2.1.1 Metoda pravidelného simplexu

U této metody hledáme maximum účelové funkce pomocí překlápějících se pravidelných simplexů. To probíhá tak, že vypočteme hodnoty účelové funkce ve všech bodech daného simplexu a následně bod s nejmenší hodnotou překlopíme přes těžiště simplexu zbývajících vrcholů, do směru největšího růstu. Pokud bychom hledali minimum funkce, překlápěl by se bod s největší hodnotou.

4.2.1.2 Metoda flexibilního simplexu

Simplex u této metody už není pravidelný, jednotlivé hrany simplexu se v průběhu výpočtu mění tak, aby se rychleji přibližoval k předpokládanému extrému. Pokud by se simplex nacházel daleko od extrému, jeho hrany se prodlouží. Naopak v jeho blízkosti se hrany zkracují. Díky flexibilitě simplexu nedochází v blízkosti extrému k zacyklení.

4.2.1.3 Metoda cyklické záměny parametru

U této metody si nejdříve vybereme jednu proměnnou a ostatní necháme konstantní, pomocí jednorozměrné optimalizace najdeme pro tuto proměnou částečné optimum a zafixujeme její hodnotu, následně budeme stejný postup provádět s další proměnnou, dokud se všechny proměnné nevystřídají.

5 GRADIENTNÍ ITERAČNÍ METODY

5.1 Jednorozměrný gradientní iterační metody

Na rozdíl od předchozí komparativní metody, gradientní metody využívají pro svůj běh odhad nebo výpočet gradientu funkce, respektive derivace, v bodě odhadu extrému. Iterační gradientní metody jsou tedy takové, ve kterých se mezivýsledky jednotlivých iteračních kroků zvyšují úměrně gradientu účelové funkce. Obecně lze princip gradientních iteračních metod vyjádřit vztahem:

$$x_{(k+1)} = x_{(k)} + \lambda_{(k)} * \nabla f(x_{(k)}).$$

Parametr λ ovlivňuje svou hodnotou metodu a hledaný extrém. Bude-li $\lambda_i < 0$, bude postup funkce směřovat k minimu, tudíž bude platit vztah $f\left(x_{(k+1)}\right) < f\left(x_{(k)}\right)$. V opačném případě pro $\lambda_i > 0$ bude postup funkce směřovat k maximu a platí $f\left(x_{(k+1)}\right) > f\left(x_{(k)}\right)$.[3]

Zároveň, bude-li $\lambda_{(k)}$ v průběhu výpočtu konstantní, jedná se o gradientní metodu krátkého skoku, pokud ovšem bude $\lambda_{(k)}$ v každém svém kroku optimální, jedná se o metodu dlouhého kroku. [7]

Mezi nejznámější jednorozměrné gradientní metody patří Newtonova metoda a

metoda regula falsi.

5.1.1 Postup řešení

5.1.1.1 Newtonova metoda

Newtonova metoda je založená na rozvoji funkce v Taylorově řadě působící na okolní příslušné iterace.

Účelovou funkci $f(x)$ nejprve aproximujeme prvními třemi členy Taylorova rozvoje, v pevně daném bodě x_{ok} a z okolí pravděpodobného výskytu extrému x_{opt} . [10]

$$f(x) = f(x_{ok}) + f'(x_{ok})(x - x_{ok}) + \frac{1}{2} f''(x_{ok})(x - x_{ok})^2$$

Derivováním vztahu a vyřešením podmínky extrému $f'(x_{opt}) = 0$ následně obdržíme vztah:

$$x_{opt} = x_{ok} - \frac{f'(x_{ok})}{f''(x_{ok})}$$

Za předpokladu odhadu extrému v dané iteraci, blízko jeho skutečné hodnoty.[1]

Finální iterační vztah bude vypadat takto:

$$x_{(k+1)} = x_{(k)} - \frac{f'(x_{(k)})}{f''(x_{(k)})}$$

5.1.1.2 Metoda regula falsi

Tato metoda je předchozí Newtonově metodě velice podobná. Rozdíl, ale mezi nimi je v tom, že druhá derivace v podílu je zde nahrazena jejím odhadem:

$$f''(x_{(k)}) = \frac{f'(x_{(k)}) - f'(x_{(k-1)})}{x_{(k)} - x_{(k-1)}}$$

Po dosazení tohoto odhadu lze iterační krok metody regula falsi definovat vztahem:

$$x_{(k+1)} = x_{(k)} - f'(x_{(k)}) \frac{x_{(k)} - x_{(k-1)}}{f'(x_{(k)}) - f'(x_{(k-1)})}$$

5.2 Vícerozměrný gradientní iterační metody

Mezi nejznámější vícerozměrné gradientní iteráční metody patří metoda největšího spádu, které se budu věnovat v této kapitole. Metoda největšího spádu, popřípadě metoda nejstrmějšího směru, je metoda, kdy hledáme směr d, ve kterém funkce nejrychleji klesá, a pohybujeme se jejím směrem. Ve chvíli kdy funkce přestane klesat, zastavíme na aktuálním bodě a opět najdeme směr největšího spádu.

Nejstrmější směr poklesu lze vyjádřit vztahem:

$$d = \nabla f(x)$$

Tato metoda se podle zadaného λ, také označuje jako metoda krátkého, popřípadě dlouhého kroku.

5.2.1 Postup řešení

Máme zadanou účelovou funkci $f(x)$, s počátečním bodem $x_{(1)}$ a počtem iteračních kroků N, popřípadě minimální velikostí skoku $\varepsilon > 0$. Na začátku je počet iteraci $i = 1$, tudíž $x_{(i)} = x_{(1)}$.

Vypočítáme $\nabla f(x_{(i)})$. Pokud dále použijeme metodu krátkého kroku, budeme se řídit vztahem:

$$x_{(i+1)} = x_{(i)} + \lambda\, \nabla f\left(x_{(i)}\right), \lambda < 0$$

Pokud se jedná o metodu dlouhého kroku, nejprve přepočítáme optimální velikost kroku λ_{opt} v bodě x:

$$\lambda_{opt,(i)} = \underset{\lambda_{(i)} \in R}{arg}\ \ extr\, p\left(\lambda_{(i)}\right)$$

$$p\left(\lambda_{(i)}\right) = f\left(x_{(i)} + \lambda_{(i)} \nabla f\left(x_{(i)}\right)\right)$$

Potom můžeme dosadit do vztahu $x_{(i+1)} = x_{(i)} + \lambda_{opt,(i)}\, \nabla f\left(x_{(i)}\right)$. [1]

Postup opakujeme do té doby, dokud počet iterací i nedosáhne stanoveného počtu kroků N, nebo dokud $\left\|\lambda_{opt,(i)}\, \nabla f\left(x_{(i)}\right)\right\|$ nebude menší, než velikost skoku ε.

6 LINEÁRNÍ PROGRAMOVÁNÍ

6.1 Lineární programování

Lineární programování, nebo také lineární optimalizace je metoda umožňující hledání optimálního řešení při daném kritériu a daných omezujících podmínkách.

Základní formulace lineárního programování lze vyjádřit vztahem:

$$\max f(x) = \sum_{i=1}^{n} c_i\, x_i = c^T x$$

$$c^T x \in R, x \in R^n$$

Při daném omezení ve tvaru $Ax \leq b$ je v těchto úlohách Ax ve tvaru matice $m \times n$. Také zde platí podmínka nezápornosti u x_i.[6]

$$\begin{matrix} a_{11}x_1 + & \dots & +a_{1n}x_n \leq b_1 \\ \vdots & \ddots & \vdots \\ a_{m1}x_1 + & \dots & +a_{mn}x_n \leq b_m \end{matrix}$$

Typicky je tato metoda používaná v ekonomicky zaměřeních úlohách, kde $f(x)$ značí maximální zisk, x_i objem výroby, c_i pevně danou cenu a $Ax \leq b$ omezující podmínku, například čas, dostupný prostor, finance, zaměstnance atd.

6.1.1 Postup řešení

Nejprve soustavu nerovnic v omezení $Ax \leq b$ převedeme do kanonického tvaru tím, že je upravíme na rovnost přidáním proměnné „vůle“ $x_{n+1}, x_{n+2,} \dots, x_{n+m} \geq 0, \tilde{A}\tilde{x} = b$. Následně dané rovnice převedeme do matice. [9]

$$\begin{matrix} a_{11}x_1 + & \dots & +a_{1n}x_n \leq b_1 \\ \vdots & \ddots & \vdots \\ a_{m1}x_1 + & \dots & a_{mn}x_n \leq b_m \end{matrix} \Rightarrow \begin{matrix} a_{11}x_1 + & \dots & +a_{1n}x_n + x_{n+1} = b_1 \\ \vdots & \ddots & \vdots \\ a_{m1}x_1 + & \dots & +a_{mn}x_n + x_{n+m} = b_m \end{matrix}$$

$$\Downarrow$$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Nyní máme soustavu m lineárních algebraických rovnic o $n + m$ neznámých, abychom zjistili, jestli má rovnice řešení, použijeme podmínku řešitelnosti $rank\ \tilde{A} = rank\ (\tilde{A}, b)$.

Nyní určíme počet základních řešení, ten zjistíme tak, že vezmeme všechny bazické (závislé) proměnné n a nebazické (volitelné) proměnné m a dáme je do vztahu $\binom{n+m}{m}$.

Základní přípustné řešení rozumíme takové, které obsahuje nezáporné prvky. To získáme pomocí eliminační metody použitou na soustavu rovnic.

Optimální řešení, potom rozumíme takové řešení, které extremalizuje účelovou funkci.

Pro optimální řešení, respektivě základní přípustné řešení, platí dvě podmínky:

- Vektor je základním přípustným řešením právě tehdy, když je vrcholem polyedru omezený.
- Existuje-li optimální řešení úlohy lineárního programování, pak je i základním přípustným řešením.

6.2 Metoda simplexové tabulky

Tato metoda vychází ze základní úlohy lineárního programování, proto je začátek postupu stejný jako v předchozím případě. Při kontrole podmínek nezápornosti musí při použití této metody platit podmínka, že pravá strana rovnice b, musí být taktéž nezáporná. Po převedení na kanonický tvar, vytvoříme simplexovou tabulku.[2]

Tabulka 1 Simplexová tabulka

x_1	x_2	...	x_n	x_{n+1}	x_{n+2}	...	x_{n+m}	omezení
a_{11}	a_{12}	...	a_{1n}	1	0	...	0	b_1
a_{21}	a_{22}	...	a_{2n}	0	1	...	0	b_2
...	...	...	...	...	...	...	...	...
a_{m1}	a_{m2}	...	a_{mn}	0	0	...	1	b_m
$-c_1$	$-c_2$	...	$-c_m$	0	0	...	0	f

6.2.1 Postup řešení

Nejprve převedeme rovnici na kanonický tvar a výsledné hodnoty zaneseme do simplexové tabulky.

Následně otestujeme, jestli její výsledek má optimální řešení. To zjistíme tehdy, když budou všechna čísla v posledním řádku tabulky nezáporná. Výjimku zde tvoří pomocné proměnné a poslední sloupec tabulky.

Nyní určíme klíčový sloupec, to je sloupec s nejmenší hodnotou v posledním řádku, označíme jej indexem j_k. Výjimku opět tvoří pomocné proměnné a poslední sloupec tabulky.

Po nalezení klíčového sloupce hledáme klíčový řádek i_k, ten vypočítáme z podílů hodnot omezení a čísel v klíčovém sloupci, jako minimum z nezáporných podílů:

$$i\_k = \arg\min_i \beta_i = \frac{b_i}{a_{ij\_k}} \geq 0, i = 1,2,\dots,m$$

Výpočet klíčového řádku nezahrnuje poslední řádek. [9]

Následuje eliminace klíčového prvku na pozici $[i\_k, j\_k]$. Pomocí standartní řádkové úpravy změníme hodnotu klíčového prvku na jedna a zbylé hodnoty v klíčovém sloupci na nulu, touto metodou změníme klíčový sloupec na jednotkový vektor. Po tomto kroku opět dojde na kontrolu optimálního řešení. V případě že tabulka optimální řešení neobsahuje, celý postup se opakujeme, dokud ho nedosáhneme. [1]

Pokud tabulka již obsahuje optimální řešení, výsledek úlohy se zjistí následujícím postupem:

- Hodnota čísla v dolním řádku a posledním sloupci určuje hodnotu účelové funkce f_{opt}.
- Hodnota proměnných, ke kterým nenáleží jednotkový vektor, je nulová.
- Hodnota proměnných, ke kterým jednotkový vektor náleží, odpovídá hodnotě posledního sloupce v řádku, kde je ve vektoru hodnota jedna. [1]

6.3 Primární a duální model lineárního programování

K primárnímu modelu lineárního programováni, lze přidat takzvaný duální model lineárního programování. Jak bylo řečeno v předchozí části, primární model je definovaný jako:

$$\max f(x) = c^T x$$

$$Ax \leq b$$

$$x_i \geq 0$$

Duální nebo také symetrický model je definovaný jako:

$$\min g(y) = b^T x$$

$$A^T y \geq c$$

$$y_j \geq 0$$

Pro duální úlohy platí věta o souměrné dualitě. Ta nám říká, že pokud má jeden z modelů řešení s konečnou hodnotou účelové funkce, má ho i druhý model a obě funkce se rovnají. Pokud ale nastane situace, kdy jeden z modelů má řešení s nekonečnou hodnotou účelové funkce, druhý model nemá přípustné řešení. [8]

Musíme si také dávat pozor, jestli jeden z modelů nemá více optimálních řešení, v tom případě je řešení úlohy v degenerovaném tvaru.

6.3.1 Postup řešení

Při řešení duálních úloh, si musíme dát pozor na typ nerovnosti v primárním modelu. Pokud jsou všechny nerovnosti typu $Ax \leq b$, sestavení duálního modelu pokračuje podle definice. Pokud tomu tak není, můžeme primární model upravit podle těchto podmínek:

- Pokud se jedná o nerovnost $Ax \geq b$, obrátíme hodnoty čísel vynásobením čísla -1.
- Pokud je ve tvaru rovnosti $Ax = b$, zavedou se místo ní dvě nerovnosti, jedna typu $Ax \geq b$ a druhá $Ax \leq b$. Z nově vzniklou nerovností typu $Ax \geq b$ se vypořádáme podle prvního kroku.

Tento problém lze také vyřešit pomocí přídavných proměnných, díky nimž lze přepsat nerovnice na rovnice.

7 DYNAMICKÉ PROGRAMOVÁNÍ

7.1 Úloha dynamického programování

Základním principem dynamického programování je rekurzivní dělení úlohy na menší části. To nám umožnuje rozložit zadanou úlohu na menší pod úkoly, jejichž řešení si ukládáme pro další použití. Řešení úloh dynamického programování, je založeno na Bellmanově principu optimality. Tento princip nám říká, že podstrategie optimální strategie je opět optimální. Tento princip můžeme také formulovat jako:

Optimální strategie z libovolného bodu řešení do cíle, nezáleží na způsobu, jakým se řešení dostalo do tohoto bodu. Přičemž cílem je zde myšleno řešení celého programu. [6]

Při použití Bellmanova principu v dynamickém programování, musí mít úloha vždy dva průchody: přímý a zpětný.

Následující příklady budeme řešit ve tvaru nelineární separovatelné účelové funkce s lineárním omezením. To je možné vyjádřit vztahem:

$$\operatorname{ext} f(x) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) = \sum_{i=1}^{n} f_i(x_i)$$

Při omezení ve tvaru:

$$\sum_{i=1}^{n} a_i x_1 \leq b$$

$$x_i \geq 0, i = 1,2,\ldots,n$$

7.2 Tabulková forma dynamického programování

Princip této metody je založen na výše zmíněných postupech, kdy budeme pomocí tabulky řešit úlohu v diskrétních krocích.

Tabulka 2 Tabulka dynamické metody

b_j	$x_1(b_j)$	$F_1(b_j)$	$x_2(b_j)$	$F_2(b_j)$	...	$x_n(b_j)$	$F_n(b_j)$
$0 = b_0$	$x_{opt,1,0}$	$F_1(b_0)$	$x_{opt,2,0}$	$F_2(b_0)$	...	$x_{opt,n,0}$	$F_n(b_0)$
b_1	$x_{opt,1,1}$	$F_1(b_1)$	$x_{opt,2,1}$	$F_2(b_1)$	...	$x_{opt,n,1}$	$F_n(b_1)$
...	...	...	...	...	...	...	...
b_{m-1}	$x_{opt,1,m-1}$	$F_1(b_{m-1})$	$x_{opt,2,m-1}$	$F_2(b_{m-1})$	...	$x_{opt,n,m-1}$	$F_{n,}(b_{m-1})$
b_m	$x_{opt,1,m}$	$F_1(b_m)$	$x_{opt,2,m}$	$F_2(b_m)$	...	$x_{opt,n,m}$	$F_n(b_m)$

První sloupec je tvořen hodnotami, které je možno dosadit jako proměnné, tyto hodnoty získáváme z podmínek úlohy. [1] Platí pro ně omezení ve tvaru:

$$0 = b_0$$

$$b_0 < b_1 < \cdots < b_m = b$$

$$b_{j+1} - b_j = h_j$$

$$j = 0,1, \dots, m-1$$

Hodnota h udává vzdálenost kroku, která je konstantní.

Při první etapě vytvoříme dva sloupce. První sloupec bude obsahovat hodnoty proměnné a druhý hodnoty etapy účelové funkce. Při každé další etapě budeme vytvářet opět dva sloupce, kde pro výpočet účelové funkce budeme používat výsledky z předchozí metody.

7.2.1 Postup řešení

Nejprve rozdělíme zdroje b podle vztahu výše, inicializujeme počitadlo průchodů $i = 1$ a definujeme $F_{0,j}(x_0) = 0$.

Pro každé dělení zdroje b_j nalezneme jeho optimální rozdělení mezi proměnnými.

7.3 Síťová forma dynamického programování

Síťoví model je typ úlohy, který nám umožnuje hledat nejkratší cestu v grafu $G = \langle V, E \rangle$. Grafem G rozumíme dvojici množin vrcholů V a hran E.

Graf můžeme označit za orientovaný, ohodnocený nebo souvislý.

Orientovaným grafem rozumíme to, že mezi vrcholy existují hrany, které mají vždy udaný směr. Vrcholy tohoto grafu jsou uspořádané konce $E \subseteq V \times V$.

Pokud je každé hraně přiřazeno číslo, je graf ohodnocený. Jestli mezi každými dvěma vrcholy existuje nepřerušená cesta, kde každý vrchol se projde jenom jednou, je graf souvislý. [1]

7.3.1.1 Borůvkův algoritmus

Jedná se o algoritmus pro hledání minimální kostry v grafu. Tato metoda je pojmenovaná po Otakaru Borůvkovi, který ji použil pro návrh elektrické sítě na Moravě.

7.3.2 Postup řešení

Pro hledání nejkratší cesty, použijeme Dijkstrův algoritmus.

1. Ke každému vrcholu přiřadíme vzdálenost od počátečního vrcholu, výchozí vrchol označíme hodnotou 0, ostatní jako nekonečno.
2. Výchozí vrchol označíme jako aktivní, zbytek za nenavštívené a vytvoříme z nich množinu.
3. Nyní zkontrolujeme sousední vrcholy aktivního uzlu a vypočítáme jejich vzdálenost od počátečního bodu. Pokud je aktuální vypočítaná velikost menší než dříve zaznamenaná, přepíšeme hodnotu vrcholu.
4. Jakmile zkontrolujeme všechny sousedy aktuálního vrcholu, označíme ho za navštívený a vyjmeme ho z množiny nenavštívených, k tomuto vrcholu se už nebudeme vracet.
5. Nyní vybereme z množiny nenavštívených vrcholů, vrchol s nejmenší vzdáleností od počátečního bodu a označíme ho za aktivní.
6. Pokud jsme se dostali k cílovému bodu, ukončíme algoritmus, v opačném případě pokračujeme od bodu 3. [14]

7.3.2.1 Borůvkův algoritmus

Na začátku cyklu jsou všechny vrcholy považované za samostatné komponenty. Následně projdeme jednotlivé komponenty a pomocí nejkratší hrany je spojíme dohromady. Po prvním kroku máme vytvořené komponenty, které jsou složené minimálně ze dvou vrcholů. Pokračujeme ve spojování komponentů, dokud nejsou všechny vrcholy spojené dohromady. [12] [13]

I. PRAKTICKÁ ČÁST

8 JEDNOROZMĚRNÝ VOLNÝ EXTRÉM

8.1.1 Příklad 1

Nalezněte lokální extrémy funkce $f(x) = 5 - 4x^2 + 2x^3$

8.1.2 Příklad 2

Nalezněte lokální extrémy funkce $f(x) = -\frac{x^2+2x}{1+5x^3}$

8.1.3 Příklad 3

Analyticky nalezněte všechny lokální extrémy funkce $f(x) = |x^2 + e|$

8.2 Postup řešení

8.2.1 Příklad 1

Nejprve vypočítáme první derivaci funkce $f(x)$, následný výsledek položíme roven nule a zjistíme stacionární body.

$$f(x) = 5 - 4x^2 + 2x^3$$

$$f'(x) = 6x^2 - 8x = 2x(3x - 4)$$

$$2x(3x - 4) = 0 \Leftrightarrow x_{0,1} = 0, x_{0,2} = \frac{4}{3}$$

Nyní vypočítáme druhou derivaci funkce $f(x)$ a zjistíme její hodnoty ve stacionárních bodech.

$$f''(x) = 12x - 8$$

$$f''(0) = -8$$

$$f''\left(\frac{4}{3}\right) = 8$$

Podle druhé věty je ve funkci $f(x)$, v bodě $x_{0,1} = 0$ ostré lokální maximum a v bodě $x_{0,2} = \frac{4}{3}$ je ostré lokální minimum.

8.2.2 Příklad 2

Vypočítáme první derivaci funkce $f(x)$.

$$f(x) = -\frac{x^2 + 2x}{1 + 5x^3}$$

$$f'(x) = -\frac{(x^2 + 2x)'(1 + 5x^3) - (x^2 + 2x)(1 + 5x^3)'}{(1 + 5x^3)^2}$$

$$f'(x) = -\frac{-5x^4 - 20x^3 + 2x + 2}{(1 + 5x^3)^2}$$

Nyní vypočítáme stacionární bod.

$$\frac{-5x^4 - 20x^3 + 2x + 2}{(1 + 5x^3)^2} = 0 \Leftrightarrow x_{0,1} \approx 0{,}51175, x_{0,2} \approx -3{,}98110$$

Následně vypočítáme druhou derivaci funkce $f(x)$ a zjistíme její hodnoty ve stacionárních bodech.

$$f''(x) = -\frac{2(25x^6 + 150x^5 - 35x^3 - 60x^2 + 1)}{(1 + 5x^3)^3}$$

$$f''(0{,}51175) = 5{,}8777$$

$$f''(-3{,}98110) = -0{,}00316$$

Funkce $f(x)$ má v bodě $x_{0,1} = 0{,}51175$ ostré lokální minimum a v bodě $x_{0,2} = -3{,}98110$ je ostré lokální maximum.

8.2.3 Příklad 3

Nejprve vypočítáme 1. derivaci $f(x)$

$$f(x) = |x^2 + e|$$

$$f(x) = -x, x < 0$$

$$f(x) = x, x > 0$$

$$f'(x) = -2x, x < 0$$

$$f'(x) = 2x, x > 0$$

Jelikož nejde stacionární bod určit a v $x_0 = 0$ nelze derivovat $f(x)$, využijeme čtvrtou větu. Na levém okolí bodu $x_0 = 0$ je první derivace záporná a na pravém okolí je kladná, z toho vyplívá, že na levém okolí je funkce $f(x)$ klesající a na pravém rostoucí. Podle toho tedy určíme, že v $x_0 = 0$ se nachází lokální minimum.

8.3 Cvičení

8.3.1 Příklad 1

Nalezněte extrémy funkce $f(x) = \frac{x^2+2x}{2+6x^2}$

8.3.2 Příklad 2

Nalezněte extrémy funkce $f(x) = e^x - 2x + e^{-x} - 7x^3$

8.3.3 Příklad 3

Nalezněte extrémy funkce $f(x) = \frac{x^2-e^x}{2-2x^3}$

9 KOMPARATIVNÍ ITERAČNÍ METODY

Nalezněte minimum funkce $f(x) = -\frac{x^2+2x}{1+5x^3}$, pomocí Fibonacciho metody a zlatého řezu na intervalu $I = \langle 0{,}7 \rangle$.

9.1.1 Příklad 1

Pro Fibonacciho metodu je zadaná přesnost $\varepsilon = 0{,}4$

9.1.2 Příklad 2

Pro metodu zlatého řezu je zadaný počet iteračních kroků $i = 7$

9.2 Postup řešení

9.2.1 Příklad 1

Pro řešení prvního příkladu nejprve podle odhadu vypočítáme počet kroků.

$$F_N \geq \frac{2}{\varepsilon}(b-a)$$
$$F_N \geq \frac{2}{0{,}4}(7-0)$$
$$F_N \geq 35$$

F_N nám zde udává N-tý prvek Fibonacciho posloupnosti, budeme tedy hledat prvek větší nebo roven číslu 35.

$$F_0 = 1, F_1 = 1, F_2 = 2, F_3 = 3, \dots, F_8 = 34, F_9 = 55$$

První prvek, který splňuje podmínku je $F_9 = 55$, tedy $N = 9$. Počet iteračních kroků i, vypočítáme podle vztahu $i = N - 2$. Počet iteračních kroků tedy bude 7.

Nyní vypočítáme vnitřní body $x_{1,(1)}$ a $x_{2,(1)}$ funkce $f(x)$.

$$x_{1,(1)} = b_{(1)} - \frac{F_8}{F_9}\left|b_{(1)} - a_{(1)}\right| = 7 - \frac{34}{55} * 7 = 2{,}6727$$

$$x_{2,(1)} = a_{(1)} + \frac{F_8}{F_9}\left|b_{(1)} - a_{(1)}\right| = 0 + \frac{34}{55} * 7 = 4{,}3272$$

Po vypočítání těchto bodů, vypočítáme jejich funkční hodnoty.

$$f\left(x_{1,(1)}\right) = -\frac{2{,}6727^2 + 2 * 2{,}6727}{1 + 5 * 2{,}6727^3} = -0{,}1294$$

$$f\left(x_{2,(1)}\right) = -\frac{4{,}3272^2 + 2 * 4{,}3272}{1 + 5 * 4{,}3272^3} = -0{,}0674$$

Protože hledáme minimum zadané funkce, ponecháme si v novém intervalu bod s nejnižší funkční hodnotou a druhý bod nahradíme.

Nový interval bude tedy vypadat $I_{(2)} = \langle a_{(2)} = a_{(1)}; b_{(2)} = x_{2,(1)} \rangle = \langle 0; 4{,}3272 \rangle$. Toto je konec první iterace.

V druhé iteraci znovu vypočítáme, tentokrát podle nového intervalu, vnitřní body a jejich funkční hodnoty.

$$x_{1,(2)} = b_{(2)} - \frac{F_7}{F_8}\left|b_{(2)} - a_{(2)}\right| = 4{,}3272 - \frac{21}{34} * 4{,}3272 = 1{,}6545$$

$$x_{2,(2)} = a_{(2)} + \frac{F_7}{F_8}\left|b_{(2)} - a_{(2)}\right| = 0 + \frac{21}{34} * 4{,}3272 = 2{,}6727$$

$$f\left(x_{1,(2)}\right) = -\frac{1{,}6545^2 + 2 * 1{,}6545}{1 + 5 * 1{,}6545^3} = -0{,}2557$$

$$f\left(x_{2,(2)}\right) = -\frac{2{,}6727^2 + 2 * 2{,}6727}{1 + 5 * 2{,}6727^3} = -0{,}1294$$

Znovu ohraničíme bod s nižší funkční hodnotou a vytvoříme nový interval $I_{(3)} = \langle a_{(3)} = a_{(2)}; b_{(3)} = x_{2,(2)} \rangle = \langle 0; 2{,}6727 \rangle$. Celý postup budeme opakovat, až do sedmé iterace. Celí postup je znázorněn následující tabulkou.

Tabulka 3 Fibonacciho metoda, příklad

i	$a_{(i)}$	$b_{(i)}$	$x_{1,(i)}$	$x_{2,(i)}$	$f(x_{1,(i)})$	$f(x_{2,(i)})$
1	0	7	2,6727	4,3273	-0,1295	-0,0674
2	0	4,3273	1,6545	2,6727	-0,2557	-0,1295
3	0	2,6727	1,0182	1,6545	-0,4895	-0,2557
4	0	1,6545	0,6364	1,0182	-0,7331	-0,4895
5	0	1,0182	0,3818	0,6364	-0,7114	-0,7331
6	0,3818	1,0182	0,6364	0,7636	-0,7331	-0,6541
7	0,3818	0,7636	0,5091	0,6364	-0,7696	-0,7331

Po sedmé iteraci je tedy konečný interval $\langle 0{,}3818; 0{,}6364\rangle$, jehož střední hodnota je 0,5091. Pomocí analytické metody, jsme v předchozím případě vypočítali minimum v bodě 0,51175.

9.2.2 Příklad 2

Obecný postup zlatého řezu je takřka totožný s Fibonacciho metodou. Rozdíl mezi nimi je ale v tom, že na místo zmenšování původního intervalu pomocí podílu hodnot Fibonacciho posloupnosti, je v metodě zlatého řezu použita konstantní hodnota $\frac{\sqrt{5}-1}{2} \approx 0{,}618$. Výsledky shrnuje následující tabulka.

Tabulka 4 metoda zlatého řezu, příklad

i	$a_{(i)}$	$b_{(i)}$	$x_{1,(i)}$	$x_{2,(i)}$	$f(x_{1,(i)})$	$f(x_{2,(i)})$
1	0	7	2,6738	4,3262	-0,1294	-0,0674
2	0	4,3262	1,6525	2,6738	-0,2562	-0,1294
3	0	2,6738	1,0213	1,6525	-0,4878	-0,2562
4	0	1,6525	0,6312	1,0213	-0,7357	-0,4878
5	0	1,0213	0,3901	0,6312	-0,7190	-0,7357
6	0,3901	1,0213	0,6312	0,7802	-0,7357	-0,6428
7	0,3901	0,7802	0,5391	0,6312	-0,7675	-0,7357

Po sedmé iteraci je konečný interval $\langle 0{,}3901; 0{,}6312\rangle$, jehož střední hodnota je 0,5106.

9.3 Cvičení

9.3.1 Příklad 1

Pro funkci $f(x) = e^x - 2x + e^{-x} - 7x^3$ nalezněte pomocí Fibonacciho metody její maximum na intervalu $I = \langle -4,2 \rangle$. Přesnost je zadaná na $\varepsilon = 0{,}5$.

9.3.2 Příklad 2

Pomocí metody zlatého řezu nalezněte minimum funkce $f(x) = \frac{x^2 - e^x}{2 - 2x^3}$ na intervalu $I = \langle -3,2 \rangle$. Počet iteračních kroků je $i = 8$.

9.3.3 Příklad 3

Libovolnou metodou zjistěte maximum funkce $f(x) = -\frac{x^2 + 2x}{1 + 5x^3}$ na intervalu $I = \langle -5,2 \rangle$.

10 LINEÁRNÍ PROGRAMOVÁNÍ

10.1 Příklady

Amatérský pěstitel konopí David, pěstuje a prodává marihuanu. U sebe doma a v pronajaté garáži má pět různých druhů: Matavation, Rocklock, Shaman, Instakus a Bílou vdovu. Cena, za které jednotlivé odrůdy prodává je: 1620 Kč, 690 Kč, 625 Kč, 410 Kč a 540 Kč, bráno popořadě. Každý den, musí David dávat přesné množství vody jednotlivým rostlinám. Spotřeba vody je: 3,5l, 3l, 3,5l, 1,5l a 0,5l, bráno popořadě. Každý den může použít jenom omezené množství vody a to 250l. U sebe doma pěstuje: Matavation, Rocklock, Instakus a Bílou vdovu, přičemž náklady na elektřinu jsou pro Matavation 5kWh, Rocklock 1kWh, Instakus 0,5kWh a Bílou vdovu 1kWh. V garáži se nachází Rocklock, Shaman a Instakus, náklady jsou 2kWh, 2kWh a 1,5kWh. Aby Davida nezatkla policie, může u sebe doma denně spotřebovat maximálně 210kWh a v garáži 300kWh. Určete optimální strategii pro maximální možný výdělek při zadaných omezeních.

10.2 Postup řešení

Nejprve sestavíme matematický model problému:

$$f(x) = 1620x_1 + 690x_2 + 625x_3 + 410x_4 + 540x_5$$

Účelová funkce $f(x)$ zde vyjadřuje výdělek a x značí počet kusů.

Dále vyjádříme jednotlivá omezení, nejdříve pro vodu a poté pro energii:

$$3{,}5x_1 + 3x_2 + 3{,}5x_3 + 1{,}5x_4 + 0{,}5x_5 \leq 250$$

$$5x_1 + 1x_2 + 0{,}5x_4 + 0{,}5x_5 \leq 210$$

$$2x_2 + 2x_3 + 1{,}5x_4 \leq 300$$

Nyní úlohu převedeme do standardizovaného tvaru a vytvoříme výchozí simplexovou tabulku:

$$3{,}5x_1 + 3x_2 + 3{,}5x_3 + 1{,}5x_4 + 0{,}5x_5 + x_6 = 250$$

$$5x_1 + 1x_2 + 0{,}5x_4 + 0{,}5x_5 + x_7 = 210$$

$$2x_2 + 2x_3 + 1{,}5x_4 + x_8 = 300$$

Tabulka 5 Výchozí simplexová tabulka

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Omezení
3,5	3	3,5	1,5	0,5	1	0	0	250
5	1	0	0,5	1	0	1	0	210
0	2	2	1,5	0	0	0	1	300
-1620	-690	-625	-410	-540	0	0	0	0

Protože v posledním řádku tabulky se nacházejí záporné hodnoty, nenachází se zde optimální řešení. Nalezneme klíčový sloupec.

$$\min(-1620, -690, -625, -410, -540) = -1690$$

Klíčový sloupec je tedy v prvním sloupci. Klíčový řádek vypočítáme z podílů hodnot omezení a čísel v klíčovém sloupci, jako minimum z nezáporných podílů.

$$\min_{\geq 0}\left(\frac{250}{3,5}, \frac{210}{5}\right) = \frac{210}{5} = 42$$

Protože ve třetím řádku se nachází nula, nebudeme tento řádek používat. Klíčový řádek je tedy druhý řádek, z toho vyplívá, že klíčový prvek je číslo 5.

Tabulka 6 Výchozí simplexová tabulka s vyznačením klíčovým řádkem, sloupcem a prvkem

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Omezení
3,5	3	3,5	1,5	0,5	1	0	0	250
5	1	0	0,5	1	0	1	0	210
0	2	2	1,5	0	0	0	1	300
-1620	-690	-625	-410	-540	0	0	0	0

Provedeme eliminaci klíčového prvku. Vynásobíme druhý řádek číslem 0,2, abychom v klíčovém prvku získali číslo 1. K ostatním řádkům poté přičteme takový násobek prvního řádku, aby ostatní čísla v klíčovém sloupci byly 0.

Tabulka 7 Simplexová tabulka po první eliminaci

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Omezení
0	2,3	3,5	1,15	-0,2	1	-0,7	0	103
1	0,2	0	0,1	0,2	0	0,2	0	42
0	2	2	1,5	0	0	0	1	300
0	-366	-625	-248	-216	0	324	0	68040

V posledním řádku tabulky se pořád vyskytují záporná čísla, tudíž řešení není ještě optimální. Proto znovu budeme opakovat celí proces. Nyní je nejmenší hodnota posledního řádku -625, tudíž klíčový sloupec je třetí sloupec. Protože v druhém řádku klíčového sloupce se opět nachází nula, nebudeme tento řádek používat při hledání klíčového řádku a použijeme pouze první a třetí řádek.

$$\min_{\geq 0}\left(\frac{103}{3,5}, \frac{300}{2}\right) = \frac{103}{3,5} = \frac{206}{7}$$

Klíčový řádek je tedy první řádek. Provedeme eliminaci klíčového prvku.

Tabulka 8 Simplexová tabulka po druhé eliminaci

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Omezení
0	0,657	1	0,329	-0,057	0,286	-0,2	0	29,429
1	0,2	0	0,1	0,2	0	0,2	0	42
0	0,686	0	0,843	0,114	-0,571	0,4	1	241,143
0	44,714	0	-42,643	-251,714	178,571	199	0	86432,857

Tabulka 9 Simplexová tabulka obsahující optimální řešení

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Omezení
0,286	0,714	1	0,357	0	0,286	-0,143	0	41,429
5	1	0	0,5	1	0	1	0	210
-0,571	0,571	0	0,786	0	-0,571	0,286	1	217,143
1258,571	296,429	0	83,214	0	178,571	450,714	0	139292,857

Po třetí eliminaci nejsou v posledním řádku už žádné záporné hodnoty, tabulka již obsahuje optimální řešení.

Hodnota v pravém dolním rohu odpovídá hodnotě účelové funkce $f(x_{opt}) = 139292,857$, což znamená, že Davidův denní výdělek bude činit 139 292,857 Kč.

Hodnotu proměnných určíme podle sloupců, ve kterých se nachází jednotkový vektor. V řádku, ve kterém máme hodnotu 1, odpovídá dané hodnotě proměnné. Zbylé proměnné mají nulovou hodnotu.

$$x_3 = 41,429, x_5 = 210, x_8 = 214,143$$

To nám značí, že se vyplatí pěstovat Shamana v množství 41,4 rostlin a Bílou vdovu v množství 210 rostlin. Proměnná x_8 nám ukazuje nedočerpané prostředky, konkrétně nám z proměnné vyplívá, že v garáži je energetická rezerva 214,143kWh.

10.3 Cvičení

10.3.1 Cvičení 1

Budeme vycházet z předchozí úlohy. Po zvýšený sankcí za držení a pěstování konopí, se zvýšila jejich poptávka. Cena za Bílou vdovu je nyní 720kč, za Rocklocka 890 kč a Shamana stojí 920kč. Naopak Matavation klesl na 1230 Kč.

10.3.2 Cvičení 2

Franta tiskne u sebe doma na zakázku 3D modely na své 3D tiskárně. Modely klasifikuje do tří kategorií na malé, střední a velké. Malé se tisknou přibližně 24h, střední se tisknou přibližně 48h a velké se tisknou přibližně 72h, přičemž Franta posílá poštou hotové modely jednou týdně. Cena jednotlivých modelů je popořadě 500 Kč, 800 Kč a 1000 Kč. Pro tvorbu modelů používá 1,75mm filament z ST-PLA, na jednotlivé modely přibližně spotřebuje 30g/kus, 70g/kus a 100g/kus. Přičemž jeho zásoba filamentu na celý týden je 800g. Tiskárna při tvorbě modelů má nárok na energii 2, 6, 12 kWh/kus. Franta má ve smlouvě uvedeno, že jeho týdenní spotřeba energie nesmí přesáhnout 200kWh. Jaký bude jeho ideální výrobní program?

11 TABULKOVÁ FORMA DYNAMICKÉHO PROGRAMOVÁNÍ

11.1 Příklady

Prodejce obuvi Al Bunda, po každé své výplatě odevzdává část peněz své ženě a dětem jako kapesné. Peníze nerozděluje přesně, ale podle nátlaku rodiny. Manželka Peggy, která ví, kolik Al vydělává, požaduje peníze ve vztahu $4x^2$, kde x značí peníze. Nejstarší dcera Kelly požaduje ve vztahu $2x^2$, jinak řekne Pegg, kam si Al schovává zbylé peníze. Syn Bud na otce žádnou páku nemá, a proto dostává peníze ve vztahu x^2. Al ve snaze zachránit nějaké peníze vypozoroval, že pokud přinese aspoň čtyři dolarové bankovky rodina je spokojená. Jaký minimální počet peněz musí Al odevzdat svojí milující rodině?

počet kroků je $h = 1$.

11.2 Postup řešení

Nejprve ze zadaných parametrů vytvoříme matematický model zadání:

$$\min f(x_1, x_2, x_3) = 4x_1^2 + 2x_2^2 + x_3^2$$

Omezení a počet kroků je zadáno následovně:

$$x_1 + x_2 + x_3 = 4$$

$$h = 1$$

Ze zadaného omezení vyplívá, že $b = 4$. Jelikož je velikost kroku 1, bude hodnota b_k od 0 po 4. Do prvního sloupce $x_1(b)$ zapíšeme hodnoty b_k, druhý sloupec F_1 tvoří část rovnice funkce, v tomto případě bude hodnota $F_1 = 4x_1^2$.

Tabulka 10 Tabulka dynamického programování I

b_k	x_1	F_1
0	0	0
1	1	4
2	2	16
3	3	36
4	4	64

Nyní vypočítáme další dvojsloupec. Hodnotu x_2 získáme rozkladem b_k, hodnota $F_2 = 4x_1^2 + 2x_2^2$, v tuto chvíli do tabulky zapíšeme hodnoty, které jsou minimální.

Tabulka 11 Výpočet hodnot I

1	1	0	4
	0	1	2
2	2	0	16
	1	1	6
	0	2	8
3	3	0	36
	2	1	18
	1	2	12
	0	3	18
4	4	0	64
	3	1	38
	2	2	24
	1	3	22
	0	4	32

Tabulka 12 Tabulka dynamického programování II

b_k	x_1	F_1	x_2	F_2
0	0	0	0	0
1	1	4	0	2
2	2	16	1	6
3	3	36	2	12
4	4	64	1	22

Hodnotu posledního dvojsloupce získáme stejnou metodou jako v předchozím případě s tím rozdílem, že budeme využívat hodnoty z předchozího výsledku. Pro výpočet F_3 použijeme předchozí minimální hodnoty a bude mít hodnotu $F_3 = F_2 + x_3^2$.

Tabulka 13 Výpočet hodnot II

1	2	0	2
	0	1	1
2	6	0	6
	2	1	3
	0	2	4
3	12	0	12
	6	1	7
	2	2	6
	0	3	9
4	22	0	22
	12	1	13
	6	2	10
	2	3	11
	0	4	16

Tabulka 14 Výpočet hodnot III

b_k	x_1	F_1	x_2	F_2	x_3	F_3
0	0	0	0	0	0	0
1	1	4	0	2	1	1
2	2	16	1	6	1	3
3	3	36	2	12	2	6
4	4	64	1	22	2	10

Z poslední hodnoty tabulky odečteme minimální hodnotu funkce pro splnění podmínky $f = 10$.

$$x_3 = 2, x_1 + x_2 = 2 \Rightarrow x_1, x_2 = 1$$

$$f(x_1, x_2, x_3) = 4x_1^2 + 2x_2^2 + x_3^2 = 10$$

Pegg a Kelly dostanou od Ala každá po dvou bankovkách a Bud dostane jednu bankovku. Celkově přijde Al 10$ z každé výplaty.

11.3 Cvičení

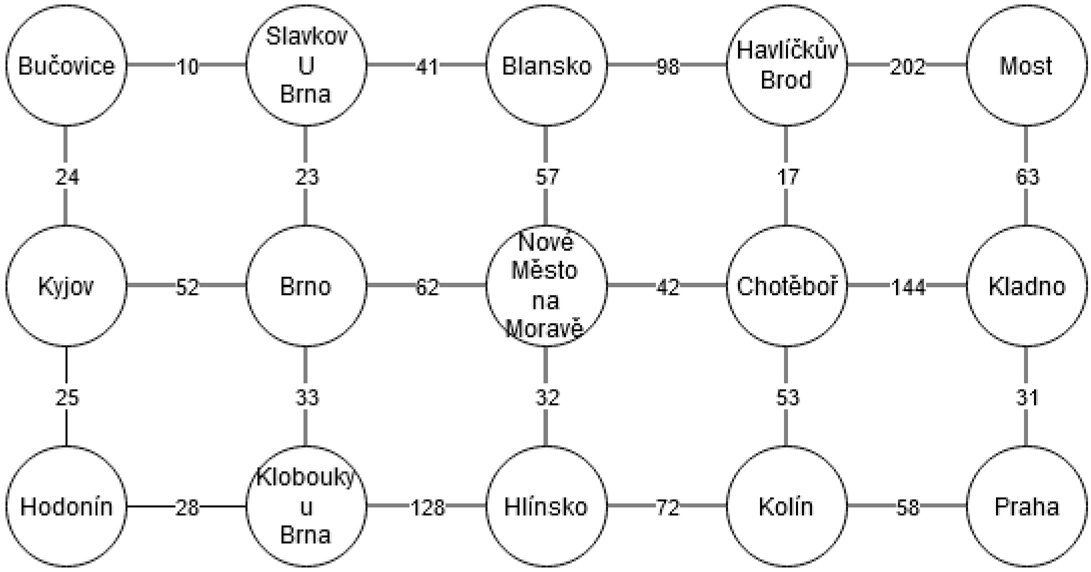
Tomáš je studentem chemické školy. Po školním experimentu, ke kterému škola dodala potřebné prostředky, Tomáš některé chemikálie neodevzdal. Konkrétně si nechal sodík, mangan a baryum. Tomáš se v rámci ekologické likvidace přebytků, rozhodl vyrobit ohňostroj. Protože Tomáš má chemikálie v kapalné formě, musí je míchat přímo se střelným prachem. Podle tabulek zjistil, že aby směs správně hořela, musí se chemikálie míchat ve správném poměru. Sodík se míchá ve vztahu $0{,}5x^2$ $[g]$, kde x značí množství střelného prachu, mangan ve vztahu $1{,}25x^2$ a baryum ve vztahu $2{,}5x^2$. Pro svůj první experiment se Tomáš rozhodl, že použije 10g chemikálii a bude je dávkovat po 2g. Kolik gramů jednotlivých chemikálii musí minimálně použít, aby vytvořil ohňostroj, a kolik střelného prachu při tom spotřebuje?

12 SÍŤOVÁ FORMA DYNAMICKÉHO PROGRAMOVÁNÍ

12.1 Příklady

12.1.1 Příklad 1

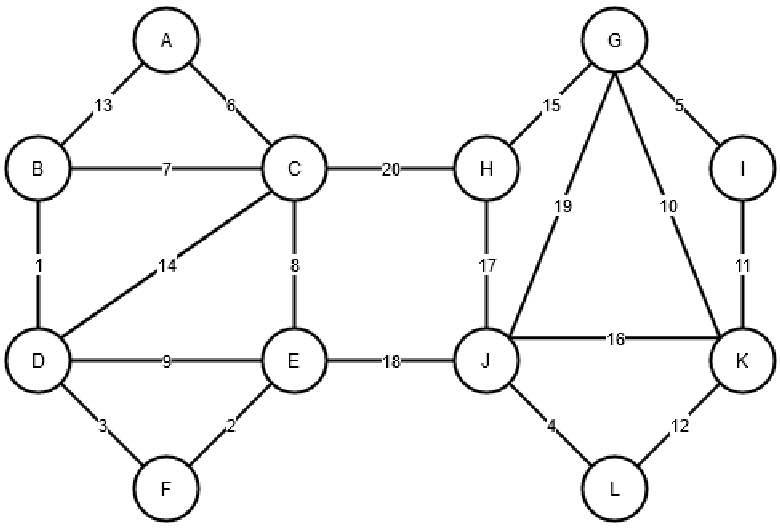
Milan jede z Hodonína autem do Mostu, jelikož nemá dálniční známku, musí jet po okrscích. Kolik kilometrů musí ujet a přes kolik měst projede?



Obrázek 1 zadání úlohy síťové formy

12.1.2 Příklad 2

Pomocí Borůvkova algoritmu vytvořte kostru spojující všechny vrcholy.



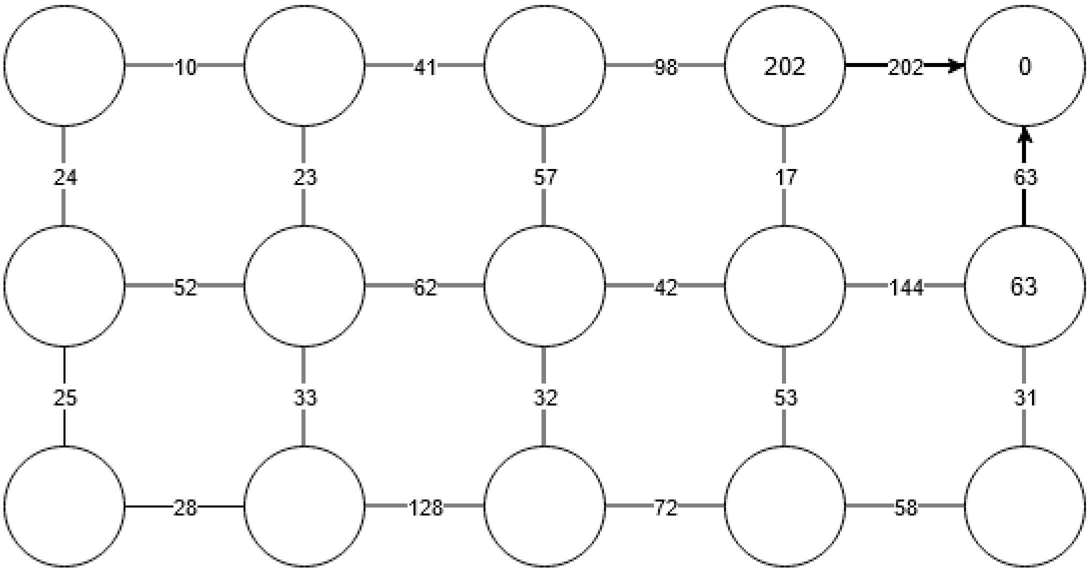
Obrázek 2 zadání úlohy Borůvkův algoritmus

12.2 Postup řešení

12.2.1 Příklad 1

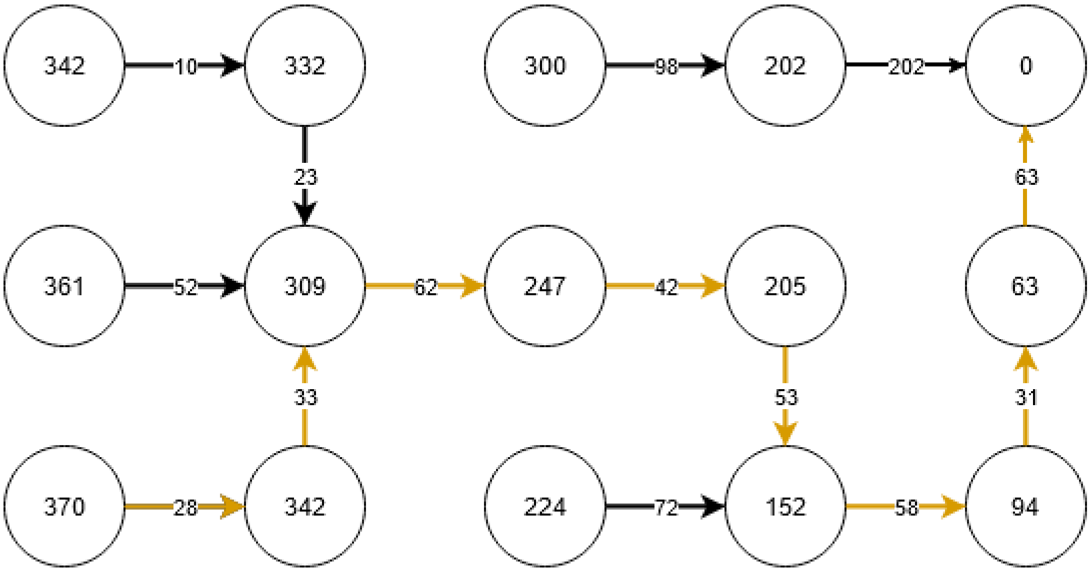
Označíme výchozí vrchol za aktivní a nastavíme mu hodnotu nula, ostatní za nekonečno (v tomto případě jsou prázdné).

Vypočítáme hodnoty sousedních vrcholů aktivního vrcholu od počátečního vrcholu, pokud je nově vypočítaná vzdálenost menší, zapíšeme ji do vrcholu, pokud ne, hodnota vrcholu zůstává nezměněná. Poté označíme aktivní vrchol za navštívený, k tomuto bodu se už nevracíme. Nyní označíme nenavštívený vrchol s nejmenší hodnotou jako aktivní. V tomto případě je to vrchol s hodnotou 63.



Obrázek 3 síťová úloha po prvním projití

Tento postup opakujeme do té doby, dokud není navštíven i výsledný vrchol.

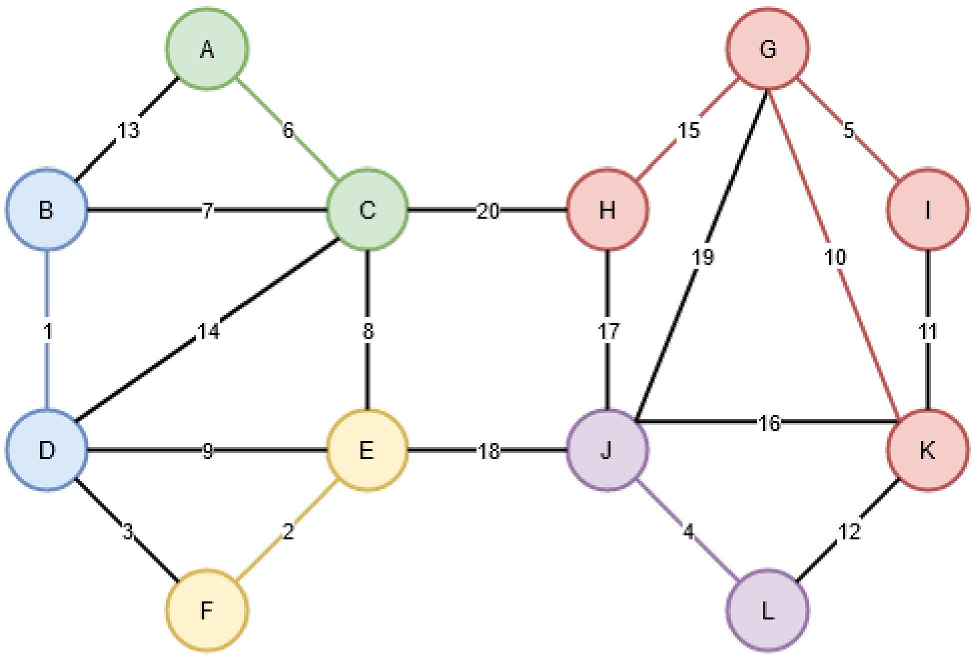


Obrázek 4 finální podoba síťové úlohy

Milan při cestě z Hodonína do Mostu projede přes sedm měst a ujede 370Km.

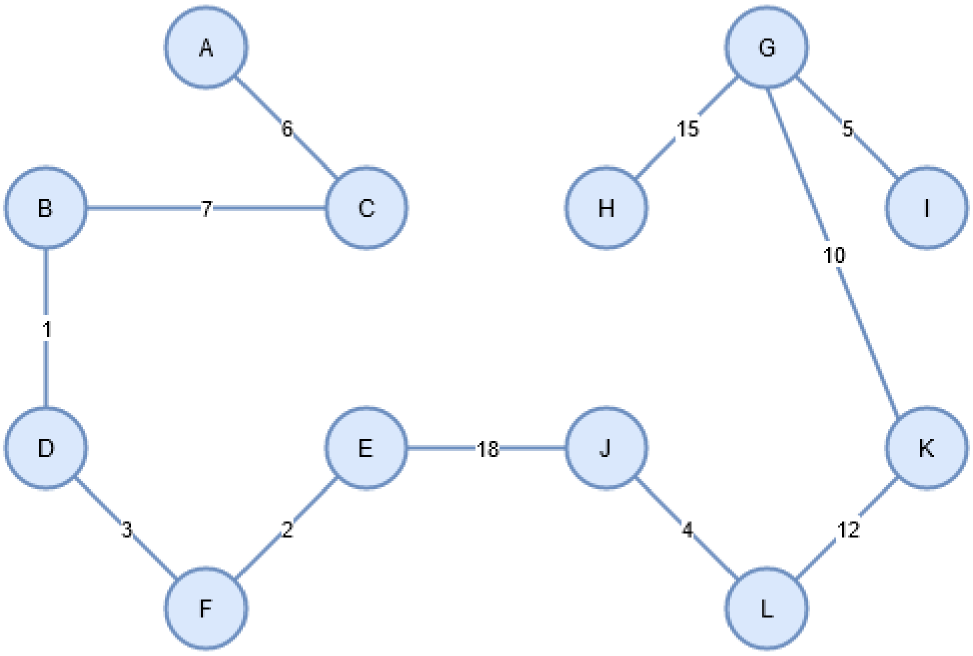
12.2.2 Příklad 2

V první části algoritmu, projdeme postupně všechny vrcholy a vyznačíme jejich hrany s nejmenší hodnotou. Tímto způsobem byly vytvořeny skupinky komponentů.



Obrázek 5 Borůvkův algoritmus po první fázi

V druhé části půjdeme postupně po jednotlivých komponentech a znovu vyznačíme hrany s nejmenší hodnotou. Tento proces opakujeme do té doby, dokud nejsou všechny vrcholy propojené.



Obrázek 6 Finální podoba Borůvkova algoritmu

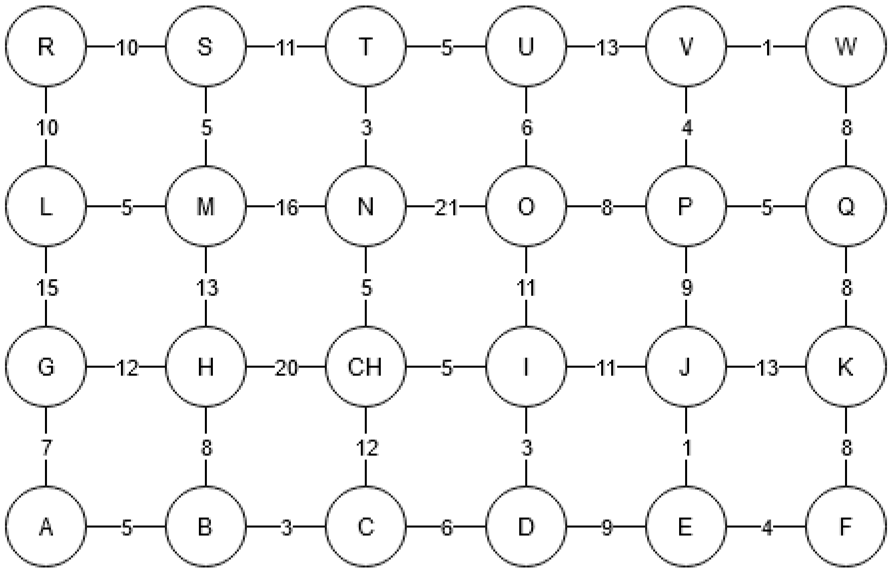
12.3 Cvičení

12.3.1 Příklad 1

Budeme vycházet z předchozí síťové úlohy. Milanův kamarád David, potřebuje jet ze Slavkova u Brna do Hlinska. Milan při cestě z Mostu zpátky do Hodonína se tedy zastaví pro Davida a odveze ho do Hlinska. Poté bude pokračovat zpátky do Hodonína, kolik kilometrů navíc projede?

12.3.2 Příklad 2

Najděte nejkratší cestu mezi počátečním bodem A a konečným bodem W.



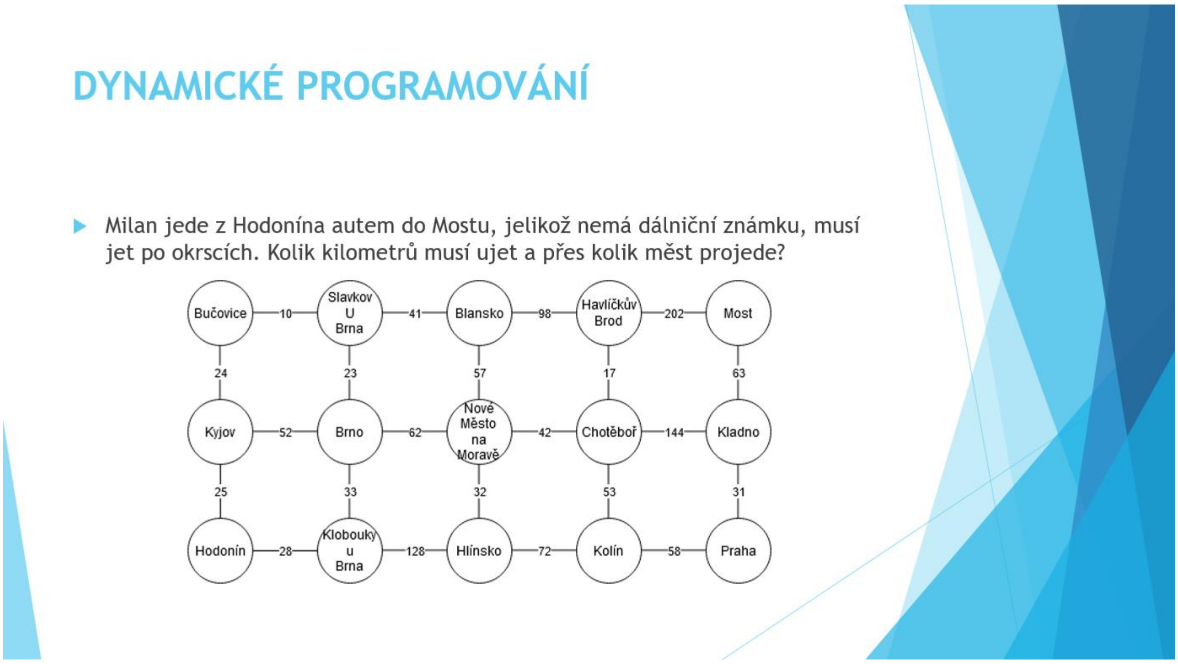
Obrázek 7 zadání příklad 2

13 PREZENTACE VÝSLEDKŮ

Součástí práce je prezentace, která shrnuje praktickou část. Prezentace je rozdělena do pěti kategorií:

1. Jednorozměrný volný extrém
2. Komparativní iterační metody
3. Lineární programování
4. Tabulková forma dynamického programování
5. Síťová forma dynamického programování.

Stejně jako v praktické části, i prezentace obsahuje zadání příkladů a jejich následné řešení. Zároveň jsou zde uvedené neřešené cvičení, které slouží k procvičení dané optimalizační úlohy. Prezentace neobsahuje teoretickou část práce, zaměřuje se pouze na výslednou praktickou část a její prezentaci.



Obrázek 8 Ukázka prezentace

ZÁVĚR

Touto prací jsem chtěl přispět k výuce předmětu Optimalizace a pomoct studentům pochopit zadaná témata a způsoby řešení zadaných úloh. V první části práce jsem se zabýval různými typy optimalizačních metod a způsoby jejich řešení. Snažil jsem se, ale hlavně dané metody vysvětlit, co možná nejvíce srozumitelně a pochopitelně, aby čtenář daným metodám porozuměl a pochopil je. Druhá část práce se věnovala praktickému řešení daných úloh. Zde byla snaha ukázat různé využití optimalizačních metod a jejich širokou aplikovatelnost. Druhá část práce navíc obsahuje neřešené příklady pro procvičení zadaných úkolů. Součástí práce je i prezentace, která shrnuje výsledky z praktické části a v mírně upravené formě ji prezentuje. Prezentace obsahuje všechny příklady a cvičení z praktické části

SEZNAM POUŽITÉ LITERATURY

[1] PEKAŘ, Libor. Optimalizace [online]. Elektronický interaktivní učební text FAI UTB ve Zlíně. Zlín, 2014 [cit. 2019-11-15], 181 s. Dostupné z: https://moodle.utb.cz/mod/resource/view.php?id=205912

[2] MAŇAS, Miroslav. Optimalizační metody. 1. vyd. Praha: SNTL, 1979, 257 s.

[3] TEMLÍK, Petr. Optimalizace [online]. Zlín, 2005 [cit. 2019-11-15]. Dostupné z: https://stag.utb.cz. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Roman Prokop.

[4] VENKATARAMAN, P. Applied Optimization with Matlab Programming. New York: John Wiley & Sons, 2002, 398 s. ISBN 0471349585

[5] VÍTEČKOVÁ, Miluše, JEDLIČKA, David. Statická optimalizace systémů [online]. VŠB-TU v Ostravě, 2003 [cit. 2019-11-15]. Dostupné z: http://books.fs.vsb.cz/StatickaOptimalizace/index.htm.

[6] HÁJEK, Michal. Soubor úloh ke cvičení do předmětu Optimalizace - lineární a dynamické programování. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009, 50 s. Dostupné také z: http://hdl.handle.net/10563/9772. Tomas Bata University in Zlín. Faculty of Applied Informatics, Ústav aplikované informatiky. Vedoucí práce Pekař, Libor.

[7] KOČICA, Martin. Iterační metody optimalizace. Zlín: Univerzita Tomáše Bati ve Zlíně, 2006, 68 s. Dostupné také z: http://hdl.handle.net/10563/921. Tomas Bata University in Zlín. Faculty of Applied Informatics, Ústav aplikované informatiky. Vedoucí práce Prokop, Roman.

[8] FRANCÍREK, Pavel. *Lineární modely: Lineární optimalizace* [online]. Brno, 2018 [cit. 2020-07-29]. Dostupné z: http://www.math.muni.cz/~xfrancirekp/vyuka/dvanacte_cviceni/dvanacte_cviceni.pdf. Výukoví materiál. Masarikova univerzita.

[9] MATOUŠEK, Jiří. Lineární programování: Úvod pro informatiky [online]. Praha, 2006 [cit. 2020-07-29]. Dostupné z: https://iti.mff.cuni.cz/series/2006/311.pdf. Výukoví materiál. KAM MFF UK.

[10] DOSTÁL, Zdeněk a Petr BEREMLIJSKI. Metody optimalizace - interaktivní verze [online]. Ostrava, 2012 [cit. 2020-07-29]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/metody_optimalizace_obr.pdf. Učební text. Technická univerzita Ostrava.

[11] HAMHALTER, Jan a Jaroslav TIŠER. Diferenciální počet funkcí více proměnných. [online]. Praha, 2005 [cit. 2020-07-29]. Dostupné z: https://math.feld.cvut.cz/tiser/web7.pdf. Školní skripta. České Vysoké Učení Technické Fakulta elektrotechnická.

[12] Borůvkův algoritmus. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020, 23.6.2020 [cit. 2020-07-23]. Dostupné z: https://cs.wikipedia.org/wiki/Bor%C5%AFvk%C5%AFv_algoritmus

[13] *Algoritmy.net: Borůvkův algoritmus* [online]. Praha: algoritmy.net, 2019 [cit. 2020-08-06]. Dostupné z: https://www.algoritmy.net/article/1396/Boruvkuv-algoritmus#

[14] HOLČÍK, Jiří a Martin KOMENDA. Dijkstrův algoritmus. *Matematická biologie: e-learningová učebnice* [online]. Brno: Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity, 2015 [cit. 2020-08-07]. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=zaklady-informatiky-pro-biology--teoreticke-zaklady-informatiky--teorie-grafu--optimalizacni-ulohy-nad-grafy--dijkstruv-algoritmus

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

f, g	značení funkce
i, j	index pozice prvku
R	označení množin reálných čísel
$\nabla f(x)$	gradient funkce
$H(f(x_0))$	Hessova matice
ε	požadovaná přesnost popřípadě velikost skoku
m, n	rozměry matice

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: PREZENTACE

Prezentace je rozdělena do pěti kategorií, Jednorozměrný volný extrém, Komparativní iterační metody, Lineární programování, Tabulková forma dynamického programování a Síťová forma dynamického programování. Každá kapitola obsahuje příklady, řešení příkladu a jeho postup a nakonec neřešené cvičení.