

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103003/I/2014/1407092347

**METÓDY PRÍPUSTNÝCH SMEROV PRE
RIEŠENIE ÚLOH NELINEÁRNEHO
PROGRAMOVANIA**

Diplomová práca

2014

Bc. Marek Ottinger

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

**METÓDY PRÍPUSTNÝCH SMEROV PRE
RIEŠENIE ÚLOH NELINEÁRNEHO
PROGRAMOVANIA**

Diplomová práca

Študijný program: Operačný výskum a ekonometria

Študijný odbor: 6258 Kvantitatívne metódy v ekonómii

Školiace pracovisko: Katedra operačného výskumu a ekonometrie

Vedúci záverečnej práce: prof. Ing. Michal Fendek, PhD.

Bratislava 2014

Bc. Marek Ottinger

Čestné vyhlásenie

**Čestne vyhlasujem, že záverečnú prácu som vypracoval
samostatne a že som uviedol všetku použitú literatúru.**

Dátum: 22.04.2014

.....

(podpis študenta)

Pod'akovanie

Chcel by som sa poďakovať prof. Ing. Michalovi Fendekovi, PhD. za odbornú pomoc, cenné rady a pripomienky, ktoré mi poskytol pri písaní diplomovej práce, ako aj všetkým ostatným, ktorí mi akokoľvek pomohli.

ABSTRAKT

OTTINGER, Marek: *Metódy prípustných smerov pre riešenie úloh nelineárneho programovania*. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra operačného výskumu a ekonometrie. – Vedúci záverečnej práce: prof. Ing. Michal Fendek, PhD.. – Bratislava: FHI EU, 2014, 50s.

Cieľom práce je popis niekoľkých optimalizačných metód, určených na riešenie nelineárnych úloh. Uvedené sú príklady uplatnenia takýchto metód v praxi. V metodologickej časti práce sú popísané teoretické východiská týchto metód a v praktickej časti je ukázané konkrétne riešenie vybranej metódy. Výsledkom práce je procedúra, umožňujúca pozorovanie priebehu výpočtového procesu vybranej metódy v jednotlivých iteráciách a analýzu vplyvu zmeny vstupných parametrov na výpočet. V práci je vysvetlená problematika hľadania optimálneho portfólia s rizikovými cennými papiermi, pre ktorú je uvedený modelový príklad riešený metódou redukovaných gradientov s využitím vlastnej procedúry v prostredí MS Excel.

Kľúčové slová:

metóda redukovaných gradientov, výber portfólia s rizikovými cennými papiermi, VBA, MS Excel, nelineárne programovanie, metódy prípustných smerov

ABSTRACT

OTTINGER, Marek: *Feasible direction methods for solving of nonlinear programming problem*. – University of Economics in Bratislava. Faculty of Economics and Informatics; Department of Operations Research and Econometrics. – Thesis supervisor: prof. Ing. Michal Fendek, PhD.. – Bratislava: FHI EU, 2014, 50s.

Aim of this theses is to describe some of the optimalization methods, which are intended for solving nonlinear problems. There are shown few examples for application of such methods in real life. In methodological part of work are described theoretical resources of these methods and in practical part is shown certain solution of a chosen method. Result of this work is a procedure, which let us watch the computational process of the chosen method during each iterative and to analyse impact of input changes on results. Issues of portfolio selection problem with risky securities are described in this theses, for which is then constructed a model example. The example is then solved according to reduced gradient method using author's own procedure in MS Excel environment.

Keywords:

reduced gradient method, portfolio selection with risky securities, VBA, MS Excel, nonlinear programming, feasible direction methods

Obsah

Úvod.....	1
1. Súčasný stav riešenej problematiky	3
1.1 Modely optimálneho plánovania	3
1.2 Problém produktového mixu s cenovou elasticitou	4
1.3 Úlohy hospodárskeho plánovania	4
1.4 Prepravný problém s úsporami z objemu	5
1.5 Výber portfólia s rizikovými cennými papiermi.....	6
2. Cieľ práce.....	10
3. Metódy riešenia úloh nelineárneho programovania.....	11
3.1 Metóda prípustných smerov.....	11
3.2 Projekčné metódy pre nelineárne ohraničenia	16
3.3 Metódy aproximačného programovania	22
3.4 Algoritmus redukovaných gradientov	26
4. Tvorba procedúry pre algoritmus redukovaných gradientov	33
4.1 Návrh vstupov a výstupov pre procedúru	33
4.2 Popis tvorby zdrojového kódu	36
5. Riešenie hypotetického príkladu.....	41
5.1 Formulácia úlohy	41
5.2 Riešenie úlohy pomocou GAMS	42
5.3 Riešenie úlohy pomocou procedúry redgrad	44
Záver	47
Zoznam použitej literatúry	49
Prílohy.....	50

Úvod

V súčasnosti môžeme pozorovať výrazný vývojový posun v oblasti informačných technológií, ktorých spôsoby aplikácie sa rozširujú do mnohých oblastí hospodárstva. Náročné matematické algoritmy sa používajú pri modelovaní rôznych situácií zo skutočného sveta. Či už v oblasti priemyslu, logistiky, riadenia hospodárstva alebo investičného rozhodovania, uplatňovanie optimalizačných algoritmov vedie k zvýšeniu efektivity rozhodovania.

Okrem toho, že si technologický pokrok vyžaduje hľadanie spôsobov a metód pre riešenie širokého spektra problémov, poskytuje aj množstvo nástrojov pre ich riešenie. Výkony dnešných počítačov umožňujú dosiahnuť výpočty takých úloh, o ktorých sa ešte v nedávnej minulosti uvažovalo iba v teoretickej rovine. Ambíciou tejto práce je poukázať na možnosti riešenia optimalizačných algoritmov použitím všeobecne dostupných nástrojov, medzi ktoré patrí aj MS Excel.

Pri modelovaní problémov z praxe sa často vyskytujú rôzne typy optimalizačných problémov. Môžu byť lineárne alebo nelineárne, s ohraničeniami alebo bez ohraničení, celočíselné alebo neceločíselné. Funkcie v optimalizačnom probléme môžu, ale nemusia byť diferencovateľné, môžu, ale nemusia byť konvexné. Niekedy uvažujeme optimalizáciu s neistotou, tiež známu ako stochastickú optimalizáciu, kde sú funkcie dané na základe určitej pravdepodobnosti.

Táto práca je zameraná na popis metód určených pre riešenie nelineárnych optimalizačných úloh. V prvej časti práce sú predstavené niektoré príklady praktických problémov, ktoré môžu byť modelované ako úlohy nelineárneho programovania. Dôraz je kladený predovšetkým na rozdiely pri tvorbe matematických modelov pre tieto problémy. Okrem iných, je tu detailne popísaný aj problém hľadania optimálneho portfólia s rizikovými cennými papiermi, ktorý v práci ďalej skúmame. V ďalšej kapitole sú potom priblížené matematické nástroje, ktoré nám umožňujú tieto problémy riešiť. Popísané sú dve skupiny metód, metódy v ktorých sa pristupuje k riešeniu nelineárnych ohraničení priamo a metódy, pri ktorých hľadáme riešenie s využitím aproximačných algoritmov. Získané poznatky hodnotíme a z uvedených optimalizačných algoritmov vyberáme metódu redukovaných gradientov, ktorá je vhodná pre hľadanie optimálneho portfólia. Ďalej

popisujeme tvorbu výpočtového algoritmu pre túto metódu, ktorý sme navrhli prostredníctvom jazyka VBA pre MS Excel. V práci ďalej stanovíme hypotetický problém hľadania optimálneho portfólia s rizikovými cennými papiermi. Stanovenú úlohu riešime pomocou nástroja GAMS, kde dosiahneme optimálne riešenie. Následne popisujeme zadanie hypotetickej úlohy do nami navrhnutého algoritmu. Na základe výstupov vytvorenej procedúry *redgrad* hodnotíme úspešnosť aplikácie metódy redukovaných gradientov na uvedenom príklade. Práca svojim obsahom smeruje k vysvetleniu metódy hľadania optimálneho portfólia, pri čom sa zameriava na popis výpočtového procesu založeného na metóde redukovaných gradientov.

1. Súčasný stav riešenej problematiky

Problematika nelineárneho programovania (NLP) má veľmi široký rozsah uplatnenia a v posledných desaťročiach zaznamenala významný pokrok. Vznikajú stále nové modely v rôznych oblastiach života a existuje široká škála riešení pre vzniknuté optimalizačné úlohy. Problémy nelineárnej optimalizácie sa objavujú napríklad v oblasti dopravy, telekomunikácie, chemického priemyslu, hospodárstva, marketingu, výživy, ale aj fyziky či mechaniky. V tejto časti bližšie uvedieme niektoré príklady a priblížime spôsob riešenia pomocou metód nelineárneho programovania.

Existuje niekoľko skupín problémov, na ktoré malo nelineárne programovanie významný dopad. Lasdon a Waren [5] popisujú tri oblasti aplikácie nelineárnych modelov, ropný a chemický priemysel, nelineárne sieťové úlohy, modely ekonomického plánovania. Prvá kategória zahŕňa problémy plánovania produkcie a distribúcie, miešanie produktov, optimalizáciu chodu rôznych výrobných jednotiek a problémy obnovy ropných a plynových zásobníkov. Problematika nelineárnych sietí sa sústreďuje na optimalizáciu úloh dodávky elektrickej energie, vody a plynu v modeloch inžinierskych sietí a vyvážených modeloch mestských dopravných tokov. Tretia skupina poskytuje náhľad na viacero aplikácií v oblasti ekonomického plánovania, príklady zahrňujúce optimálne riadenie a dynamické ekonometrické modely, modely hospodárskeho plánovania pre jedno obdobie, formulácie úloh použitých v energetike a poľnohospodárstve, či modely systému plánovania vyššieho vzdelávania. V tejto kapitole poukážeme na niektoré v súčasnosti riešené problémy a načrtujeme postupy ich riešenia.

1.1 Modely optimálneho plánovania

NLP modely boli tiež použité pri optimalizácii plánovania v priemysle. Uvažujme problém kontroly teploty tekutiny v nepretržitom procese spracovania. Často je potrebné, aby bol prúd vychádzajúci z jednej výrobnej jednotky ochladený alebo zahriaty na určitú teplotu, podľa požiadaviek nadväzujúcej jednotky. Pece, chladiče, výmenníky tepla sú používané na zmenu teploty toku. V jednom výrobnom zariadení sa nachádza mnoho takýchto tokov prúdiacich cez komplexnú štruktúru výhrevných a chladiacich zariadení. Vzťahy medzi teplotami prúdov a zariadení sú nelineárne, ako aj funkcia nákladov na prevádzku výhrevných a chladiacich zariadení. Bryant (1979) so spoločnosti Standard Oil popísal optimalizačný model plánovania, ktorý udáva parametre pre výmenníky tepla, pece

a chladiče pre zostavenie daného procesu a tepelnej konfigurácie. Systém hodnotí existujúci systém, ako aj budúce možné zostavené jednotky. Z dôvodu komplexnosti celého systému bol výpočtový čas modelu príliš dlhý. Autori preto vyvinuli aproximačnú funkciu, ktorá je nenáročná na výpočet. Keď je nájdené jej optimum, hodnota pôvodnej účelovej funkcie je vypočítaná vychádzajúc z optimálneho riešenia aproximovanej úlohy.

1.2 Problém produktového mixu s cenovou elasticitou

Pri zostavovaní produktového mixu je cieľom nájsť optimálnu štruktúru produkcie pre produkty podniku, vzhľadom na obmedzenia zdrojov potrebných pre výrobu produktov, pri maximalizácii celkového zisku podniku. V niektorých prípadoch je daná fixná hodnota zisku vzhľadom na produkt, výsledná účelová funkcia je teda lineárna. V mnohých úlohách zaoberajúcich sa produktovým mixom však vystupujú určité faktory, vnášajúce do účelovej funkcie nelineárne závislosti. Veľký výrobca môže napríklad naraziť na problém cenovej elasticity, kedy sa možnosti predaja výrobkov znižujú pri raste cien a naopak. Cenovo-dopytová krivka je v takom prípade zakrivená, vzťah medzi cenou tovaru $p(x)$ a predaným množstvom x je nelineárny. Firma potom dosahuje zisk z predaja x jednotiek na úrovni $xp(x)$ mínus náklady na produkciu a distribúciu. Ak sú teda náklady na produkciu a distribúciu výrobkov fixné, vyjadrené ako c , zisk firmy z výroby x výrobkov je daný nelineárnou funkciou $P(x) = xp(x) - cx$. Účelovú funkciu potom vyjadríme ako súčet takýchto nelineárnych funkcií pre všetky produkty.

Ďalším dôvodom pre vznik nelinearity v účelovej funkcii býva tiež skutočnosť, že hraničné náklady na produkciu ďalšej jednotky závisia od úrovne produkcie. Napríklad, keď hraničné náklady pri zvýšení produkcie klesnú v dôsledku efektívnejšej produkcie s bohatšími skúsenosťami. Podobný jav môže byť spôsobený aj nákupom drahších zariadení pre výrobu väčšieho objemu produktov, ktoré umožňujú výrobu s nižšími nákladmi.

1.3 Úlohy hospodárskeho plánovania

Modely nelineárneho programovania je možné aplikovať aj v oblasti hospodárskeho plánovania. Medzi ne patria aj modely zaoberajúce sa úlohami plánovania v rozvojových krajinách. Príkladom je trieda modelov "Yulgok", ktorá bola popísaná Inmanom (1978). Pre každý ekonomický sektor sú definované premenné ako produkcia, fixné investície, export, import, hromadenie zásob. Ohraničenia sú tvorené štyrmi základnými vzťahmi. Produkcia musí pokryť alebo presiahnuť dopyt, pri čom nesmie

prekročiť kapacity sektora, investície sú limitované dostupnými fondmi a rovnováha platieb je obmedzená. Úlohou je maximalizovať spotrebu domácností výstupov so všetkých sektorov. Všetky premenné reprezentujú hodnoty rôznych ekonomických agregátov v konečnom roku osemročného plánovacieho horizontu (1974-1981), za predpokladu, že fixné investície, produkcia a hrubý domáci produkt budú rásť konštantnou mierou. Neznáme miery rastu majú byť volené optimalizáciou, čo vedie k exponenciálnym nelinearitám v mnohých podmnožinách ohraničení.

Pri s sektoroch hospodárstva má Yulgokský model $4s+5$ ohraničení, z toho $2s+1$ nelineárnych a $6s+4$ premenných, z toho $2s+2$ nelineárnych. Verzia modelu obsahujúca 53 sektorov bola použitá Korejským ministerstvom energetiky a národných zdrojov na vyšetrenie záležitostí ohľadom medzisektorovej alokácie kumulatívnych investícií za obdobie 1974-1981 a medzsektorového modelu zahraničného obchodu v roku 1981. V týchto aplikáciách bola použitá schéma rekurzívneho lineárneho programovania.

Ďalšou významnou triedou modelov plánovania sú PROLOG modely, popísané Nortonom (1978) a Inmanom (1977). Ide o cenovo-endogénne, statické, čiastkové, rovnovážne modely. Verzia s jedným spotrebiteľom, bez investícií a bez verejného sektora v ktorej je dopyt po komoditách funkciou cien všetkých komodít a reálneho príjmu spotrebiteľov. Zohľadnený je aj vzťah medzi dostupnými produkčnými zdrojmi a cenou komodity. Cieľom takéhoto modelu je maximalizácia celkového objemu spotrebovaných surovín mínus hodnota použitých zdrojov, ktorá je tiež nelineárna.

Významnou časť úloh v rámci metód nelineárneho programovania môže byť formulovaná prostredníctvom kvadratických úloh, ktoré si vyžadujú optimalizáciu kvadratickej funkcie vzhľadom na lineárne ohraničenia. Princípy podporujúce tieto modely pochádzajú pôvodne od Samuelsona (1952) a boli uplatňované Takayamom (1971) a Manneom (1979). Aplikácie modelovali produkciu a distribúciu uhlia, všeobecné energetické modely a modely poľnohospodárskeho plánovania.

1.4 Prepravný problém s úsporami z objemu

Typickou aplikáciou pre dopravný problém je určenie optimálneho plánu pre dopravu tovarov z rôznych zdrojov do rôznych cieľov, s danými ohraničeniami dopytu a ponuky, vzhľadom na minimalizáciu prepravných nákladov. Pri lineárnych úlohách sa predpokladá, že sú náklady na prepravu fixné, bez ohľadu na prepravované množstvo. V skutočnosti však tieto náklady nemusia byť fixné. Pri veľkých dodávkach sú niekedy

možné úspory z objemu, pri ktorých môžu hraničné náklady spojené s pridaním dodatočnej prepravovanej jednotky klesať. Výsledné dopravné náklady pre x prepravovaných jednotiek sú potom dané nelineárnou funkciou $C(x)$, ktorá je po častiach lineárna funkcia so sklonom zodpovedajúcim hraničným nákladom. Každá kombinácia zdroja a cieľa zodpovedá rovnakej funkcii prepravných nákladov. Náklady na prepravu x_{ij} jednotiek zo zdroja i ($i = 1, 2, \dots, m$) do cieľa j ($j = 1, 2, \dots, n$) sú potom dané nelineárnou funkciou $C_{ij}(x_{ij})$. Účelová funkcia pre riešenie problému má potom tvar

$$f(x) = \sum_{i=1}^m \sum_{j=1}^n C_{ij}(x_{ij}). \quad (1.4.1)$$

Hoci je účelová funkcia nelineárna, ohraničenia sú obyčajne stále lineárne, v rovnakom tvare, ako pri lineárnych úlohách.

1.5 Výber portfólia s rizikovými cennými papiermi

Investor s daným dôchodkom obvykle stojí pred rozhodnutím, akým spôsobom investovať do rôznych aktív. Takéto rozhodovanie je spojené s hodnotením viacerých faktorov. Pri investovaní do rizikových aktív investor vždy akceptuje nejaké riziko. Riziko je možnosť, že sa skutočný výnos bude odlišovať od očakávaného výnosu. To, aká je veľkosť rizika, ktoré je investor ochotný akceptovať možno popísať prostredníctvom investorových preferencií meraných aplikáciou koncepcie očakávanej užitočnosti. Za predpokladu, že sa investor pokúša dosiahnuť čo najvyšší zisk a riadiť pritom úroveň svojho vystavenia sa riziku, môžeme cieľ investora sformulovať týmito dvoma spôsobmi:

- a) pri danej miere rizika zaistiť čo najvyšší očakávaný výnos,
- b) pre požadovanú úroveň miery výnosu zaistiť očakávaný výnos s najnižším možným rizikom.

Problém investičného rozhodovania možno formulovať ako úlohu nelineárneho programovania. Nelinearita v takýchto úlohách pochádza so zohľadňovania rizika. Hoci existujú rôzne modely pre výber portfólia, široko uplatňovaná je metóda kvadratickej optimalizácie, pri ktorej sa minimalizuje riziko.

V súčasnosti je bežnou praxou, že profesionálni manažéri veľkých akciových portfólií používajú počítačové modely čiastočne založené na nelineárnom programovaní. Preto že sa investori orientujú na očakávaný výnos ako aj na mieru rizika spojenú s ich

investíciou. Metódy nelineárneho programovania sa sústreďujú na zostavenie takého portfólia, ktoré za určitých predpokladov predstavuje optimálnu rovnováhu medzi týmito dvoma faktormi. Tento postup je s prevažnej miery založený na prelomovom výskume Harryho Markowitza a Williama Sharpeho, ktorý im pomohol získať Nobelovu cenu za ekonómiu v roku 1990.

Model popisujúci tento problém môže byť formulovaný ako úloha nelineárneho programovania, v ktorej máme n rôznych cenných papierov pre zostavenie portfólia a rozhodovacie premenné x_j ($j=1,2,\dots,n$) ktoré predstavujú množstvo, v akom je nakúpený j -ty cenný papier. Nech μ_j a σ_{jj} vyjadrujú priemer a varianciu výnosu, respektíve môžeme tiež povedať, že σ_{jj} vyjadruje mieru rizika j -teho cenného papiera. Pre $i = 1, 2, \dots, n$ ($i \neq j$) σ_{ij} vyjadruje kovarianciu výnosov medzi cenným papierom i a cenným papierom j . Pre variačno-kovariačnú maticu S platí vzťah

$$\{S\}_{ij} = \sigma_{ij}^2 = \frac{1}{T} \sum_{t=1}^T (r_i(t) - \mu_i)(r_j(t) - \mu_j) \quad (1.5.1)$$

kde r predstavuje výnos investície i v roku t .

Preto že by bolo príliš náročné odhadnúť všetky hodnoty σ_{ij} , zvyčajný postup je založený na presných predpokladoch ohľadom správania sa trhu, ktoré umožňujú výpočet σ_{ij} priamo z σ_{ii} a σ_{jj} . Očakávanú hodnotu $R(x)$ a varianciu $V(x)$ potom definujeme ako

$$R(x) = \sum_{j=1}^n \mu_j x_j \quad (1.5.2)$$

$$V(x) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j, \quad (1.5.3)$$

kde $V(x)$ vyjadruje riziko spojené s portfóliom. Jedným zo spôsobom ako zohľadniť rovnováhu medzi týmito dvoma faktormi, je použitie $V(x)$ účelovej funkcie, ktorú minimalizujeme a súčasne ako ohraničenie zohľadňujeme $R(x)$, ktoré nesmie byť nižšie ako minimálny prijateľný očakávaný výnos. Úplný tvar nelineárneho modelu má potom formu

$$\min V(x) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j \quad (1.5.4)$$

pri ohraničeníach

$$\sum_{j=1}^n \mu_j x_j \geq L \quad (1.5.5)$$

$$\sum_{j=1}^n P_j x_j \leq B \quad (1.5.6)$$

kde $x_j \geq 0$, pre $j = 1, 2, \dots, n$. L predstavuje najmenší prijateľný očakávaný výnos, P_j je cena j -teho cenného papiera a B je objem financií v rozpočte určenom pre portfólio. Uvedená formulácia je vhodná v prípade, že hľadáme riešenie vyjadrené absolútnymi veličinami. V takom prípade sú B , L , x a P vyjadrené peňažnými jednotkami.

V praxi môže nastať situácia, kedy pri hľadaní optimálnej kombinácie portfólií neuvažujeme s konkrétnymi hodnotami výnosov, zdrojov či investícií. Naším cieľom môže byť pomerné rozdelenie investícií do jednotlivých cenných papierov tak, aby sme dosiahli čo najväčší percentuálny výnos. Premenná x_j bude teda vyjadrovať, časť investície vloženú do i -teho cenného papiera. V takom prípade neplatí (1.5.6), namiesto toho budeme uvažovať, že celkový objem investícií (rozpočet) tvorí 100%. Z dôvodu zachovania pestrosti portfólia býva niekedy kladená požiadavka, obmedzujúca výšku investície do jedného cenného papiera. Maximálnu možnú investíciu označíme ako h_j . Takúto úlohu môžeme zapísať ako

$$\min x^T S x \quad (1.5.7)$$

s ohraničeniami

$$\sum_i x_i = 1 \quad (1.5.8)$$

$$\sum_i \mu_i x_i \geq R \quad (1.5.9)$$

kde $x_i \geq 0$, $x_i \leq h_i$.

Nevýhodou tejto formulácie je, že sa ťažko vyberá vhodná hodnota L pre dosiahnutie vhodnej rovnováhy medzi $R(x)$ a $V(x)$. Namiesto skončenia výberom iba jedného L býva zvyčajným postupom parametrické nelineárne programovanie, pre tvorbu optimálneho riešenia funkcie L , ako širokej množiny hodnôt. Ďalším krokom je preskúmanie hodnôt $R(x)$ a $V(x)$ pre tieto riešenia, ktoré sú optimálne pri určitej úrovni L a následný výber variantu, ktorý poskytuje najlepšiu rovnováhu medzi týmito hodnotami. Takýto postup sa často uvádza ako ten, pri ktorom vznikajú riešenia na hranici efektívnej množiny v dvojrozmernom grafe $R(x)$, $V(x)$, na ktorom sa nachádza prípustné x . Každé optimálne x je efektívne v tom zmysle, že žiadne iné prípustné riešenie nie je aspoň tak dobré s rovnakým pomerom R a V a zároveň je lepšie s iným pomerom R a V .

2. Cieľ práce

Hlavným cieľom práce je popísanie niektorých optimalizačných metód, určených na riešenie nelineárnych úloh. Uvedením niekoľkých príkladov uplatnenia takýchto metód v praxi sa pokúsime zdôrazniť ich význam. V metodologickej časti práce popisujeme teoretické východiská týchto metód a v praktickej časti ukazujeme konkrétne riešenie vybranej metódy. Výsledkom práce je procedúra, ktorá umožňuje užívateľovi sledovať priebeh výpočtového procesu vybranej metódy v jednotlivých iteráciách a analyzovať tak vplyv zmeny vstupných parametrov na výpočet.

Čiastkovým cieľom práce je vysvetlenie problematiky výberu optimálneho portfólia s rizikovými cennými papiermi. Problém je popísaný v teoretickej rovine, formulovaný v podobe hypotetického príkladu a následne riešený dvoma rôznymi metódami nelineárnej optimalizácie.

Nemenej dôležitým cieľom je aj vysvetlenie postupu tvorby, pomocou ktorého je možná realizácia optimalizačného algoritmu s využitím bežne dostupných nástrojov výpočtovej techniky. Opis tvorby výpočtovej procedúry v prostredí MS Excel umožňuje študujúcemu pochopiť väzby medzi teoretickým vyjadrením vybranej metódy a jej interpretáciou v jazyku VBA.

3. Metódy riešenia úloh nelineárneho programovania

V prvej časti tejto kapitoly budú popísané metódy, ktoré riešia ohraničenia priamo. Prezentované metódy sú rozšírením techník odvodených od problémov bez ohraničení, pri ktorých sú predstavené špeciálne kroky zohľadňujúce ohraničenia. Tieto kroky najskôr nájdu prípustný smer, ktorý ukazuje na ohraničené minimum, potom prejdú k hľadaniu vzhľadom na prípustnú množinu. Vo všeobecnosti platí, že aj keď sú uplatnené penalizačné metódy, lineárne ohraničenia by mali byť riešené priamo a nemali by byť transformované na rôzne rozšírené účelové funkcie. Žiadna takýto predpoklad však neplatí o nelineárnych ohraničeniach, hoci by sa zdalo, že vzhľadom na súčasný stav výpočtových techník, penalizačné metódy môžu byť vhodné pre tento typ problému. Priamy prístup, proces nasledovania aktívneho nelineárneho ohraničenia, ako aj ohraničenia rovnosti je obyčajne zložitá a časovo náročná operácia, ktorej je lepšie sa vyhnúť. Možnou alternatívou by mohla byť nejaká postupná linearizácia nelineárnych ohraničení, ktorej výsledok vo všeobecnosti spočíva v riešení postupnosti lineárnych úloh.

Projekčné metódy a metódy prípustných smerov pre úlohy nelineárnych ohraničení sú orientované výhradne na ohraničenú optimalizáciu. Tieto metódy buď transformujú ohraničený problém na neohraničený, alebo rozširujú neohraničené metódy na ohraničené problémy primeranou úpravou hľadania smeru a dĺžky kroku. Okrem nich predstavíme aj metódy ohraničenej optimalizácie, ktoré sú orientované na lineárne programovanie v tom zmysle, že pozostávajú z krokov, v ktorých sú riešené čiastkové úlohy lineárneho programovania, alebo ak sa problém po zvážení stane lineárnym, algoritmus sa stane verziou dobre známej simplexovej metódy. Metódy popísané v tejto kapitole, ako aj ďalšie metódy pre riešenie nelineárnych optimalizačných úloh popisuje Avriel vo svojej práci *Nelineárne programovanie, analýzy a metódy* [1], kde je okrem bližšieho popisu metód možné nájsť niekoľko modelových príkladov pre aplikáciu popísaných algoritmov.

3.1 Metóda prípustných smerov

Ukážme si teraz základné postupy riešenia veľkého množstva úloh matematického programovania. Predpokladajme problém

$$\min f(x) \tag{3.1.1}$$

pri ohraničeniach

$$x \in X \subset R^n, \quad (3.1.2)$$

kde X je uzavretá spojitá množina a f je spojito diferencovateľná funkcia, ktorá má ohraničené úrovňové množiny. Algoritmus pre riešenie tejto úlohy sa nazýva metóda prípustných smerov ak môže byť popísaný nasledovným spôsobom:

Pre $k=0,1,\dots$, je daný bod $x^k \in X$ nájdeme smerový vektor z^{k+1} kde

$$(z^{k+1})^T \nabla f(x^k) < 0 \quad (3.1.3)$$

a existuje také $\bar{a}_{k+1} > 0$ ktoré spĺňa $x^k + a_{k+1} z^{k+1} \in X$ pre všetky $0 \leq a_{k+1} \leq \bar{a}_{k+1}$, teda z^{k+1} je prípustný pokles, tiež nazývaný “použiteľný smer“. Určme a_{k+1}^* takým kritériom, že

$$x^{k+1} = x^k + a_{k+1}^* z^{k+1} \in X. \quad (3.1.4)$$

Týmto spôsobom pokračujeme pokým nie je splnená niektorá zo zastavovacích podmienok, ideálne podmienka konvergenie.

Metódy prípustných smerov boli odvodené Zoutendijkom a môžu byť nájdené v mnohých jeho prácach. Väčšina neohraničených algoritmov popísaných v predošlých kapitolách, rovnako ako metódy zaoberajúce sa ohraničenými úlohami popísané v tejto kapitole, sú vlastne špeciálnym prípadom metódy prípustných smerov. Navyše simplexová metóda lineárneho programovania môže byť tiež považovaná za metódu prípustných smerov. Treba podotknúť, že pre všeobecnosť metódy prípustných smerov je otázne, či sú myšlienky a závery popísané v tejto kapitole skutočne rozšírením metód neohraničenej optimalizácie, teda či vôbec patria do tejto kapitoly. Rozbor metód prípustných smerov teda obmedzíme iba na prípady lineárne ohraničených nelineárnych úloh

$$(LOP) \quad \min f(x) \quad (3.1.5)$$

vzhľadom na

$$\sum_{j=1}^n a_{ij} x_j - b_i = 0, \quad i = 1, \dots, m \quad (3.1.6)$$

$$\sum_{j=1}^n a_{ij}x_j - b_i \geq 0, \quad i = m+1, \dots, p, \quad (3.1.7)$$

definovaných vyššie. V každej iterácii metódy prípustných smerov je potrebné vykonať dva základné kroky:

1. výber prípustného smeru poklesu.
2. voľba dĺžky kroku prípustným smerom.

Počnúc prvým, budeme musieť ukázať, že pre daný bod $x^k \in X$, spĺňajúci (3.1.8) a (3.1.9), sa musí prípustný smer poklesu z x^k nachádzať v $Z^1(x^k) \cap Z^2(x^k)$. Ďalej definujeme

$$Z^1(x^k) = \left\{ z: \sum_{j=1}^n a_{ij}z_j = 0, j = 1, \dots, m; \sum_{j=1}^n a_{ij}z_j \geq 0, i \in I(x^k) \right\} \quad (3.1.8)$$

$$Z^2(x^k) = \{z: z^T \nabla f(x^k) < 0\}, \quad (3.1.9)$$

kde

$$I(x^k) = \left\{ i: \sum_{j=1}^n a_{ij}x_j^k = b_i, i = m+1, \dots, p \right\}. \quad (3.1.10)$$

Vo všeobecnosti platí, že z daného bodu $x^k \in X$ existuje viacero prípustných smerov poklesu a teda najstrmší z nich môže byť vybratý riešením optimalizačnej úlohy

$$(NPS) \quad \min \quad z^T \nabla f(x^k) \quad (3.1.11)$$

vzhľadom na $z \in Z^i(x^k)$, a

$$\|z\| \leq 1. \quad (3.1.12)$$

Pokiaľ je $Z^i(x^k)$ kónická, je pre dosiahnutie konečného riešenia nevyhnutné ohraničiť $\|z\|$. Všimnime si, že $\|z\|$, Euklidovská norma z je nelineárnou funkciou, teda (2.1.12) je

nelineárnym ohraňčením. Vieme že $Z^1(x^k) \cap Z^2(x^k) = \emptyset$, teda neexistuje ťiaden prípustný smer z x^k práve vtedy a len vtedy, ak existuje vektor $\lambda \geq 0$ a $\mu \in R^m$ kde

$$\nabla f(x^k) = \sum_{i=1}^m \mu_i a^i + \sum_{i \in I(x^k)} \lambda_i a^i, \quad (3.1.13)$$

kde $a^i = (a_{i1}, \dots, a_{in})^T$. Inými slovami, je možné nájsť prípustný smer poklesu z x^k vtedy a len vtedy, ak $\nabla f(x^k)$ nie je kónicky rozložený vektorom a^i , ktorý zodpovedá aktívnym ohraňčeniam. Predpokladajme, že tieto predpoklady platia. Nech je t^{k+1} bodom kuťela, ktorý je najbliťšie k $\nabla f(x^k)$, teda

$$(t^{k+1})^T [t^{k+1} - \nabla f(x^k)] = 0. \quad (3.1.14)$$

Vektor najstrmšieho prípustného poklesu, ktorý je riešením (NPS), je daný ako

$$z^{k+1} = \frac{t^{k+1} - \nabla f(x^k)}{\|t^{k+1} - \nabla f(x^k)\|}. \quad (3.1.15)$$

Algoritmus prípustných smerov ako aj niektoré dodatočné procedúry boli navrhnuté McCormickom, Polakom, Zangwillom, Zoutendijkom a ďalšími. Ku koncu tejto kapitoly uvedieme Zautendijkovu metódu, ktorá zabráňuje ‘‘cikcakovému‘‘ priebehu algoritmu tým, že nezachováva aktívne ohraňčenia aktívnymi, pokiaľ to nie je nevyhnutné.

1. Predpokladajme, že $x^0 \in X$. Vypočítame $\nabla f(x^0)$. Nech $I_0 = K_0 = \emptyset$, poloťíme $k = 0$ a zvolíme malé nejaké malé číslo $\epsilon > 0$.

2. Zvolíme niektorú metódu pre voľbu prípustného smeru a nájdeme z^{k+1} pri splnení podmienok

$$(a^i)^T z^{k+1} = 0, \quad i = 1, \dots, m \quad (3.1.16)$$

$$(a^i)^T z^{k+1} \begin{cases} = 0, & i \in I_k \\ \geq 0, & i \in I(x^k), i \notin I_k. \end{cases} \quad (3.1.17)$$

$$(\nabla f(x^{h+1}) - \nabla f(x^h))^T z^{k+1} = 0, \quad h \in K_k. \quad (3.1.18)$$

$$(z^{k+1})^T \nabla f(x^k) \leq -\epsilon. \quad (3.1.19)$$

3. Pokiaľ je úloha hľadania smeru nekonzistentná, teda ťiadne z^{k+1} nespĺňa tieto podmienky, potom

I. Ak $I_k \cup K_k \neq \emptyset$, vynecháme najstarší index z I_k alebo z K_k a prejdeme na krok 2.

II. Ak $I_k \cup K_k = \emptyset$, skončíme. Bolo dosiahnuté optimálne riešenie.

4. Určíme $\alpha_{k+1}^* = \min\{\alpha_{k+1}', \alpha_{k+1}''\}$. Ak $\alpha_{k+1}^* = +\infty$ skončíme, riešenie je neohraničené.

5. Vypočítame

$$x^{k+1} = x^k + \alpha_{k+1}^* z^{k+1} \quad (3.1.20)$$

a $\nabla f(x^{k+1})$.

6. Zvážime možnosti

I. Ak $\alpha_{k+1}^* = \alpha_{k+1}'$, potom nech $I_{k+1} = I_k$ a $K_{k+1} = K_k \cup \{k\}$.

II. Ak $\alpha_{k+1}^* = \alpha_{k+1}''$, potom nech

$$I_{k+1} = I_k \cup \{i: (a^i)^T x^{k+1} = 0 \text{ a } (a^i)^T x^k > 0, i = m+1, \dots, p\} \quad (3.1.21)$$

a nech $K_{k+1} = \emptyset$.

III. $\alpha_{k+1}^* = \alpha_{k+1}''$, postupujeme rovnako ako v 6.II. s tým rozdielom, že $K_{k+1} = \{k\}$.

7. Nech $k = k + 1$ a prejdeme ku kroku 2.

Všimnime si, že (3.1.18) bolo pridané pre dosiahnutie združených smerov, pokiaľ nebolo nájdené optimálne riešenie, teda

$$(z)^T \nabla f(x^k) > -\epsilon \quad (3.1.22)$$

pre všetky prípustné z , alebo množinu aktívnych ohraňení sa mení. V praxi sa ϵ postupne znižuje až k veľmi malému kladnému číslu, aby sa zabezpečila približná optimálnosť. Pre kvadratické účelové funkcie, $\epsilon = 0$ v každej iterácii.

Ak nie je dostupný žiadny počiatočný prípustný bod, môžeme použiť prvú fázu lineárneho programovania, v ktorej riešime pomocnú lineárnu úlohu

$$\min \sum_{i=1}^p \xi_i \quad (3.1.23)$$

pri ohraňeniach

$$\sum_{j=1}^n a_{ij}x_j + \xi_i = b_i, \quad i = 1, \dots, m \quad (3.1.24)$$

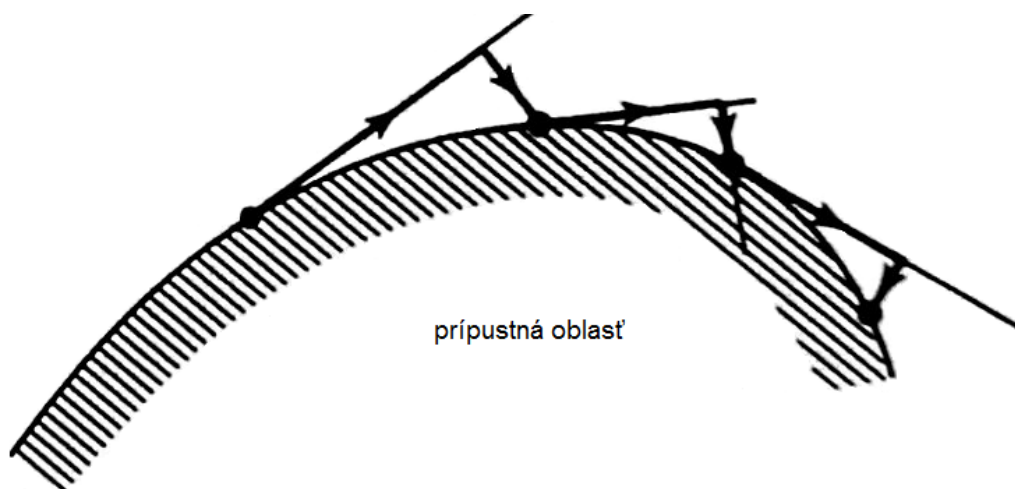
$$\sum_{j=1}^n a_{ij}x_j + \xi_i \geq b_i, \quad i = m + 1, \dots, p \quad (3.1.25)$$

$$\xi \geq 0. \quad (3.1.26)$$

Ak je optimálnym riešením takejto úlohy (x^*, ξ^*) , kde $\xi^* = 0$, potom zadáním $x^0 = x^*$ môže začať algoritmus prípustných smerov s nelineárnou účelovou funkciou.

3.2 Projekčné metódy pre nelineárne ohraničenia

Niektoré tu uvedené metódy boli rozšírené na matematické úlohy s nelineárnymi ohraničeniami. Takýto spôsob riešenia je však omnoho zložitejší ako pri lineárnych ohraničeniach. Ak sú ohraničenia lineárne, smer hľadania, pri ktorom je účelová funkcia minimalizovaná, leží buď vo vnútri množiny prípustných riešení, alebo v prieniku nadrovín zodpovedajúcim aktívnym ohraničeniam. V oboch prípadoch nehrozí žiadne riziko opustenia prípustnej množiny posunom zdaného bodu o malú vzdialenosť v smere hľadania. Problém so zobrazovacími metódami a metódami prípustných smerov, vznikajúci pri nelineárnych ohraničeniach spočíva v obmedzenosti smeru hľadania na prienik dotýkajúcich sa nadrovín aktívnych nelineárnych ohraničení. V takom prípade môže akýkoľvek pohyb v smere hľadania viesť k neprípustnému bodu a pre návrat do prípustnej množiny sú potrebné zvláštne kroky. Opakovaný pohyb po smere hľadania a návraty do prípustnej množiny spôsobujú “výšivkový” jav znázornený na obrázku 3.2.1.



Obr. 3.2.1 Vyšívanie nasledovaním aktívnych nelineárnych ohraničení

Priame riešenie tohto problému spôsobuje značné spomalenie algoritmu. Z tohto dôvodu sa nepriame riešenie nelineárnych ohraničení prostredníctvom penalizačných funkcií javí výhodnejšie, pokiaľ teda nelineárne ohraničenia nespádajú pod nejaký špeciálny typ štruktúry, ako pri geometrickom programovaní. Metódy popísané ďalej môžu byť tiež použité pri riešení určitých nelineárne ohraničených úlohách, miera konvergencie algoritmu je však vo všeobecnosti pomalá. Napriek tomu v tejto kapitole uvedieme stručné zopakovanie niektorých rozšírení techník neohraničenej optimalizácie na úlohy nelineárne ohraničeného matematického programovania.

Uvažujme všeobecný nelineárny problém

$$\min f(x) \quad (3.2.1)$$

pri ohraničeniach

$$g_i(x) \geq 0, \quad i = 1, \dots, m \quad (3.2.2)$$

$$h_j(x) = 0, \quad j = 1, \dots, p \quad (3.2.3)$$

kde f , g_i , h_j sú spojito diferencovateľné funkcie a najmenej jedno z ohraničení je nelineárne. Metóda pre riešenie tejto úlohy je rozšírením Goldfarbovho algoritmu.

Ohraničenie nerovnosti $g_i(x) \geq 0$ linearizované v bode x^k , je dané ako

$$g_i(x^k) + (x - x^k)^T \nabla g_i(x^k) \geq 0 \quad (3.2.4)$$

a podobne linearizované ohraničenie rovnosti $h_j(x) = 0$ je

$$h_j(x^k) + (x - x^k)^T \nabla h_j(x^k) = 0. \quad (3.2.5)$$

Ak $g_i(x^k) = 0$, linearizované ohraničenie (3.2.4) definuje uzavretý polpriestor, tvorený dotyčnicou nadroviny ku hladine $g_i(x) = 0$ v x^k . Ak $h_j(x^k) = 0$, potom body vyhovujúce (1.5.5) ležia v nadrovine dotýkajúcej sa $h_j(x) = 0$ v x^k .

Počas výpočtov musia zostať všetky ohraničenia rovnosti aktívne. Nelineárne ohraničenie nazývame aktívnym v bode x^k , ak je zodpovedajúce lineárne ohraničenie aktívne v bode x^k . Ak je niektoré z g_i alebo h_i lineárne, sú riešené rovnakým spôsobom ako v Goldfarbovom algoritme. Daným prípustným bodom x^k určujeme množinu aktívnych ohraničení v x^k a počítame z^{k+1} , smer hľadania, ktorý je vhodne zobrazený do prieniku

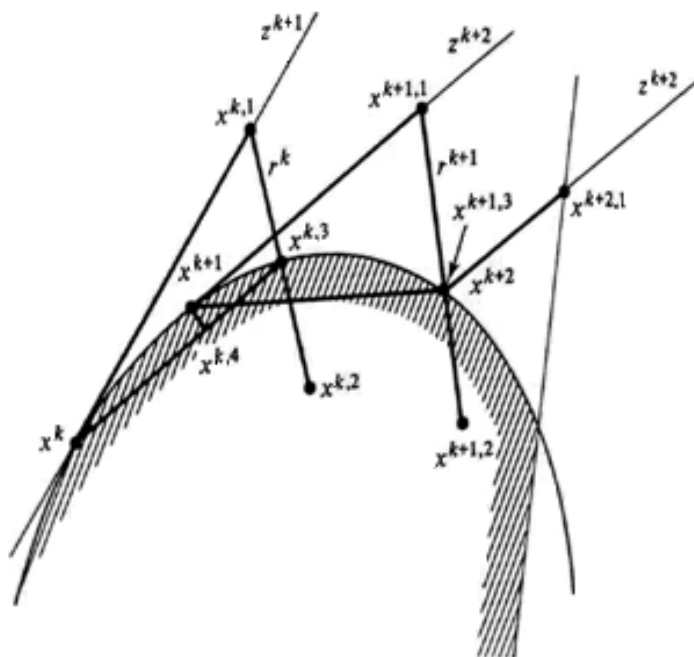
nadrovín zodpovedajúcemu aktívnym ohraňčeniam. Ďalšou úlohou je zistiť \bar{a}_{k+1} pre dĺžku kroku v smere z^{k+1} . Je vyhodnotená vzdialenosť k najbližšiemu linerizovanému ohraňčeniu v smere z^{k+1} . Nech je táto vzdialenosť θ . Potom majme

$$\bar{a}_{k+1} = \min\{c, \theta\}, \quad (3.2.6)$$

kde c je nejaká kladná konštanta, napríklad $c=2$. Takéto \bar{a}_{k+1} bude obmedzovať dĺžku kroku a zabráni pohybu ďaleko mimo prípustnej množiny. Ďalej nech

$$x^{k,1} = x^k + \bar{a}_{k+1}z^{k+1} \quad (3.2.7)$$

a porovnáme všetky ohraňčenia pre $x^{k,1}$. Ak je niektoré z ohraňčení narušené, vektor návratového smeru r^k je určený tak, že ukazuje naspäť do prípustnej množiny. Uskutoční sa pohyb týmto smerom a je nájdený vnútorný bod $x^{k,2}$, nasledovný interpolačnou schémou, ktorá určí bod $x^{k,3}$ na hranici prípustnej množiny. Pre tento účel bola Beamerom a Wildeom vyvinutá zaujímavá interpolačná metóda. Minimum z^k je potom hľadané na úsečke $x^k, x^{k,3}$. Ak je takéto minimum, povedzme $x^{k,4}$ nájdené, a ak je prípustná množina konvexná, potom sa $x^{k,4}$ nachádza vo vnútri prípustnej množiny. V ďalšom kroku sa opäť musíme posunúť smerom k ohraňčeniu v smere najstrmšieho poklesu, do ďalšieho bodu x^{k+1} . Ak nie je medzi x^k a $x^{k,3}$ nájdené žiadne minimum, potom začína nové hľadanie z bodu $x^{k,3}$ v smere z^{k+1} . Typická situácia je ukázaná na obrázku 3.2.2.



Obr. 2.2.2 Nasledovanie aktívnych nelineárnych ohraňčení Daviesovou metódou

Popíšme si teraz algoritmu prípustných smerov pre nelineárne ohraničenia odvodený Zoutendijkom. Uvažujme problém

$$\min f(x) \quad (3.2.8)$$

vzhľadom na

$$g_i(x) \geq 0, \quad i = 1, \dots, m \quad (3.2.9)$$

kde f a g_i sú spojito diferencovateľné. Pre všetky prípustné x a $\epsilon \geq 0$ platí

$$X_\epsilon(x) = \{i: g_i(x) \leq \epsilon\} \quad (3.2.10)$$

a X je prípustná množina. V k -tej iterácii sú dané prípustný bod x^k a $\epsilon^k > 0$. Riešime lineárnu úlohu pre výber smeru.

$$\min \sigma \quad (3.2.11)$$

pri

$$\sigma \geq z^T \nabla f(x^k) \quad (3.2.12)$$

$$z^T g_i(x^k) \geq \sigma, \quad i \in X_{\epsilon^k}(x^k) \quad (3.2.13)$$

$$|z_j| \leq 1, \quad j = 1, \dots, n. \quad (3.2.14)$$

Nech optimálne riešenie $\sigma^{k+1} \in R, z^{k+1} \in R$. Ak $\sigma^{k+1} \leq -\epsilon^k$, potom $\epsilon^{k+1} = \epsilon^k$; inak $\epsilon^{k+1} = \frac{1}{2}\epsilon^k$. Po nájdení prípustného smeru musíme zvoliť dĺžku kroku α_{k+1}^* . Tú vypočítame ako

$$\alpha_{k+1}^* = \min \{\alpha_{k+1}', \alpha_{k+1}''\} \quad (3.2.15)$$

kde

$$\alpha_{k+1}' = \max \{\alpha_{k+1}': (z^{k+1})^T \nabla f(x^k + \alpha_{k+1} z^{k+1}) \leq 0\} \quad (3.2.16)$$

$$\alpha_{k+1}'' = \max \{\alpha_{k+1}'': (x^k + \alpha_{k+1} z^{k+1}) \in X\}. \quad (3.2.17)$$

Pre $\alpha_{k+1}' < +\infty$, $\nabla f(x^k + \alpha_{k+1} z^{k+1}) = 0$. Ďalej vypočítame $x^{k+1} = x^k + \alpha_{k+1}^* z^{k+1}$. Táto metóda konverguje k riešeniu úlohy danej v (3.2.8) a (3.2.9), kde bod x^* nazývame

riešením, ak z neho nevieme určiť žiaden prípustný smer poklesu. Výpočet α_{k+1}^* môže byť samozrejme veľmi komplikovaný a môže byť potrebných viacero iteračných nástrojov.

Viacero úspešných realizácií metód penalizačných funkcií a algoritmov pre lineárne ohraničené nelineárne úlohy viedli k dvom významným prácam, v ktorých sú nelineárne ohraničené úlohy riešené sériou lineárne ohraničených úloh, použitím Lagrangeovej penalizačnej funkcie.

Znovu uvažujme úlohu s ohraňením nerovnosti

$$\min f(x) \quad (3.2.18)$$

$$g_i(x) \geq 0, \quad i = 1, \dots, m \quad (3.2.19)$$

kde f a g_i sú spojitó diferencovateľné, konvexné a konkávne funkcie $x \in R^n$. Metóda Rosena a Kreusera môže byť popísaná nasledovne: Začneme v ľubovoľnom $x^0 \in R^n$. V k -tej iterácii máme množinu indexov $I(x^k)$ definovanú ako

$$I(x^k) = \{i: g_i(x^k) \leq 0, \quad i = 1, \dots, m\}. \quad (3.2.20)$$

Vypočítame λ^k ako

$$\lambda^k = J^+(x^k) \nabla f(x^k), \quad (3.2.21)$$

kde

$$J^+(x^k) = \left[\left(J(x^k) \right)^T J(x^k) \right]^{-1} (J(x^k))^T \quad (3.2.22)$$

kde $J(x^k)$ predstavuje Jacobiho maticu (maticu parciálnych derivácií) funkcie g_i v bode x^k pre $i \in I(x^k)$, teda matica, ktorej riadky sú $\nabla g_i(x^k)$, $i \in I(x^k)$. Potom $J^+(x^k)$ je všeobecne inverzná k $J(x^k)$. Matica $(J(x^k))^T J(x^k)$ má byť nesignulárna. Ďalej riešime lineárne ohraňený problém

$$\min \left\{ f(x) - \sum_{\lambda_i^k > 0} \lambda_i^k [g_i(x) - g_i(x^k) - (x - x^k)^T \nabla g_i(x^k)] \right\} \quad (3.2.23)$$

vzhľadom na

$$g_i(x^k) + (x - x^k)^T \nabla g_i(x^k) \geq 0, \quad i = 1, \dots, m \quad (3.2.24)$$

napríklad Goldfarbovou metódou. Nech je riešením x^{k+1} . Ak x^{k+1} spĺňa pre úlohu (3.2.18) a (3.2.19) podmienky Kuhna-Tuckera prvého rádu s určitou toleranciou, skončíme, inak položíme $k=k+1$ a opakujeme predošlé kroky. Je možné ukázať, že táto metóda konverguje kvadratickou mierou, ak sú splnené určité predpoklady regularity. Všimnime si, že sa v (3.2.20) nikdy neobjavia lineárne ohraničenia, sú však zahrnuté v (3.2.24).

V podobnej metóde navrhuje Robins riešenie (3.2.18) a (3.2.19) s pridaným ohraničením rovnosti

$$h_j(x) = 0, \quad j = 1, \dots, p \quad (3.2.25)$$

kde h_j , rovnako ako f a g_i sú dvojnásobne spojito diferencovateľné, nasledovne: V iterácii k máme vektor $x^k \in R^n$ a vektory $\lambda^k \in R^m, \lambda^k \geq 0, \mu^k \in R^p$. Riešime ohraničenú úlohu

$$\min_x \left\{ f(x) - \sum_{i=1}^m \lambda_i^k [g_i(x) - \tilde{g}_i(x_k, x)] - \sum_{j=1}^p \mu_j^k [h_j(x) - \tilde{h}_j(x_k, x)] \right\} \quad (3.2.26)$$

pri ohraničeniach

$$\tilde{g}_i(x^k, x) = g_i(x^k) + (x - x^k)^T \nabla g_i(x^k) \geq 0, \quad i = 1, \dots, m \quad (3.2.27)$$

$$\tilde{h}_j(x^k, x) = h_j(x^k) + (x - x^k)^T \nabla h_j(x^k) = 0, \quad j = 1, \dots, p. \quad (3.2.28)$$

Nech je riešením x^{k+1} . Získame zodpovedajúce Lagrangeove multiplikátory $\lambda^{k+1} \geq 0, \mu^{k+1}$, tejto lineárne ohraničenej úlohy. Ak je viac ako jeden z nich trojitý $(x^{k+1}, \lambda^{k+1}, \mu^{k+1})$, vyberieme spomedzi nich ten, ktorý je podľa určitých štandardov najbližšie k (x^k, λ^k, μ^k) . Ak $(x^{k+1}, \lambda^{k+1}, \mu^{k+1})$ spĺňajú nutné podmienky optimálnosti Kuhna-Tuckera prvého rádu pre úlohu definovanú v (3.2.18), (3.2.19) a (3.2.25) v určitej tolerancii, skončíme. Inak položíme $k=k+1$ a preriešime (3.2.26) s danými ohraničeniami. Počiatočný bod algoritmu x^0 musí byť prípustný. Znova, za vhodných predpokladov vzhľadom na funkciu definujúcu úlohu, algoritmus konverguje k optimu kvadraticky.

3.3 Metódy aproximačného programovania

Všeobecne najjednoduchší postup pri riešení úloh nelineárneho programovania sériou lineárnych úloh je aproximácia každej nelineárnej funkcie vyskytujúcej sa v úlohe lineárnou funkciou a riešenie výslednej lineárnej úlohy. Okamžitá námietka je, že lineárna aproximácia je použiteľná iba lokálne, teda iba v bodoch, okolo ktorých sa nelineárne funkcie aproximujú alebo v ich malom okolí. Pre vyvinutie použiteľného algoritmu musíme teda obmedziť pohyb z daného bodu, okolo ktorého sú funkcie aproximované, robia sa teda iba malé kroky. Z tohto dôvodu sa metódy popísané v tejto časti nazývajú aj “malo-krokové” gradientové metódy, naproti “veľko-krokovým” metódam, ktoré boli popísané vyššie.

Prvá tu uvedená metóda aproximačného programovania bola odvodená Griffithom a Stewartom. Uvažujme všeobecnú úlohu nelineárneho programovania s ohraničenými premennými.

$$\min f(x) \tag{3.3.1}$$

vzhľadom na

$$g_i(x) \geq 0, \quad i = 1, \dots, m \tag{3.3.2}$$

$$h_l(x) = 0, \quad l = 1, \dots, p \tag{3.3.3}$$

$$u_j \geq x \geq 0, \quad j = 1, \dots, n \tag{3.3.4}$$

kde f , g_i , h_l sú diferencovateľné funkcie a horné ohraničenia u_j sú dané. Ak sú zavedené nenulové spodné ohraničenia, môžu byť jednoducho transformované na predošlú formu jednoduchou zmenou premenných.

Predpokladajme, že \bar{x} je prípustným riešením ohraničenej úlohy. Linearizáciou každej nelineárnej funkcie pri \bar{x} dostaneme

$$f(x) \cong \tilde{f}(x, \bar{x}) = f(\bar{x}) + (x - \bar{x})^T \nabla f(\bar{x}) \tag{3.3.5}$$

$$g_i(x) \cong \tilde{g}_i(x, \bar{x}) = g_i(\bar{x}) + (x - \bar{x})^T \nabla g_i(\bar{x}), \quad i = 1, \dots, m \tag{3.3.6}$$

$$h_l(x) \cong \tilde{h}_l(x, \bar{x}) = h_l(\bar{x}) + (x - \bar{x})^T \nabla h_l(\bar{x}), \quad l = 1, \dots, p. \tag{3.3.7}$$

Uvedieme nový vektor y definovaný ako

$$y = x - \bar{x}. \quad (3.3.8)$$

Nakoľko je y všeobecne neohraničený v znamienku, zavedieme vzťah

$$y_j = y_j^+ - y_j^-, \quad j = 1, \dots, n \quad (3.3.9)$$

kde

$$y_j^+ \geq 0 \quad y_j^- \geq 0, \quad j = 1, \dots, n \quad (3.3.10)$$

$$y_j^+ = \begin{cases} y_j & \text{ak } y_j \geq 0 \\ 0 & \text{ak } y_j < 0 \end{cases} \quad (3.3.11)$$

$$y_j^- = \begin{cases} -y_j & \text{ak } y_j \leq 0 \\ 0 & \text{ak } y_j > 0. \end{cases} \quad (3.3.12)$$

Početnosť $|y_j|$ reprezentuje vzdialenosť x_j od \bar{x}_j , zodpovedajúceho komponentu aproximujúceho bodu \tilde{x} . Túto vzdialenosť budeme ohraničovať nerovnosťou

$$|y_j| \leq m_j, \quad j = 1, \dots, n \quad (3.3.13)$$

kde m_j predstavujú dané ohraničenia. Nakoľko je vektor x tiež ohraničený, môžeme vyvodiť

$$0 \leq y_j^+ \leq \min\{m_j, u_j - \bar{x}_j\} \quad (3.3.14)$$

$$0 \leq y_j^- \leq \min\{m_j, \bar{x}_j\}. \quad (3.3.15)$$

Aproximácia lineárnej úlohy je daná

$$\min \tilde{f}(x, \bar{x}) = f(\bar{x}) + (y^+)^T \nabla f(\bar{x}) - (y^-)^T \nabla f(\bar{x}) \quad (3.3.16)$$

vzhľadom na

$$\tilde{g}_i(x, \bar{x}) = g_i(\bar{x}) + (y^+)^T \nabla g_i(\bar{x}) - (y^-)^T \nabla g_i(\bar{x}) \geq 0, \quad i = 1, \dots, m \quad (3.3.17)$$

$$\tilde{h}_l(x, \bar{x}) = h_l(\bar{x}) + (y^+)^T \nabla h_l(\bar{x}) - (y^-)^T \nabla h_l(\bar{x}) = 0, \quad l = 1, \dots, p \quad (3.3.18)$$

a tiež na vyššie uvedené nerovnice.

Griffithov a Stewartov algoritmus pozostáva z iteratívnej schémy, v ktorej je riešená predošlá lineárna úloha začínajúc v bode \bar{x} . Ak je optimálne riešenie tejto aproximačnej úlohy x_L^* prípustné pre ohraničenú úlohu, potom je použité ako nový bod aproximácie pre ďalšiu lineárnu úlohu, s rovnakými ohraničeniami, aké boli použité v predošlej úlohe. Ak je však x_L^* naopak neprípustné, nie je prijaté a lineárna úloha je znovu riešená s redukovanými hodnotami m_j . Bod sa nazýva optimálnym pre ohraničenú úlohu vtedy, keď je prípustný v rámci určitej tolerancie a ak sú splnené kritéria konvergenzie na za sebou idúcich hodnotách y a $f(x)$.

Hodnoty ohraničení m_j výrazne ovplyvňujú úspešnosť aplikácie algoritmu. Ak sú príliš malé, uskutočnené kroky sú tiež veľmi malé a progres je pomalší. Na druhú stranu, príliš veľké hodnoty m_j môžu spôsobiť neprípustnosť riešenia, následkom čoho sú potrebné postupné redukcie ohraničení. Pri presune z jedného aproximačného bodu do iného sú všetky lineárne funkcie znovu aproximované a musí byť riešená celkom nová lineárna úloha. Metóda nie je obmedzená na konvexné úlohy a ich zovšeobecnenia. Hoci konvergencia k lokálnemu optimu nebola preukázaná ani pre konvexné úlohy, v praxi v prípade konvexnosti metóda zvyčajne konverguje. Griffith-Stewartova metóda je užitočná predovšetkým v úlohách nelineárnej optimalizácie s mnohými premennými a s možným nedostatkom predpokladu konvexnosti, keď je len málo nelineárnych ohraničení pridaných k lineárnym. V takýchto úlohách je často nutné vylepšiť nejaké konkrétne riešenie a teoretické predpoklady konvergenzie sú nepodstatné.

Je potrebné zmieniť sa o tom, že nelineárne úlohy, v ktorých sú funkcie separovateľné, ako úlohy s účelovou funkciou v tvare

$$f(x) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n) \quad (3.3.19)$$

pri lineárnych ohraničeniach, môžu byť bez problémov riešené lineárnou aproximáciou, ktorá využíva niektoré modifikácie simplexovej metódy alebo lineárneho programovania. Úspešná implementácia metód aproximačného programovania závisí od účinného riešenia aproximačného podproblému. Účinný algoritmus kvadratického programovania a skutočnosť, že nelineárne funkcie môžu byť aproximované efektívnejšie pomocou kvadratických vyústili k vývoju pokročilejších metód aproximačného programovania. Príkladom tejto triedy algoritmov je Fletcherova metóda nadrovín.

Táto metóda je navrhnutá pre riešenie ohraničených, prípadne neohraničených úloh. Fletcher neuvažoval nelineárne ohraničené úlohy, možno preto, že takéto ohraničenia by sa dali lepšie riešiť nepriamo, teda ich transformáciou do účelovej funkcie. Uvažujme problém

$$\min f(x) \quad (3.3.20)$$

s ohraničeniami

$$A^T x \geq b, \quad (3.3.21)$$

kde f je diferencovateľná funkcia, A je matica rozmeru $n \times m$ a b je vektor rozmeru m . Nech X udáva množinu $x \in R^n$ spĺňajúcu ohraničenia (3.3.21). V každej iterácii metódy je realizovaná kvadratická aproximácia $\nabla f_q(x, x^k)$ v bode x^k , ktorý zodpovedá f v bode x^k a taktiež $\nabla f_q(x^k) = \nabla f(x^k)$. Hessián funkcie f_q je popísaný danou symetrickou maticou B_k . Každá čiastková úloha pozostáva z riešenia kvadratickej úlohy, v ktorej je f_q minimalizovaná vzhľadom na (3.3.21) a vzhľadom na ohraničenie, ktoré obmedzuje možnosť presunu z daného bodu x^k v tvare

$$\|x - x^k\|_\infty = \max_{1 \leq j \leq n} |x_j - x_j^k| \leq h^k. \quad (3.3.22)$$

Dôležitou vlastnosťou tejto metódy je, že poskytuje symetrický odhad h^k , (3.3.22) teda definuje hyperkocku, pre ktorú je f_q v určitom zmysle platnou aproximáciou f .

Ďalšou dôležitou vlastnosťou je aktualizácia B_k matice v každej iterácii. Rovnako ako v nejakom kvázi-Newtonovskom algoritme pre neohraničenú optimalizáciu, B_k je aktualizovaná prostredníctvom druhostupňovej korekcie, pre ktorú platí vzťah

$$B_k p^k = y^k \quad (3.3.23)$$

kde rovnosti $p^k = x^k - x^{k-1}$ a $y^k = \nabla f(x^k) - \nabla f(x^{k-1})$ sú splnené. Navrhovaný vzťah pre aktualizáciu je daný ako

$$B_k = B_{k-1} + \frac{(\gamma^k - B_{k-1} p^k)(p^k)^T + p^k(\gamma^k - B_{k-1} p^k)^T}{(p^k)^T p^k} - \frac{p^k(\gamma^k - B_{k-1} p^k) p^k (p^k)^T}{[(p^k)^T p^k]^2}. \quad (3.3.24)$$

Tento aktualizálny vzťah nezaručuje, že B_k bude kladne definitná v prípade, že B_{k-1} je kladne definitná. Následkom tejto skutočnosti môže byť situácia, kedy jednotlivé podúlohy vedú k bodom spĺňajúcim nutne podmienky optimálnosti Kuhna-Tuckera, nie sú však globálnym minimom. Hoci Fletcherov algoritmus kvadratického programovania môže tieto prípady vyriešiť, je omnoho vhodnejšie aby boli vzťahy pre aktualizáciu upravené do takej podoby, kde bude počiatočná matica B_0 kladne definitná a všetky ďalšie B_k budú tiež pozitívne definitné. Tento krok môže byť uskutočnený vynásobením druhého a tretieho prvkov pravej strany rovnice (3.3.24) vhodne zvolenou konštantou θ , $0 \leq \theta \leq 1$, teda

$$\det B_k \geq \alpha \det B_{k-1}, \quad (3.3.25)$$

kde ako príklad $\alpha = 0.1$. Detailnejší popis tejto metódy je uvedená v práci Avriela [2].

3.4 Algoritmus redukovaných gradientov

Táto kapitola je zameraná na riešenie ohraničených nelineárnych úloh metódami, ktoré sú podobné postupu riešenia simplexovej metódy pre lineárne programovanie. Pre opakovanie najskôr uvedieme metódu redukovaných gradientov pre lineárne ohraničenia a potom jej zovšeobecnenie pre úlohy s nelineárnymi ohraničeniami. Uvažujme lineárne ohraničený problém

$$\min f(x) \quad (3.4.1)$$

vzhľadom na

$$Ax = b \quad (3.4.2)$$

$$x \geq 0, \quad (3.4.3)$$

kde f je spojitou diferencovateľná reálna funkcia, A je matica $m \times n$, b je vektor rozmeru m a $m \leq n$. Vektor premenných x môže byť rozdelený na dva subvektory $x = (x^B, x^N)^T$, kde $x^B = (x_1^B, \dots, x_m^B)^T$ je vektor bazických, alebo závislých premenných a $x^N = (x_1^N, \dots, x_{n-m}^N)^T$ je vektor nebazických, nezávislých premenných. Podobne aj matica A je rozdelená na $A = [B, C]$, kde predpokladáme, bez straty všeobecnosti, že prvých m riadkov A zodpovedá bazickým premenným. Ďalej predpokladáme, že B , submatica k matici A rozmeru $m \times m$ zodpovedajúca vektoru x^B je nesusingulárna. Potom môžeme zapísať

$$Bx^B + Cx^N = b \quad (3.4.4)$$

a

$$x^B = B^{-1}b - B^{-1}Cx^N. \quad (3.4.5)$$

Ďalej predpokladáme, že bázické premenné sú nedegeneratívne, teda platí, že $x^B > 0$. Nebázické premenné sú nazývané nezávislými, pokiaľ tým, že do nich priradíme číselné hodnoty, dostaneme jedinečné riešenie vzťahu (3.4.4).

Základná myšlienka metódy redukovaných gradientov spočíva v eliminácii x^B (ako funkcie x^N) pomocou (3.4.5) a uvažovanie optimalizačnej úlohy len v závislosti od x^N . Táto myšlienka bola uplatnená v diferenčných algoritmoch odvodených Wildom a Beightlerom pod pojmom “ohraničené deriváty,” vo Wolfovej metóde redukovaných gradientov a v Zangwillovej konvex-simplexovej metóde.

Zo vzťahu (3.4.5) dostaneme redukované gradienty $r \in R^{n-m}$ pomocou vzťahu

$$r(x^N) = \nabla_{x^N} f(x^B(x^N), x^N) - (B^{-1}C)^T \nabla_{x^B} f(x^B(x^N), x^N). \quad (3.4.6)$$

Keby sme sa teraz z aktuálnej hodnoty x^N mohli pohnúť v smere záporne redukovaného gradientu bez porušenia nezáporných ohraničení vektora x , prejavilo by sa zníženie funkčnej hodnoty f . Tento krok je vykonaný nasledovne:

Je daný prípustný x^k , pre $i = 1, \dots, n - m$ vypočítame

$$z_i^{N,k+1} = \begin{cases} 0 & \text{ak } x_i^{N,k} = 0 \text{ a } r_i(x^{N,k}) > 0 \\ -r_i(x^{N,k}) & \text{za iných okolností} \end{cases} \quad (3.4.7)$$

a nech

$$z^{B,k+1} = -B^{-1}Cz^{N,k+1}. \quad (3.4.8)$$

V prípade, že je vektor $z^{N,k+1}$ nulový, znamená to, že sa z daného bodu už nemáme kam posunúť, nemôžeme už teda nájsť lepšie riešenie a algoritmus končí. V opačnom prípade počítame $z^{k+1} = (z^{B,k+1}, z^{N,k+1})^T$. Ďalší bod, $x^{k+1} = (x^{B,k+1}, x^{N,k+1})^T$, je daný vzťahom

$$x^{k+1} = x^k + \alpha_{k+1}^* z^{k+1}, \quad (3.4.9)$$

kde α_{k+1}^* je dané vzťahmi

$$\alpha_{k+1}^1 = \max \{ \alpha_{k+1} : x^{B,k} + \alpha_{k+1} z^{B,k+1} \geq 0 \} \quad (3.4.10)$$

$$\alpha_{k+1}^2 = \max \{ \alpha_{k+1} : x^{N,k} + \alpha_{k+1} z^{N,k+1} \geq 0 \} \quad (3.4.11)$$

a

$$f(x^k + \alpha_{k+1}^* z^{k+1}) = \min_{\alpha_{k+1}} \{ f(x^k + \alpha_{k+1} z^{k+1}) : 0 \leq \alpha_{k+1} \leq \min(\alpha_{k+1}^1, \alpha_{k+1}^2) \}. \quad (3.4.12)$$

Ak $\alpha_{k+1}^* < \alpha_{k+1}^1$, nech je x^{k+1} definované vzťahom (2.4.9). Inak

$$x_l^{B,k} + \alpha_{k+1}^1 z_l^{B,k+1} = 0 \quad (3.4.13)$$

pre niektoré l , a x_l^B je vynechané z vektora bázických premenných na výmenu s najväčšou kladnou nebázickou premennou. Algoritmus končí keď $\|z^{k+1}\| \leq \epsilon$, kde $\epsilon > 0$ je malé číslo.

Metóda redukovaných gradientov môže zlyhať v konvergencii k bodu ktorý spĺňa podmienky Kuhna-Tuckera kôli jej klukateniu sa. Bolo navrhnutých niekoľko techník pre potlačenie tohto javu. Jedna z nich navrhuje zmenu pri výbere $z_i^{N,k+1}$ jej vynulovaním, v prípade že $x_i^{N,k} \leq \epsilon$ a $r_i(x^{N,k}) > 0$, kde $\epsilon > 0$ je nejaká vopred daná hodnota, čím nahradíme vzťah (3.4.7). Ďalší postup navrhol McCormick. Hodnota z^{k+1} je podľa neho vypočítaná na základe $\nabla f(x^{k-1})$ v prípade, že α_k^* nie je neohraničené minimum funkcie $f(x^{k-1} + \alpha_k z^k)$ pri z^k .

Ak je f lineárnou funkciou, metóda redukovaných gradientov sa stáva simplexovou metódou lineárneho programovania pomocou jednoduchej úpravy pravidiel pre zmenu báz. Ak nie sú prítomné obmedzenia, metóda redukovaných gradientov sa stáva technikou najstrmšieho poklesu a teda na ňu môžeme nahliadať ako na rozšírenie neohraničenej techniky. McCormick navrhol úpravu metódy redukovaných gradientov, ktorá sa dá považovať za rozšírenie DFP maticového algoritmu pre lineárne ohraňenia. Preukázal konvergenciu tejto metódy k bodu, ktorý vyhovuje podmienkam Kuhna-Tuckera v superlineárnej miere, za splnenia určitých miernych podmienok.

Abadie a Carpentier zovšeobecniili metódu redukovaných gradientov na úlohy s nelineárnymi ohraňeniami. Uvažujme úlohu s ohraňenou premennou

$$(\text{NOU}) \quad \min f(x) \quad (3.4.14)$$

vzhľadom na

$$h_i(x) = 0, \quad i = 1, \dots, m \quad (3.4.15)$$

$$\beta \geq x \geq \alpha, \quad (3.4.16)$$

kde f, h_1, h_2, \dots, h_m majú byť spojito diferencovateľné funkcie $x \in R^n$ a $m \leq n$.

Každý prípustný bod x^0 je považovaný za nedegeneratívny, čo znamená že x^0 môže byť rozdelený na dva subvektory, $x = (x^{B,0}, x^{N,0})^T$ kde $x^{B,0} \in R^m, x^{N,0} \in R^{n-m}$. Podobne aj $\alpha = (\alpha^{B,0}, \alpha^{N,0})^T, \beta = (\beta^{B,0}, \beta^{N,0})^T$, zároveň platí

$$\beta^{B,0} > x^{B,0} > \alpha^{B,0} \quad (3.4.17)$$

a vektory

$$\nabla_{x^B} h_i(x^0) = \left[\frac{\partial h_i(x^0)}{\partial x_1^{B,0}}, \dots, \frac{\partial h_i(x^0)}{\partial x_m^{B,0}} \right]^T, \quad i = 1, \dots, m \quad (3.4.18)$$

sú lineárne nezávislé. Matica $\Delta^B(x^0)$ rozmeru $m \times m$ ktorej riadky sú $\nabla_{x^B} h_i(x^0)$ je teda nesingulárna. Jednoduchým spôsobom sa dá overiť, že nutné podmienky optimálnosti Kuhna-Tuckera pre x^0 v úlohe (NOU) potvrdzujú existenciu vektorov $\mu^0 \in R^m, \lambda^0 \in R^{n-m}$ takých, že

$$\nabla_{x^N} f(x^0) - \sum_{i=1}^m \mu_i^0 \nabla_{x^N} h_i(x^0) - \lambda^0 = 0 \quad (3.4.19)$$

$$\nabla_{x^B} f(x^0) - \sum_{i=1}^m \mu_i^0 \nabla_{x^B} h_i(x^0) = 0 \quad (3.4.20)$$

kde

$$\lambda_j^0 \geq 0 \quad \text{ak} \quad x_j^{N,0} = \alpha_j^{N,0} \quad (3.4.21)$$

$$\lambda_j^0 = 0 \quad \text{ak} \quad \beta_j^{N,0} > x_j^{N,0} > \alpha_j^{N,0} \quad (3.4.22)$$

$$\lambda_j^0 \leq 0 \quad \text{ak} \quad x_j^{N,0} = \beta_j^{N,0}. \quad (3.4.23)$$

Teda

$$\mu^0 = [\Delta^B(x^0)]^{-1} \nabla_{x^B} f(x^0) \quad (3.4.24)$$

a

$$\lambda^0 = \nabla_{x^N} f(x^0) - \Delta^N(x^0) [\Delta^B(x^0)]^{-1} \nabla_{x^B} f(x^0), \quad (3.4.25)$$

kde $\Delta^N(x^0)$ je matica rozmeru $(n - m) \times m$ matica, ktorej riadky sú tvorené vektormi $\nabla_{x^N} h_i(x^0)$. Analogicky k (3.4.6), λ^0 je redukovaným gradientom ak sú ohraničenia lineárne. Pre každé prípustné x^0 môžeme teraz pomocou danej rovnice vypočítať λ^0 a v prípade, že dosiahnuté λ^0 spĺňa tiež podmienky (3.4.21) až (3.4.23), potom sú nutné podmienky optimálnosti danej (NOU) funkcie splnené v bode x^0 . Zovšeobecnená metóda redukovaných gradientov je založená na iteratívnom presune z prípustného x^0 do prípustného x^1, x^2, \dots pokiaľ nie je dosiahnutý bod x^l , kde λ^l vypočítaná z (3.4.25) spĺňa podmienky (3.4.21) až (3.4.23). Predpokladajme, že máme prípustné x^k a zodpovedajúce λ^k nespĺňa tieto tri vzťahy. Je potom potrebné zmeniť hodnotu nebázického vektora tak, že

$$x^{N,k+1} = x^{N,k} + \theta_{k+1} z^{k+1}, \quad \theta_{k+1} \geq 0, \quad (3.4.26)$$

kde $z_j^{k+1} = 0$ ak

$$x_j^{N,k} = \alpha_j^{N,k} \text{ a } \lambda_j^k > 0, \quad \text{alebo} \quad x_j^{N,k} = \beta_j^{N,k} \text{ a } \lambda_j^k < 0 \quad (3.4.27)$$

Inak buď položíme $z_j^{x+1} = -\lambda_j^k$ alebo $z_j^{x+1} = 0$, vzhľadom na pravidlá závisiace na partikulárnej verzii použitej metódy.

1. Pri GRG verzii položíme $z_j^{x+1} = -\lambda_j^k$ vždy, keď neplatí (3.4.27). Táto voľba je rovnaká, ako v prípade, kedy sú ohraničenia lineárne.

2. V prípade GRGS verzie najskôr nájdeme index s pomocou

$$|\lambda_s^x| = \max_j |\lambda_j^k|, \quad (3.4.28)$$

kde indexy j majú väčší rozsah ako tie, pre ktoré neboli splnené (3.4.27). Potom nech

$$z_j^{k+1} = \begin{cases} 0 & j \neq s \\ -\lambda_s^k & j = s \end{cases} \quad (3.4.29)$$

Táto verzia sa zhoduje so simplexovou metódou, v prípade že optimalizačným problémom je lineárna úloha.

3. V GRGC verzii cyklicky určujeme $z_j^{x+1} = -\lambda_j^k$. Cyklus pozostáva z n iterácií. Pre $k = 1, \dots, n$ určíme

$$z_j^{k+1} = \begin{cases} 0 & j \neq k \\ -\lambda_s^k & j = k \end{cases} \quad (3.4.30)$$

za predpokladu, že k nie je indexom bázekej premennej a že neplatí (3.4.27). V takom prípade je iterácia vynechaná. V momente keď $k = n$ sa vrátíme ku $k = 1$, a tak ďalej.

Keď sme už za použitia jednej z metód určili $z^{k+1} \in R^{n-m}$, vezmeme pokusnú hodnotu dĺžky kroku θ_{k+1} , kde $\theta_{k+1}^1 > 0$ a definujeme

$$\tilde{x}_j^{N,k} = \begin{cases} \beta_j^{N,k} & \text{ak } x_j^{N,k} + \theta_{k+1}^1 z_j^{k+1} > \beta_j^{N,k} \\ \alpha_j^{N,k} & \text{ak } x_j^{N,k} + \theta_{k+1}^1 z_j^{k+1} < \alpha_j^{N,k} \\ x_j^{N,k} + \theta_{k+1}^1 z_j^{k+1} & \text{za iných okolností.} \end{cases} \quad (3.4.31)$$

Pre dosiahnutie prípustnosti riešime rovnicu m nelineárnych rovníc

$$h_i(x^B, \tilde{x}^{N,k}) = 0, \quad i = 1, \dots, m \quad (3.4.32)$$

pre m premenných x_1^B, \dots, x_m^B iteratívnou procedúrou podobnou Newtonovej metóde. V l -tej iterácii Newtonovej metódy potom dostaneme vzťah

$$x^{B,l+1} = x^{B,l} - [\Delta^N(x^{B,l}, \tilde{x}^{N,k})]^{-1} h(x^{B,l}, \tilde{x}^{N,k}), \quad (3.4.33)$$

kde $h = (h_1, \dots, h_m)^T$ a $x^{B,l} = x^{B,k}$ pre $l = 0$. Vzťah (2.4.33) sa používa, pokiaľ nenastane jeden z nasledovných prípadov:

1. Miera $\|h(x^{B,l}, \tilde{x}^{N,k})\|$ rastie v niekoľkých po sebe idúcich iteráciách. V takom prípade v určitej miere znížime θ_{k+1}^l , prepočítame (3.4.31) a vrátime sa k riešeniu (3.4.32).

2. Dostaneme

$$f(x^{B,l}, \tilde{x}^{N,k}) > f(x^{B,l-1}, \tilde{x}^{N,k}). \quad (3.4.34)$$

Korekcia prebieha rovnako ako v prípade 1.

3. Pre niektoré l leží bod $(x^{B,l}, \tilde{x}^{N,k})$ mimo ohraničení definovaných v (3.4.16). Nájďme bod $\tilde{x}^{B,l}$ ležiaci na úsečke medzi $x^{B,l-1}$ a $x^{B,l}$ pre ktorý $\tilde{x}_r^{B,l} = \alpha_r^{B,l}$ alebo $\tilde{x}_r^{B,l} = \beta_r^{B,l}$ pre nejaké r a prevedieme zmenu v aktuálnej báze tým že $x_r^{B,l}$ nahradíme $x_s^{N,l}$,

kde index s je daný pravidlom podobným tomu, ktoré sa používa pri simplexovej metóde. Môžeme napríklad uplatniť vzťah

$$|\Delta^N(\tilde{x})_s(\Delta^B(\tilde{x}))_r^{-1}|v_s = \max_p\{|\Delta^N(\tilde{x})_p(\Delta^B(\tilde{x}))_r^{-1}|v_p\} \quad (3.4.35)$$

kde $\Delta^N(\tilde{x})_p$ je riadkový vektor $\left(\frac{\partial h_1(\tilde{x})}{\partial x_p^N}, \dots, \frac{\partial h_m(\tilde{x})}{\partial x_p^N}\right)$ vypočítaný pre $\tilde{x} = (\tilde{x}^{B,l}, \tilde{x}^{N,k})$, $(\Delta^B(\tilde{x}))_r^{-1}$ je r -tý riadok matice $(\Delta^B(\tilde{x}))^{-1}$ a

$$v_p = \min\{(\tilde{x}_p^{N,k} - \alpha_p^{N,k}), (\beta_p^{N,k} - \tilde{x}_p^{N,k})\}. \quad (3.4.36)$$

Riešenie (2.2.31) sa teraz realizuje s novou bázou.

4. Iterácie konvergujú, teda

$$\|h(x^{B,l^*}, \tilde{x}^{N,k})\| \leq \epsilon \quad (3.4.37)$$

pre nejaké l^* , kde ϵ predstavuje malé kladné číslo. Ak

$$\beta^{B,l^*} > x^{B,l^*} > \alpha^{B,l^*}, \quad (3.4.38)$$

položíme $x^{k+1} = (x^{B,l^*}, x^{N,k})$ a nájdeme nový smer hľadania. Ak je niektorý z komponentov r z x^{B,l^*} na dolnej alebo hornej hranici, zmeníme bázu rovnako ako v prípade 3. Výpočty končia nájdením bodu spĺňajúceho nutné podmienky optimálnosti.

4. Tvorba procedúry pre algoritmus redukovaných gradientov

Problém hľadania optimálneho portfólia s rizikovými aktívami je úlohou s lineárnymi ohraničeniami, v ktorej je účelová funkcia nelineárnou funkciou. Práve tieto vlastnosti sú dôležité pri voľbe vhodnej aproximačnej metódy pre jej riešenie. Spomedzi metód popísaných v kapitole 3 sa ako najvhodnejšia javí algoritmus redukovaných gradientov, popísaná v 3.4. Pomocou programovacieho jazyka VBA sme vytvorili procedúru *redgrad*, ktorá umožňuje riešenie jednoduchých nelineárnych úloh použitím algoritmu redukovaných gradientov.

Prostredie tabuľkového procesora MS Excel je výkonným nástrojom pre jednoduchú prácu s dátami vo forme matíc či vektorov, poskytuje tak vhodné podmienky nie len pre výpočet riešenia zadaných úloh, ale aj analýzu a ďalšie spracovanie získaných výsledkov. Primárnym cieľom, s ktorým bola procedúra *redgrad* navrhovaná je nielen nájsť riešenie zadaného optimalizačného problému, ale aj popísať priebeh algoritmu redukovaných gradientov. Okrem hodnôt koreňov a účelovej funkcie ukazuje algoritmus aj hodnoty čiastkových výsledkov, ako sú hodnoty redukovaného gradientu, smeru kroku, dĺžky kroku, či iných parametrov popísaných v podkapitole 3.4. Procedúra umožňuje užívateľovi nahliadnuť do výpočtového procesu v každej jednej iterácii, čím umožňuje hlbšie pochopenie algoritmu. Kompletný zdrojový kód tejto procedúry s dopĺňujúcimi poznámkami je uvedený v prílohe č. 1. V prílohe č. 2 je uvedený modelový príklad, slúžiaci ako návod pre použitie algoritmu. V tejto časti práce sa budeme ďalej venovať detailnejšiemu popisu procedúry *redgrad* a postupu jej tvorby.

4.1 Návrh vstupov a výstupov pre procedúru

Budeme predpokladať, že optimalizačný problém, ktorý chceme riešiť zodpovedá (3.4.1), (3.4.2) a (3.4.3). Nakoľko je procedúra zameraná predovšetkým na popis priebehu algoritmu, príprava vstupných údajov je ponechaná na užívateľovi. Pri volaní procedúry je potrebné, aby užívateľ zadal gradient účelovej funkcie ako vstupný parameter. Príklad formulácie vstupných údajov je uvedený na Obr. 4.1.1.

	A	B	C	D	E	F	G	H
1		x1	x2	x3	x4	x5		
2								
3								
4	UF							
5								
6	DF							
7								
8				A			b	
9								
10								
11								
12	az							
13								

Obr. 4.1.1 – Formulácia vstupných údajov pre úlohu 5×2

Podobne ako je tomu pri volaní *solveru* v MS Excel, aj procedúra *redgrad* vyžaduje, aby pred jej volaním užívateľ naformuloval vzťahy medzi bunkami. Konkrétne ide o vzťahy medzi hodnotou účelovej funkcie, premennými a posunom, v tvare

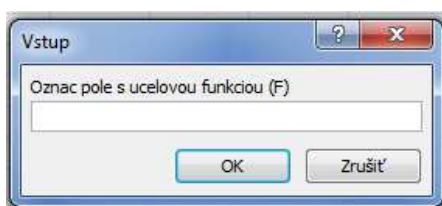
$$f(x) = x^{k-1} + \alpha_{k-1}^* z^{k-1} \quad (4.1.1)$$

a vyjadrenie gradientu účelovej funkcie pomocou buniek, v ktorých sú umiestnené premenné.

Tab. 4.1.1 – popis polí pre vstup

označenie	pole	význam
x1 x2 x3 x4 x5	B2:F2	x_j
UF	B4	$f(x)$
DF	B6:F6	$\nabla f(x)$
A	B9:F10	A
b	G9:G10	b
az	B12:F12	$\alpha_k^* z^k$

Procedúra po spustení vyzve užívateľa prostredníctvom série dialógových okien aby zadal všetky vstupné parametre tak, ako sú uvedené v Tab. 4.1.1.



Obr. 4.1.2 Príklad dialógového okna

Sled príkazov, slúžiaci na vymedzenie polí pre výstup neboli bližšie popísané, nakoľko nijak priamo nesúvisia s výpočtom algoritmu redukovaných gradientov. Výstup algoritmu má tvar vyobrazený na Obr. 4.1.3.

	A	B	C	D	E	F	G
18	k						
19							
20	B						
21							
22							
23	N						
24							
25							
26	BDF						
27							
28	r						
29							
30	zj						
31							
32	zN						
33							
34	zB						
35							
36	zk						
37							
38	alfa						
39							
40	alfak						
41							

Obr. 4.1.3 – Formulácia výstupných údajov pre úlohu 5×2

Tab. 4.1.1 – Popis polí pre výstup

označenie	pole	význam
k	B18	k
B	B20:C21	B
N	B23:D24	C
BDF	B26:C26	∇f^B
r	B28:F28	$r(x)$
zj	B30:F30	z_j
zN	B32:D32	z^N
zB	B34:C34	z^B
zk	B36:F36	z^k
alfa	B38	α_{max}
alfak	B40	α^k

4.2 Popis tvorby zdrojového kódu

Ešte pred začiatkom výpočtového procesu musí algoritmus definovať pomocné premenné, ktoré budú použité pri ďalších výpočtoch. Týmito parametrami sú počet premenných v účelovej funkcii (l), počet ohraňení (m), celkový počet premenných (n), súradnice bunky pre výpis výstupu (V_x, V_y), počet iterácií (k) a konštantu predstavujúcu najväčšie dosiahnuteľné číslo pri výpočtoch (inf). Dá sa povedať, že hodnota ϵ z kapitoly 3.4 predstavujúca veľmi malé číslo je v tejto procedúre definovaná ako $1/inf$.

```
l = az.Count
m = B.Count
n = X.Count
Vx = vyp.Row
Vy = vyp.Column
k = 0
inf = 1000000000
```

Výpočtový cyklus (iterácia) začína zápisom čísla iterácie do príslušného pola.

```
For t = 1 To ite
k = k + 1
Cells(Vx, Vy + 1) = k
```

Indexy n premenných je potrebné rozdeliť na dve časti, množinu indexov báziických premenných (BI) s rozmerom m a množinu indexov nebáziických premenných (NI) s rozmerom $n-m$. Pre zjednodušenie výpočtu budú v množine BI uložené indexy premenných x s najvyššou hodnotou, v množine NI indexy s najnižšou hodnotou. Pri tomto postupe bola použitá pomocná premenná $DFset$ ako náhrada za X . Hľadanie indexu BI bolo navrhnuté v tvare

```
For i = 0 To 100
    DFset(i) = -inf
Next i

For j = 1 To n
    DFset(j) = X(j)
Next j

For i = 1 To m
    For j = 1 To n
        If (DFset(j) = Application.Max(DFset)) Then
            BI(i) = j
            DFset(j) = -inf
            Exit For
        End If
    Next j
Next i
```

a pre IN v tvare

```
For i = 0 To 100
    DFset(i) = inf
Next i

For j = 1 To n
    premennej
    DFset(j) = X(j)
Next j

For i = 1 To n - m
    For j = 1 To n
        If (DFset(j) = Application.Min(DFset)) Then
            NI(i) = j
            DFset(j) = inf
            Exit For
        End If
    Next j
Next i
```

Následne boli indexy BI zoradené vzostupne.

```
For i = 1 To m
    For j = 1 To m
        If (BI(i) < (BI(j))) Then
            p = BI(i)
            q = BI(j)
            BI(i) = q
            BI(j) = p
            Exit For
        End If
    Next j
Next i
```

Obdobný postup bol uplatnený aj pre hľadanie indexov NI .

```
For i = 1 To n - m
    For j = 1 To n - m
        If (NI(i) < (NI(j))) Then
            p = NI(i)
            q = NI(j)
            NI(i) = q
            NI(j) = p
            Exit For
        End If
    Next j
Next i
```

Pomocou získaných indexov bola matica A (A) rozdelená na B (RB) a C (RN). Z gradientu $\nabla f(x)$ (DF) bol vyčlenený vektor $\nabla f(x)^B$ ($RBDF$).

```

For i = 1 To m
  RBDF(i) = DF(BI(i))
  For j = 1 To m
    RB(i, j) = A(i, BI(j))
  Next j

  For j = 1 To n - m
    RN(i, j) = A(i, NI(j))
  Next j
Next i

```

Z dôvodu zjednodušenia výpočtov v priebehu procedúry a redukcie počtu premenných bol vzťah (3.4.6) pre výpočet redukovaného gradientu upravený na

$$r^k = \nabla f(x^k) - \nabla f(x^k)^B B^{-1} A \quad (4.2.1)$$

Keďže pri výpočte dochádza k odčítavaniu matíc, procedúra musí vytvoriť zápornú jednotkovú maticu (*RNEG*).

```

For i = 1 To n
  For j = 1 To n
    If (i = j) Then
      RNEG(i, j) = -1
    Else
      RNEG(i, j) = 0
    End If
  Next j
Next i

```

S využitím funkcie *MMult* následne dôjde k výpočtu redukovaného gradientu *r* (*Rr*).

```

Rr = Application.MMult(RBDF, Application.MMult(Application.
MInverse(RB), A))
Rr = Application.MMult(Rr, RNEG)
For i = 1 To n
  Rr(i) = Rr(i) + DF(i)
Next i

```

Na základe podmienok (3.4.7) sú vypočítané hodnoty z_j (Rzj).

```

For j = 1 To n
  If Rr(j) > 0 Then
    Rzj(j) = (-X(j)) * Rr(j)
  Else
    Rzj(j) = -Rr(j)
  End If
Next j

```

Ako je známe z popisu metódy, v prípade že je vektor z_j nulový, bolo v predošlej iterácii nájdené optimálne riešenie a algoritmus končí. V takom prípade sú všetky ďalšie nedopočítané výsledky nahradené označením *N/A*.

```

If Application.Sum(Rzj) < 1 / inf Then
  If Application.Sum(Rzj) > -1 / inf Then
    RzN = "N/A"
    RzB = "N/A"
    Rzk = "N/A"
    Ralfa = "N/A"
    Ralfak = "N/A"
    az = "N/A"
    Exit For
  End If
End If

```

Vektor z^N je vyčlenený zo z_j pomocou indexu NI .

```

For j = 1 To n - m
  RzN(j) = Rzj(NI(j))
Next j

```

Podľa vzťahu (3.4.8) je potom pomocou funkcií *MMult* a *Transpose* vypočítaný vektor z^B .

```

RNT = Application.Transpose(RN)
RzB = Application.MMult(RzN, RNT)
RzB = Application.MMult(RzB, Application.MMult(RNEG2, Application.
MInverse(RB)))

```

Spojením vektorov z^N a z^B pomocou indexov NI a BI potom dostaneme smer posunu z^k .

```

For i = 1 To m
  Rzk(BI(i)) = RzB(i)
Next i
For j = 1 To n - m
  Rzk(NI(j)) = RzN(j)
Next j

```

Ďalším krokom v algoritme je výpočet dĺžky kroku v smere z^k . Aby sme našli hornú hranicu pre dĺžku kroku α_{max} popísaný vo vzťahu (3.4.12) ako $\min(\alpha_{k+1}^1, \alpha_{k+1}^2)$ použijeme špeciálny vzťah, vhodný pre zjednodušenie výpočtu

$$\alpha_{max} = \begin{cases} \min_{1 \leq j \leq n} \left\{ \frac{-x_{jk}}{z_{jk}} : z_{jk} < 0 \right\} & ak \quad z_k < 0 \\ \infty & ak \quad z_k \geq 0 \end{cases} \quad (4.2.2)$$

vyjadrený kódom ako

```

Ralfa = inf
For j = 1 To n
  If Rzk(j) < 0 Then
    If (-X(j) / Rzk(j)) < Ralfa Then
      Ralfa = (-X(j) / Rzk(j))
    End If
  End If
Next j

```

Pre získanie hodnoty α_k^* ako optimálneho riešenia úlohy (3.4.12) používame *solver* zabudovaný v MS Excel. Procedúra najskôr vytvorí vzťah medzi polom *az*, *Ralfak* a *Rzk*, pomenuje polia a následne vykoná sled príkazov pre nájdenie optimálneho riešenia.

```
adr = Cells(14, 1).Address
For j = 1 To 1
    az(j).FormulaArray = "=" & Ralfak.Address & "*" & Rzk(j).
Address & ""
Next j
```

```
F.Name = "SF"
Ralfak.Name = "Salfak"
Ralfa.Name = "Salfa"
```

```
solverreset
SolverOk SetCell:="SF", MaxMinVal:=2, ValueOf:="0", ByChange:=
"Salfak"
SolverAdd CellRef:="Salfak", Relation:=1, FormulaText:="Salfa"
SolverAdd CellRef:="Salfak", Relation:=3, FormulaText:="0"
SolverOk SetCell:="SF", MaxMinVal:=2, ValueOf:="0",
ByChange:="Salfak"
SolverSolve True
```

Iterácia končí výpočtom nového riešenia x^k .

```
For i = 1 To n
    X(i) = X(i) + (Ralfak * Rzk(i))
Next i
```

5. Riešenie hypotetického príkladu

Ako sme uviedli v kapitole 3, poznáme niekoľko metód, ktorými môžeme nájsť optimálne riešenie úlohy nelineárneho programovania, akou je aj úloha pre výber portfólia popísaná v podkapitole 1.5. V tejto kapitole popíšeme spôsob tvorby takejto úlohy na hypotetickom príklade štyroch cenných papierov. Následne potom nájdeme optimálne riešenie tejto úlohy pomocou výpočtového softvéru GAMS. V podkapitole 5.3 prevedieme výpočet tohto príkladu metódou redukovaných gradientov s využitím procedúry *redgrad*.

5.1 Formulácia úlohy

Uvažujme problém hľadania optimálneho portfólia, pri ktorom sa investor rozhoduje, v akom pomere rozdelí svoje investície medzi 4 cenné papiere (*A*, *B*, *C*, *D*). Investor chce pri minimálnom riziku dosiahnuť výnos aspoň 15%, pri čom z dôvodu diverzifikácie nechce do jedného aktíva investovať viac ako polovicu zdrojov. Známe sú výnosy jednotlivých cenných papierov r_j^t v období 10 rokov.

Tab. 5.1.1 – Výnosy cenných papierov v čase

t	A	B	C	D
1	1.04	1.64	1.28	1.04
2	1.13	0.96	1.32	1.03
3	1.16	0.98	0.97	1.06
4	1.12	1.5	1.29	1.52
5	1.11	0.88	1.31	1.04
6	1.15	1.73	1.25	1.05
7	1.04	1.34	1.17	1.19
8	1.14	0.77	0.98	1.04
9	1.05	0.88	0.94	1.45
10	1.06	1.82	1.19	1.08

Na základe údajov z Tab. 5.1.1 sme určili priemerné výnosy $\mu = (1,1; 1,25; 1,17; 1,15)$.

Na základe (1.5.1) sme vypočítali variačno-kovariačnú maticu *S*.

Tab. 5.1.2 – Variačno-kovariačná matica *S*

0.00227	-0.00519	-0.00032	-0.00201
-0.00519	0.15991	0.02577	0.00069
-0.00032	0.02577	0.02271	-0.00406
-0.00201	0.00069	-0.00406	0.03358

Dosadením parametrov úlohy do (1.5.7), (1.5.8) a (1.5.9) sme dostali formuláciu hypotetickej úlohy v tvare

$$\begin{aligned} \min f(x) = & 0,00227x_1^2 + 0,15911x_2^2 + 0,02271x_3^2 + 0,03358x_4^2 - 0,01038x_1x_2 \\ & - 0,00064x_1x_3 - 0,00402x_1x_4 + 0,05154x_2x_3 + 0,00138x_2x_4 \\ & - 0,00812x_3x_4 \end{aligned} \quad (5.1.1)$$

pri ohraničeníach

$$1,1x_1 + 1,25x_2 + 1,17x_3 + 1,15x_4 \geq 1,15 \quad (5.1.2)$$

$$x_1 + x_2 + x_3 + x_4 = 1 \quad (5.1.3)$$

kde $0,5 \geq x_1, x_2, x_3, x_4 \geq 0$.

5.2 Riešenie úlohy pomocou GAMS

Cieľom tvorby algoritmu v GAMS bolo nájsť optimálne portfólio na základe znalosti priemerných výnosov μ a variačno-kovariačnej matice S . Takéto riešenie bolo dosiahnuté zadáním nasledovného kódu:

```
sets          i / 1*4/
              alias(i,j);

parameters r(i) / 1 1.1, 2 1.25, 3 1.17, 4 1.15 /;

table S(i,j)
      1          2          3          4
1    0.00227    - 0.00519    -0.00032    -0.00201
2    -0.00519    0.15991     0.02577     0.00069
3    -0.00032    0.02577     0.02271     -0.00406
4    -0.00201    0.00069     -0.00406     0.03358

variables x(i), OBJ;

EQUATIONS OHR1,OHR2,QP, DH, HH;
QP.. OBJ =E= (SUM (j, SUM(i, (S(i,j)*x(i)*x(j)))));
OHR1.. SUM(i,(r(i)*x(i))) =G= 1.15;
OHR2.. SUM(i, x(i)) =E= 1;
DH(i).. x(i) =G= 0;
HH(i).. x(i) =L= 0.5;

MODEL PORTQP/QP,OHR1,OHR2, DH, HH/;
SOLVE PORTQP USING NLP MINIMIZING OBJ;
```

Výsledky riešenia sú takéto (uvádzame len časť výstupu):

MODEL STATISTICS

BLOCKS OF EQUATIONS	5	SINGLE EQUATIONS	11
BLOCKS OF VARIABLES	2	SINGLE VARIABLES	5
NON ZERO ELEMENTS	21	NON LINEAR N-Z	4
DERIVATIVE POOL	10	CONSTANT POOL	26
CODE LENGTH	66		

S O L V E S U M M A R Y

MODEL	PORTQP	OBJECTIVE	OBJ
TYPE	NLP	DIRECTION	MINIMIZE
SOLVER	CONOPT	FROM LINE	26

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 2 Locally Optimal
 **** OBJECTIVE VALUE 0.0065

RESOURCE USAGE, LIMIT	0.016	1000.000
ITERATION COUNT, LIMIT	9	2000000000
EVALUATION ERRORS	0	0

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU QP	.	.	.	1.000
---- EQU OHR1	1.150	1.150	+INF	0.274
---- EQU OHR2	1.000	1.000	1.000	-0.302

---- EQU DH

	LOWER	LEVEL	UPPER	MARGINAL
1	.	0.299	+INF	.
2	.	0.075	+INF	.
3	.	0.372	+INF	.
4	.	0.255	+INF	EPS

---- EQU HH

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	0.299	0.500	EPS
2	-INF	0.075	0.500	.
3	-INF	0.372	0.500	.
4	-INF	0.255	0.500	.

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	0.299	+INF	.
2	-INF	0.075	+INF	.
3	-INF	0.372	+INF	.
4	-INF	0.255	+INF	.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR OBJ	-INF	0.007	+INF	.

Z uvedeného výstupu vyplýva, že bolo nájdené optimálne riešenie úlohy po 9 iteráciach. Hodnota účelovej funkcie nájdeného riešenia $f(x^*) = 0.0065$ s koreňmi $x^* = (0,299; 0,075; 0,372; 0,255)$.

Investor dosiahne optimálne rozdelenie portfólia ak investuje 29,9% zdrojov do aktíva A, 7,5% do aktíva B, 37,2% do aktíva C a 25,5% do aktíva D, pri čom dosiahne výnos 15% s mierou rizika 0,65%.

5.3 Riešenie úlohy pomocou procedúry redgrad

Pre riešenie úlohy výberu optimálneho portfólia procedúrou *redgrad* potrebujeme formuláciu úlohy v tvare (5.1.1). Táto procedúra však pracuje iba s ohraničeniami rovnosti, je teda potrebné, aby sme ohraničenie (5.1.2) upravili na

$$1,1x_1 + 1,25x_2 + 1,17x_3 + 1,15x_4 - x_5 = 1,15. \quad (5.3.1)$$

Ohraničenie (5.1.3) vyhovuje požiadavkám procedúry, môže teda zostať v pôvodnom tvare. Z riešenia dosiahnutého v GAMS vidíme, že ohraničenie $x_1, x_2, x_3, x_4 \leq 0,5$ nemá vplyv na riešenie úlohy. Pre zjednodušenie ho teda z úlohy vynecháme. Ešte pred volaním výpočtového algoritmu je potrebné určiť gradient účelovej funkcie

$$\begin{aligned} \nabla f(x) = & (0,00454x_1 - 0,01038x_2 - 0,00064x_3 - 0,00402x_4; \\ & -0,01038x_1 + 0,31982x_2 + 0,05154x_3 + 0,00138x_4; \\ & -0,00064x_1 + 0,05154x_2 + 0,04542x_3 - 0,00812x_4; \\ & -0,00402x_1 + 0,00138x_2 - 0,00812x_3 + 0,06716x_4). \end{aligned} \quad (5.3.2)$$

Účelová funkcia musí byť ďalej doplnená podľa vzťahu (4.1.1) aby mohli byť neskôr vytvorené vzťahy medzi jednotlivými vstupnými množinami.

$$\begin{aligned} f'(x^{k+1}) = & 0,00227(x_1^2 + \alpha_k^* z_1^k) + 0,15911(x_2^2 + \alpha_k^* z_2^k) + 0,02271(x_3^2 + \alpha_k^* z_3^k) \\ & + 0,03358(x_4^2 + \alpha_k^* z_4^k) - 0,01038(x_1 + \alpha_k^* z_1^k)(x_2 + \alpha_k^* z_2^k) \\ & - 0,00064(x_1 + \alpha_k^* z_1^k)(x_3 + \alpha_k^* z_3^k) - 0,00402(x_1 + \alpha_k^* z_1^k)(x_4 \\ & + \alpha_k^* z_4^k) + 0,05154(x_2 + \alpha_k^* z_2^k)(x_3 + \alpha_k^* z_3^k) + 0,00138(x_2 \\ & + \alpha_k^* z_2^k)(x_4 + \alpha_k^* z_4^k) - 0,00812(x_3 + \alpha_k^* z_3^k)(x_4 + \alpha_k^* z_4^k) \end{aligned} \quad (5.3.3)$$

Ako východiskový prípustný bod sme si zvolili $x^0 = (0,1; 0,4; 0,4; 0,1; 0,043)$. Zadanie vstupných parametrov do pracovného hárka MS Excel je vyobrazené na Obr. 5.2.1.

	A	B	C	D	E	F	G	H
1		x1	x2	x3	x4	x5		
2		0.1	0.4	0.4	0.1	0.043		
3								
4	UF							
5								
6	DF							
7								
8		A					b	
9	dg1	1.1	1.25	1.17	1.15	-1	1.15	
10	dg2	1	1	1	1	0	1	
11								
12	az							
13								

Obr. 5.3.1 – Formulácia úlohy pre hľadanie optimálneho portfólia

Medzi bunkami boli podľa vzťahov (5.3.2) a (5.3.3) vytvorené väzby:

$$B4 = ((0.00227*(B2+B12)^2)-(0.01038*(B2+B12)*(C2+C12))-(0.00064*(B2+B12)*(D2+D12))-(0.00402*(B2+B12)*(E2+E12))+(0.15911*(C2+C12)^2)+(0.05154*(C2+C12)*(D2+D12))+(0.00138*(C2+C12)*(E2+E12))+(0.02271*(D2+D12)^2)-(0.00812*(D2+D12)*(E2+E12))+(0.03358*(E2+E12)^2))$$

$$B6 = (0.00454*B2)-(0.01038*C2)-(0.00064*D2)-(0.00402*E2)$$

$$C6 = -(0.01038*B2)+(0.31982*C2)+(0.05154*D2)+(0.00138*E2)$$

$$D6 = -(0.00064*B2)+(0.05154*C2)+(0.04542*D2)-(0.00812*E2)$$

$$E6 = -(0.00402*B2)+(0.00138*C2)-(0.00812*D2)+(0.06716*E2)$$

Úloha je teraz pripravená na zavolanie procedúry *redgrad*. Vstupné parametre sme vložili prostredníctvom dialógových okien tak, ako to popisuje Tab. 5.3.1.

Tab. 5.3.1 – Zadávanie vstupov

požiadavka	vstup
(F)	"B4"
(DF)	"B6:E6"
(A)	"B9:F10"
(B)	"G9:G10"
(X)	"B2:F2"
(az)	"B12:E12"
výpis	"A18"

Po zadání počtu iterací $k = 100$ sme prostredníctvom procedúry *redgrad* dosiahli výsledok $x^{100} = (0,210; 0,021; 0,520; 0,367; 0,020)$. Hodnota účelovej funkcie nájdeného riešenia je $f(x^{100}) = 0.0094$. Kompletný výstup algoritmu po 100 iteráciách ukazuje Obr. 5.3.2 a Obr. 5.3.3.

	A	B	C	D	E	F	G	H
1		x1	x2	x3	x4	x5		
2		0.210412836	0.021302508	0.520252935	0.366586	0.020342986		
3								
4	UF	0.009443423						
5								
6	DF	-0.001072483	0.031948608	0.021616478	0.019579	0		
7								
8		A					b	
9	dg1	1.1	1.25	1.17	1.15	-1	1.15	
10	dg2	1	1	1	1	0	1	
11								
12	az	0.000394328	7.22682E-05	-0.008063484	0.008806			
13								

Obr. 5.3.2 – Výsledky po 100 iteráciách

	A	B	C	D	E	F	G
18	k	100					
19							
20	B	1.17	1.15				
21		1	1				
22							
23	N	1.1	-1	1.25			
24		1	0	1			
25							
26	BDF	0.022050757	0.018923561				
27							
28	r	-0.012148853	-0.002226514	1.35308E-16	1.56E-16	0.156359797	
29							
30	zj	0.012148853	0.002226514	-7.14857E-17	-5.6E-17	-0.003197051	
31							
32	zN	0.012148853	-0.003197051	0.002226514			
33							
34	zB	-0.248428222	0.271317088				
35							
36	zk	0.012148853	0.002226514	-0.248428222	0.271317	-0.003197051	
37							
38	alfa	2.126636077					
39							
40	alfak	0.032458002					
41							

Obr. 5.3.3 – Výsledky po 100 iteráciách

Záver

Práca poskytla prehľad niektorých metód pre riešenie úloh nelineárnej optimalizácie. V metodologickej časti bola popísané projekčné metódy pre nelineárne ohraničenia, metóda prípustných smerov pre nelineárne ohraničenia, metódy aproximačného programovania a algoritmus redukovaných gradientov. V popise bol uvedený základný princíp týchto metód, ako aj niektoré ich rozšírenia. Pre nadobudnutie lepšieho prehľadu o možnostiach uplatnenia nelineárnych optimalizačných algoritmov bolo uvedených niekoľko príkladov praktických aplikácií z praxe. Ukázali sme, že je možné tieto metódy využiť v modeloch optimálneho plánovania, v problémoch produktového mixu s cenovou elasticitou, v niektorých problémoch hospodárskeho plánovania, pri riešení prepravných problémov s úsporami z objemu, ako aj pri výbere portfólia s rizikovými cennými papiermi.

Preštudovaním uvedených optimalizačných úloh a rozborom popísaných metód pre nelineárnu optimalizáciu sme si pre ďalšie skúmanie vybrali problém výberu portfólia s rizikovými faktormi, nariadenie ktorej sme použili algoritmus redukovaných gradientov.

Jedným z cieľov práce bol návrh softvérového riešenia pre vybranú metódu nelineárnej optimalizácie. Pre tento účel bola navrhnutá procedúra *redgrad*, umožňujúca riešenie jednoduchých nelineárnych úloh z lineárnymi ohraničeniami využitím aproximačného algoritmu redukovaných gradientov. Procedúra je navrhnutá tak, aby okrem výpočtu koreňov úlohy a hodnoty účelovej funkcie umožnila užívateľovi náhľad do priebehu algoritmu. Výstup procedúry ukazuje čiastkové výsledky, ako napríklad smer nájdený prípustný smer posunu či dĺžku kroku v prípustnom smere. Užívateľovi je dovolené, aby pri volaní výpočtovej procedúry zadal počet iterácií, po ktorých sa výpočet zastaví. Pri opätovnom volaní procedúry môže tento výpočet pokračovať ďalej, znova o toľko iterácií, koľko užívateľ zvolí. Procedúra *redgrad* je teda vhodným nástrojom pre analýzu výpočtového procesu algoritmu redukovaných gradientov. Zmenou počiatočného bodu pre výpočet je možné pozorovať zmeny v dosiahnutom riešení, vyplývajúce z vlastností tohto algoritmu.

Aby sme si overili či sme pre riešenie úlohy výberu portfólia zvolili vhodný algoritmus, skonštruovali sme hypotetický príklad, v ktorom sme hľadali optimálnu kombináciu štyroch aktív na základe pozorovania ich výnosov v desiatich obdobiach. Na

základe teoretických predpokladov sme zostavili optimalizačnú úlohu, ktorú sme vyriešili pomocou výpočtového nástroja GAMS. Týmto spôsobom sme dosiahli optimálne riešenie $x^* = (0,299; 0,075; 0,372; 0,255; 0)$ s hodnotou účelovej funkcie $f(x^*) = 0.0065$. Hypotetickú úlohu sme ďalej upravili do tvaru, ktorý požaduje procedúra *redgrad*. Riešením 100 iterácií dosiahli výsledok $x^{100} = (0,210; 0,021; 0,520; 0,367; 0,020)$ a hodnotu účelovej funkcie $f(x^{100}) = 0.0094$. Môžeme konštatovať, že sme aproximačnou metódou redukovaných gradientov nedosiahli optimálne riešenie, je však viditeľné, že sa toto riešenie k optimálnemu riešeniu blíži. Po prepočte väčšieho počtu iterácií ($k = 1000$) sa dosiahnuté riešenie od optimálneho vzdďaľuje. Zadaním iného počiatočného bodu $x_2^0 = (\frac{29}{75}; \frac{17}{150}; \frac{2}{5}; \frac{1}{10}; 0)$ sme už po niekoľkých iteráciách dosiahli výsledok vzdďaľujúci sa od optima $x_2^{10} = (0,771; 0,027; 0,046; 0,089; 0)$. Nepresnosť algoritmu môže byť spôsobená nepresnosťami vo výpočtovom procese, ktoré sa s každou iteráciou kumulujú. Zlyhanie v konvergencii k optimálnemu bodu môže byť spôsobené aj “cikcakovitým” priebehom algoritmu, ktorého odstránenie procedúra nerieši.

Pre ukážku výpočtového procesu pri ktorom procedúra *redgrad* nájde optimálne riešenie bol skonštruovaný modelový príklad popísaný v prílohe 2. Príklad je dostatočne jednoduchý na to, aby sa dal prepočítať ručne. Jeho riešením môže študujúci hlbšie pochopiť priebeh procedúry *redgrad* a overiť správnosť výpočtov.

Zoznam použitej literatúry

- [1] AVRIEL, M. 1976. *Nonlinear Programming Analysis and Methods*. Englewood Cliffs, N.J. : Prentice-hall, 1976. 512 s. ISBN: 0-486-43227-0
- [2] BIEGLER, L. T. 2001. *Nonlinear Programming: Concepts, Algorithms and Applications*. Pittsburgh, PA. : Carnagie Mellon University, 2001. 91 s. [online] dostupné na: <http://cepac.cheme.cmu.edu/pasilectures/biegler/BieglerLecture.pdf>
- [3] E-BOOKSPDF.ORG. 6.252 *Nonlinear Programming Lecture 9: Feasible Direction Methods Lecture Outline*. [online] dostupné na: <http://www.e-bookspdf.org/>
- [4] FENDEK, M. 1998. *Nelineárne optimalizačné modely a metódy*. Bratislava. : Ekonóm, 1998. 254 s. ISBN: 9788022509350
- [5] LASDON, L. S. – WAREN, A. D. 1980. *Survey of Nonlinear Programming Applications*. Operations Research. 28(5) : 1029-1073 [online] dostupné na: <http://pubsonline.informs.org/doi/pdf/10.1287/opre.28.5.1029>
- [6] MITRADJIEVA-DANEVA, M. 2007. *Feasible Direction Methods for Constrained Nonlinear Optimization Suggestions for Improvements*. Linköping, Sweden. : Linköping University, 2007. 27 s. ISBN: 978-91-85715-11-4 [online] dostupné na: <http://liu.diva-portal.org/smash/get/diva2:23509/FULLTEXT01.pdf>
- [7] MLYNAROVIC, V. 1995. *Finančné modelovanie*. Bratislava. : Ekonóm, 1995. 251 s. ISBN: 9788022506779
- [8] OUYANG, H. – Gray, A. 2010. *Fast Stochastic Frank-Wolfe Algorithms for Nonlinear SVMs*. Georgia. : College of Computing, 2010. 12 s. [online] dostupné na: http://www.cc.gatech.edu/~houyang/doc/sdm10_sfw.pdf
- [9] VIRGINIA TECH. *Gradient Projection and Reduced Gradient Methods*. Blacksburg, Virginia. : College of Engineering. [online] dostupné na: <http://www2.esm.vt.edu/~zgurdal/COURSES/4084/4084-Docs/LECTURES/GradProj.pdf>