

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Najmenšie štvorce regulované l_1 normou - metódy a aplikácie

DIPLOMOVÁ PRÁCA

Bratislava 2013

Roman Kukumberg

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ MATEMATIKY A ŠTATISTIKY



Najmenšie štvorce regulované l_1 normou - metódy a aplikácie

DIPLOMOVÁ PRÁCA

Roman Kukumberg

Vedúci diplomovej práce: RNDr. Mária Trnovská, PhD.

Študijný odbor: 1114 aplikovaná matematika

Študijný program: Ekonomická a finančná matematika

BRATISLAVA 2013



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Roman Kukumberg
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

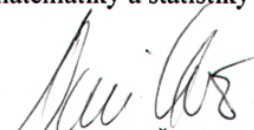
Názov: Najmenšie štvorce regulované l_1 normou - metódy a aplikácie

Cieľ: Cieľom je analyzovať rôzne prístupy na riešenie úlohy najmenších štvorcov regulovaných l_1 normou, ich porovnanie, naštudovanie rôznych možných praktických aplikácií a numerická implementácia.

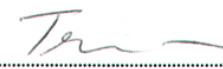
Vedúci: RNDr. Mária Trnovská, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.

Dátum zadania: 25.01.2012

Dátum schválenia: 26.01.2012


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


.....
študent


.....
vedúci práce

Čestné prehlásenie

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne s využitím teoretických vedomostí a s použitím uvedenej odbornej literatúry.

Bratislava, 24.04.2013

.....

Roman Kukumberg

Pod'akovanie

Aj touto cestou by som sa chcel poďakovať svojej vedúcej RNDr. Márii Trnovskej, PhD. za cenné rady, ochotu a čas pri písaní tejto diplomovej práce. Chcel by som sa taktiež poďakovať mojej rodine a priateľom za ich podporu počas doterajšieho štúdia.

Abstrakt

KUKUMBERG, Roman: Najmenšie štvorce regulované l_1 normou - metódy a aplikácie [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; vedúci práce: RNDr. Mária Trnovská, PhD., Bratislava, 2013

Diplomová práca sa zaoberá tzv. úlohou najmenších štvorcov regulovaných l_1 normou, ktorá má využitie v mnohých problémoch v praxi. Napríklad sa využíva pri rekonštrukcii signálu a zaostrovaní obrázkov. Zaoberáme sa dvoma konkurenčnými metódami konvexnej optimalizácie na riešenie danej úlohy, ktorými sú proximal gradient metóda a metódy vnútorného bodu. Proximal gradient metódu aplikujeme na neohraničený minimalizačný problém a metódu vnútorného bodu na formuláciu problému v tvare úlohy kvadratického programovania. Metódy použijeme na numerické riešenie dvoch konkrétnych aplikácií úlohy l_1 regulovaných najmenších štvorcov, spomenieme ich výhody/ nevýhody.

Kľúčové slová: najmenšie štvorce, l_1 regulácia, l_1 regulované najmenšie štvorce, proximal gradient metóda, proximal operátor, primárno duálna metóda vnútorného bodu, l_1 filtrovanie, Truss topology design

Abstrakt

KUKUMBERG, Roman: l_1 regularized least squares - methods and applications [Diploma thesis], COMENIUS UNIVERSITY; Faculty of Mathematics, Physics and Informatics; Department of Applied Mathematics and Statistics ; thesis supervisor: RNDr. Mária Trnovská, PhD., Bratislava, 2013

Diploma thesis deals with the so-called l_1 -regularized least squares, which is used in many problems in praxis. For example it is used in signal recovery and image denoising. We deal with two competing methods of convex optimization for solving this problem, which are proximal gradient method and interior-point method. Proximal gradient method is applied to solve unconstrained minimization problem and interior-point method is applied to solve quadratic program formulation of the problem. We use methods to find numerical solution of two concrete applications of l_1 -regularized least squares, we mention their advantages and disadvantages.

Key words: least squares, l_1 regularization, l_1 -regularized least squares, proximal gradient method, proximal operator, primal-dual interior-point method, l_1 filtering, Truss topology design

Obsah

1	Úvod	1
2	Úloha l_1 regulovaných najmenších štvorcov	2
2.1	Vlastnosti úlohy l_1 regulovaných najmenších štvorcov	3
2.2	Bi-kriteriálna formulácia a trade-off krivka	5
2.3	Zovšeobecnenie úlohy	7
2.4	Podmienky optimality	7
2.5	Duálna úloha	9
2.6	Aplikácie l_1 regulácie a úlohy l_1 regulovaných najmenších štvorcov . . .	11
3	Metódy riešenia	16
3.1	Proximal-gradient metóda	16
3.1.1	Proximal operátor	17
3.1.2	Algoritmus proximal-gradient metódy	20
3.1.3	Zrýchlená proximal gradient metóda	22
3.1.4	Spádová verzia zrýchlenej proximal gradient metódy	24
3.1.5	Nová modifikácia proximal gradient metódy	24
3.2	Metóda vnútorného bodu	26
3.2.1	Všeobecná formulácia úlohy	27
3.2.2	Primárno-duálny hľadací smer	28
3.2.3	Algoritmus primárno-duálnej metódy vnútorného bodu	30
4	Aplikácie	32
4.1	l_1 Trend Filtering	32
4.1.1	Trend filtering	32
4.1.2	Hodrick-Prescott filtering	33
4.1.3	l_1 trend filtering	34
4.1.4	Duálna úloha	37
4.1.5	Riešenie pomocou primárno-duálnej metódy vnútorného bodu	39
4.1.6	Riešenie pomocou Proximal gradient metódy	41
4.1.7	Experiment - porovnanie metód/prístupov riešenia	45
4.2	Truss topology design	51
4.2.1	Odvodenie modelu	52

4.2.2	Najjednoduchší TTD problém	54
4.2.3	Riešenie pomocou Proximal gradient metódy	57
4.2.4	Riešenie pomocou primárno-duálnej metódy vnútorného bodu .	58
4.2.5	Experimenty a porovnania metód/prístupov	62
5	Záver	71
6	Príloha - zdrojové kódy programov	75

1 Úvod

V práci nesúcej názov *Najmenšie štvorce regulované l_1 normou* sa budeme snažiť o analýzu a porovnanie rôznych prístupov na riešenie úlohy najmenších štvorcov regulovaných l_1 normou, a taktiež uvedieme rôzne možné praktické aplikácie danej úlohy. Na dvoch vybraných aplikáciách otestujeme nami naprogramované metódy, prípadne vykonáme iný numerický experiment.

V optimalizácii existuje mnoho problémov, ktoré využívajú l_1 regularizáciu, či už priamo alebo nepriamo. Mnohé z nich sa dajú naformulovať v tvare úlohy l_1 regulovaných najmenších štvorcov. Jednou z prvých oblastí, kde sa začala skúmať a využívať táto formulácia a regularizácia pomocou l_1 normy bola štatistika (regresia). V roku 1996 R. Tibshirani predstavil v [8] tzv. LASSO (Least Absolute Selection And Shrinkage Operator), ktorý je presne v tvare optimalizačnej úlohy, ktorej sa venujeme v práci. l_1 regulované najmenšie štvorce, s ich schopnosťou odhaliť riedke reprezentácie sú značne rozšírené napríklad v oblasti Machine Learning [29][37], v geofyzike [22],[23] a v problematike spracovania signálu [27],[24]. l_1 regularizácia sa stala populárnou v klasifikačných metódach ako Support Vector Machines a Logistická regresia.

Práca pozostáva z troch hlavných častí. V prvej časti práce sa oboznámime s problematikou l_1 regulovaných najmenších štvorcov, uvedieme jej niektoré vlastnosti, odvodíme podmienky optimality a Lagrangeovu duálnu úlohu k úlohe vo všeobecnejšom tvare. Taktiež si ako motiváciu spomenieme mnohé uplatnenia najmenších štvorcov regulovaných l_1 normou v praxi. V druhej časti si predstavíme dve konkurenčné metódy konvexnej optimalizácie, ktoré sa dajú na numerické riešenie daných úloh použiť. Konkrétne sa jedná o Proximal gradient metódu a Metódu vnútorného bodu. V neskoršej časti práce sa budeme venovať aplikáciám ako navrhovanie mechanických konštrukcií (Truss Topology Design) a l_1 filtrovanie podkladového trendu. Uvedieme si o nich niektoré poznatky a vlastnosti. Na tieto problémy sa budeme snažiť čo najlepšie aplikovať už spomínané metódy konvexnej optimalizácie. Metódu proximal gradient aplikujeme na pôvodný neohraničený problém, zatiaľ čo metódu vnútorného bodu budeme aplikovať na ekvivalentnú formuláciu (alebo duálnu úlohu) úlohy v tvare kvadratického programovania. Účinnosť metód v praxi otestujeme v numerických experimentoch, kde sa pokúsime objasniť, prečo v danom prípade funguje jedna metóda lepšie ako druhá.

2 Úloha l_1 regulovaných najmenších štvorcov

Mnoho problémov, s ktorými sa stretávame v optimalizácii sa dá naformulovať v tvare úlohy l_1 regulovaných najmenších štvorcov. V tejto kapitole si uvedieme niektoré poznatky o tejto formulácii a aj jej mnohé aplikácie. Kapitola bola spracovaná na základe informácii z [1], [2], [3], [8] a [14]. Jednou z prvých oblastí, kde sa začala skúmať a využívať táto formulácia a regularizácia pomocou l_1 normy bola štatistika (regresia). V roku 1996 R. Tibshirani predstavil v [8] tzv. LASSO (Least Absolute Selection And Shrinkage Operator).

Uvažujme lineárny model nasledujúceho tvaru

$$y = Ax + e$$

kde $x \in R^n$ je hľadaný vektor neznámych, $y \in R^m$ je vektor pozorovaní, $e \in R^m$ je šum v pozorovaniach a $A \in R^{m \times n}$ je takzvaná dátová matica. V prípade klasickej dátovej matice s rozmermi $m \geq n$ a lineárne nezávislými stĺpcami, môžeme riešenie x ľahko určiť riešením problému najmenších štvorcov, minimalizovaním štvorcov reziduí

$$\text{Min } \|Ax - y\|_2^2.$$

Avšak, ak počet pozorovaní m nie je dosť veľký v porovnaní s počtom premenných n , jednoduchá regresia vedie k tzv. over-fittingu a štatistický model popisuje náhodnú chybu (šum) namiesto skúmaného podkladového vzťahu. Takýto model následne poskytuje zlé predikcie. Štandardná technika na zabránenie over-fittingu je tzv. Tichonovova regularizácia, čo predstavuje regularizáciu pomocou l_2 normy. Úloha má tvar

$$\text{Min } \|Ax - y\|_2^2 + \lambda \|x\|_2^2$$

kde $\lambda > 0$ je regularizačný parameter. Výsledná regresia sa taktiež nazýva hrebeňová regresia (ridge regression). Problém l_2 regularizovaných najmenších štvorcov má analytické riešenie, ktoré je lineárnou funkciou vektora pozorovaní y . Avšak hrebeňová regresia má nevýhodu, že v prípade $n \gg m$ produkuje hodnoty koeficientov pre všetkých n premenných. Aj táto nevýhoda viedla k využívaniu regularizácie úlohy pomocou l_1 normy, z čoho vzniká úloha l_1 regulovaných najmenších štvorcov, ktorej sa budeme v práci venovať.

V úlohe l_1 regulovaných najmenších štvorcov zameníme sumu štvorcov používanú v Tichonovovej regularizácii za súčet absolútnych hodnôt, čiže l_1 normu. Úloha má tvar

$$\text{Min } \|Ax - y\|_2^2 + \lambda \|x\|_1 \quad (1)$$

kde $\lambda > 0$ je regularizačný parameter. Táto úloha má vždy riešenie, ktoré nemusí byť nutne jedinečné. Výsledná úloha sa v štatistike nazýva LASSO regresia alebo LASSO regularizácia. Rozsiahly výskum ukázal, že regresia regularizovaná l_1 normou je omnoho lepšia ako hrebeňová regresia, hlavne vtedy, ak je počet pozorovaní menší ako počet parametrov.

2.1 Vlastnosti úlohy l_1 regulovaných najmenších štvorcov

Uvedieme si niektoré základné vlastnosti l_1 regulovaných najmenších štvorcov a uvedieme taktiež niektoré odlišnosti od úlohy regulovanej l_2 normou.

1. Zaujímavou vlastnosťou úlohy l_1 regulovaných najmenších štvorcov je limitné správanie pre regularizačný parameter $\lambda \rightarrow 0$. Limitné riešenie úlohy je ten bod, ktorý má spomedzi všetkých riešení spĺňajúcich $A^T(Ax - y) = 0$ najmenšiu veľkosť v zmysle l_1 normy.
2. Tak ako v prípade Tichonovovej regularizácie, optimálne riešenie l_1 regulovaných najmenších štvorcov sa blíži k nulovému vektoru, keď regularizačný parameter $\lambda \rightarrow \infty$. Avšak narozdiel od Tichonovovej regularizácie táto konvergencia nastáva pre konečnú hodnotu regularizačného parametra $\lambda_{max} = \|2A^T y\|_\infty$. Pre všetky hodnoty $\lambda \geq \lambda_{max}$ je optimálne riešenie úlohy rovné 0. Hodnota λ_{max} pochádza z podmienok optimality úlohy.
3. Regularizačná cesta úlohy l_1 regulovaných najmenších štvorcov, teda trieda riešení úlohy ako sa parameter λ mení na $(0, \infty)$, má po častiach lineárnu vlastnosť: Existujú hodnoty $0 = \lambda_k < \dots < \lambda_1 = \lambda_{max}$ také, že regularizačná cesta je po častiach lineárnou funkciou v R^n

$$x^* = \frac{\lambda_i - \lambda}{\lambda_i - \lambda_{i+1}} x^{(i+1)} + \frac{\lambda - \lambda_{i+1}}{\lambda_i - \lambda_{i+1}} x^{(i)}$$

$$\lambda_{i+1} \leq \lambda \leq \lambda_i, \quad i = 1, \dots, k - 1$$

pričom $x^{(i)}$ rieši úlohu l_1 regulovaných najmenších štvorcov s hodnotou regularizačného parametra $\lambda = \lambda_i$.

4. Jednou z najdôležitejších vlastností úlohy l_1 regulovaných najmenších štvorcov je, že úloha typicky podáva riedky vektor riešenia x , čiže vektor x má pomerne málo nenulových zložiek (častokrát iba m , ak $m \leq n$). Táto vlastnosť je spôsobená práve prítomnosťou jednotkovej normy v účelovej funkcii. V bežných prípadoch platí, že pri zväčšovaní regularizačného parametra λ pribúdajú nulové zložky riešenia, riešenie sa stáva redšie. Penalta v tvare l_1 normy prikladá relatívne veľkú váhu na najmenšie zložky vektora a menšie váhy väčším zložkám. Riedke riešenia majú častokrát dobrú interpretáciu a šetria pamäť.

Uvedomme si, čo nám táto vlastnosť spôsobí v prípade LASSO regresie - väčšina parametrov (premenných) je položená nulovej hodnote. Tento jav je možné chápať tak, že LASSO sa postará o výber vhodného modelu za nás, nenulové (signifikantné) budú parametre prislúchajúce k dôležitým premenným. Tiež možno pozorovať, za akých hodnôt λ premenné vstupujú do modelu. Z uvedeného dôvodu sú regularizačné metódy používané aj na výber modelu. Naopak v hrebeňovej regresii ostávajú všetky parametre nenulové.

Úlohu l_1 regulovaných najmenších štvorcov môžeme interpretovať aj ako problém optimálneho dizajnu, kde n premenných x_1, \dots, x_n sú dizajnovacie premenné, ktorých hodnoty chceme určiť. Vektor $h = Ax$ dáva vektor m výsledkov, o ktorých predpokladáme, že sú lineárnymi funkciami dizajnových premenných x . Vektor y predstavuje vektor cieľových alebo požadovaných výsledkov. Cieľom je vybrať vektor dizajnovacích premenných, ktorý spĺňa čo najpresnejšie požadované výsledky $Ax \approx y$, a pritom vektor chceme čo najjednoduchší v zmysle l_1 normy (nájdeme riedke riešenie). Keďže meriame kvalitu dizajnu normou odchýlky medzi dosiahnutými a cieľovými výsledkami, možno úlohu l_1 regulovaných najmenších štvorcov označiť za problém hľadania najlepšieho dizajnu. Napríklad úloha Truss Topology Design sa dá zapísať v tvare úlohy l_1 regulovaných najmenších štvorcov. Problematika TTD sa zaoberá dizajnovaním optimálnych konštrukcií. Matica A predstavuje reakciu konštrukcie na prítomnosť hrán konštrukcie. Vhodným rozložením prípustných hrán (dizajnovacie premenné) chceme kompenzovať záťaž pôsobiacu na konštrukciu (y), pričom nemá zmysel dizajnoviť konštrukciu zo všetkých možných priečok. Preto chceme, aby čo najviac priečok dostalo nulovú váhu. Optimalizácia úlohy l_1 regulovaných najmenších štvorcov vyberie vhodné hrany na postavenie štruktúry, ktorá odolá zadaným silám.

Nelinearita

Zatiaľ, čo úloha regularizovaná l_2 normou má explicitné riešenie (l_2 regresia nepridáva na zložitosti problému), ktoré je lineárnou funkciou vektora pozorovaní y a je pomerne ľahko spočítateľné, úloha l_1 regulovaných najmenších štvorcov má riešenie, ktoré nie je lineárne v y . Taktiež neexistuje analytická formula pre optimálne riešenie úlohy. Riešenie úlohy l_1 regulovaných najmenších štvorcov musí byť preto počítané numericke. Účelová funkcia úlohy l_1 regulovaných najmenších štvorcov je konvexná (suma konvexných funkcií-noriem), ale nie je diferencovateľná ako v prípade Tichonovovskej regularizácie, čo je dôsledkom výmeny sumy štvorcov za l_1 normu (suma absolútnych hodnôt). Z dôvodu nediferencovateľnosti účelovej funkcie je riešenie úlohy náročnejšie ako riešenie úlohy regulovanej l_2 normou. Na optimalizáciu nediferencovateľných konvexných funkcií existujú metódy ako napríklad subgradientné metódy či proximal gradient metóda. K optimalizácii daného problému sa však dá pristúpiť aj iným spôsobom. Nediferencovateľnosti účelovej funkcie sa dá vyhnúť vhodnou transformáciou na konvexný kvadratický problém s lineárnymi ohraničeniami, kde už je účelová funkcia diferencovateľná. Takáto formulácia sa dá riešiť napríklad metódami vnútorného bodu.

2.2 Bi-kriteriálna formulácia a trade-off krivka

Úloha l_1 regulovaných najmenších štvorcov sa dá vnímať taktiež ako úloha bi-kriteriálnej optimalizácie, kde máme dve konvexné účelové funkcie, ktoré chceme minimalizovať. Máme maticu $A \in R^{m \times n}$ a vektor $y \in R^m$, chceme vybrať riešenie $x \in R^n$, pričom berieme do úvahy dva ciele (účelové funkcie):

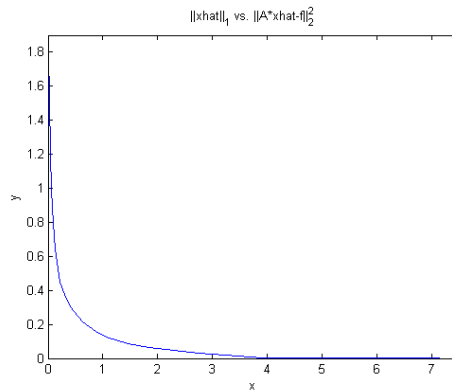
- $F_1(x) = \|Ax - y\|_2^2$ - miera blízkosti y k Ax v zmysle l_2 normy (veľkosť reziduí)
- $F_2(x) = \|x\|_1$ - miera veľkosti riešenia x v zmysle l_1 normy (\sim riedkosť riešenia)

Naším cieľom je nájsť vektor x , ktorý robí reziduá malé a nie je príliš veľký.

Účelovú funkciu úlohy l_1 regulovaných najmenších štvorcov možno teda vnímať ako váženú sumu dvoch našich cieľov, pričom regularizačný parameter λ predstavuje relatívnu váhu funkcie $\|x\|_1$ vzhľadom k funkcii $\|Ax - y\|_2^2$. Váha vyjadruje naše preferencie - berieme veľké λ , ak chceme funkciu $\|x\|_1$ malú a opačne.

Prirodzene nás môže napadnúť otázka, aká je dobrá hodnota regularizačného parametra λ . Odpoveď na danú otázku nám dá tzv. trade off analýza. Trade off analýza študuje, o koľko horšie si musíme viesť v jednom ciele na to, aby sme si počínali lepšie

v druhom. Nájdeme tzv. trade off krivku medzi dvoma cieľmi, ktorá ukazuje, koľko musíme "zaplatiť" v hodnote jednej účelovej funkcie, aby sme dosiahli zlepšenie v hodnote druhej účelovej funkcie. Taktiež ukazuje, aká množina hodnôt účelovej funkcie je dosiahnuteľná. Ako sa hodnota parametra λ mení od 0 do ∞ riešenia úlohy l_1 regulovaných najmenších štvorcov vykresľujú optimálnu trade off krivku (krivka je parametrizovaná parametrom λ). Ak uvažujeme trade off krivku medzi $F_1(x)$ a $F_2(x)$, X-sová os predstavuje optimálne hodnoty funkcie $F_2(x)$ a os Y predstavuje optimálne hodnoty $F_1(x)$. Tzv. koleno (knee) trade off krivky reprezentuje v mnohých aplikáciach dobré kompromisné riešenie. Trade off krivku si ukážeme v nasledujúcom príklade pre úlohu Truss Topology Design, ktorej formulácia je v tvare úlohy l_1 regulovaných najmenších štvorcov. Krivku sme vykreslili pre 100 rôznych hodnôt parametra λ . Poznamenajme, že sa jedná o úlohu malých rozmerov, pretože vyčísľovanie krivky je náročné.



Obr. 1: Trade off krivka medzi $\|Ax - f\|_2^2$ a $\|x\|_1$ pre úlohu TTD o rozmeroch 56x265. Xsová os predstavuje optimálne hodnoty funkcie $\|x\|_1$ a os Y predstavuje optimálne hodnoty $\|Ax - f\|_2^2$. Na obrázku taktiež vidieť, aká množina hodnôt účelových funkcií je dosiahnuteľná. Tzv. koleno (knee) trade off krivky reprezentuje v mnohých aplikáciach dobré kompromisné riešenie. Avšak v prípade problému TTD je dôležitejšie, aby daná konštrukcia ustála danú záťaž f , a preto parameter λ volíme v neskoršej kapitole 0.0001, ktoré na tomto obrázku dáva bod (4.66,0).

2.3 Zovšeobecnenie úlohy

V nasledujúcej kapitole si ukážeme podmienky optimality k úlohe l_1 regulovaných najmenších štvorcov. Vzhľadom na neskoršie riešenie podobných formulácii si však pod-

mienky optimality odvodíme pre úlohu vo všeobecnejšom tvare, pod ktorý budú neskôr spadať aj nami riešené úlohy. Budeme sa venovať úlohám l_1 -filtering a *truss topology design*, ktorých formulácie vyzerajú nasledovne:

$$\text{Min } \frac{1}{2}\|y - x\|_2^2 + \lambda\|Dx\|_1, \quad (\text{L1})$$

kde $D \in R^{(n-2) \times n}$ je matica diferencie druhého rádu a $\lambda > 0$ je kladný parameter.

$$\text{Min } \|Ax - f\|_2^2 + \lambda\|x\|_1, \quad (\text{TTD})$$

kde $\lambda > 0$ je parameter, matica $A \in R^{m \times n}$ a vektor $f \in R^m$ sú vstupné dáta. Riešené úlohy (L1) a (TTD) majú zreteľné spoločné črty. Úloha (TTD) je presne v tvare úlohy l_1 regulovaných najmenších štvorcov. V oboch prípadoch sa jedná o optimalizáciu na celom priestore (voľná optimalizácia). Minimalizujeme účelovú funkciu skladajúcu sa z diferencovateľnej a nediferencovateľnej časti. Diferencovateľnú časť predstavuje kvadratická funkcia, nediferencovateľnú tvorí l_1 norma (súčet absolútnych hodnôt zložiek vektora) prenasobená kladným parametrom λ . Všeobecná úloha daného typu môže vyzeráť nasledúco:

$$\text{Min } \frac{1}{2}\|Ax - b\|_2^2 + \alpha\|Wx\|_1, \quad (\text{VSU})$$

kde $\alpha > 0$ je kladný parameter, $A \in R^{m \times n}$, $b \in R^m$ a $W \in R^{l \times n}$. Pre budúce potreby zavedieme ešte jeden technický predpoklad - budeme požadovať, aby hodnosť matice $W_{l \times n}$, $l \leq n$ bola rovná menšiemu rozmeru l . Dôsledkom tohto predpokladu je, že symetrická matica $WW^T \in R^{l \times l}$ má plnú hodnosť $h(WW^T) = l$. Matica WW^T je preto regulárna a existuje k nej inverzná matica.

Ak by sme chceli získať ekvivalentnú úlohu s (L1), stačí za matice a parametre položiť $A = I_{n \times n}$, $b = y$, $\alpha = \lambda$ a $W = D_{(n-2) \times n}$, podobne vieme položením $A = A_{m \times n}$, $b = f_{m \times 1}$, $\alpha = \frac{\lambda}{2}$ a $W = I_{n \times n}$ dostať úlohu ekvivalentnú k (TTD), resp. k úlohe l_1 regulovaných najmenších štvorcov.

2.4 Podmienky optimality

Účelová funkcia problému (VSU) je konvexná, pretože je súčtom konvexných funkcií, avšak nie je diferencovateľná kvôli prítomnosti jednotkovej normy. Z toho dôvodu použijeme podmienky optimality prvého rádu definované pomocou subgradientov (analógia pre gradient). Viac informácií o subgradientoch a dôkazy nasledujúcich viet je možné nájsť v [14] a [21].

Definícia 2.1. Nech je $f(x) : R^n \rightarrow R$ konvexná funkcia. Vektor $g \in R^n$ nazývame subgradient funkcie f v bode $x \in D(f)$ ak platí:

$$f(z) \geq f(x) + g^T(z - x), \quad \forall z \in D(f)$$

Poznámka 2.1. Pre konvexnú diferencovateľnú funkciu f je subgradient rovný gradientu funkcie pre každé x . Subgradient však existuje aj v bodoch x , v ktorých funkcia nie je diferencovateľná. V týchto bodoch nemusí byť subgradient jednoznačne definovaný, t.j. existuje množina vektorov, ktoré sú subgradientami danej funkcie v bode x . Táto konvexná množina sa nazýva subdiferenciál v bode x a značí sa $\partial f(x)$.

Veta 2.1. (*Minimum nediferencovateľnej funkcie*) Nech funkcia $f(x)$ je konvexná a subdiferencovateľná v bode \hat{x} . Funkcia nadobúda v bode \hat{x} minimum práve vtedy, keď množina subdiferenciálu funkcie f v bode \hat{x} obsahuje nulovú hodnotu, t.j. $0 \in \partial f(\hat{x})$.

Veta 2.2. (*Reťazové pravidlo*) Nech funkcia $f(x)$ je konvexná. Potom pre funkciu definovanú $h(x) = f(Ax + b)$ je subdiferenciál daný vzťahom $\partial h(x) = A^T \partial f(Ax + b)$.

Použitím oboch predošlých viet nájdeme podmienky optimality pre úlohu (VSU). Z Vety 2.1. vieme, že funkcia nadobúda minimum, keď nulový vektor patrí do subdiferenciálu funkcie $f(x)$

$$\frac{\partial f(x)}{\partial x} = A^T(Ax - b) + \alpha W^T u \ni 0, \quad u \in R^l$$

$$u_i = \begin{cases} 1 & (Wx)_i > 0 \\ -1 & (Wx)_i < 0 \\ [-1, 1] & (Wx)_i = 0 \end{cases}$$

V ďalších krokoch využijeme elementárne operácie a dôsledok dodatočného predpokladu úlohy (VSU) - existenciu inverznej matice $(WW^T)^{-1}$.

$$\begin{aligned} -A^T(Ax - b) &\in \alpha W^T u \\ WA^T(b - Ax) &\in \alpha WW^T u \\ (WW^T)^{-1}WA^T(b - Ax) &\in \alpha u \end{aligned}$$

Dostávame nutné a postačujúce podmienky pre minimum účelovej funkcie (VSU):

$$((WW^T)^{-1}WA^T(b - Ax))_i \in \begin{cases} \alpha & (Wx)_i > 0 \\ -\alpha & (Wx)_i < 0, \quad i = 1, \dots, l \\ [-\alpha, \alpha] & (Wx)_i = 0 \end{cases}$$

Pre úlohu l_1 -filtering (L1) sú podmienky nasledovné:

$$((DD^T)^{-1}D(y-x))_i \in \begin{cases} \lambda & (Dx)_i > 0 \\ -\lambda & (Dx)_i < 0, \quad i = 1, \dots, n-2 \\ [-\lambda, \lambda] & (Dx)_i = 0 \end{cases}$$

Poznamenajme, že z podmienok optimality pre úlohu (L1) vyplývajú/sa dajú dokázať niektoré spomínané vlastnosti l_1 filtrovania ako tzv. maximum fitting error bound (horné ohraničenie chyby), vlastnosť lineárnej rozširiteľnosti a taktiež sa pomocou nich dá nájsť λ_{max} , pri ktorom je riešenie úlohy (L1) už najlepším afínnym riešením.

Podobne dostaneme podmienky optimality pre úlohu (TTD), resp. pre úlohu l_1 regulovaných najmenších štvorcov, z ktorých sa takisto dá určiť hodnota λ_{max} :

$$(A^T(b-Ax))_i \in \begin{cases} \frac{\lambda}{2} & x_i > 0 \\ -\frac{\lambda}{2} & x_i < 0, \quad i = 1, \dots, n \\ [-\frac{\lambda}{2}, \frac{\lambda}{2}] & x_i = 0 \end{cases}$$

2.5 Duálna úloha

V nasledujúcej časti odvodíme Lagrangeovu duálnu úlohu k všeobecnej úlohe (VSU). Za účelom odvodenia duálnej úlohy vykonáme jeden umelý krok - zavedieme novú premennú $z \in R^m$ a dodáme nové ohraničenie $z = Ax - y$. Danou substitúciou dostaneme ekvivalentnú formuláciu úlohy v tvare minimalizačnej úlohy s ohraničením:

$$\begin{aligned} \text{Min } & \frac{1}{2}\|z\|_2^2 + \alpha\|Wx\|_1 \\ & z = Ax - y \end{aligned}$$

Pre danú formuláciu vieme jednoducho zostrojiť Lagrangeovu funkciu, kde $v \in R^m$ je duálna premenná prislúchajúca k ohraničeniu $z = Ax - y$ v tvare rovnosti.

$$L(x, z, v) = \frac{1}{2}\|z\|_2^2 + \alpha\|Wx\|_1 + v^T(Ax - y - z)$$

Lagrangeovu duálnu funkciu získame ako $G(v) = \inf_{x,z} L(x, z, v)$. Lagrangeova funkcia je zrejme konvexná funkcia v premenných x, z a teda podľa Vety 2.1. funkcia nadobúda minimum, ak $0 \in \partial L(x)$.

$$\frac{\partial L}{\partial x} = \alpha W^T u + A^T v \ni 0, \quad u_i = \begin{cases} 1 & (Wx)_i > 0 \\ -1 & (Wx)_i < 0 \\ [-1, 1] & (Wx)_i = 0 \end{cases}$$

$$\frac{\partial L}{\partial z} = z - v = 0$$

Využitím predpokladu úlohy (VSU) sa prvý uvedený vzťah dá po násobení maticou $(WW^T)^{-1}W$ upraviť na tvar

$$((WW^T)^{-1}WA^T v)_i \in \begin{cases} -\alpha & (Wx)_i > 0 \\ \alpha & (Wx)_i < 0, \quad i = 1, \dots, l \\ [-\alpha, \alpha] & (Wx)_i = 0 \end{cases}$$

Všimnime si, že v optime je súčet členov (v Lagrangeovej funkcii) $\alpha\|Wx\|_1$ a $v^T Ax$ vždy rovný nulovej hodnote, o čom nám hovorí prvý vzťah. Po použití oboch podmienok optimality dostávame

$$\inf_{x,z} L(x, z, v) = \begin{cases} -\frac{1}{2}v^T v - v^T y & ((WW^T)^{-1}WA^T v)_i \in [-\alpha, \alpha] \\ -\infty & \text{inak} \end{cases}$$

Lagrangeova duálna úloha má potom tvar

$$\begin{aligned} \text{Max} \quad & -\frac{1}{2}v^T v - y^T v \\ & -\alpha \mathbf{1} \leq (WW^T)^{-1}WA^T v \leq \alpha \mathbf{1}, \end{aligned} \tag{2}$$

čo je úloha konvexného kvadratického programovania s kvadratickou účelovou funkciou a lineárnymi ohraničeniami. Hovoríme, že premenná $v \in R^m$ je duálne prípustná, ak spĺňa ohraničenia duálnej úlohy $-\alpha \mathbf{1} \leq (WW^T)^{-1}WA^T v \leq \alpha \mathbf{1}$. Ľubovoľný duálne prípustný bod v nám dáva dolné ohraničenie pre optimálnu hodnotu p^* primárneho problému, čiže $-\frac{1}{2}v^T v - v^T y \leq p^*$. Navyše, optimálne hodnoty primárnej a duálnej úlohy sa rovnajú, pretože primárny problém spĺňa Slaterovu podmienku (regularita). Slaterova podmienka je v tomto prípade len prípustnosť a tá je triviálne splnená. Platí teda aj $p^* = d^*$.

2.6 Aplikácie l_1 regulácie a úlohy l_1 regulovaných najmenších štvorcov

V optimalizácii existuje mnoho problémov, ktoré využívajú l_1 regularizáciu, či už priamo alebo nepriamo. Mnohé z nich sa dajú naformulovať v tvare úlohy l_1 regulovaných najmenších štvorcov. Ako motiváciu, a taktiež ako demonštráciu výhod l_1 normy voči

l_2 norme si v tejto podkapitole stručnejšie uvedieme dve aplikácie l_1 regularizácie. Obe z nich sa dajú napísať vo forme úlohy l_1 regulovaných najmenších štvorcov - konkrétne sa jedná o rekonštrukciu signálu a rekonštrukciu/zaostrovanie/odšumovanie obrázkov. V neskoršej časti práce sa budeme podrobnejšie sústreďiť na aplikácie l_1 -filtering (l_1 filtrovanie) a Truss Topology Design (TTD, dizajn mechanických konštrukcií). l_1 filtering je metóda na odhadnutie podkladového trendu v časových radoch a TTD sa zaoberá optimálnym dizajnom mechanických konštrukcií.

l_1 regulované najmenšie štvorce, s ich schopnosťou odhaliť riedke reprezentácie sú značne rozšírené v oblasti Machine Learning [29][37], v štatistike [8], v geofyzike [22],[23] a v problematike spracovania signálu [27],[24]. Medzi aplikácie l_1 regulovaných najmenších štvorcov patrí napríklad LASSO regresia [8] (ktorá predstavuje konkurenciu klasickej Tichonovovej regularizácii, taktiež sa využíva na výber modelu), spracovanie prirodzeného jazyka (Natural Language Processing) [36], rekonštrukcia signálu [27], [28], zobrazovanie magnetickej rezonancie [3], komprimované snímanie (Compressed sensing) [27] a vlnková transformácia (Wavelet tresholding/transform) [30]. Úloha nachádza aplikáciu taktisto pri detekcii prichádzajúcich radarových signatúr [35], pri epidemiologických štúdiách, na analýzu genetických dát, na kompresiu dát a taktiež na dekódovanie lineárnych kódov [34].

l_1 regularizácia sa stala populárnou v klasifikačných metódach ako Support Vector Machines, Boosting a Logistická regresia, l_1 penalty sú využívané taktiež v Multi-layer Perceptions (neurónové siete učené backpropagáciou), v zovšeobecnených aditívnych modeloch (Generalized Additive Models), Probit regresii. Myšlienka l_1 regularizácie bola využitá aj v optimalizácii portfólia [31], riadiacom dizajne [32] a pri výbere grafických modelov [33].

Rekonštrukcia signálu - l_1 vs l_2 norma

V problematike rekonštrukcie signálu máme daný signál reprezentovaný vektorom $x \in R^n$, pričom jeho zložky predstavujú namerané hodnoty signálu v pravidelných časových intervaloch, t.j. $x_i = f(t_i)$. Zvyčajne sa predpokladá, že sa signál v čase príliš nemení $x_i \approx x_{i+1}$. Môžu však nastávať skoky v signále, v izolovaných bodoch nemusí byť táto podmienka nutne splnená. Predpokladajme však, že prijatý signál x_{sum} je znečistený

neznámym šumom v , ktorý sa v čase dosť mení.

$$x_{sum} = x + v$$

Cieľom je zrekonštruovať/odšumiť prijatý signál, resp. odhadnúť pôvodný nezašumený signál. Formulácia takéhoto problému vyzerá nasledujúco

$$\text{Min } \|x - x_{sum}\|_2 + \gamma\Phi(x), \quad \gamma > 0$$

kde $\Phi(x)$ je nejaká regularizačná funkcia, ktorá meria nedostatok hladkosti odhadovaného signálu, resp. predstavuje mieru blízkosti $x_i \approx x_{i+1}$. Porovnáme 2 typy regularizačných funkcií za účelom porovnania správania a vlastností l_1 normy a l_2 normy. Ak je pôvodný signál hladký, je vhodné využiť kvadratické vyhladenie, kde

$$\Phi_q(x) = \|Dx\|_2^2 = \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2$$

Ak sa hodnoty signálu môžu meniť skokovito, je vhodné použiť totálnu variáciu

$$\Phi_{tv}(x) = \|Dx\|_1 = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

$$D = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \in R^{(n-1) \times n}$$

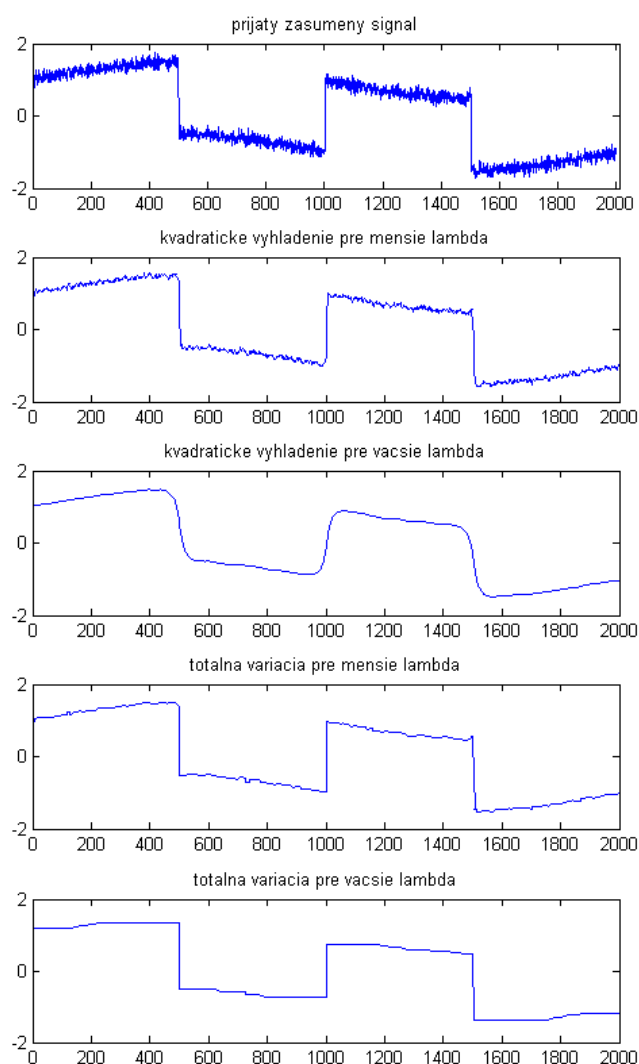
Najväčšia výhoda l_1 normy voči l_2 norme spočíva v tom, že l_1 norma prikladá relatívne veľkú váhu na najmenšie zložky vektora a menšie váhy väčším zložkám (outlierom). Z toho dôvodu je veľmi robustná voči outlierom a využíva sa napríklad aj v tzv. robustnej regresii.

Zaostrovanie obrázkov

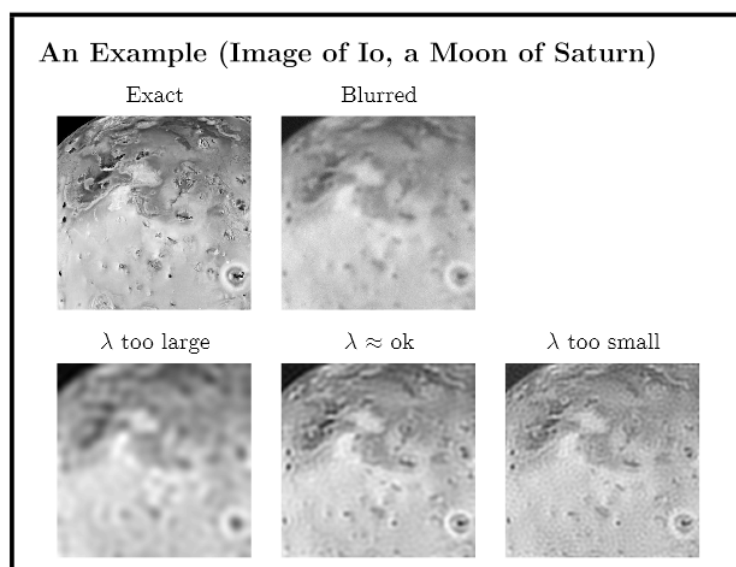
Úloha rekonštrukcie obrázkov sa dá formulovať ako minimizačná úloha

$$\text{Min } \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|Wx\|_1, \quad \lambda > 0$$

kde matica A je lineárna transformácia reprezentujúca rozostrenie, $b \in R^{mn}$ je rozmazaný obrázok vo veľkosti $m \times n$ pixelov a ortogonálna matica W reprezentuje vlnkovú transformáciu (wavelet).



Obr. 2: Rekonštrukcia signálu: Na obrázku možno pozorovať riešenie úlohy pre rôzne λ pre kvadratické vyhladenie ako aj totálnu variáciu. Obrázok pre malé lambda v prípade kvadratického vyhladenia je stále pomerne veľa zašumený a už v tomto štádiu možno pozorovať mierne zahľadenie výrazných zmien. Pre väčšie lambda je rekonštruovaný signál nepoužiteľný nereprezentuje dobre pôvodný nezašumený signál, zhladenie je príliš veľké. V prípade totálnej variácie obrázky dopadli lepšie - pre obe voľby lambda sú výrazné zmeny pôvodného signálu dobre znázornené. Najlepší rekonštruovaný signál sme dosiahli pomocou totálnej variácie pre malé lambda. Daný obrázok sme sami zhotovili v programe cvx, kde sme kvadratické vyhladenie ako aj totálnu variáciu použili na dáta pre rôzne voľby parametra λ .



Obr. 3: Rekonštrukcia obrázkov: Na obrázku možno pozorovať riešenie úlohy pre rôzne λ . Vidíme, že voľba regularizačného parametra je pri odšumovaní dôležitá.

3 Metódy riešenia

Pre optimálne riešenie úlohy l_1 regulovaných najmenších štvorcov neexistuje analytická formula, preto musí byť počítané numericky. Účelová funkcia úlohy l_1 regulovaných najmenších štvorcov je konvexná (súčet konvexných funkcií), ale nie je diferencovateľná. V zásade existujú dva hlavné prístupy na získanie riešenia danej úlohy. Tieto prístupy sa líšia práve v tom, ako sa vysporiadavame s nediferencovateľnosťou účelovej funkcie. Prvá možnosť je priama, optimalizujeme pôvodnú nediferencovateľnú účelovú funkciu bez ohraničení. Na optimalizáciu funkcií rozložiteľných na diferencovateľnú a nediferencovateľnú časť je obzvlášť vhodná proximal gradient metóda. Druhá možnosť ako sa vysporiadať s nediferencovateľnosťou účelovej funkcie je vhodná transformácia úlohy. Nediferencovateľnosti účelovej funkcie sa dá vyhnúť transformáciou na konvexný kvadratický problém s lineárnymi ohraničeniami, kde už je účelová funkcia diferencovateľná. Takáto formulácia sa dá riešiť napríklad metódami vnútorného bodu. Z uvedených dôvodov si v nasledujúcej kapitole predstavíme tieto dve konkurenčné metódy konvexného programovania. V ďalšej časti práce sa pokúsime tieto metódy naprogramovať a priamo aplikovať na nami riešené úlohy. Nasledujúca podkapitola bola spracovaná na základe [11], [15] a [21]. Uvádzame v nej len dôležité vlastnosti proximal gradient metódy. Dôkazy uvedených tvrdení a viet, analýza konvergenzie a ďalšie informácie o proximal gradient metóde sú podrobne spracované v mnou napísanej bakalárskej práci [21].

3.1 Proximal-gradient metóda

V optimalizácii sa často stretávame s potrebou riešiť úlohy s nediferencovateľnými konvexnými funkciami. Bežný postup, ktorým sa dajú úlohy takéhoto typu riešiť, je použitie subgradientných metód konvexného programovania, ktoré ako optimalizačný smer využívajú subgradients. Subgradientné metódy majú však niekoľko nevýhod - sú pomalé a taktiež nemajú dobré ukončovacie kritériá. Subgradientné metódy nie sú spádovými metódami, nerastúcosť účelovej funkcie počas optimalizácie nie je zaručená. Väčšina spôsobov voľby dĺžky kroku takisto nezaručuje konvergenciu subgradientných metód. Vzhľadom na uvedené nevýhody subgradientných metód boli vytvorené efektívnejšie metódy na riešenie úloh s nediferencovateľnými konvexnými funkciami rôznej štruktúry. Neohraničené úlohy s účelovou funkciou rozložiteľnou na konvexnú diferenco-

vateľnú a konvexnú nediferencovateľnú časť, vedú k zavedeniu tzv. proximal operátora a metódy proximal gradient. Tejto metóde konvexného programovania a jej rôznym modifikáciám sa budeme venovať v nasledujúcej kapitole. Všetky verzie proximal gradient metódy konvergujú rýchlejšie ako klasické subgradientné metódy, tri z verzií sú spádové metódy, hodnota účelovej funkcie po každom kroku klesá alebo ostáva rovnaká.

Predpoklady

Proximal gradient metóda sa využíva na optimalizáciu neohraničeného problému, pričom účelová funkcia $f(x)$ sa dá rozložiť na súčet dvoch funkcií.

$$\text{Min } f(x) = g(x) + h(x) \quad (3)$$

kde:

- Funkcia $g(x) : R^n \rightarrow R$ je diferencovateľná, konvexná.
- Funkcia $h(x)$ je uzavretá ¹ konvexná. Môže byť nediferencovateľná, najlepšie taká, pri ktorej je proximal operátor nenáročný, jednoduchý.

3.1.1 Proximal operátor

Základným prvkom algoritmu proximal gradient metódy je tzv. proximal operátor. Uvedieme si jeho dve ekvivalentné definície, spomenieme niektoré vlastnosti.

Definícia 3.1. Nech funkcia h je konvexná. Potom proximal operátor funkcie h je definovaný ako:

$$\text{prox}_h(x) = \underset{u}{\operatorname{argmin}} \left(h(u) + \frac{1}{2} \|u - x\|_2^2 \right) \quad (4)$$

Veta 3.1. (Definícia pomocou subgradientu): Proximal operátor $\hat{u} = \text{prox}_h(x)$ konvexnej funkcie h v bode x možno ekvivalentne definovať ako:

$$x - \hat{u} \in \partial h(\hat{u}) \quad (5)$$

kde $\partial h(\hat{u})$ je subdiferenciál funkcie h v bode \hat{u} .

Pre úplnosť uvedieme niektoré vlastnosti proximal operátora, mnohé napomáhajú pri odvodení proximal operátorov pre rôzne funkcie.

¹Konvexná funkcia sa nazýva uzavretá, ak má uzavretý konvexný epigraf.

Tvrdenie 3.1. *Proximal operátor $\text{prox}_h(x)$ pre konvexú funkciu h existuje a je jednoznačne určený pre každé x .*

Tvrdenie 3.2. *Ak je $h : R^{n_1} \times R^{n_2} \rightarrow R$ separovateľná funkcia $h(u_1, u_2) = h_1(u_1) + h_2(u_2)$, potom platí $\text{prox}_h(x_1, x_2) = (\text{prox}_{h_1}(x_1), \text{prox}_{h_2}(x_2))$.*

Poznamenajme, že predošlé tvrdenie je veľmi užitočné, namiesto hľadania celého vektora nám v prípade separovateľných funkcií stačí hľadať jednotlivé zložky proximal operátora.

Tvrdenie 3.3. *(Škálovanie a posúvanie argumentu): Ak je funkcia $h(u) = f(tu + a)$, $t \neq 0$ získaná škálovaním a posúvaním argumentu funkcie f , potom sa proximal operátor funkcie h dá zapísať ako:*

$$\text{prox}_h(x) = \frac{1}{t} [\text{prox}_{t^2 f}(tx + a) - a]$$

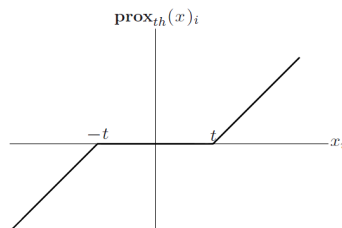
Tvrdenie 3.4. *(Nerozpínavosť): Proximal operátor je Lipschitzovsky spojitá funkcia s Lipschitzovskou konštantou $L = 1$. Ak označíme $u = \text{prox}_h(x)$ a $\hat{u} = \text{prox}_h(\hat{x})$ potom platí vzťah $\|u - \hat{u}\|_2 \leq \|x - \hat{x}\|_2$*

Príklady proximal operátora

V nasledujúcich príkladoch si ukážeme explicitné vyjadrenia proximálneho operátora pre rôzne tvary funkcií $h(u)$. Často budeme zobrazovať proximal operátor $\text{prox}_{th}(x)$ funkcie $th(u)$ z dôvodu štruktúry algoritmu proximal gradient metódy, kde premenná t predstavuje dĺžku kroku.

Poznamenajme, že pre niektoré funkcie je proximal operátor ťažko explicitne vyjadriteľný, čo znemožňuje následné efektívne využitie proximal gradient metódy.

Funkcia $h(u)$:	Proximal operátor:
$h(u) = 0$	$prox_h(x) = x$
Kvadratická funkcia: $\frac{1}{2}u^T Au + b^T u + c$	$prox_{th}(x) = (tA + I)^{-1}(x - tb)$
Logaritmická bariéra: $-\sum_{i=1}^n \log u_i$	$prox_{th}(x)_i = \frac{x_i + \sqrt{x_i^2 + 4t}}{2}, i = 1, \dots, n.$
Jednotková norma: $\ u\ _1 = \sum_{i=1}^n u_i $	$prox_{th}(x)_i = \begin{cases} x_i - t & x_i \geq t \\ x_i + t & x_i \leq -t \\ 0 & x_i \leq t \end{cases}$
Euklidovská norma: $\ u\ _2$	$prox_{th}(x) = \begin{cases} x(1 - \frac{t}{\ x\ _2}) & \ x\ _2 \geq t \\ 0 & \text{inak} \end{cases}$
Indikátorová funkcia: $I_C(u) = \begin{cases} 0 & u \in C \\ \infty & \text{inak} \end{cases}$	$prox_h(x) = \underset{u \in C}{\operatorname{argmin}} \ u - x\ _2^2 = P_C(x)$



Obr. 4: Proximálny operátor jednotkovej normy

3.1.2 Algoritmus proximal-gradient metódy

Algoritmus základnej proximal-gradient metódy na riešenie úlohy $\text{Min } g(x) + h(x)$ pozostáva z opakovania hlavnej iterácie až po dosiahnutie požadovanej presnosti. Proximal-gradient iterácia vyzerá nasledovne:

$$x^{(k)} = \text{prox}_{t_k h}(x^{(k-1)} - t_k \nabla g(x^{(k-1)}))$$

Premenná $t_k > 0$ je dĺžka kroku, ktorú volíme buď konštantnú alebo ju budeme určovať v každej iterácii.

Poznámka 3.1. Ak za nediferencovateľnú časť zvolíme $h(u) = 0$, dostaneme klasickú gradientnú metódu. Minimalizačná úloha sa zjednoduší na tvar $\text{Min } g(x)$ a iterácia je nasledovného tvaru:

$$x^{(k)} = x^{(k-1)} - t_k \nabla g(x^{(k-1)})$$

Poznámka 3.2. Ak za funkciu zvolíme $h(u) = I_C(u)$. Minimalizačná úloha sa dá chápať ako $\text{Min}_{x \in C} g(x)$. Iterácia sa zmení na tvar gradientnej projekčnej metódy, proximal operátor sa správa ako projekcia.

$$x^{(k)} = P_C(x^{(k-1)} - t_k \nabla g(x^{(k-1)}))$$

Zápis hlavnej iterácie proximal gradient metódy sa dá prepísať taktiež ako

$$x^{(k)} = x^{(k-1)} - t_k G_{t_k}(x^{(k-1)}),$$

pričom $G_t(x) = \frac{1}{t}(x - \text{prox}_{th}(x - t \nabla g(x)))$ je smer, ktorý môžeme chápať ako subgradient funkcie f , o čom hovorí aj nasledujúca veta.

Veta 3.2. : Pre smer $G_t(x)$ definovaný ako $G_t(x) = \frac{1}{t}(x - \text{prox}_{th}(x - t \nabla g(x)))$ platí:

$$G_t(x) \in \nabla g(x) + \partial h(x - tG_t(x)). \quad (6)$$

Poznámka 3.3. Veta nám vlastne hovorí o tom, že smer $G_t(x)$ patrí do subdiferenciálu funkcie h v bode x^k novej iterácie posunutom o gradient funkcie g v bode predchádzajúcej iterácie x^{k-1} . Taktiež je vidno fakt, že bod x je minimom funkcie $f(x) = g(x) + h(x)$ práve vtedy, keď $G_t(x) = 0$. Táto skutočnosť vyplýva priamo z podmienok optimality diferencovateľných a nediferencovateľných funkcií.

Voľba približne optimálneho kroku

Cieľom algoritmu je určiť optimálnu dĺžku kroku t_k (line search) v hlavnej iterácii proximal gradient metódy:

$$x^{(k)} = \text{prox}_{t_k h} (x^{(k-1)} - t_k \nabla g(x^{(k-1)})) = x^{(k-1)} - t_k G_{t_k}(x^{(k-1)})$$

Algoritmus:

Začínáme s krokom $t = \hat{t} > 0$ a opakujeme $t = \beta t$ ($0 < \beta < 1$) až pokým náš krok splní nerovnosť:

$$g(x^{(k)}) = g(x^{(k-1)} - t G_t(x^{(k-1)})) \leq g(x^{(k-1)}) - t \nabla g(x^{(k-1)})^T G_t(x^{(k-1)}) + \frac{t}{2} \|G_t(x^{(k-1)})\|_2^2$$

Ako počiatočný krok t_0 volíme ľubovoľnú kladnú hodnotu. Algoritmus vyžaduje výpočet jedného proximal operátora za iteráciu. Line search nerovnosť je inšpirovaná analýzou konvergenencie.

Konvergenca proximal-gradient metódy

Proximal gradient metóda konverguje rýchlejšie ako klasické subgradientné metódy, ktoré majú zložitosť iba $O(1/\epsilon^2)$. Konvergenca metódy je garantovaná pre voľbu konštantného kroku $t = 1/L$ tak ako aj pre voľbu kroku t_k určeného v každej iterácii. Aby však metóda konvergovala potrebujeme splniť nasledujúce predpoklady.

Predpoklady:

- Optimálna funkčná hodnota \hat{f} je konečná a funkcia ju nadobúda v bode \hat{x} , ktorý nemusí byť jednoznačný.
- Gradient funkcie ∇g je Lipschitzovsky spojitá funkcia s konštantou $L > 0$

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L \|x - y\|_2, \quad \forall x, y \in R^n.$$

Veta 3.3. *Proximal gradient je spádová metóda, pre hodnotu účelovej funkcie v nasledujúcom bode platí $f(x^+) \leq f(x) - \frac{t}{2} \|G_t(x)\|_2^2$. Navyše, vzdialenosť priebežného riešenia od optimálneho sa každou iteráciou znižuje, pretože platí vzťah $\|x^+ - \hat{x}\|_2^2 \leq \|x - \hat{x}\|_2^2$.*

Poznámka 3.4. Veta nám hovorí o tom, že pri každej novej iterácii sa funkčná hodnota účelovej funkcie zmenší alebo ostane rovnaká (ak sme v optime). Funkčná hodnota sa za každú iteráciu zlepšuje prinaajmenšom o hodnotu $\frac{t}{2} \|G_t(x)\|_2^2$, tento jav je dostatočný na to, aby garantoval konvergenziu metódy.

Veta 3.4. (*Konvergencia proximal gradient metódy*) *Nech sú splnené predpoklady konvergencie. Proximal gradient metóda dosahuje ϵ presné riešenie po $O(1/\epsilon)$ krokoch. Pri voľbe konštantného kroku $t = 1/L$ platí vzťah*

$$f(x^{(k)}) - \hat{f} \leq \frac{1}{2kt} \|x^{(0)} - \hat{x}\|_2^2$$

a pri určovaní kroku v každej iterácii $t = t_i \geq t_{min}$ (line search) platí

$$f(x^{(k)}) - \hat{f} \leq \frac{1}{2kt_{min}} \|x^{(0)} - \hat{x}\|_2^2$$

Pri oboch spôsoboch voľby kroku sa teda rozdiel priebežnej funkčnej hodnoty od optimálnej znižuje najmenej rýchlosťou $1/k$. Na nájdenie ϵ -suboptimálneho bodu spĺňajúceho $f(x^{(k)}) - \hat{f} \leq \epsilon$ pomocou klasickej proximal gradient metódy potrebujeme $O(1/\epsilon)$ iterácii.

3.1.3 Zrýchlená proximal gradient metóda

Zložitosť základnej proximal gradient metódy sa dá vhodnou úpravou vylepšiť na hodnotu $O(1/\sqrt{\epsilon})$. Upravený algoritmus s lepšou zložitouťou sa nazýva zrýchlená proximal gradient metóda. Od klasického algoritmu sa líši v tom, že do proximálneho operátora v každej iterácii vstupujú namiesto hodnôt $x^{(k-1)}$ hodnoty $y^{(k-1)}$, ktoré sú získané extrapoláciou z predošlých hodnôt $x^{(k-1)}$ a $x^{(k)}$. Preto je zrýchlená metóda nazývaná tiež ako proximal gradient metóda s extrapoláciou. Zrýchlená metóda dosahuje približne rovnakú časovú náročnosť za iteráciu ako základná proximal gradient metóda.

Algoritmus zrýchlenej PGM:

Zvolíme štartovací bod $x^{(0)} \in D(h)$ a položíme $y^{(0)} = x^{(0)}$, potom pre $k \geq 1$:

$$\begin{aligned} x^{(k)} &= \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) \\ y^{(k)} &= x^{(k)} + \frac{k-1}{k+2} (x^{(k)} - x^{(k-1)}) \end{aligned}$$

Tak ako v prípade klasickej metódy krok t_k býva konštantný alebo určený v každej iterácii.

Všimnime si, že prvá iterácia ($k=1$) zrýchlenej metódy je zhodná s prvou iteráciou pôvodnej metódy.

Voľba približne optimálneho kroku

Cieľom je určiť optimálnu dĺžku kroku t_k (line search) v časti metódy:

$$x^{(k)} = \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) = y^{(k-1)} - t_k G_{t_k}(y^{(k-1)})$$

Algoritmus:

Začíname s krokom $t = t_{k-1}$ z predošlej iterácie a opakujeme $t = \beta t$ ($0 < \beta < 1$) až pokiaľ náš krok splní nerovnosť:

$$g(x^{(k)}) = g(y^{(k-1)} - t G_t(y^{(k-1)})) \leq g(y^{(k-1)}) - t \nabla g(y^{(k-1)})^T G_t(y^{(k-1)}) + \frac{t}{2} \|G_t(y^{(k-1)})\|_2^2$$

Ako počiatočný krok t_0 volíme ľubovoľnú kladnú hodnotu. Poznamenajme, že takto definovaný algoritmus voľby dĺžky kroku implikuje nerastúcosť kroku $t_k \leq t_{k-1}$. Jedným rozdielom oproti základnej metóde je to, že v nerovnosti používame namiesto pôvodných hodnôt x extrapolované hodnoty y .

Konvergencia zrýchlenej proximal gradient metódy

Rovnako ako v prípade základnej metódy musia byť pre konvergenciu metódy pri voľbe ľubovoľného koku (fixný/line search) splnené predpoklady:

- Optimálna funkčná hodnota \hat{f} je konečná a funkcia ju nadobúda v bode \hat{x} , ktorý nemusí byť jednoznačný.
- Gradient funkcie ∇g je Lipschitzovsky spojitá funkcia s konštantou $L > 0$

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L \|x - y\|_2, \quad \forall x, y.$$

Veta 3.5. (Konvergencia zrýchlenej proximal gradient metódy) *Nech sú splnené predpoklady konverencie. Zrýchlená proximal gradient metóda potrebuje $O(1/\sqrt{\epsilon})$ iterácií na nájdenie ϵ -presného riešenia splňajúceho $f(x^{(k)}) - \hat{f} \leq \epsilon$. Pri voľbe konštantného kroku $t_i = t = 1/L$ platí*

$$f(x^{(k)}) - \hat{f} \leq \frac{2}{(k+1)^2 t} \|x^{(0)} - \hat{x}\|_2^2$$

a pri kroku $t_{i-1} \geq t_i \geq t_{\min}$ určenom v každej iterácii (line search) platí

$$f(x^{(k)}) - \hat{f} \leq \frac{2}{(k+1)^2 t_{\min}} \|x^{(0)} - \hat{x}\|_2^2$$

Uviedli sme si, že rozdiel $f(x^{(k)}) - \hat{f}$ sa znižuje prinajmenšom rýchlosťou $1/k^2$. Pri oboch spôsoboch voľby kroku dosahuje algoritmus proximal gradient metódy s extrapoláciou zložitosť $O(1/\sqrt{\epsilon})$. Zrýchlená metóda však nezaručuje nerastúcosť účelovej funkcie $f(x^{(k)}) \leq f(x^{(k-1)})$ tak ako pôvodná verzia. Táto vlastnosť všeobecne nie je platná, metóda sa dostáva k optimálnej hodnote často pomocou "skokov". Z uvedeného dôvodu existuje ďalšia modifikácia proximal gradient metódy, ktorá vlastnosť spádovosti spĺňa. Nazýva sa spádová (descent) verzia zrýchlenej proximal gradient metódy.

3.1.4 Spádová verzia zrýchlenej proximal gradient metódy

Nasledujúca modifikácia algoritmu zaručuje nerastúcosť účelovej funkcie.

$$\begin{aligned} z^{(k)} &= \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) \\ x^{(k)} &= \begin{cases} z^{(k)} & f(z^{(k)}) \leq f(x^{(k-1)}) \\ x^{(k-1)} & \text{inak} \end{cases} \\ v^{(k)} &= x^{(k-1)} + \frac{1}{\theta_k} (z^{(k)} - x^{(k-1)}) \\ y^{(k)} &= (1 - \theta_{k+1})x^{(k)} + \theta_{k+1}v^{(k)} \end{aligned}$$

kde premenná θ_k je definovaná ako $\theta_k = \frac{2}{k+1}$. Všimnime si, že ak nie je splnená podmienka $f(z^{(k)}) \leq f(x^{(k-1)})$, premenná x_k ostáva konštantná, až pokiaľ sa nenájde lepšie suboptimálne riešenie. Ak je podmienka splnená, tak iterácia je ekvivalentná s iteráciou zrýchlenej metódy.

3.1.5 Nová modifikácia proximal gradient metódy

Nasledujúcu úpravu sme vytvorili na základe poznatkov o klasickej a zrýchlenej proximal gradient metóde, využili sme predovšetkým fakt, že klasická proximal gradient metóda je spádová a platí v jej prípade vzťah $\|x^+ - \hat{x}\|_2^2 \leq \|x - \hat{x}\|_2^2$. Modifikácia je svojou stavbou podobná descent verzii, taktiež má vlastnosť nerastúcosť účelovej funkcie, ktorá zrýchlenej proximal gradient metóde chýba. Najväčší rozdiel oproti spádovej verzii je ten, že premenná x neostáva za neplatnosti podmienky konštantnou, ale

v každej iterácii sa mení, približuje sa optimálnemu riešeniu.

$$\begin{aligned}
 z^{(k)} &= \text{prox}_{t_k h} (y^{(k-1)} - t_k \nabla g(y^{(k-1)})) \\
 x^{(k)} &= \begin{cases} z^{(k)} & f(z^{(k)}) \leq f(x^{(k-1)}) \\ \text{prox}_{t_k h} (x^{(k-1)} - t_k \nabla g(x^{(k-1)})) & \text{inak} \end{cases} \\
 y^{(k)} &= x^{(k)} + \frac{k-1}{k+2} (x^{(k)} - x^{(k-1)})
 \end{aligned}$$

Pokiaľ je podmienka nerastúcnosti splnená, používa sa klasická zrýchlená metóda, kvôli jej vysokej rýchlosti konvergencie. Ak daná podmienka neplatí, využívame pôvodnú proximal gradient metódu na predchádzajúce $x^{(k-1)}$, ktorá je vždy spádová. Táto úprava je založená na tom, že v klasickom proximal gradient algoritme je účelová funkcia nerastúca a vzdialenosť priebežného od optimálneho riešenia sa znižuje každou iteráciou. Vďaka tomu priebežné riešenie $x^{(k)}$ nebude nikdy stagnovať, od čoho očakávame rýchlejšie dosiahnutie suboptimálneho riešenia.

3.2 Metóda vnútorného bodu

V tejto kapitole sa budeme venovať metódam vnútorného bodu pre riešenie všeobecných úloh konvexnej optimalizácie. Presnejšie sa budeme venovať základnej primárno-duálnej metóde vnútorného bodu. Kapitola bola spracovaná na základe poznatkov z [14], [16] a [18].

Metódy vnútorného bodu sú založené na riešení postupnosti parametrizovaných úloh na voľný extrém, resp. úloh s ohraničeniami typu lineárnych rovností pomocou Newtonovej metódy, pričom optimálne riešenie každej parametrizovanej úlohy slúži ako štartovací bod pre ďalšiu úlohu. Postupnosť týchto miním (ktoré spolu tvoria tzv. centrálnu trajektóriu) konverguje k optimálnemu riešeniu pôvodnej úlohy.

Túto metódu v 60. tých rokoch rozpracovali Fiacco a McCormick (Nonlinear Programming, [38]) a nazvali ju SUMT (Sequential Unconstrained Minimization Technique, resp. metóda postupného riešenia úloh na voľný extrém). Bola však dokázaná konvergencia iba presných riešení parametrizovaných úloh a navyše sa metódy aplikovali na zložité úlohy, ktoré bolo zložité analyzovať. Až v roku 1984 N. Karmarkar navrhol tzv. projektívny algoritmus na riešenie úloh lineárneho programovania, ktorý bol polynomiálny. Ukázalo sa, že Karmarkarov algoritmus úzko súvisí so spomínanou metódou SUMT. Následne sa metódy začali skúmať na jednoduchých úlohách konvexného programovania a boli dokázané mnohé dôležité vlastnosti (konvergencia v okolí centrálnej trajektórie, polynomiálnosť algoritmov,...)

Hlavnou myšlienkou bariérových metód vnútorného bodu je, že pôvodná úloha je riešená aplikovaním Newtonovej metódy (alebo inej metódy voľnej optimalizácie) na postupnosť parametrizovaných úloh. Primárno-duálne metódy vnútorného bodu však, narozdiel od bariérových metód, riešia podmienky optimality pôvodnej úlohy pomocou Newtonovej metódy (alebo inej metódy voľnej optimalizácie) aplikovanej na postupnosť perturbovaných podmienok optimality.

Uvedená idea metód vnútorného bodu je základom algoritmov, ktoré sa riadia podľa nasledovnej schémy:

1. pre danú hodnotu parametra t nájsť približné riešenie parametrizovanej úlohy (podmienok optimality) pomocou nejakej metódy voľnej optimalizácie zo zadaného štartovacieho bodu.
2. zväčšiť hodnotu parametra t a postup opakovať, pričom štartovacím bodom pre

túto minimalizáciu je vypočítané približné riešenie predchádzajúcej úlohy.

Primárno-duálne metódy vnútorného bodu sú častokrát účinnejšie ako bariérové metódy, hlavne v prípadoch, keď je vyžadovaná veľká presnosť, pretože sú schopné dosiahnuť lepšiu ako lineárnu konvergenciu. Pre úlohy lineárneho, kvadratického, geometrického a semidefinitného programovania špecializované primárno-duálne metódy vnútorného bodu prekonávajú bariérovú metódu. Jednou z výhod primárno-duálnych algoritmov je fakt, že dokážu pracovať aj vtedy, keď je problém prípustný, ale nie striktné prípustný. Pre všeobecnú konvexnú optimalizáciu sú primárno-duálne metódy témou aktívneho výskumu.

Napriek niektorým rozdielom sú primárno-duálne metódy vnútorného bodu veľmi podobné bariérovej metóde. V primárno-duálnej metóde je iba jeden cyklus (iterácia), nie je tu prítomné rozlišovanie vnútorných a vonkajších iterácii ako v bariérovej metóde. V každej iterácii primárno-duálnej metódy sú aktualizované primárne aj duálne premenné. Hľadacie smery (search directions) v primárno-duálnej metóde sú získané z Newtonovej metódy aplikovanej na modifikované Karush-Kuhn-Tuckerove (KKT) podmienky, hľadacie smery sú príbuzné hľadacím smerom v bariérovej metóde. V primárno-duálnej metóde nie sú primárne a duálne premenné počas iterácií nevyhnutne prípustné.

3.2.1 Všeobecná formulácia úlohy

Uvažujme nasledovnú úlohu konvexného programovania

$$\begin{aligned} \text{Min } & f_0(x) \\ & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & Ax = b, \end{aligned} \tag{7}$$

kde $f_0, \dots, f_m : R^n \rightarrow R$ sú konvexné, dvakrát spojitاً diferencovateľné funkcie, a $A \in R^{p \times n}$ je matica s hodnotou $h(A) = p < n$. Úlohou je teda nájsť minimum konvexnej účelovej funkcie $f_0(x)$ na konvexnej množine prípustných riešení

$$P = \{x : Ax = b, f_i(x) \leq 0 \quad i = 1, \dots, m\}$$

Poznamenajme, že pod daný tvar úlohy spadajú mnohé úlohy - napríklad úlohy lineárneho a kvadratického programovania.

Predpokladáme, že úloha je riešiteľná, optimálne riešenie x^* existuje, pričom optimálna hodnota účelovej funkcie je $p^* = f_0(x^*)$. Taktiež predpokladáme, že existuje $x \in P$ taký, ktorý spĺňa $Ax = b$ a $f_i(x) < 0$ $i = 1, \dots, m$ (x je vnútorný bod množiny). Vnútro množiny prípustných riešení P je neprázdne, čo znamená, že je splnená Slaterova podmienka. Preto existujú duálne optimálne $\lambda^* \in R^m$, $v^* \in R^p$, ktoré spolu s optimálnym riešením x^* spĺňajú Karush-Kuhn-Tuckerove podmienky, ktoré sú nutnými aj postačujúcimi podmienkami optimality

$$\begin{aligned} \nabla f_0(x^*) + Df(x^*)^T \lambda^* + A^T v^* &= 0 \\ \text{diag}(\lambda^*) f(x^*) &= 0 \\ Ax^* = b, \quad f(x^*) \leq 0, \quad \lambda^* \geq 0, \end{aligned} \tag{8}$$

kde $f : R^n \rightarrow R^m$ sú ohraničenia a $Df(x)$ je Jakobián

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix}, \quad Df(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{pmatrix}$$

Metódy vnútorného bodu riešia pôvodnú konvexnú úlohu (alebo jej KKT podmienky) aplikovaním Newtonovej metódy na postupnosť úloh s ohraničeniami v tvare rovnosti (alebo na postupnosť modifikovaných KKT podmienok).

3.2.2 Primárno-duálny hľadací smer

Základným prvkom primárno-duálnej metódy vnútorného bodu je tzv. hľadací smer (search direction), ktorý je smerom optimalizácie/posúvania primárnych a duálnych premenných. Hľadacie smery v primárno-duálnej metóde sú získané z Newtonovej metódy aplikovanej na systém nelineárnych rovníc $r_t(x, \lambda, v) = 0$, ktorým sú modifikované Karush-Kuhn-Tuckerove (KKT) podmienky

$$r_t(x, \lambda, v) = \begin{pmatrix} r_{dual} \\ r_{cent} \\ r_{pri} \end{pmatrix} = \begin{pmatrix} \nabla f_0(x) + Df(x)^T \lambda + A^T v \\ -\text{diag}(\lambda) f(x) - \frac{1}{t} \mathbf{1} \\ Ax - b \end{pmatrix}, \tag{9}$$

kde podmienka komplementarity je upravená o člen $\frac{1}{t} \mathbf{1}$, pričom $t > 0$ je kladný parameter. Ak premenné x, λ, v spĺňajú $r_t(x, \lambda, v) = 0$ a $f(x) < 0$, potom $x = x^*(t)$, $\lambda = \lambda^*(t)$ a $v = v^*(t)$ sú body centrálnej trajektórie. Presnejšie, x je primárne prípustné a λ ,

v sú duálne prípustné s duálnou mezderou m/t . Prvá časť r_t , r_{dual} , sa nazýva duálne rezíduum, prostredná časť r_{cent} sa nazýva centrálné rezíduum a posledná časť r_{pri} sa nazýva primárne rezíduum.

Všimnime si, ak parameter $t \rightarrow \infty$, systém rovníc $r_t(x, \lambda, v) = 0$ sa zredukuje na KKT podmienky pre našu pôvodnú úlohu. Základnou myšlienkou je používať Newtonove kroky riešenia systému $r_t(x, \lambda, v) = 0$ pre rastúcu postupnosť parametra t . Preto sa teraz sústredíme na Newtonov krok pre riešenie nelineárneho systému rovníc $r_t(x, \lambda, v) = 0$ pre fixnú hodnotu parametra t , v bode (x, λ, v) , ktorý spĺňa $f(x) < 0$, $\lambda > 0$. Označíme súčasný bod $y = (x, \lambda, v)$ a Newtonov krok ako $\Delta y = (\Delta x, \Delta \lambda, \Delta v)$. Newtonov krok je potom určený lineárnymi rovnicami

$$r_t(y + \Delta y) \approx r_t(y) + Dr_t(y)\Delta y = 0,$$

kde $Dr_t(y)$ predstavuje Jakobián. Je teda potrebné riešiť systém lineárnych rovníc $Dr_t(y)\Delta y = -r_t(y)$, čo v pôvodnom značení predstavuje systém

$$\left(\begin{array}{ccc|c} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Df(x)^T & A^T & -\nabla f_0(x) - Df(x)^T \lambda - A^T v \\ -diag(\lambda)Df(x) & -diag(f(x)) & 0 & diag(\lambda)f(x) + \frac{1}{t} \mathbf{1} \\ A & 0 & 0 & -Ax + b \end{array} \right) \quad (\text{PD})$$

Definícia 3.2. Primárno-duálny hľadací smer $\Delta y_{pd} = (\Delta x_{pd}, \Delta \lambda_{pd}, \Delta v_{pd})$ úlohy (7) je definovaný ako riešenie systému lineárnych rovníc (PD).

Poznámka 3.5. Ak vektor x spĺňa $Ax = b$ (primárne rezíduum je nulové), potom platí $A\Delta x_{pd} = 0$ a vektor Δx_{pd} definuje prípustný smer - pre ľubovoľné s je splnený systém $A(x + s\Delta x_{pd}) = b$.

Poznámka 3.6. Systém lineárnych rovníc (PD) je možné riešiť elimináciou a dá sa využiť štruktúra úlohy - ako riedkosť alebo diagonalita. Newtonov krok je v takýchto prípadoch možné dostať efektívnejšie. Keď je pôvodný problém riedky, čo znamená, že účelová funkcia a každé ohraničenie závisí iba na malom počte premenných, potom sú aj všetky gradienty a Hessove matice riedke. Z toho dôvodu je aj systém (PD) riedky (má veľa nulových zložiek) a dá sa rýchlejšie vyriešiť metódami určenými pre prácu s riedkymi maticami. Podobne sa systém zjednoduší v prípade separovateľnej účelovej funkcie a lineárnych ohraničeniach. V tomto prípade je matica $\nabla^2 f_0(x)$ diagonálna a výrazy $\nabla^2 f_i(x)$ zanikajú.

Náhrada duálnej medzery

V primárno-duálnej metóde vnútorného bodu nie sú k -te hodnoty primárnych a duálnych premenných $(x^{(k)}, \lambda^{(k)}, v^{(k)})$ nevyhnutne prípustné. To samozrejme neplatí pre ich limitné hodnoty, keď algoritmus skonverguje. Kvôli tomuto poznatku však nastáva problém, že nemôžeme ľahko vyhodnotiť duálnu medzeru $\eta^{(k)}$, prislúchajúcu ku k -temu kroku algoritmu, ako je to zaužívané pri bariérovej metóde.

Definícia 3.3. (surrogate duality gap): Pre každé x , ktoré spĺňa $f(x) < 0$ a $\lambda \geq 0$, je náhradná duálna medzera rovná

$$\hat{\eta}(x, \lambda) = -f(x)^T \lambda$$

Náhradná medzera bude duálnou medzerou, ak sú premenné x a λ, v primárne, resp. duálne prípustné ($r_{pri} = 0, r_{dual} = 0$). Hodnota parametra t , ktorá prislúcha náhradnej duálnej medzere je $m/\hat{\eta}$.

Určenie dĺžky kroku

Voľba dĺžky kroku v primárno-duálnej metóde vnútorného bodu je založená na norme rezíduí a je modifikovaná, aby zabezpečila nerovnosti $\lambda > 0$ a $f(x) < 0$. Označme súčasný bod ako y a bod nasledujúcej iterácie ako y^+ , potom

$$y^+ = y + s\Delta y_{pd}.$$

Ako prvé vypočítame maximálnu kladnú dĺžku kroku, ktorá spĺňa $\lambda^+ \geq 0$

$$s^{max} = \sup\{s \in [0, 1] | \lambda + s\Delta\lambda \geq 0\} = \min\{1, \min\{-\lambda_i/\Delta\lambda_i | \Delta\lambda_i < 0\}\}$$

Následne položíme $s = 0.99s^{max}$, a násobíme dĺžku kroku s parametrom $\beta \in (0, 1)$, až pokým splníme podmienky

$$f(x^+) < 0, \quad \|r_t(x^+, \lambda^+, v^+)\|_2 \leq (1 - \alpha s)\|r_t(x, \lambda, v)\|_2.$$

Parameter α sa zvyčajne volí od 0.01 do 0.1, parameter β je typicky z rozpätia od 0.3 do 0.8.

3.2.3 Algoritmus primárno-duálnej metódy vnútorného bodu

Algoritmus primárno-duálnej metódy vnútorného bodu sa zastaví, keď sú všetky premenné prípustné (primárne aj duálne s toleranciou ϵ_{feas}) a zároveň hodnota náhradnej

duálnej medzery je menšia ako tolerancia ϵ . Keďže má primárno-duálna metóda častokrát lepšiu ako lineárnu konvergenciu, je bežné položiť tolerancie malé.

Vstup: vektor x , ktorý splníva $f(x) < 0$; vektor $\lambda > 0$; skalár $\mu > 1$; požadované miery presnosti $\epsilon > 0$, $\epsilon_{feas} > 0$

Opakuj kroky 1., 2., 3., 4., až pokým $\|r_{pri}\|_2 \leq \epsilon_{feas}$, $\|r_{dual}\|_2 \leq \epsilon_{feas}$ a $\hat{\eta} \leq \epsilon$

1. Určenie t . Nastavenie $t = \mu m / \hat{\eta}$.
2. Výpočet primárno-duálneho hľadacieho kroku $\Delta y_{pd} = (\Delta x_{pd}, \Delta \lambda_{pd}, \Delta v_{pd})$.
3. Určenie dĺžky kroku $s > 0$.
4. Aktualizácia premenných $y = y + s \Delta y_{pd}$.

4 Aplikácie

V prvej kapitole sme spomenuli, že existuje mnoho úloh využívajúcich reguláciu pomocou l_1 normy, prípadne sa dajú naformulovať na tvar úloh najmenších štvorcov regulovaných l_1 normou. Vyberieme dve aplikácie, ktoré nás dovedli k potrebe riešiť úlohu v tvare najmenších štvorcov regulovaných l_1 normou. Konkrétne sa jedná o tzv. l_1 Trend Filtering a Truss Topology Design. l_1 filtrovanie je metóda na odhadnutie podkladového trendu v časových radoch a TTD sa zaoberá optimálnym dizajnom mechanických konštrukcií. Spomenieme si o nich niektoré poznatky a na záver podkapitol otestujeme nami naprogramované metódy z predošlej kapitoly na daných úlohách.

4.1 l_1 Trend Filtering

Jednou z možných aplikácií l_1 regulácie, ktoré sme objavili, je metóda nesúca názov l_1 Trend Filtering. Jedná sa o metódu pre modelovanie podkladového trendu v dátach (v časových radoch), ktorá je variáciou Hodrick-Prescottovho filtra. l_1 Trend filter vznikol zmenou účelovej funkcie HP filtra, kde sa zmení jej druhý penalizačný člen - suma štvorcov druhých diferencií časového radu sa zamení za sumu ich absolútnych hodnôt (l_1 norma). Vďaka tejto modifikácii je už účelová funkcia zložená z diferencovateľnej a nediferencovateľnej časti, dostávame na prvý pohľad typickú stavbu úlohy riešiteľnej pomocou proximal gradient metódy.

V nasledujúcich sekciách si načrtne myšlienku problému Trend filtering, oboznámime sa s HP filtrom a jeho niektorými vlastnosťami. Predovšetkým si predstavíme problém l_1 trend filtering, jeho vlastnosti, iné formulácie a dva spôsoby jeho riešenia. Kapitola je napísaná na základe zdrojov [4], [5] a [20].

4.1.1 Trend filtering

Daný je časový rad y_t , $t = 1, \dots, n$. Predpokladáme, že sa skladá z málo meniaceho trendového komponentu x_t a viac meniaceho náhodného komponentu z_t , niekedy nazývaného aj ako cyklický komponent alebo rezíduá.

$$y_t = x_t + z_t \quad t = 1, \dots, n$$

Cieľom úlohy trend filtering je odhadnúť podkladový trend alebo odhadnúť náhodný komponent, čo je ekvivalentná úloha. Od odhadnutého trendu požadujeme, aby bol

hladký (málo meniaci sa) a odhadnutý náhodný komponent (rezíduá) chceme čo najmenší. Trend filtering teda môžeme považovať za optimalizačnú úlohu s dvomi protichodnými cieľmi, ktoré budú neskôr vážené parametrom λ . Trend filtering nachádza uplatnenie napríklad v makroekonómii, v geofyzike a v analýze finančných časových radov. Existuje viacero metód na odhadovanie podkladového trendu, avšak my sa budeme venovať l_1 trend filtering, ktorý je modifikáciou známeho Hodrick-Prescottovho filtra. HP filter si objasníme v nasledujúcej časti.

4.1.2 Hodrick-Prescott filtering

HP filter sa využíva na získanie zhladenej krivky (trendový komponent) x_t dobre reprezentujúcej daný časový rad y_t . V prípade Hodrick-Prescottovho filtra sa odhadovaný podkladový trend volí tak, aby minimalizoval účelovú funkciu, ktorá je váženou sumou dvoch dobre interpretovateľných komponentov. Prvý komponent predstavuje veľkosť rezíduí a druhý, penalizačný komponent predstavuje hladkosť odhadovaného trendu.

$$\frac{1}{2} \sum_{t=1}^n (y_t - x_t)^2 + \lambda \sum_{t=2}^{n-1} (x_{t-1} - 2x_t + x_{t+1})^2$$

Regularizačný parameter $\lambda \geq 0$ sa používa na riadenie kompromisu medzi veľkosťou rezíduí a hladkosťou riešenia. Často je označovaný aj ako zhladzovací parameter, keďže kontroluje hladkosť odhadovaného trendu. Argumentom druhého komponentu je druhá diferenciacia časového radu v čase $t + 1$, je nulová práve vtedy, keď ležia tri body x_{t-1}, x_t a x_{t+1} na priamke. Preto druhá časť účelovej funkcie môže byť nulová len vtedy, keď je x_t afínne, teda je tvaru $x_t = \alpha + \beta t$. Účelovú funkciu môžeme zapísať taktiež maticovo:

$$\text{Min } \frac{1}{2} \|y - x\|_2^2 + \lambda \|Dx\|_2^2, \quad \lambda \geq 0 \quad (\text{HP})$$

kde $x, y \in R^n$ a $D \in R^{(n-2) \times n}$ je matica diferencie druhého rádu...

$$D = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \end{pmatrix}$$

Riešenie problému HP filtering možno vyjadriť explicitne ako

$$x^{hp} = (I + 2\lambda D^T D)^{-1} y$$

Z uvedeného vzťahu vidíme, že odhadnutý trend pomocou HP filtra je lineárnou funkciou časového radu (dát) y . Všimnime si taktiež vlastnosť konvergencie odhadu k originálnym dátam, keď $\lambda \rightarrow 0$. Pre $\lambda \rightarrow \infty$ konverguje odhad trendu k najlepšiemu afínnemu riešeniu fitujúcemu dáta, druhá časť účelovej funkcie musí byť nulová, pretože sa jej prikladá veľká váha.

4.1.3 l_1 trend filtering

V prípade l_1 filtra sa odhadovaný podkladový trend x_t volí tak, aby minimalizoval účelovú funkciu skladajúcu sa z váženej sumy dvoch častí. Prvá časť predstavuje veľkosť reziduí a druhá predstavuje hladkosť odhadovaného trendu. l_1 trend filter vznikol modifikáciou účelovej funkcie HP filtra, kde sa suma štvorcov druhých diferencií časového radu zamení za sumu ich absolútnych hodnôt (l_1 norma). Minimalizovaná účelová funkcia preto vyzerá nasledovne:

$$\frac{1}{2} \sum_{t=1}^n (y_t - x_t)^2 + \lambda \sum_{t=2}^{n-1} |x_{t-1} - 2x_t + x_{t+1}|$$

Tak ako v prípade HP filtra sa regularizačný parameter $\lambda \geq 0$ používa na korigovanie veľkosti reziduí verzus hladkosti riešenia. Taktiež sa dá účelová funkcia zapísať maticovo

$$\text{Min } \frac{1}{2} \|y - x\|_2^2 + \lambda \|Dx\|_1, \quad \lambda > 0 \quad (\text{L1})$$

kde $x, y \in R^n$ a $D \in R^{(n-2) \times n}$ je matica diferencie druhého rádu.

Vlastnosti

Rovnako ako v prípade HP filtrovania je účelová funkcia l_1 filtrovania striktne konvexná, existuje jednoznačné riešenie daného problému. Účelová funkcia však stratila vlastnosť diferencovateľnosti kvôli prítomnosti jednotkovej normy. Riešenie (L1) problému nevieme vyjadriť analytickou formulou tak ako v prípade HP filtrovania, odhadovaný trend už nie je lineárnou funkciou originálnych dát. Pre regularizačný parameter $\lambda \rightarrow 0$ konverguje riešenie k originálnym dátam a pre $\lambda \rightarrow \infty$ konverguje odhad trendu k najlepšiemu afínnemu riešeniu fitujúcemu dáta, kde druhá časť účelovej funkcie je nulová, pretože sa jej prikladá veľká váha. Narozdiel od problému HP filtering k tomuto javu dochádza skôr, konvergencia k najlepšiemu afínnemu riešeniu nastáva už pre konečnú hodnotu parametra λ ,

$$\lambda_{max} = \|(DD^T)^{-1}Dy\|_\infty$$

Pre $\lambda \geq \lambda_{max}$ sa už hľadané riešenie nemení, ostáva najlepším afínnym.

Lineárnosť po častiach

Najväčší rozdiel oproti HP filtrovaniu je, že l_1 filter produkuje odhady podkladového trendu, ktoré sú hladké v tom zmysle, že sú po častiach lineárnou funkciou v čase t (mení sa počet spájacích bodov, lineárnych častí). Z tohto dôvodu je l_1 filtrovanie vhodné na analýzu časových radov s podkladovým po častiach lineárnym trendom. Vzniknuté uzly (spájacie body), zmeny sklonu odhadnutého trendu je možné interpretovať ako náhle zmeny, udalosti v podkladovej dynamike časového radu. Preto je možné l_1 filtrovanie považovať za detekciu alebo odhadovanie zmien v podkladovom lineárnom trende časového radu. Čiastková linearita odhadnutého trendu je spôsobená práve pridaním jednotkovej normy do minimalizovanej účelovej funkcie. Riešenie takéhoto problému má zvyčajne argument v l_1 norme riedky (l_1 norma má sklony ťahať argument k nulovej hodnote). V našom prípade to spôsobuje, že argument Dx , čiže diferenciu druhého rádu odhadnutého trendu, má mnoho nulových členov, čo znamená, že odhadnutý trend musí byť po častiach lineárny.

Pri lineárnosti riešenia po častiach existujú celočíselné časy

$1 = t_1 < t_2 < \dots < t_{p-1} < t_p = n$, pre ktoré platí

$$x_t^{l_1} = \alpha_k + \beta_k t, \quad t_k \leq t \leq t_{k+1}, \quad k = 1, \dots, p-1$$

Inými slovami, v každom intervale $[t_i, t_{i+1}]$ je x^{l_1} afínnou funkciou času t . Parametre α_k, β_k určujúce afínnu funkciu na každom intervale nie sú nezávislé, musia podávať zhodné hodnoty funkcie v spájacích bodoch t_2, \dots, t_{p-1} (výsledný trend bude spojitá funkcia).

$$\alpha_k + \beta_k t_{k+1} = \alpha_{k+1} + \beta_{k+1} t_{k+1}, \quad k = 1, \dots, p-1$$

Ak $p = 2$, tak nemáme žiadne spájacie body. V tomto prípade sú časy $t_1 = 1, t_2 = n$ a riešenie je afínné. V prípade opačného extrému, keď máme spájací bod v každom časovom bode, sú časy $t_i = i, \quad i = 1, \dots, n$ a riešenie zodpovedá pôvodnému časovému radu (prípád $\lambda = 0$). Počet spájacích bodov zvyčajne s rastúcim regularizačným parametrom λ klesá.

Na l_1 filtrovanie sa možno pozerať tak, že produkuje segmentovú lineárnu regresiu s afínnym fitom na každom segmente s rovnakými hodnotami v uzloch. l_1 filtrovanie nájde segmentáciu aj afínný fit na každom segmente riešením jediného optimalizačného

problému.

Výhody l_1 filtra oproti HP

Základný dôvod, prečo preferovať odhad trendu pomocou l_1 filtering voči trendu získanému HP filtrom, môže byť práve vlastnosť l_1 filtra produkovať riešenia, ktoré sú po častiach lineárne. Ďalší veľmi dobrý dôvod je to, že štrukturálne zlomy a outliers v dátach neskresľujú výsledky l_1 filtra tak ako v prípade HP filtra. Pri použití HP filtra na dáta, ktoré obsahujú štrukturálny zlom, dôjde k 'zhladeniu'/zmierneniu zlomu a k rozšíreniu jeho efektu dopredu aj dozadu v čase. Z toho dôvodu odhadovaný trend pomocou HP filtra už nemusí v oblasti štrukturálneho zlomu dobre reprezentovať pôvodné dáta (bude vychýlený), čo ukazuje aj Obr.5. Navyiac l_1 filter nevyžaduje žiadnu znalosť o počte a lokáciách štrukturálnych zlomov v pôvodnom časovom rade.

Podmienky optimality

Účelová funkcia problému (L1) je konvexná (súčet kvadratickej funkcie a normy), avšak nie je diferencovateľná kvôli prítomnosti jednotkovej normy. Z uvedeného dôvodu použijeme podmienky optimality prvého rádu definované pomocou subgradientov. Využijeme výsledok z kapitoly 2.2, kde sme odvodili nutné a postačujúce podmienky optimality pre účelovú funkciu úlohy l_1 filtering. Podmienky vyzerajú nasledovne

$$((DD^T)^{-1}D(y-x))_i \in \begin{cases} \lambda & (Dx)_i > 0 \\ -\lambda & (Dx)_i < 0, \quad i = 1, \dots, n-2 \\ [-\lambda, \lambda] & (Dx)_i = 0 \end{cases}$$

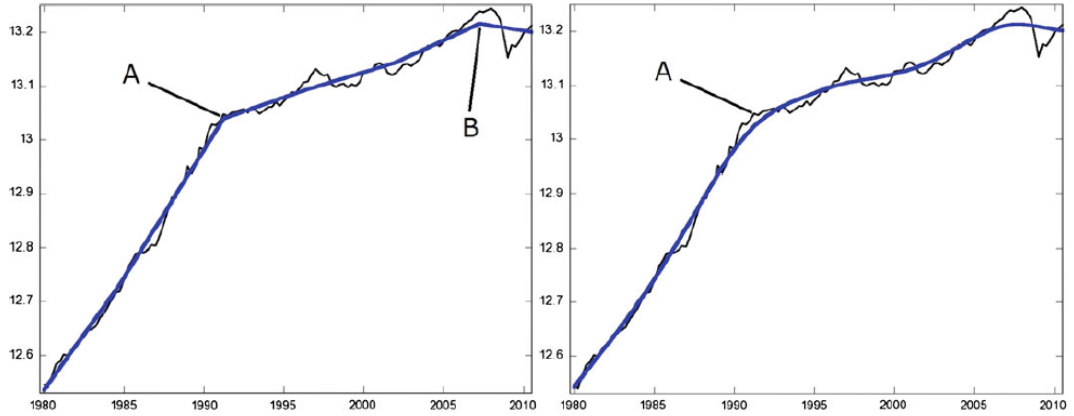
Ak definujeme zložky vektora $u \in R^{n-2}$ ako

$$u_i = \begin{cases} 1 & (Dx)_i > 0 \\ -1 & (Dx)_i < 0 \\ [-1, 1] & (Dx)_i = 0 \end{cases}$$

môžno predošlé podmienky prepísať ekvivalentne ako

$$y-x \in \lambda D^T u.$$

Z daných podmienok optimality sa dajú odvodiť niektoré zaujímavé vlastnosti l_1 filtrovania, niektoré z nich sme už spomenuli.



Obr. 5: Výhoda l_1 vs. HP: Produkčná medzera Japonska 1980-2010. Produkčná medzera je počítaná ako percentuálna odchýlka aktuálnej produkcie od potenciálnej produkcie. Potenciálna produkcia nie je priamo pozorovateľná, musí sa odhadovať. Na obrázku je vykreslený logaritmus HDP Japonska očistený od sezónnosti. Vidíme, že Japonská ekonomika zažila štrukturálne zmeny (bod A). Parameter λ l_1 filtrovania bol zvolený tak, aby štvorce rezíduí boli rovnaké ako v prípade HP filtra pri voľbe $\lambda = 1600$, čo je bežne používaná hodnota pre štvrtročné dáta. l_1 filter zachytil štrukturálnu zmenu v ekonomike Japonska oveľa lepšie ako HP filter, ktorý danú zmenu zahladil (odhad je vychýlený). Logaritmus HDP Japonska je vhodné modelovať po častiach lineárnym trendom. Obrázky pochádzajú zo zdroja [5].

Horné ohraničenie chyby

Pre každý vektor $u \in R^{n-2}$, ktorého zložky pochádzajú z intervalu $u_i \in [-1, 1]$ určite platí nerovnosť $-4\lambda \leq \lambda(D^T u)_i \leq 4\lambda$. Keďže podľa podmienky optimality vektor $y - x$ patrí do množiny $\lambda D^T u$, tak aj jeho každá zložka musí spĺňať predošlú nerovnosť, z čoho následne vyplýva vzťah horného ohraničenia chyby (maximum fitting error bound)

$$\|y - x\|_\infty \leq 4\lambda \quad (10)$$

Vzťah (10) implikuje konvergenciu odhadnutého trendu k originálnym dátam, keď regularizačný parameter $\lambda \rightarrow 0$.

Z podmienok optimality sa dá taktiež odvodiť hodnota regularizačného parametra λ_{max} , pri ktorej je nami odhadnutý trend už najlepším afínnym riešením reprezentujúcim pôvodné dáta. Pre riešenie, ktoré je afínné, určite platí $Dx^a = 0$. Preto sa

podmienka optimality pre riešenie x^a zredukuje na tvar

$$((DD^T)^{-1}Dy)_i \in [-\lambda, \lambda], \quad i = 1, \dots, n-2$$

Z čoho priamo vyplýva vzťah $\|(DD^T)^{-1}Dy\|_\infty \leq \lambda$, a preto za hodnotu λ_{max} , pri ktorej je už riešenie afínne, možno položiť $\lambda_{max} = \|(DD^T)^{-1}Dy\|_\infty$.

Mohlo by nás taktiež zaujímať, za akých podmienok možno odhadnutý trend pri nemennom λ iba extrapolovať (rozšíriť), ak nám do originálneho časového radu pribudne nové pozorovanie. Aj túto otázku je možné zodpovedať pomocou odvodených podmienok optimality pre úlohu l_1 filtering. Stačí iba overiť, či odhadnutý trend spolu s extrapolovanou hodnotou spĺňa podmienku

$$\|(DD^T)^{-1}D\left(\begin{pmatrix} y \\ y_{n+1} \end{pmatrix} - \begin{pmatrix} x^{l1} \\ 2x_n^{l1} - x_{n-1}^{l1} \end{pmatrix}\right)\|_\infty \leq \lambda$$

Z predošlej nerovnice je možné odvodiť dolnú aj hornú hranicu pre nové pozorovanie y_{n+1} , aby stačilo použiť pri riešení extrapoláciu.

4.1.4 Duálna úloha

V tejto časti odvodíme Lagrangeovu duálnu úlohu k úlohe l_1 filtering, ktorá bude neskôr kľúčová pre riešenie daného problému pomocou metódy vnútorného bodu. Na to, aby sme odvodili duálnu úlohu potrebujeme najprv vykonať jeden umelý krok - zavedieme novú premennú $z \in R^{n-2}$ a pridáme nové ohraňenie v tvare rovnosti $z = Dx$. Danou substitúciou dostaneme ekvivalentnú formuláciu l_1 filtering problému v tvare minimalizačnej úlohy s ohraňením:

$$\begin{aligned} \text{Min} \quad & \frac{1}{2}\|y - x\|_2^2 + \lambda\|z\|_1 \\ & z = Dx \end{aligned}$$

Pre ekvivalentnú formuláciu vieme zostrojiť Lagrangeovu funkciu, kde $v \in R^{n-2}$ je duálna premenná prislúchajúca k ohraňeniu v tvare rovnosti $z = Dx$.

$$L(x, z, v) = \frac{1}{2}\|y - x\|_2^2 + \lambda\|z\|_1 + v^T(Dx - z)$$

Lagrangeovu duálnu funkciu získame ako $G(v) = \inf_{x,z} L(x, z, v)$. Lagrangeova funkcia je konvexná v premenných x, z a teda podľa Vety 2.1. funkcia nadobúda minimum, ak

do subdiferenciálu funkcie patrí nulový vektor

$$\frac{\partial L}{\partial x} = x - y + D^T v = 0 \Rightarrow x = y - D^T v$$

$$\frac{\partial L}{\partial z_i} = -v_i + \lambda \begin{cases} 1 & z_i > 0 \\ -1 & z_i < 0 \\ [-1, 1] & z_i = 0 \end{cases} \ni 0 \Rightarrow v_i \in \begin{cases} \lambda & z_i > 0 \\ -\lambda & z_i < 0 \\ [-\lambda, \lambda] & z_i = 0 \end{cases}$$

Prvý vzťah nám hovorí ako dostať z riešenia v^{l_1} duálnej úlohy riešenie x^{l_1} primárnej úlohy, ktorú chceme riešiť. Všimnime si taktiež, že v optime je súčet členov (v Lagrangeovej funkcii) $\lambda \|z\|_1$ a $-v^T z$ vždy rovný nulovej hodnote, čo je vidieť z druhého vzťahu. Po použití oboch podmienok dostávame

$$\inf_{x,z} L(x, z, v) = \begin{cases} -\frac{1}{2} v^T D D^T v + v^T D y & v_i \in [-\lambda, \lambda] \\ -\infty & \text{inak} \end{cases}$$

Lagrangeova duálna úloha má potom tvar

$$\begin{aligned} \text{Min } & \frac{1}{2} v^T D D^T v - y^T D^T v \\ & - \lambda \mathbf{1} \leq v \leq \lambda \mathbf{1} \end{aligned} \tag{DL1}$$

Výsledný duálny problém (DL1) je úlohou konvexného kvadratického programovania s kvadratickou účelovou funkciou a lineárnymi ohraňeniami.

V nasledujúcom texte najprv predstavíme obvyklý spôsob riešenia úlohy l_1 trend filtering, v ktorom sa využíva metóda vnútorného bodu použitá na duálnu úlohu (DL1). Náplňou druhej časti je pokus o aplikovanie proximal gradient metódy na primárnu úlohu (L1). Tieto dva rôzne prístupy porovnáme, spomenieme ich výhody/nevýhody.

4.1.5 Riešenie pomocou primárno-duálnej metódy vnútorného bodu

Na riešenie duálnej úlohy (DL1) konvexného (kvadratického) programovania existuje mnoho metód. V našom prípade dosiahneme riešenie úlohy pomocou primárno-duálnej metódy vnútorného bodu, z ktorého jednoduchou transformáciou $x^{l_1} = y - D^T v^{l_1}$ získame hľadané riešenie úlohy l_1 filtering. V nasledujúcom texte budeme považovať za primárnu úlohu (DL1) s primárnou premennou v (aj keď sú duálne k úlohe (L1)).

Vďaka použitiu primárno-duálnej metódy vnútorného bodu budú primárne aj duálne

premenné aktualizované v každej iterácii.

Primárno-duálny hľadací smer

Základom primárno-duálnej metódy vnútorného bodu je tzv. hladací smer. Hladací smer je určený Newtonovým krokom pre systém nelineárnych rovníc $r_t(v, \mu_1, \mu_2) = 0$, ktorým sú vlastne modifikované Karush-Kuhn-Tuckerové (KKT) podmienky. Preto ako prvý krok odvodíme KKT podmienky pre úlohu (DL1), ktoré následne modifikujeme na požadovaný tvar.

Lagrangeova funkcia pre úlohu (DL1) je daná ako

$$L(v, \mu_1, \mu_2) = \frac{1}{2}v^T DD^T v - y^T D^T v + \mu_1^T (v - \lambda \mathbf{1}) + \mu_2^T (-v - \lambda \mathbf{1})$$

kde $\mu_1, \mu_2 \geq 0_{n-2}$ sú duálne premenné prislúchajúce k ohraničeniam $v \leq \lambda \mathbf{1}$ a $-v \leq \lambda \mathbf{1}$ úlohy (DL1). KKT podmienky budú vyzerat

$$\begin{array}{ccc} v & \mu_1 & \mu_2 \\ & \mu_{1i} \geq 0 & \mu_{2i} \geq 0 \\ \frac{\partial L}{\partial v} = 0 & \frac{\partial L}{\partial \mu_{1i}} \mu_{1i} = 0 & \frac{\partial L}{\partial \mu_{2i}} \mu_{2i} = 0 \quad i = 1, \dots, n-2 \\ & \frac{\partial L}{\partial \mu_{1i}} \leq 0 & \frac{\partial L}{\partial \mu_{2i}} \leq 0 \end{array}$$

$$\mu_1 \geq 0, \mu_2 \geq 0, v - \lambda \mathbf{1} \leq 0, -v - \lambda \mathbf{1} \leq 0$$

$$DD^T v - Dy + \mu_1 - \mu_2 = 0 \tag{KKT}$$

$$diag(\mu_1)(v - \lambda \mathbf{1}) = 0$$

$$diag(\mu_2)(-v - \lambda \mathbf{1}) = 0$$

Pridaním parametra $t > 0$ do podmienok komplementarity (posledné dve rovnice KKT podmienok) získame modifikovaný systém rovníc (mKKT).

$$r_t(v, \mu_1, \mu_2) = \begin{pmatrix} DD^T v - Dy + \mu_1 - \mu_2 \\ -diag(\mu_1)(v - \lambda \mathbf{1}) - \frac{1}{t} \mathbf{1} \\ -diag(\mu_2)(-v - \lambda \mathbf{1}) - \frac{1}{t} \mathbf{1} \end{pmatrix} \tag{mKKT}$$

Nech je premenná v je primárne prípustná a premenné μ_1, μ_2 sú duálne prípustné. Ak premenné v, μ_1, μ_2 spĺňajú systém rovníc $r_t(v, \mu_1, \mu_2) = 0$, potom $v = v^*(t)$, $\mu_1 = \mu_1^*(t)$ a $\mu_2 = \mu_2^*(t)$. Ak $t \rightarrow \infty$, tak sa systém rovníc $r_t(v, \mu_1, \mu_2) = 0$ redukuje na

pôvodné KKT podmienky. Hlavnou myšlienkou primárno-duálnej metódy je použiť Newtonovské kroky na vyriešenie série nelineárnych rovníc $r_t(v, \mu_1, \mu_2) = 0$ pre rastúcu postupnosť hodnôt parametra t .

Teraz uvažujme Newtonovský krok pre riešenie nelineárnych rovníc $r_t(v, \mu_1, \mu_2) = 0$ pre fixnú hodnotu parametra t v bode (v, μ_1, μ_2) , ktorý spĺňa $\mu_1 > 0$, $\mu_2 > 0$, $v - \lambda \mathbf{1} < 0$, $-v - \lambda \mathbf{1} < 0$. Označme súčasný bod $w = (v, \mu_1, \mu_2)$ a Newtonovský krok $\Delta w = (\Delta v, \Delta \mu_1, \Delta \mu_2)$. Potom je Newtonovský krok určený lineárnymi rovnicami

$$r_t(w + \Delta w) \approx r_t(w) + Dr_t(w)\Delta w = 0$$

teda je potrebné riešiť sústavu lineárnych rovníc $Dr_t(w)\Delta w = -r_t(w)$, kde Dr_t predstavuje Jakobián r_t . V premenných v, μ_1, μ_2 potom dostávame rovnice

$$\left(\begin{array}{ccc|c} DD^T & I & -I & -DD^T v + Dy - \mu_1 + \mu_2 \\ -diag(\mu_1) & -diag(v - \lambda \mathbf{1}) & 0 & diag(\mu_1)(v - \lambda \mathbf{1}) + \frac{1}{t} \mathbf{1} \\ diag(\mu_2) & 0 & diag(v + \lambda \mathbf{1}) & diag(\mu_2)(-v - \lambda \mathbf{1}) + \frac{1}{t} \mathbf{1} \end{array} \right)$$

Keďže by na získanie kroku bolo neefektívne riešiť daný systém rovníc, chceme eliminovaním premenných $\Delta \mu_1, \Delta \mu_2$ z prvej rovnice dostať redukovaný systém trojuholníkového tvaru. Prenásobením druhej rovnice inverznou maticou $diag(\mu_1)^{-1}$, tretej rovnice maticou $diag(\mu_2)^{-1}$ a substitúciou $J_1 = diag(\mu_1)^{-1}diag(v - \lambda \mathbf{1})$, $J_2 = diag(\mu_2)^{-1}diag(-v - \lambda \mathbf{1})$ získame

$$\left(\begin{array}{ccc|c} DD^T & I & -I & -DD^T v + Dy - \mu_1 + \mu_2 \\ -I & -J_1 & 0 & (v - \lambda \mathbf{1}) + \frac{1}{t} diag(\mu_1)^{-1} \mathbf{1} \\ I & 0 & -J_2 & (-v - \lambda \mathbf{1}) + \frac{1}{t} diag(\mu_2)^{-1} \mathbf{1} \end{array} \right)$$

Druhú(tretiu) rovnicu pre násobíme inverznou maticou J_1^{-1} ($-J_2^{-1}$), tieto inverzie sú ľahko spočítateľné, pretože sa jedná o diagonálne matice.

$$\left(\begin{array}{ccc|c} DD^T & I & -I & -DD^T v + Dy - \mu_1 + \mu_2 \\ -J_1^{-1} & -I & 0 & J_1^{-1}(v - \lambda \mathbf{1}) + \frac{1}{t} diag(v - \lambda \mathbf{1})^{-1} \mathbf{1} \\ -J_2^{-1} & 0 & I & J_2^{-1}(v + \lambda \mathbf{1}) - \frac{1}{t} diag(-v - \lambda \mathbf{1})^{-1} \mathbf{1} \end{array} \right)$$

Využijeme fakt, že $J_1^{-1}(v - \lambda \mathbf{1}) = diag(v - \lambda \mathbf{1})^{-1}diag(\mu_1)(v - \lambda \mathbf{1}) = \mu_1$ (podobne pre μ_2). Následne pripočítame druhú aj tretiu rovnicu k prvej. Tým dostaneme systém

$$\left(\begin{array}{ccc|c} DD^T - J_1^{-1} - J_2^{-1} & 0 & 0 & -DD^T v + Dy + \frac{1}{t} diag(v - \lambda \mathbf{1})^{-1} \mathbf{1} - \frac{1}{t} diag(-v - \lambda \mathbf{1})^{-1} \mathbf{1} \\ -J_1^{-1} & -I & 0 & \mu_1 + \frac{1}{t} diag(v - \lambda \mathbf{1})^{-1} \mathbf{1} \\ -J_2^{-1} & 0 & I & -\mu_2 - \frac{1}{t} diag(-v - \lambda \mathbf{1})^{-1} \mathbf{1} \end{array} \right)$$

Na nájdenie Newtonovského kroku nám stačí vyriešiť redukovaný systém (11), ktorého riešenie Δv využijeme na dopočítanie ďalších dvoch zložiek Newtonovho kroku.

$$(DD^T - J_1^{-1} - J_2^{-1})\Delta v = -DD^T v + Dy + \frac{1}{t} \text{diag}(v - \lambda \mathbf{1})^{-1} \mathbf{1} - \frac{1}{t} \text{diag}(-v - \lambda \mathbf{1})^{-1} \mathbf{1} \quad (11)$$

Systém lineárnych rovníc (11) má aj tú výhodu, že matica $(DD^T - J_1^{-1} - J_2^{-1})$ je riedka (presnejšie 5-diagonálna). Z toho dôvodu je lacno riešiteľný. Ďalšie dva komponenty kroku $\Delta\mu_1, \Delta\mu_2$ získame zo vzťahov

$$\begin{aligned} \Delta\mu_1 &= -\mu_1 - \frac{1}{t} \text{diag}(v - \lambda \mathbf{1})^{-1} \mathbf{1} - J_1^{-1} \Delta v \\ \Delta\mu_2 &= -\mu_2 - \frac{1}{t} \text{diag}(-v - \lambda \mathbf{1})^{-1} \mathbf{1} + J_2^{-1} \Delta v \end{aligned} \quad (12)$$

kde J_1 a J_2 sú diagonálne. Primárno-duálny smer $\Delta w = (\Delta v, \Delta\mu_1, \Delta\mu_2)$ je teda definovaný ako riešenie (11) a (12). Všimnime si, že všetky používané matice sú riedke, čo významne urýchluje výpočet systému pri implementácii metódy.

4.1.6 Riešenie pomocou Proximal gradient metódy

V predošlej časti sme riešili úlohu (L1) pomocou metódy vnútorného bodu aplikovanej na jej duálnu úlohu (DL1), ktorá je úlohou konvexného programovania. V tejto časti budeme úlohu riešiť aplikovaním proximal gradient metódy priamo na primárnu úlohu (L1). Keďže sa v prípade úlohy (L1) jedná o neohraničený problém, kde minimalizujeme účelovú funkciu rozložiteľnú na diferencovateľnú konvexnú a nediferencovateľnú, uzavretú konvexú časť, môžeme na jej optimalizáciu použiť Proximal gradient metódu. Za týmto účelom sa pokúsime odvodiť základný prvok metódy - proximálny operátor pre nediferencovateľnú časť $h(u) = \lambda \|Du\|_1$. Ako prvé si treba uvedomiť, že funkcia $h(u)$ nie je separovateľná, a preto nemôžeme použiť vlastnosť proximálneho operátora, ktorú sme použili pri odvodení proximálneho operátora klasickej jednotkovej normy. Na výpočet proximálneho operátora využijeme definíciu proximálneho operátora pomocou subgradientu (Veta 3.1.).

$$u = \text{prox}_{th}(x) \Leftrightarrow x - u \in t\partial h(u)$$

$$x - u \in t\lambda\partial \|Du\|_1$$

Úpravami dostávame vzťah

$$x - u \in t\lambda D^T v$$

kde vektor $v \in R^{n-2}$ má zložky

$$v_i = \begin{cases} 1 & (Du)_i > 0 \\ -1 & (Du)_i < 0 \\ [-1, 1] & (Du)_i = 0 \end{cases}$$

Problémom je však to, že proximálny operátor u je z uvedeného vzťahu ťažko vyjadriteľný. Ak chceme použiť na riešenie úlohy (L1) proximal gradient metódu, je potrebné riešiť ekvivalentnú formuláciu daného problému. Ekvivalentná formulácia l_1 trend filtering problému vyzerá nasledovne:

$$\text{Min}_{\theta} \frac{1}{2} \|A\theta - y\|_2^2 + \lambda \sum_{i=3}^n |\theta_i| \quad (\text{eL1})$$

kde $\theta \in R^n$ je premenná a matica $A \in R^{n \times n}$ je dolná trojuholníková matica tvaru

$$A = \begin{pmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ 1 & 2 & 1 & & & \\ 1 & 3 & 2 & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & 1 & \\ 1 & n-1 & n-2 & \dots & 2 & 1 \end{pmatrix}$$

Riešenie pôvodného problému (L1) x^{l1} získame zo vzťahu $x^{l1} = A\theta^{l1}$. Poznamenajme, že ekvivalentná formulácia sa dá taktiež riešiť primárno-duálnou metódou vnútorného bodu ako v prípade pôvodnej úlohy. Problém (eL1) je však už riešiteľný aj pomocou proximal gradient metódy. Žiaľ formulácia (eL1) má pomerne nepríjemnú nevýhodu oproti klasickej formulácii. Pôvodná úloha mala jednoduchý zápis, pracovala s riedkymi maticami I, D (čo významne urýchluje následný výpočet riešenia pri veľkých rozmeroch). Vo formulácii (eL1) sa však nachádza matica A , ktorá je dolná trojuholníková, čiže jej prvky sú oveľa hustejšie zaplnené.

Ekvivalencia úloh (L1) a (eL1)

Ako prvé si ukážeme, že úlohy (L1) a (eL1) sú naozaj ekvivalentné. Úlohu (eL1) možno zapísať aj maticovo

$$\text{Min} \frac{1}{2} \|A\theta - y\|_2^2 + \lambda \|F\theta\|_1 \quad (\text{eL1v2})$$

kde matica

$$F = \begin{pmatrix} 0 & 0 & 1 & & \\ 0 & 0 & & 1 & \\ \vdots & \vdots & & & \ddots \\ 0 & 0 & & & & 1 \end{pmatrix}$$

Prvé komponenty oboch formulácií (L1) a (eL1) sa očividne rovnajú vďaka vzťahu $x = A\theta$, kde matica A má plnú hodnosť, je regulárna.

Pre druhé (penalizačné) komponenty účelových funkcií platí:

$$(L1) \Rightarrow (eL1) : \quad Dx = DA\theta = F\theta$$

$$(L1) \Leftarrow (eL1) : \quad F\theta = FA^{-1}x = Dx$$

Teda uvedené vzťahy (L1) a (eL1) sú naozaj ekvivalentné. Interpretácia koeficientov θ_i je taká, že prvá zložka ($i=1$) je rovná prvej zložke časového radu x , druhá zložka je diferenciac prvého rádu $\theta_2 = x_2 - x_1$ a ostatné zložky vektora θ sú diferenciami druhého rádu časového radu x .

Pri odvodení proximálneho operátora pre nediferencovateľnú časť účelovej funkcie (eL1) $h(u) = \lambda \sum_{i=3}^n |u_i|$ si treba uvedomiť separovateľnosť danej funkcie. Vďaka tejto vlastnosti nám stačí vyjadriť proximálny operátor pre jednotlivé zložky. Stačí si všimnúť, že funkcia h dáva pre prvé dve zložky vektora u nulovú hodnotu, preto pre $i = 1, 2$ platí $\hat{u}_i = x_i$. Pre $i \geq 3$ je postup odvodenia proximálneho operátora analogický postupu pre proximálny operátor jednotkovej normy. Podrobnejšie odvodenie proximal operátora pre l_1 normu sa nachádza v kapitole 4.2.3. Výsledný proximálny operátor pre nediferencovateľnú časť (eL1) je

$$i = 1, 2 : \quad prox_{th}(x)_i = x_i$$

$$i = 3, \dots, n : \quad prox_{th}(x)_i = \begin{cases} x_i - t\lambda & x_i \geq t\lambda \\ x_i + t\lambda & x_i \leq -t\lambda \\ 0 & |x_i| \leq t\lambda \end{cases}$$

Metóda	PDMVB	APGM	APGM	APGM
Čas v sec	0.08	5	50	500
Počet it.	84	13410	135897	1343466
Suboptimalita	2.85E-05	4.52E+04	2.06E+04	9.07E+00

Tabuľka 1: Porovnanie metód - primárno-duálna metóda vnútorného bodu (PDMVB), zrýchlená proximal gradient metóda (APGM). Riadok suboptimalita predstavuje duálnu medzeru medzi hodnotami primárnej a fixne zvolenej duálnej účelovej funkcie. Za hodnotu duálnej účelovej funkcie bola zobrazená hodnota z MVB = 64114.08 (ktorá je takmer optimálna).

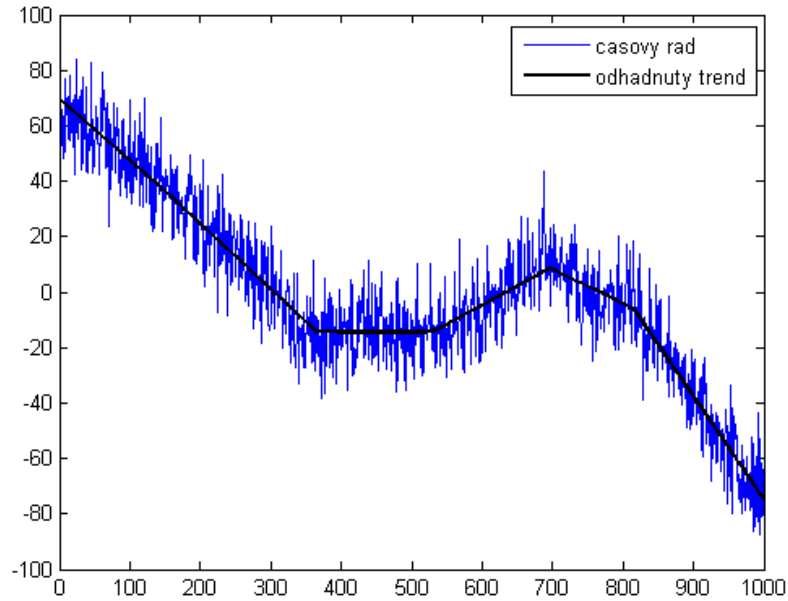
blíži k správnej optimálnej hodnote, ale na to, aby sme dosiahli dobré suboptimálne riešenie je potrebné veľa času v zrovnaní s metódou vnútorného bodu. Riešenie pomocou PG dosiahnuté za 500s však už pomerne dobre reprezentuje podkladový počastiach lineárny trend, čo možno pozorovať aj na Obr. 7. Z daného obrázka je vidieť aj to, ako sa riešenia s narastajúcim časom približujú k riešeniu pomocou MVB.

Z vyššie uvedených faktov možno skonštatovať, že pri riešení l_1 filtrovania primárno-duálna metóda vnútorného bodu vysoko prekonáva zrýchlenú proximal gradient metódu. Príčina tohto javu je nasledujúca: Zatiaľ, čo sa v algoritme primárno-duálnej metódy vnútorného bodu (pri hľadaní Newtonovho kroku) pracuje iba s riedkymi maticami, proximal gradient metóda musí pracovať s dolnou trojuholníkovou maticou A , resp. s maticou $A^T A$ (pri výpočte gradientu), ktorej prvky sú nenulové - najmenší prvok $A^T A$ je 1 a maximálny je 332833500 ($t = 1000$). Táto matica má taktiež obrovské najväčšie vlastné číslo $\alpha_{max}(A^T A) = 8.1e + 010$. Zlá vlastnosť matice A je aj tá, že je zle podmienená. Číslo podmienenosti matice je rovné

$$C(A) = \|A\| \|A^{-1}\| = 1.14E + 06$$

a v prípade matice $A^T A$ je $C(A^T A) = 8.1e + 010$. Maticu A využívame pri dopočítaní hľadaného riešenia $x = A\theta$ a v algoritme proximal gradient metódy pri počítaní gradientu $g = A^T A * \theta - A^T y$.

Z Vety 3.5. vieme, že zrýchlená proximal gradient metóda konverguje, ak je zvolený konštantný krok $t = 1/\alpha_{max}(A^T A) = 1.2e - 011$. Z tejto vety takisto vidíme, že má pre



Obr. 6: Časový rad s podkladovým po častiach lineárnym trendom. Odhadnutý trend na obrázku je získaný pomocou primárno-duálnej metódy vnútorného bodu.

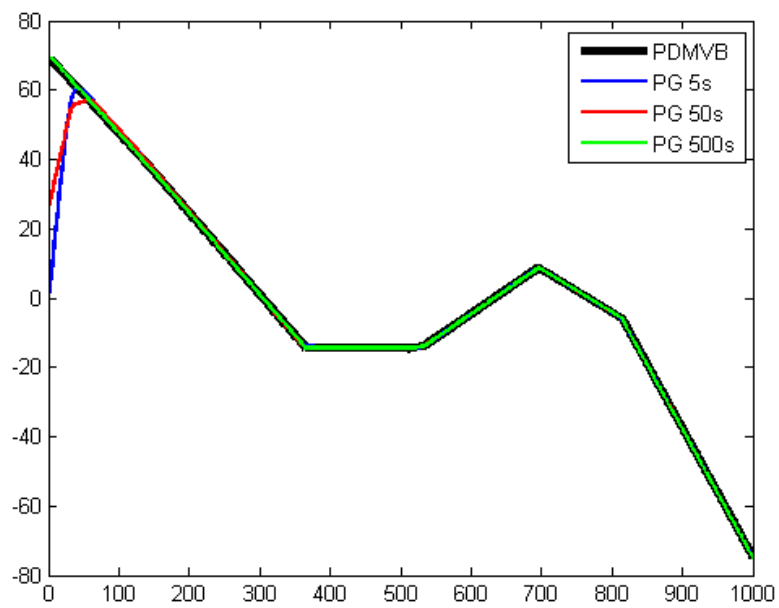
naše riešenie platí nasledovná nerovnosť

$$f(x^{(k)}) - \hat{f} \leq \frac{2}{(k+1)^2 t} \|x^{(0)} - \hat{x}\|_2^2$$

z čoho po dosadení dostávame

$$9.07 \leq \frac{2}{1343467 * 1.2e - 011} * 9.2689e + 005 = 8.3081e + 004.$$

Pre garantovanie lepšej suboptimálnosti nášho riešenia by sme potrebovali omnoho viac iterácií. Formulácia (eL1) nie je prospešná pri riešení úlohy l_1 filtrovania.



Obr. 7: Konvergencia Proximal gradient metódy. Odhadovaný trend (riešenie) získaný pomocou proximal gradient konverguje k optimálnemu riešeniu dosiahnutým pomocou PDMVB. Odhadnutý trend pomocou APGM za 500s už pomerne dobre vystihuje skutočný trend.

4.2 Truss topology design

Jednou z úloh, ktoré sa dajú naformulovať ako úloha najmenších štvorocov regulovaných l_1 normou (1), je aj úloha zvaná Truss Topology Design (TTD). Jej názov by sa dal voľne preložiť ako "dizajn rozloženia konštrukcie". V procese navrhovania mechanických štruktúr ako železničné mosty, stĺpy elektrického vedenia, Eiffelova veža, ale aj lietadlá a budovy je potrebné nadizajnovať systém pospájaných priečok - Truss, ktorý je schopný optimálnym spôsobom odolať silám, ktoré naň pôsobia. Pri písaní kapitoly sme použili zdroje [7], [9], [6] a [10].

Truss je mechanická konštrukcia zložená z elastických priečok rôznych objemov pospájaných medzi sebou v tzv. uzloch, ktoré môžu byť fixné alebo voľné. Fixné uzly sa nemôžu pohybovať, sú pevne ukotvené, môžeme si pod nimi predstaviť základy budovy alebo piliere mostu. Konštrukcia môže byť vystavená externému zaťaženiu, simultánne pôsobiacim silám, ktoré pôsobia na uzly. Pod zaťažením sa konštrukcia mierne deformuje, priečky sa natahujú/napínajú a voľné uzly menia svoju polohu, až pokým budú externé sily kompenzované pnutím konštrukcie a systém bude v ekvilibriu. V ekvilibriu deformovaná konštrukcia uskladňuje potenciálnu energiu, táto energia je taktiež nazývaná poddajnosť (angl. compliance) konštrukcie. Čím menšia je poddajnosť konštrukcie, tým je pevnejšia. Cieľom Truss topology design je zostrojiť podpernú konštrukciu (danej celkovej váhy), ktorá je schopná najlepšie ustáť záťaž v takom zmysle, že jej poddajnosť vzhľadom na danú záťaž bude čo najmenšia.

V najjednoduchšom probléme TTD máme danú:

- mriežku uzlov, ktorou je konečná množina bodov v rovine alebo v priestore podľa toho, či chceme zostrojiť konštrukciu v 2D alebo 3D. V týchto bodoch je priečkam konštrukcie povolené spájať sa,
- množinu fixných uzlov, je to podmnožina mriežky uzlov, v ktorej sa nachádzajú iba ukotvené/nehybne body, napríklad umiestnené na zemi,
- záťaž, ktorá je kolekciou externých síl (vektorov), ktoré pôsobia na uzly.

Cieľom je pre danú mriežku uzlov a pre sily, ktoré na uzly pôsobia, zostrojiť podpernú konštrukciu (danej celkovej váhy) s minimálnou poddajnosťou a maximálnou pevnosťou. Chceme, aby konštrukcia odolávala silám optimálne.

Atraktívna vlastnosť úlohy TTD je, že nájde optimálny geometrický tvar (rozlože-

nie priečok) konštrukcie, aj keď navonok pracuje len s váhami priečok konštrukcie. Optimalizačný problém rozhodne, ktoré uzly použiť, ako ich pospájať a nájde aj optimálne váhy priečok. Aj keď začneme optimalizáciu problému so všetkými uzlami pospájanými priečkami, optimalizačný proces nájde optimálnu konštrukciu, v ktorej je väčšina priečok s nulovými váhami.

4.2.1 Odvodenie modelu

Úloha TTD má mnoho formulácií, ktoré sú riešiteľné rôznymi metódami matematického programovania. Problém TTD je možné zapísať napríklad vo forme úlohy lineárneho, kvadratického či semidefinitného programovania. Formulácia TTD ako úlohy najmenších štvorcov regulovaných l_1 normou je vhodná na aplikáciu súradnicovo spádových metód (Coordinate Descent Methods), čo presvedčivo demonštruje článok P. Richtárika a M. Takáča [9]. Táto formulácia je takisto vhodná na aplikáciu Proximal gradient metódy. V kapitole si ukážeme niektoré z možných formulácií TTD problému, začneme odvodením modelu.

Aby sme mohli TTD problém zapísať ako optimalizačnú úlohu, je potrebné sa bližšie pozrieť na to, čo sa stane s konštrukciou pri aplikovaní záťaže. Uvažujme jednu priečku AB nezaťaženej konštrukcie. Pod záťažou sa uzly A a B mierne vychýlia o vektor dA resp. dB , priečka AB sa natiahne. Zanedbaním členov druhého rádu môžeme predĺženie dl priečky AB vyjadriť ako projekciu vektora $dB-dA$ na smer pôvodnej priečky:

$$dl = (dB - dA)^T(B - A)/\|B - A\|$$

Veľkosť reakčnej sily (napnutosť angl. tension) spôsobenej týmto predĺžením je podľa Hookovho zákona rovná

$$\tau = \frac{E \times dl \times S_{AB}}{\|B - A\|} = \frac{E \times dl \times t_{AB}}{\|B - A\|^2} = Et_{AB}(dB - dA)^T(B - A)\|B - A\|^{-3}$$

kde E je modul pružnosti v ťahu (Youngov modul), S_{AB} je prierez priečky AB , t_{AB} je objem priečky.

Reakčná sila v bode B prislúchajúca k napnutiu je vektor

$$\begin{aligned} -\tau(B - A)\|B - A\|^{-1} &= Et_{AB}[(dB - dA)^T(B - A)](B - A)\|B - A\|^{-4} \\ &= -t_{AB}[(dB - dA)^T\beta_{AB}]\beta_{AB} \\ \beta_{AB} &= \sqrt{E}(B - A)\|B - A\|^{-2} \end{aligned} \tag{13}$$

Všimnime si, že vektor β_{AB} nezávisí od záťaže ani od predĺženia, závisí len od pozícií spojených uzlov.

Ďalšie, čo nás bude zaujímať za účelom odvodenia modelu, je potenciálna energia uložená priečkou ako dôsledok predĺženia. Z mechaniky vieme, že potenciálna energia je rovná

$$\frac{\text{napnutosť} \times \text{predĺženie}}{2},$$

teda

$$\frac{\tau dl}{2} = \frac{Et_{AB}[(dB - dA)^T(B - A)]^2 \|B - A\|^{-4}}{2} = \frac{t_{AB}[(dB - dA)^T \beta_{AB}]^2}{2} \quad (14)$$

Na základe predošlých poznatkov z fyziky zostavíme prvý model. Uvažujme rxc mriežku uzlov, z nich M je voľných (fixných je rxc- M). Definujeme m -rozmerný priestor posunutí konštrukcie, kde $m = 2M$ (resp. $m = 3M$ ak 3D). Vektor $v \in R^m$ reprezentuje posunutie mriežky uzlov, k voľnému uzlu ν prislúcha jeden pár indexov vektora v , subvektor $v[\nu]$ predstavuje 2-rozmerné posunutie. Kolekcia externých 2D síl (zaťaženie) pôsobiaca na voľné uzly je reprezentovaná vektorom $f \in R^m$, subvektor $f[\nu]$ je sila pôsobiaca na uzol ν . Poznamenajme, že nemá zmysel hovoriť o pôsobení externých síl na fixný uzol. Povolíme, aby každý pár uzlov (kde je aspoň jeden uzol voľný) mohol byť prepojený priečkou. Nech n označuje počet takýchto potenciálnych priečok a i -ta priečka spája dva uzly $\nu_1(i)$, $\nu_2(i)$. Definujeme vektor $a_i \in R^m$ prislúchajúci k i -tej hrane nasledujúcim spôsobom

$$a_i[\nu] = \begin{cases} \beta_{\nu_1(i)\nu_2(i)} & \nu = \nu_2(i) \text{ a } \nu \text{ je voľný} \\ -\beta_{\nu_1(i)\nu_2(i)} & \nu = \nu_1(i) \text{ a } \nu \text{ je voľný} \\ 0 & \text{inak} \end{cases} \quad (15)$$

Všimnime si, že vektory a_i sú veľmi riedke, každý obsahuje najviac 4 nenulové zložky. Majme nejaké usporiadanie prípustných priečok. Potom jednotlivé mechanické konštrukcie môžeme identifikovať pomocou nezáporného vektora objemov (resp. váh) hrán $t \in R^n$, kde i -ta zložka vektora predstavuje objem i -tej prípustnej hrany konštrukcie. Uvažujme konštrukciu t a pozrime sa na reakčné sily spôsobené posunutím uzlov kontrukcie v . Zo vzťahu pre reakčnú silu v jednom bode (13) dostaneme, že komponent reakčnej sily spôsobenej i -tou priečkou vo vrchole ν je $-t_i(a_i^T v)a_i[\nu]$. Následne celková reakčná sila v uzle ν je sumou reakčných síl vo vrchole ν spôsobených všetkými

n priečkami. Kolekcia reakčných síl na uzly spôsobená posunutím v je reprezentovaná m -rozmerným vektorom f_r , ktorý závisí od v lineárne

$$f_r = - \sum_{i=1}^n t_i a_i^T v a_i = - \sum_{i=1}^n t_i a_i a_i^T v = -K(t)v \quad (16)$$

$$K(t) = \sum_{i=1}^n t_i a_i a_i^T$$

kde $K(t)$ je $m \times m$ symetrická matica zapísaná ako vážený súčet matíc s hodnotou 1 a nazýva sa matica tuhosti priečok konštrukcie. Matica závisí lineárne od objemov priečok t_i .

Pri zaťažení sa konštrukcia mierne deformuje, voľné uzly menia svoju polohu, až pokým budú externé sily kompenzované reakčnými silami ($f = -f_r$) - systém bude v ekvilibriu. Tento poznatok nám dá systém lineárnych rovníc určujúcich posunutie konštrukcie pod externou záťažou

$$K(t)v = f \quad (17)$$

Ak tento systém nemá riešenie, znamená to, že konštrukcia t nedokáže uniesť záťaž f a zničí sa.

Na dokončenie modelu potrebujeme ešte odvodiť vzťah pre potenciálnu energiu uloženú konštrukciou v ekvilibriu (poddajnosť angl. compliance). Využijeme vzťah pre potenciálnu energiu uloženú jednou priečkou (14) a zosumujeme energie uložené priečkami, aby sme dostali potenciálnu energiu pre celú konštrukciu. Z toho dôvodu je poddanosť konštrukcie t pod záťažou f rovná

$$Compl_f(t) = \frac{1}{2} \sum_{i=1}^n t_i (v^T a_i)^2 = \frac{1}{2} v^T K(t)v = \frac{1}{2} v^T f \quad (18)$$

kde v je prislúchajúce posunutie, ktoré je riešením $K(t)v = f$. Výraz pre poddanosť je možné interpretovať: *Poddajnosť je polovica mechanickej práce vykonanej externou záťažou na posunutie konštrukcie až do ekvilibría.* Poznamenajme fakt, že ak je systém $K(t)v = f$ riešiteľný, tak hodnota poddanosti konštrukcie nezávisí od jednotlivého výberu riešenia systému.

4.2.2 Najjednoduchší TTD problém

Teraz naformulujeme problém navrhovania najtuhšej konštrukcie vopred zadanej váhy. Je daná mriežka uzlov $[m, n, \{a_i \in R^m\}_{i=1}^n]$, pôsobiaca externá sila f a celkový objem

priečok $w > 0$. Treba nájsť konštrukciu $t \geq 0_n$ s minimálnou možnou poddajnosťou $Comp_f(t)$, ktorá spĺňa obmedzenie zdrojov

$$\sum_{i=1}^n t_i \leq w \quad (19)$$

Taktiež je rozumné zaviesť predpoklad:

Vektory $\{a_i \in R^m\}_{i=1}^n$ generujú celý priestor R^m .

Tento predpoklad hovorí o tom, že neexistuje nenulové posunutie v kolmé na všetky vektory a_i , t.j. neexistuje $v \neq 0$ také, že $v^T a_i = 0 \forall i$. Inými slovami, predpoklad zakazuje posúvanie celých konštrukcií.

Výsledný zápis problému TTD vyzerá nasledovne:

$$\begin{aligned} \underset{v,t}{Min} \quad & 1/2 f^T v \\ & \sum_{i=1}^n t_i a_i a_i^T v = f \\ & \sum_{i=1}^n t_i \leq w \\ & t \geq 0 \end{aligned} \quad (20)$$

Na prvý pohľad vyzerá tento problém TTD ako úloha nelineárneho programovania, prekvapujúco sa však dá prepísať na formu úlohy lineárneho programovania. Transformácia sa uskutoční pomocou novej premennej s_i , ktorá predstavuje stres(angl. stress). Pre zaťaženie konštrukciu je stres priečky rovný absolútnej hodnote napnutia(tension) delenej prierezom priečky. Čím väčšia je hodnota stresu v priečke, tým v horších podmienkach materiál priečky pracuje. Z Hookovho zákona dostaneme vzťah pre stres v i -tej priečke ako funkciu posunutia v

$$s_i = \sqrt{E} |a_i^T v| \quad (21)$$

Je daná mriežka uzlov $[m, n, \{a_i \in R^m\}_{i=1}^n]$ a na uzly pôsobí externá sila f . Úlohou je nájsť posunutie v , ktoré maximalizuje prácu $f^T v$ záťaže za platnosti ohraničenia, že všetky stesy priečok sú menšie ako 1.

$$\begin{aligned} \underset{v}{Max} \quad & f^T v \\ & |a_i^T v| \leq 1 \quad i = 1, \dots, n \end{aligned} \quad (22)$$

Duálny problém k tejto úlohe je ekvivalentný úlohe lineárneho programovania

$$\begin{aligned} \underset{x_1, \dots, x_n}{Min} \quad & \sum_{i=1}^n |x_i| \\ & \sum_{i=1}^n x_i a_i = f \end{aligned} \tag{23}$$

alebo ekvivalentne maticovo

$$\begin{aligned} \underset{x}{Min} \quad & \|x\|_1 \\ & Ax = f, \end{aligned} \tag{24}$$

kde $A = [a_1, \dots, a_n]$. Poznamenajme, že obe úlohy LP (primárna aj duálna) sú prípustné. Prípustnosť duálnej úlohy pochádza priamo z predpokladu, že vektory $\{a_i \in R^m\}_{i=1}^n$ generujú celý priestor R^m . Stĺpce matice A možno chápať ako vychýlenie uzlov prislúchajúce i -tej priečke (reakcia konštrukcie na prítomnosť hrany). Absolútna hodnota premennej x_i duálnej úlohy predstavuje váhu danej priečky.

Namiesto riešenia duálnej úlohy lineárneho programovania, môžeme penalizovať účelovú funkciu penalizačným členom $p(x) = \frac{1}{\lambda} \|Ax - f\|_2^2$, $\frac{1}{\lambda} > 0$ prislúchajúcim ohraničeniu duálnej úlohy a následne riešiť nasledujúcu neohraničenú úlohu:

$$\underset{x}{Min} \quad \|Ax - f\|_2^2 + \lambda \|x\|_1 \tag{25}$$

Všimnime si, že daná úloha je v tvare úlohy najmenších štvorcov regulovaných l_1 normou. Matica A predstavuje reakciu konštrukcie na prítomnosť hrán. Vhodným rozložením prípustných hrán (dizajnovacie premenné x) chceme kompenzovať záťaž pôsobiacu na konštrukciu (f), pričom nemá zmysel dizajnovvať konštrukciu zo všetkých možných priečok. Preto chceme, aby čo najviac priečok dostalo nulovú váhu. Formulácia TTD problému ako úlohy najmenších štvorcov regulovaných l_1 normou je vhodná predovšetkým na aplikáciu súradnicovo spádových metód [9] a takisto je vhodná na aplikáciu Proximal gradient metódy. Úlohu sa pokúsime riešiť taktiež primárno-duálnou metódou vnútorného bodu.

V nasledujúcich podkapitolách si predstavíme rôzne prístupy k riešeniu úlohy TTD. Na formuláciu TTD problému v tvare úlohy najmenších štvorcov regulovaných l_1 normou aplikujeme Proximal gradient metódu. Taktiež sa pre ekvivalentnú úlohu k tejto formulácii pokúsime odvodiť primárno duálnu metódu vnútorného bodu. Nakoniec

spomenieme, ako je možné úlohu formulovanú v tvare lineárneho programovania použiť na výpočet v prostredí Matlab. Rôzne prístupy porovnáme, spomenieme ich výhody aj nevýhody.

4.2.3 Riešenie pomocou Proximal gradient metódy

Keďže sa v prípade úlohy (25) (úloha najmenších štvorcov regulovaných l_1 normou) jedná o neohraničený problém, kde minimalizujeme účelovú funkciu rozložiteľnú na diferencovateľnú konvexnú ($\|Ax - f\|_2^2$) a nediferencovateľnú, uzavretú konvexú ($\lambda\|x\|_1$) časť, môžeme na jej optimalizáciu použiť Proximal gradient metódu. Za týmto účelom sa odvodíme základný prvok metódy - proximal operátor pre nediferencovateľnú časť $h(u) = \lambda\|u\|_1$. Proximal operátor sme definovali v kapitole 3.1.1, uviedli sme jeho dôležité vlastnosti. Ako prvé si treba uvedomiť, že funkcia $h(u)$ je separovateľná, vďaka tejto vlastnosti stačí vyjadriť proximal operátor pre jednu zložku. Na výpočet proximálneho operátora využijeme definíciu proximálneho operátora pomocou subgradientu (Veta 3.1.).

$$u_i = \text{prox}_{th_i}(x_i) \Leftrightarrow x_i - u_i \in t\partial h_i(u_i)$$

$$x_i - u_i \in t\lambda\partial|u_i|$$

$$x_i - u_i \in t\lambda \begin{cases} 1 & u_i > 0 \\ -1 & u_i < 0 \\ [-1, 1] & u_i = 0 \end{cases}$$

V prípade ak $u_i > 0$ ľahko dostaneme vzťah pre proximálny operátor $u_i = x_i - t\lambda$, aby platila podmienka musí x_i spĺňať vzťah $x_i > t\lambda$. Analogicky sa proximálny operátor odvodí pre ďalšie dva prípady. Spojením prípadov získame proximálny operátor funkcie $th(u) = t\lambda\|u\|_1$, ktorý je tvaru

$$\text{prox}_{th}(x)_i = \begin{cases} x_i - t\lambda & x_i > t\lambda \\ x_i + t\lambda & x_i < -t\lambda \\ 0 & |x_i| \leq t\lambda \end{cases}$$

Ďalšia vec, ktorá nás môže pri aplikácii proximal gradient metódy na úlohu zaujímať je voľba kroku t . Z teórie vieme, že konvergencia metódy je zaručená u konštantného kroku $t = \frac{1}{L}$ (Veta 3.4.,3.5.), kde L je Lipschitzovská konštanta gradientu funkcie

$g(x) = \|Ax - f\|_2^2$. Preto nájdeme konštantu L z Lipschitzovskej vlastnosti gradientu $\nabla g(x)$.

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in R^n$$

Po dosadení

$$\|2A^T A(x - y)\|_2 \leq 2\alpha_{max}(A^T A)\|x - y\|_2,$$

kde $\alpha_{max}(A^T A)$ predstavuje najväčšie vlastné číslo matice $A^T A$. Hľadaná Lipschitzovská konštanta L sa rovná dvojnásobku $\alpha_{max}(A^T A)$. Z toho dôvodu v algoritmoch metódy použijeme fixný krok $t = \frac{1}{2\alpha_{max}(A^T A)}$.

4.2.4 Riešenie pomocou primárno-duálnej metódy vnútorného bodu

Na riešenie úlohy l_1 filtering v predchádzajúcej kapitole bola použitá primárno-duálna metóda vnútorného bodu aplikovaná na formuláciu danej úlohy v tvare kvadratického programovania (na duálnu úlohu). Presnejšie sa jednalo o úlohu s kvadratickou účelovou funkciou a s lineárnymi ohraničeniami. Aj v prípade TTD problému sa pokúsime formuláciu v tvare úlohy najmenších štvorcov regulovaných l_1 normou previesť na takýto typ úlohy kvadratického programovania a následne budeme novú úlohu riešiť primárno-duálnou metódou vnútorného bodu. Vďaka použitiu primárno-duálnej metódy vnútorného bodu budú primárne aj duálne premenné aktualizované v každej iterácii.

Úprava úlohy

Naším cieľom je úlohu transformovať tak, aby sme sa zbavili nediferencovateľnej časti účelovej funkcie, ktorú predstavuje l_1 norma. Zavedením novej premennej u a nových ohraničení $-u_i \leq x_i \leq u_i \quad i = 1, \dots, n$ získame očividne ekvivalentnú úlohu s úlohou najmenších štvorcov regulovaných l_1 normou, navyše už má požadovanú diferencovateľnú účelovú funkciu

$$\begin{aligned} \underset{x, u}{Min} \quad & \|Ax - f\|_2^2 + \lambda \sum_{i=1}^n u_i \\ & - u_i \leq x_i \leq u_i \quad i = 1, \dots, n \end{aligned} \tag{26}$$

Získaná úloha má kvadratickú účelovú funkciu a lineárne ohraničenia. Maticovo vyzerá zápis nasledovne

$$\begin{aligned} \underset{x,u}{Min} \quad & \|Ax - f\|_2^2 + \lambda \mathbf{1}^T u \\ & x - u \leq 0 \\ & -x - u \leq 0 \end{aligned} \tag{QP}$$

Ekvivalentnú formuláciu TTD problému (QP) budeme riešiť primárno-duálnou metódou vnútorného bodu. Poznamenajme však, že predošlou úpravou úlohy narástol počet premenných na dvojnásobok pôvodnej hodnoty, čo nemusí byť prospešné pre chod algoritmu.

Primárno-duálny hľadací smer

Základným prvkom primárno-duálnej metódy vnútorného bodu je tzv. hľadací smer, ktorý je určovaný Newtonovým krokom pre systém nelineárnych rovníc $r_t(x, u, \mu_1, \mu_2) = 0$. Tento systém predstavujú modifikované Karush-Kuhn-Tuckerové (KKT) podmienky. Preto ako prvé odvodíme KKT podmienky pre úlohu (QP), ktoré následne modifikujeme na požadovaný tvar.

Lagrangeova funkcia pre úlohu (QP) je daná ako

$$L(x, u, \mu_1, \mu_2) = \|Ax - f\|_2^2 + \lambda \mathbf{1}^T u + \mu_1^T (x - u) + \mu_2^T (-x - u)$$

kde $\mu_1, \mu_2 \geq 0_n$ sú duálne premenné prislúchajúce k ohraničeniam $x \leq u$ a $-u \leq x$ úlohy (QP). KKT podmienky pre úlohu (QP) budú v tvare

$$\begin{array}{cccc} x & u & \mu_1 & \mu_2 \\ & & \mu_1 \geq 0 & \mu_2 \geq 0 \\ \frac{\partial L}{\partial x} = 0 & \frac{\partial L}{\partial u} = 0 & \frac{\partial L}{\partial \mu_{1i}} \mu_{1i} = 0 & \frac{\partial L}{\partial \mu_{2i}} \mu_{2i} = 0 \quad i = 1, \dots, n-2 \\ & & \frac{\partial L}{\partial \mu_1} \leq 0 & \frac{\partial L}{\partial \mu_2} \leq 0 \end{array}$$

$$\mu_1 \geq 0, \mu_2 \geq 0, x - u \leq 0, -x - u \leq 0$$

$$2A^T(Ax - f) + \mu_1 - \mu_2 = 0$$

$$\lambda \mathbf{1} - \mu_1 - \mu_2 = 0 \tag{KKT}$$

$$diag(\mu_1)(x - u) = 0$$

$$diag(\mu_2)(-x - u) = 0$$

Pridaním parametra $t > 0$ do podmienok komplementarity (posledné dve rovnice KKT podmienok) získame modifikovaný systém rovníc (mKKT).

$$r_t(x, u, \mu_1, \mu_2) = \begin{pmatrix} 2A^T(Ax - f) + \mu_1 - \mu_2 \\ \lambda \mathbf{1} - \mu_1 - \mu_2 \\ -diag(\mu_1)(x - u) - \frac{1}{t} \mathbf{1} \\ -diag(\mu_2)(-x - u) - \frac{1}{t} \mathbf{1} \end{pmatrix} \quad (\text{mKKT})$$

Nech sú premenné x, u primárne prípustné a premenné μ_1, μ_2 sú duálne prípustné. Ak premenné x, u, μ_1, μ_2 spĺňajú systém rovníc $r_t(x, u, \mu_1, \mu_2) = 0$, potom položíme $x = x^*(t)$, $u = u^*(t)$, $\mu_1 = \mu_1^*(t)$ a $\mu_2 = \mu_2^*(t)$. Ak $t \rightarrow \infty$, tak sa systém rovníc $r_t(x, u, \mu_1, \mu_2) = 0$ redukuje na pôvodné KKT podmienky. Hlavnou myšlienkou metódy je použiť Newtonovské kroky na vyriešenie série nelineárnych rovníc $r_t(x, u, \mu_1, \mu_2) = 0$ pre rastúcu postupnosť hodnôt parametra t .

Teraz sa bližšie pozrieme na Newtonovský krok pre riešenie nelineárnych rovníc $r_t(x, u, \mu_1, \mu_2) = 0$ pre fixnú hodnotu parametra t v bode (x, u, μ_1, μ_2) , ktorý spĺňa $\mu_1 > 0$, $\mu_2 > 0$, $x - u < 0$, $-x - u < 0$. Označme súčasný bod $w = (x, u, \mu_1, \mu_2)$ a Newtonovský krok $\Delta w = (\Delta x, \Delta u, \Delta \mu_1, \Delta \mu_2)$. Potom je Newtonovský krok určený lineárnymi rovnicami

$$r_t(w + \Delta w) \approx r_t(w) + Dr_t(w)\Delta w = 0$$

teda je potrebné riešiť sústavu lineárnych rovníc $Dr_t(w)\Delta w = -r_t(w)$, kde Dr_t predstavuje Jakobián r_t . V premenných x, u, μ_1, μ_2 potom dostávame systém lineárnych rovníc

$$\begin{pmatrix} 2A^T A & 0 & I & -I & | & -2A^T(Ax - f) - \mu_1 + \mu_2 \\ 0 & 0 & -I & -I & | & -\lambda \mathbf{1} + \mu_1 + \mu_2 \\ -diag(\mu_1) & diag(\mu_1) & -diag(x - u) & 0 & | & diag(\mu_1)(x - u) + \frac{1}{t} \mathbf{1} \\ diag(\mu_2) & diag(\mu_2) & 0 & -diag(-x - u) & | & diag(\mu_2)(-x - u) + \frac{1}{t} \mathbf{1} \end{pmatrix}$$

Dostali sme teda systém, ktorý určuje primárno-duálny smer. Keďže by bolo neefektívne riešiť daný systém rovníc, pokúsime sa ho zjednodušiť tak, aby sa čo najviac pracovalo s jednoduchými maticami - najlepšie diagonálnymi, s ktorými sa systém bude riešiť s menšou náročnosťou. Elementárnymi operáciami zredukujeme systém na trojuholníkový tvar. Pre zjednodušenie označme $p = 2A^T(Ax - f)$. Prenásobením tretej rovnice inverznou maticou $-diag(\mu_1)^{-1}$, štvrtej rovnice maticou $-diag(\mu_2)^{-1}$ a

substitúciou $J_1 = \text{diag}(\mu_1)^{-1} \text{diag}(x - u)$, $J_2 = \text{diag}(\mu_2)^{-1} \text{diag}(-x - u)$ získame

$$\left(\begin{array}{cccc|c} 2A^T A & 0 & I & -I & -p - \mu_1 + \mu_2 \\ 0 & 0 & -I & -I & -\lambda \mathbf{1} + \mu_1 + \mu_2 \\ I & -I & J_1 & 0 & -(x - u) - \frac{1}{t} \text{diag}(\mu_1)^{-1} \mathbf{1} \\ -I & -I & 0 & J_2 & x + u - \frac{1}{t} \text{diag}(\mu_2)^{-1} \mathbf{1} \end{array} \right)$$

Tretiu (štvrtú) rovnicu prenásobíme inverznou maticou J_1^{-1} (J_2^{-1}), tieto inverzie sú nenáročné, pretože sa jedná o diagonálne matice. Taktiež označme pre zjednodušenie $s_1 = \frac{1}{t} \text{diag}(x - u)^{-1} \mathbf{1}$ a $s_2 = \frac{1}{t} \text{diag}(-x - u)^{-1} \mathbf{1}$

$$\left(\begin{array}{cccc|c} 2A^T A & 0 & I & -I & -p - \mu_1 + \mu_2 \\ 0 & 0 & -I & -I & -\lambda \mathbf{1} + \mu_1 + \mu_2 \\ J_1^{-1} & -J_1^{-1} & I & 0 & -J_1^{-1}(x - u) - s_1 \\ -J_2^{-1} & -J_2^{-1} & 0 & I & -J_2^{-1}(-x - u) - s_2 \end{array} \right)$$

V ďalšom kroku využijeme fakt, že $J_1^{-1}(x - u) = \text{diag}(x - u)^{-1} \text{diag}(\mu_1)(x - u) = \mu_1$ (podobne pre μ_2). Následne pripočítame tretiu aj štvrtú rovnicu k druhej a podobne k prvej rovnici pripočítame štvrtú a odpočítame tretiu. Označme taktiež vzniknuté matice $J_p = J_1^{-1} + J_2^{-1}$ a $J_m = J_1^{-1} - J_2^{-1}$. Tým dostaneme systém

$$\left(\begin{array}{cccc|c} 2A^T A - J_p & J_m & 0 & 0 & -p + s_1 - s_2 \\ J_m & -J_p & 0 & 0 & -\lambda \mathbf{1} - s_1 - s_2 \\ J_1^{-1} & -J_1^{-1} & I & 0 & -\mu_1 - s_1 \\ -J_2^{-1} & -J_2^{-1} & 0 & I & -\mu_2 - s_2 \end{array} \right)$$

Pripočítaním $J_m J_p^{-1}$ násobku druhej rovnice k prvej a následne prenásobením druhej rovnice maticou $-J_p^{-1}$ dostaneme výsledný systém v trojuholníkovom tvare.

$$\left(\begin{array}{cccc|c} 2A^T A - J_p + J_m J_p^{-1} J_m & 0 & 0 & 0 & -p + s_1 - s_2 - J_m J_p^{-1}(\lambda \mathbf{1} + s_1 + s_2) \\ -J_p^{-1} J_m & I & 0 & 0 & J_p^{-1}(\lambda \mathbf{1} + s_1 + s_2) \\ J_1^{-1} & -J_1^{-1} & I & 0 & -\mu_1 - s_1 \\ -J_2^{-1} & -J_2^{-1} & 0 & I & -\mu_2 - s_2 \end{array} \right)$$

Na nájdenie Newtonovského kroku nám stačí vyriešiť prvú rovnicu redukovaného systému (27), ktorého riešenie Δx využijeme na dopočítanie ďalších troch zložiek Newtonovho kroku.

$$(2A^T A - J_p + J_m J_p^{-1} J_m) \Delta x = -p + s_1 - s_2 - J_m J_p^{-1}(\lambda \mathbf{1} + s_1 + s_2) \quad (27)$$

Ostatné rovnice už pracujú len s diagonálnymi maticami, ich inverzie sú jednoducho spočítateľné, čo je výhodou tohto postupu. Ďalšie tri komponenty kroku $\Delta u, \Delta\mu_1, \Delta\mu_2$ získame zo vzťahov

$$\begin{aligned}\Delta u &= J_p^{-1}(\lambda \mathbf{1} + s_1 + s_2 + J_m \Delta x) \\ \Delta\mu_1 &= -\mu_1 - s_1 + J_1^{-1}(-\Delta x + \Delta u) \\ \Delta\mu_2 &= -\mu_2 - s_2 + J_2^{-1}(\Delta x + \Delta u)\end{aligned}\tag{28}$$

kde J_1, J_2, J_p a J_m sú diagonálne.

Primal-dual search direction $\Delta w = (\Delta x, \Delta u, \Delta\mu_1, \Delta\mu_2)$ je teda definovaný ako riešenie (27) a (28).

Všimnime si, že celý systém pracuje s riedkymi maticami, čo urýchluje výpočet daného systému. Najnáročnejšiu časť riešenia predstavuje výpočet prvej zložky (27) Newtonovského kroku. V poslednej časti práce sa nachádza nami naprogramovaný funkčný program primárno-duálnej metódy vnútorného bodu, ktorý využíva vyššie uvedený systém na získanie primárno-duálneho hľadacieho kroku.

4.2.5 Experimenty a porovnaní metód/prístupov

V tejto podkapitole otestujeme nami spomínané metódy/postupy riešenia TTD úlohy. V prípade proximal gradient metódy porovnáme 4 rôzne verzie: základnú verziu, zrýchlenú verziu, spádovú verziu a novú verziu proximal gradient metódy. Experimenty sú uskutočnené v prostredí Matlab, niektoré zdrojové kódy sa nachádzajú v poslednej časti práce. Všetky pokusy sú vykonané na PC zostave so štvorjadrovým procesorom Intel Core-i7 2,66GHz s operačnou pamäťou 6GB RAM.

Za účelom zmenšenia rozmerov úlohy budeme uvažovať len také potenciálne priečky, že žiadne dve z nich sa navzájom nepretínajú viac ako v jednom bode. Tento dodatočný predpoklad nie je vôbec obmedzujúci, pretože ak sa dve priečky pretínajú viac ako v jednom bode, musia sa pretínať práve v nejakých uzloch, a preto je možné väčšiu z priečok zložiť z menších pospájaným častí. Prípustné priečky tohto typu nám vytvára nami naprogramovaný generátor mriežky.

Napriek predošlému zjednodušeniu však rozmernosť úlohy ($A_{m \times n}$) s rastúcimi rozmermi mriežky uzlov prudko rastie, kvôli pribúdajúcemu počtu prípustných hrán. Rozmery riešených úloh sú častokrát tak veľké, že metódy zlyhávajú kvôli nedostatku pamäti aj výpočet gradientu je mnohokrát časovo náročný. Pre lepšie fungovanie všetkých metód

mriežka rxr	m	n	% nenulových zložiek
5x5	50	196	7.26
20x20	800	48934	0.5
25x25	1250	119016	0.3
30x30	1800	246690	0.2
50x50	5000	1901930	0.08
100x100	20000	30398894	0.002

Tabuľka 2: Závislosť rozmerov úlohy a počtu nenulových prvkov matice A od rozmeru mriežky uzlov

je vhodné využiť vrodennú vlastnosť matice A - jej obrovskú riedkosť. Stĺpce a_i matice A sú veľmi riedke, každý obsahuje najviac 4 nenulové zložky. V prostredí Matlab je preto vhodné využiť funkciu `sparse`. V našom prípade nám táto funkcia umožnila riešiť vysoko-rozmerne úlohy, ktoré pred jej použitím nebolo možné riešiť kvôli nedostatku operačnej pamäti počítača(6GB!) a taktiež táto funkcia urýchlila samotný výpočet úloh. V prípade použitia Proximal gradient metódy bolo pomocou funkcie `sparse` dosiahnuté až štvornásobné zrýchlenie.

Atraktívnou vlastnosťou úlohy Truss topology design je, že nájde optimálny geometrický tvar (rozloženie) konštrukcie, aj keď navonok pracuje len s váhami priečok konštrukcie. Optimalizačný problém rozhodne, ktoré uzly použiť, ako ich pospájať a nájde aj optimálne váhy priečok. Aj keď začneme optimalizáciu problému so všetkými uzlami pospájanými priečkami, optimalizačný proces nájde optimálnu konštrukciu, v ktorej je väčšina priečok s nulovými váhami. Tento jav budeme pozorovať aj na nasledujúcom experimente, ktorý porovnáva všetky nami naprogramované metódy. Metódy boli na úlohu TTD aplikované v súlade s výsledkami z kapitol 4.2.3 resp 4.2.4.

Experiment č.1 - Porovnanie metód

Budeme porovnávať primárno-dáľnu metódu vnútorného bodu a 4 verzie Proximal gradient metódy (základnú verziu, zrýchlenú verziu, spádovú verziu a novú verziu). Modifikácie proximal gradient metódy však majú rôznu náročnosť za jednu iteráciu a taktiež nemajú rovnakú rýchlosť konvergencie. Napríklad základná proximal gradient metóda má najmenej časovo náročnú iteráciu, avšak má najmenšiu rýchlosť konvergencie z uvedených metód. Aby bolo porovnanie týchto odlišných metód spravodlivé, každej z nich ponúkne na vyriešenie úlohy rovnaký čas (60s, 180s, 1500s). Takisto budeme pozorovať postupné nachádzanie riešenia v čase.

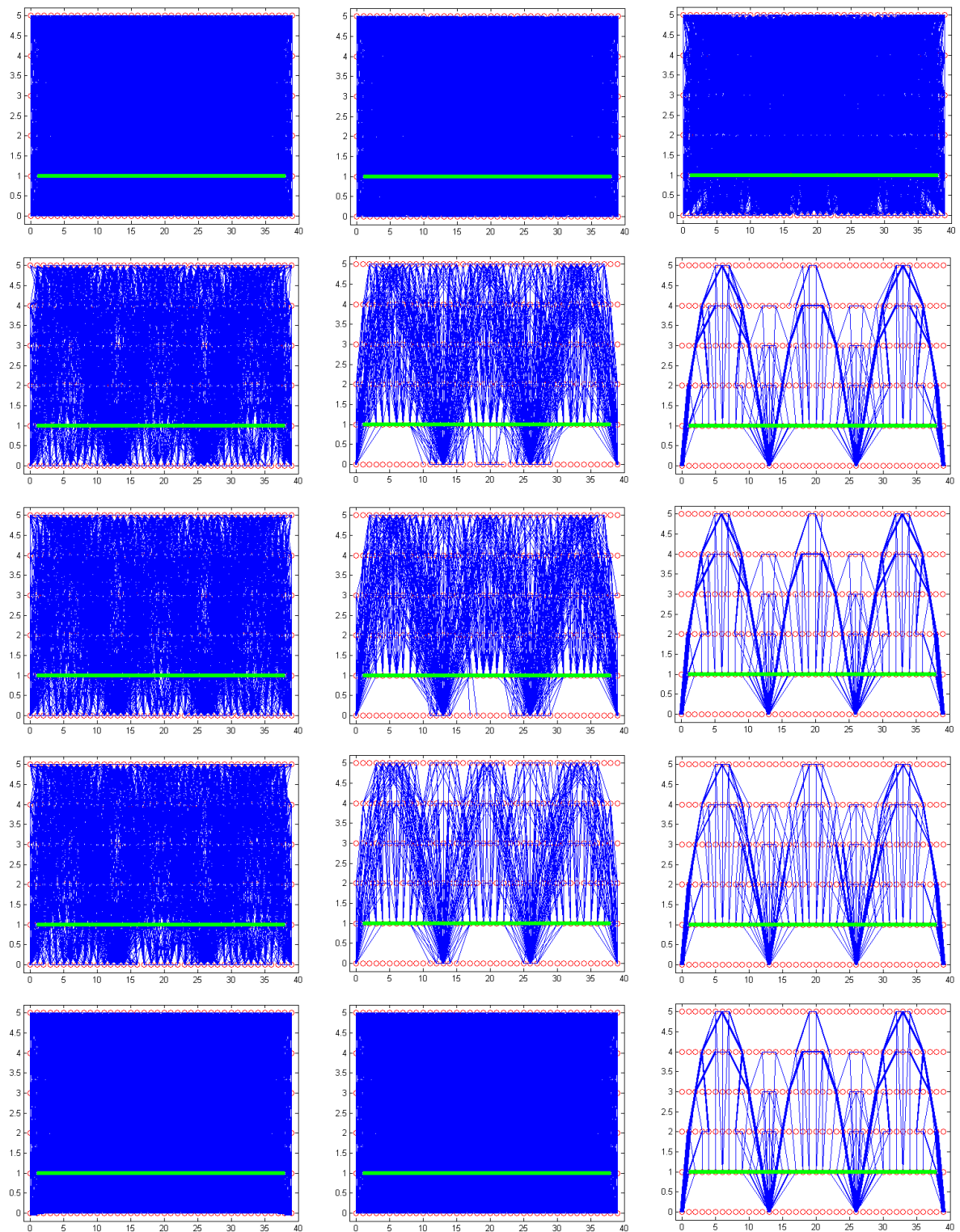
Metódy porovnáme na nami vygenerovanej úlohe TTD o veľkosti mriežky 6x40. Úloha je mostového typu s viac ako 16500 prípustnými priečkami a obsahuje 4 fixné body na spodu, ktoré reprezentujú pevné piliere mostovej konštrukcie. Na uzly vo výške 1 je aplikovaná jednotková sila smerujúca kolmo na podložku. Počiatočné riešenie sme položili nulovému vektoru a modul pružnosti v ťahu (Youngov modul) sme zvolili rovný hodnote 200, čo zodpovedá oceľovej priečke. Vďaka tomu, že máme usporiadanie prípustných priečok, vieme vykreslovať aj geometrický tvar konštrukcie. Optimalizačný proces nájde optimálnu konštrukciu, v ktorej je väčšina priečok s nulovými váhami, preto sú posledné obrázky na Obr. 8 riedke, je vidieť zreteľnú mostovú konštrukciu.

Na obrázku porovnania metód je možné pozorovať viacero faktov. Výsledný obrázok (25min) základnej verzie proximal gradient metódy (1. riadok) ani zďaleka nepripomína mostovú konštrukciu aj napriek tomu, že jeho účelová funkcia bola malá s duálnou medzerou $8.42E-04$. Rýchlosť iterácie základnej metódy teda nebola o toľko väčšia, aby prebila lepšiu konvergenciu ostatných metód. Mostovú štruktúru nepripomínajú ani suboptimálne riešenia všetkých ostatných metód pri čase 60s. Dokonca metóda vnútorného bodu, nemá žiadnu výraznú štruktúru ani po 180s, pretože stihla len 8 iterácii. Z 2. stĺpca je pozorovateľné, že zo začiatku si viedli lepšie proximal gradient metódy (až na základnú), najlepšie pôsobí 4 metóda, no duálna medzera toto pozorovanie nepotvrdila. Podľa duálnej medzery najlepšie obstála zrýchlená PG metóda. Z obrázka 4. metódy po 1500s možno vidieť, že sa so svojím riešením nedostala tak blízko k optimálnemu ako 2.,3. a 5. metóda, obrázok v porovnaní s nimi obsahuje viac nenulových priečok. Zo zvyšných 3 metód (2,3,5) je ťažké určiť víťaza - všetky finálne obrázky vyzerajú podobne, obrázky výrazne pripomínajú mostovú konštrukciu - asi najlepšie

obstála metóda vnútorného bodu a zrýchlená proximal gradient metóda. Hodnoty účelových funkcií týchto metód sú takmer identické, najlepšie skončila metóda vnútorného bodu. Metódy štartovali z nulového vektora, pri ktorom bola hodnota účelovej funkcie $f(x_0) = 38$. Poznamenajme, že hodnota účelovej funkcie klesla v priebehu experimentu v prípade proximal gradient metód veľmi rýchlo na malú hodnotu (po 1min - základná PG - 4.49E-03, ostatné PG - 2.82E-03), ale prislúchajúce obrázky ešte nepripomínajú mostovú štruktúru. Finálne obrázky majú mostovú štruktúru, je zreteľné, že väčšine prípustných priečok bola priradená nulová váha, taktiež je možné pozorovať polohu pilierov mosta - vychádzajú z nich priečky s najväčšími váhami.

Čas it.	60s		180s		1500s	
PGM	6908	1.85E-03	20704	1.64E-03	172173	8.42E-04
APGM	6829	1.75E-04	20521	3.98E-05	171198	2.01E-06
DPGM	6415	1.89E-04	19332	4.42E-05	161602	1.99E-06
NPGM	6484	1.87E-04	19484	5.91E-05	162581	4.49E-06
PDMVB	4	5.66E+00	8	4.08E-02	52	1.97E-06

Tabuľka 3: Porovnanie metód - základná verzia PG (PGM), zrýchlená PG (APGM), spádová PG (DPGM), nová PG (NPGM) a primárno-duálna MVB (PDMVB). Na obrázku je možné pozorovať dosiahnutý počet iterácii metód ako aj dosiahnutú suboptimalitu riešení v závislosti od času. Suboptimalita predstavuje duálnu medzeru medzi hodnotou primárnej účelovej funkcie a hodnotou fixne zvolenej duálnej účelovej funkcie. Za hodnotu duálnej účelovej funkcie bola zobrazená hodnota z MVB (ktorá je takmer optimálna). Proximal gradient metódy majú menšiu časovú náročnosť na 1 iteráciu, najviac iterácii stihla najjednoduchšia základná PG, ale rozdiely v prípade PG metód nie sú veľké. V prípade MVB sú iterácie časovo náročné za 25min stihla metóda iba 52 iterácii. Taktiež možno pozorovať väčšiu rýchlosť PG metód do času 180s, priebežne najlepšiu hodnotu dosahuje APGM. Finálnu hodnotu účelovej funkcie má najlepšiu PDMVB, aj keď výsledky sú tesné (APGM, DPGM). Preto je možné, že sme metóde vnútorného bodu pomohli väčším časom riešenia, pretože do času 180s PG metódy silne zdolávajú MVB. Je možné, že z dôvodu ich algoritmu nevedeli lepšie "dotlačiť" riešenie ako MVB. Ako celkového víťaza pozorovania by preto sme mohli označiť zrýchlenú proximal gradient metódu.

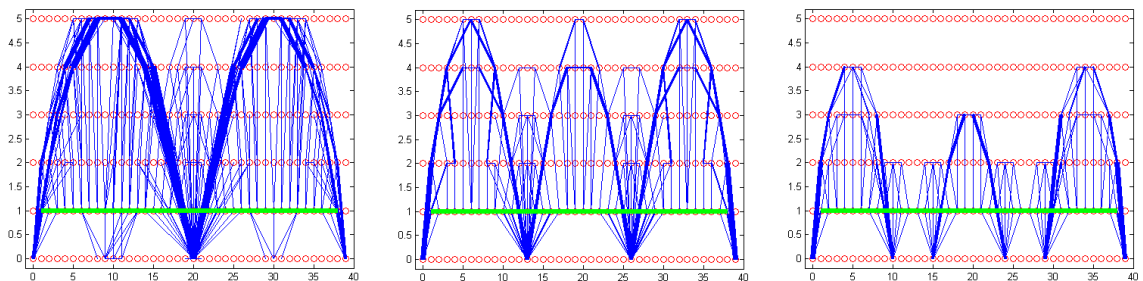


Obr. 8: Porovnanie metód: Riadky predstavujú rôzne verzie metód - základná, zrýchlená, spádová a nová Proximal gradient metóda, posledný riadok predstavuje primárno-duálna metóda vnútorného bodu. Stĺpce sú zoradené podľa času výpočtu riešenia (60s, 180s, 1500s).

Experiment č.2 - Zmena počtu fixných uzlov

V ďalšom experimente sa pozrieme na vplyv počtu fixných uzlov (pilierov) na geometrický tvar/rozloženie nosnej konštrukcie. Podobne ako v predošlom experimente použijeme veľkosť mriežky 6x40. Úloha je mostového typu s viac ako 16500 prípustnými priečkami. Na uzly vo výške 1 je aplikovaná jednotková sila smerujúca kolmo na podložku. Počiatočné riešenie sme položili nulovému vektoru ($f(x_0)=38$) a modul pružnosti v ťahu (Youngov modul) sme zvolili rovný hodnote 200, čo zodpovedá oceľovej priečke.

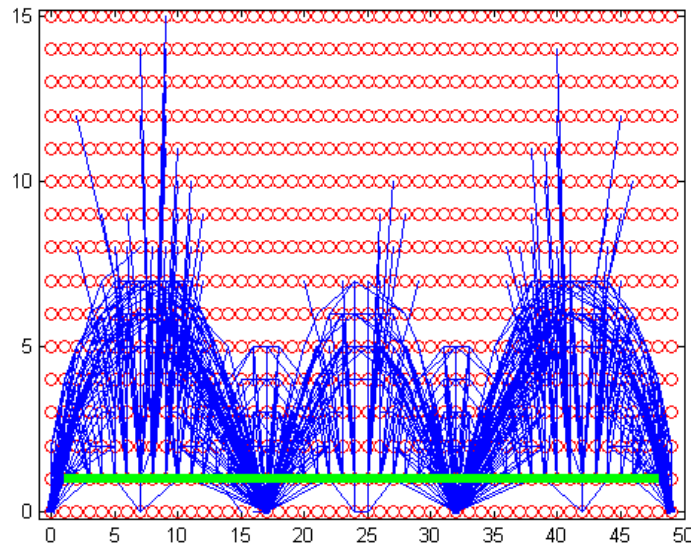
Výsledné obrázky majú zreteľnú mostovú štruktúru, väčšine prípustných priečok bola v optime priradená nulová váha. Taktisto je možné pozorovať polohu pilierov mosta - vychádzajú z nich priečky s najväčšími váhami. Všimnime si, že na obrázku s 3 piliermi je viac priečok ako na ďalších dvoch, taktiež obsahuje priečky s väčšou váhou v pomere k ostatným priečkam. Na udržanie rovnakej záťaže potrebuje silné nosné priečky. Z obrázkov je možné tiež pozorovať jav, že s pribúdajúcim počtom pilierov sa výška mostových štruktúr znižuje. V prípade 6 pilierov siaha mostová konštrukcia len do výšky 4, naopak v prípade len 3 pilierov je konštrukcia vo výške 5. Riešenia sú získané primárno-duálnou metódou vnútorného bodu, hodnoty duálnej medzery sú v poradí obrázkov nasledujúce: 1.13E-05, 1.97E-06, 1.07E-07.



Obr. 9: Vplyv počtu pilierov na tvar mostovej konštrukcie

Experiment č.3 - Veľkorozmerná úloha

V nasledujúcom experimente sa jedná o úlohu veľkých rozmerov - mriežka má rozmery



Obr. 10: Riešenie veľkorozmernej úlohy pomocou zrýchlenej proximal gradient metódy (APGM).

16x50, matica A má rozmery 1600x186116, čiže úloha obsahuje 186116 prípustných priečok. Závaž pôsobí na uzly vo výške 1. Ako počiatočný bod riešenia sme zvolili nulový vektor, $f(x_0) = 48$. Výsledný obrázok získaný zrýchlenou proximal gradient metódou má mostovú štruktúru, prislúcha účelovej funkcii o hodnote $4.22E-03$. Takisto možno zreteľne pozorovať piliere mostu. Samotný experiment v Matlabe trval 4 hodiny. Metóda dosiahla viac ako 35000 iterácií a ponúkla nám najlepšie dosiahnuté riešenie. Na tento istý problém sme sa pokúsili aplikovať aj primárno-duálnu metódu vnútorného bodu, no tá nedokázala z dôvodu zaplnenia operačnej pamäte počítača vypočítať ani prvú iteráciu. Systém pri počítaní Newtonovho kroku (27) je pri takýchto rozmeroch (186tis) časovo náročné riešiť aj napriek vrodenej riedkosti celého systému. Bezkonkurenčným víťazom pri vysokých rozmeroch je teda zrýchlená proximal gradient metóda.

Poznamenajme, že ak by sme znížili výšku mriežky na 9x50 boli by sme pravdepodobne schopní vypočítať presnejšie riešenie daného problému, prípadne by sme dosiahli riešenie za kratší čas.

5 Záver

V diplomovej práci sme sa venovali úlohe s názvom *Najmenšie štvorce regulované l_1 normou*. Urobili sme analýzu a porovnanie rôznych prístupov, ktoré možno použiť na riešenie danej úlohy, a taktiež sme uviedli rôzne možné praktické aplikácie danej úlohy. Na dvoch vybraných aplikáciách sme otestovali nami naprogramované metódy.

V jednotlivých častiach sme sa oboznámili s rôznymi poznatkami a vlastnosťami úlohy l_1 regulovaných najmenších štvorcov. Taktiež sme uviedli mnohé aplikácie daného optimalizačného problému, z ktorých sme sa bližšie venovali l_1 filtrovaniu a problematike Truss Topology Design. V druhej časti práce sme si všeobecne predstavili dve konkurenčné metódy konvexnej optimalizácie, ktoré sa dajú na numerické riešenie daných úloh použiť. Konkrétne sa jedná o Proximal gradient metódu a Metódu vnútorného bodu. Metódy sme sa neskôr snažili čo najlepšie prispôbiť na riešenie našich úloh. V prípade l_1 filtrovania sme primárno-duálnu metódu vnútorného bodu použili na riešenie duálnej úlohy daného problému, ktorá bola v tvare kvadratického programovania. Proximal gradient metóda bola použitá na riešenie ekvivalentnej formulácie l_1 filtra. Numerické experimenty jasne ukázali, že na riešenie l_1 filtrovania je vhodnejšia metóda vnútorného bodu, ktorá narozdiel od proximal gradient metódy v tomto prípade pracuje s riedkymi maticami. Na riešenie Truss topology design problému sme proximal gradient metódu použili na pôvodnú neohraničenú formuláciu problému, zatiaľ čo metódu vnútorného bodu sme aplikovali na ekvivalentnú formuláciu v tvare kvadratického programovania. Už pri rozmeroch úlohy 480×16502 bolo možné pozorovať výhodu proximal gradient metódy voči metóde vnútorného bodu. Proximal gradient metódy mali oveľa menej časovo náročnú iteráciu a konvergovali k optimálnemu riešeniu rýchlejšie ako metóda vnútorného bodu. Nízku hodnotu účelovej funkcie dosiahli za pomerne malý čas. Pri aplikovaní metód na vysokorozmernú úlohu s 186000 prípustnými priečkami sme zistili, že hľadanie Newtonovho kroku v metóde vnútorného bodu je tak náročné, že algoritmus nedokázal vypočítať ani prvú iteráciu. Proximal gradient metóda si však viedla pomerne dobre, za 4 hodiny zvládla 35000 iterácií a ponúkla suboptimálne riešenie, ktoré malo po vykreslení riedku, mostovú štruktúru. Slabina metódy vnútorného bodu by sa mohla odstrániť, tým že by sme hľadali len aproximácie Newtonovho kroku pomocou tzv. Truncated Newton metódy ako sa uvádza v článku [3].

Literatúra

- [1] P. Gong C. Zhang: A Fast Dual Projected Newton Method for l1-Regularized Least Squares, Tsinghua University, Beijing, 2011

<http://ijcai.org/papers11/Papers/IJCAI11-216.pdf>
- [2] M. Schmidt: Least Squares Optimization with L1-Norm Regularization, CS542B Project Report, December 2005

<http://www.di.ens.fr/~mschmidt/Software/lasso.pdf>
- [3] S. J. Kim: An Interior-Point Method for Large-Scale l1-Regularized Least Squares, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 1, NO. 4, DECEMBER 2007

http://www.stanford.edu/~boyd/papers/pdf/l1_ls.pdf
- [4] S. J. Kim: l1 Trend Filtering, SIAM Review, May 2009

http://www.stanford.edu/~gorin/papers/l1_trend_filter.pdf
- [5] H. Yamada: Japan's output gap estimation and l1 trend filtering, Empir Econ, June 2012
- [6] A. B. Tal and A. Nemirovski: Lectures on modern convex optimization, MPS-SIAM Series on Optimization, Philadelphia, 2001
- [7] A. B. Tal and A. Nemirovski: Robust Truss Topology Design via Semidefinite Programming, November 2004
- [8] R. Tibshirani: Regression shrinkage and selection via the lasso, J. Royal. Statist. Soc B., Vol. 58, No. 1, 1996

<http://www-stat.stanford.edu/~tibs/lasso/lasso.pdf>
- [9] P. Richtárik and M. Takáč: Efficient Serial and Parallel Coordinate Descent Methods for Huge-Scale Truss Topology Design, June 2011

<http://www.maths.ed.ac.uk/~richtarik/ttd2011.pdf>
- [10] M. Kočvara: Truss topology design with integer variables made easy, Preprint 2010/09, School of MMathematics, University of Birmingham, UK, May 2010

http://www.optimization-online.org/DB_FILE/2010/05/2614.pdf
- [11] L. Vandenbergh: Algorithms for large-scale convex optimization - lecture notes, course at Technical University of Denmark, August 23–27, 2010

<http://www.ee.ucla.edu/~vandenbe/shortcourses.html>

- [12] M. Hamala: Konspekty k prednáškam z nelineárneho programovania
- [13] S. Boyd and L. Vandenberghe, Subgradients, Notes for EE364b, Stanford University, 2007
- [14] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge university press, 2004
www.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf
- [15] D. P. Bertsekas, Lecture slides on convex analysis and optimization based on 6.253 class lectures at the Mass. Institute of technology Cambridge, spring 2010
web.mit.edu/dimitrib/www/Convex_Slides_2010.pdf
- [16] M. Halická: Dvadsať rokov moderných metód vnútorného bodu, Bratislava, 2004
<http://www.iam.fmph.uniba.sk/institute/halicka/clanoklatex.pdf>
- [17] L. Vandenberghe: First-order methods - lecture notes, OPTEC/K.U.Leuven, July 6–7, 2009
<http://www.ee.ucla.edu/~vandenbe/shortcourses.html>
- [18] M. Hamala, M. Trnovská: Nelineárne programovanie, teória a algoritmy, EPOS Bratislava 2013
- [19] M. Trnovská: Konvexna optimalizácia - Prednášky, Bratislava 2013
<http://www.iam.fmph.uniba.sk/institute/trnovska/konvexoptim.html>
- [20] H. Ahumada: Hodrick-Prescott Filter in Practice, 1999
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.3365>
- [21] R. Kukumberg: Proximal-gradient, metóda konvexného programovania
<http://www.iam.fmph.uniba.sk/efm/bakalarky/2011/kukumberg/bakalarka.pdf>
- [22] J. Claerbout and F. Muir. Robust modeling of erratic data. *Geophysics*, 38(5):826–844, 1973
- [23] H. Taylor, S. Banks, and J. McCoy. Deconvolution with the l_1 norm. *Geophysics*, 44(1):39–52, 1979
- [24] S. Chen and D. Donoho. Basis pursuit. In *Proceedings of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44, 1994
- [25] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

- [26] J. Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, 2005.
- [27] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [28] D. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions On Information Theory*, 52(1):6–18, 2006.
- [29] A. d’Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming, 2005. In L. Saul, Y. Weiss and L. Bottou, editors, *Advances in Neural Information Processing Systems*, 17, pp. 41-48, MIT Press.
- [30] D. Donoho, I. Johnstone, G. Kerkycharian, and D. Picard. Wavelet shrinkage: Asymptopia? *J. R. Statist. Soc. B.*, 57(2):301–337, 1995.
- [31] M. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152(1):341–365, 2006.
- [32] A. Hassibi, J. How, and S. Boyd. Low-authority controller design via convex optimization. In *Proceedings of the IEEE Conference on Decision and Control*, pages 140–145, 1999.
- [33] J. Dahl, V. Roychowdhury, and L. Vandenberghe. Maximum likelihood estimation of Gaussian graphical models: Numerical implementation and topology selection.
- [34] E. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [35] S. Sardy, A. Bruce, and P. Tseng. Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries. Technical report, Seattle, WA, 1998.
- [36] Joshua Goodman. Exponential priors for maximum entropy models. In *HLT-NAACL*, pages 305–312, 2004.
- [37] C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.
- [38] A.V. Fiacco, G.P. McCormick: *Nonlinear programming: Sequential Unconstrained Minimization Techniques*

6 Príloha - zdrojové kódy programov

```
%Proximal operator l1 filter
function [x]=proxl11(x,t)
m=size(x);
x(3:m)=sign(x(3:m)).*max(0,abs(x(3:m))-t);
%%%%
%PDMVB l1 filter
function [xi]=l1ipm(y,lambda)
%parametre
e=1e-9;
ef=1e-9;
alfa=0.01;
beta=0.5;
mu=2;
maxls=30;
n=size(y,1);

D=gallery('tridiag',ones(n-1,1),-2*ones(n,1),ones(n-1,1));
D([1,n],:)=[];
DDT=D*D';
Dy=D*y;

j=ones(n-2,1);
v=zeros(n-2,1);
u1=j;
u2=j;

t=1e-10;
f1=v-lambda*j;
f2=-v-lambda*j;
gap=Inf;
mmax=1000;
for i=1:mmax
    t=max(t,2*n*mu/gap);
    iJ1=sparse(1:(n-2),1:(n-2),u1./f1);
    iJ2=sparse(1:(n-2),1:(n-2),u2./f2);
    s1=1/t*(1./f1);
    s2=1/t*(1./f2);

    dv=(DDT-iJ1-iJ2)\(-DDT*v+Dy+s1-s2); %newton
    du1=-u1-s1-iJ1*dv;
    du2=-u2-s2+iJ2*dv;

    rd=DDT*v-Dy+u1-u2;
    rc=[-u1.*f1-1/t; -u2.*f2-1/t];
    res=[rd;rc];
    %LINE SEARCH
    neg1 = (du1 < 0);
```

```

neg2 = (du2 < 0);
s = 1;
if (any(neg1))
    s = min( s, min(-u1(neg1)./du1(neg1)) );
end
if (any(neg2))
    s = min( s, min(-u2(neg2)./du2(neg2)) );
end
s=0.99*s;
for li = 1:maxls
    nv    = v  + s*dv;
    nu1   = u1 + s*du1;
    nu2   = u2 + s*du2;
    nf1   = nv-lambda*j;
    nf2   =-nv-lambda*j;
    %nove rezidua
    nrd=DDT*nv-Dy+nu1-nu2;
    nrc=[-nu1.*nf1-1/t; -nu2.*nf2-1/t];
    nres=[nrd;nrc];
    if ( max(max(nf1),max(nf2)) < 0 && ...
        norm(nres) <= (1-alfa*s)*norm(res) )
        break;
    end
    s = beta*s;
end
% aktualizacia
v = nv; u1 = nu1; u2 = nu2; f1 = nf1; f2 = nf2;
gap=-f1'*u1-f2'*u2;
% stop
if ((gap <= e)&&(norm(rd)<=ef))
    status = 'vyriesene';
    disp(status);
    xi=y-D'*v;
    1/2*norm(xi-y)^2+lambda*sum(abs(D*xi))
    i
    return;
end;
end
xi=y-D'*v;
1/2*norm(xi-y)^2+lambda*sum(abs(D*xi))
if (i >= mmax)
    status = 'dosiahnuty maxit';
    disp(status);
    return;
end
%%%%%%
%APGM l1 filter
function [x,k]=accwh(b,lambda,cas)
n=size(b,1);

```



```

%tvorba matice A
A1=zeros(n-1);
for i=1:n-1
    A1(i:n-1,i)=1:n-i;
end
A=zeros(n);
A(:,1)=ones(n,1);
A(2:n,2:n)=A1;
t=1/max(svd(A))^2;
G=A'*A;
h=A'*b;
x0=zeros(n,1);
%cas=30;
x=x0;
y=x0;
k=0;
t1 = tic;
while (toc(t1)<cas)
    k=k+1;
    xs=x;
    gg=G*y-h;
    [x]=proxl11(y-t*gg,t*lambda);
    y=x+((k-1)/(k+2))*(x-xs);
end
x=A*x;
%%%%%
%Proximal operator l1 norma
function [x]=proxnorml1(x,t)
x=sign(x).*max(0,abs(x)-t);
%%%%%
%Generator mriezky
function [H,S,l] = gen(r,c)
%vrcholy a ich suradnice
p=0;
S=zeros(r*c,2);
for i=1:r
    for j=1:c
        p=p+1;
        S(p,:)=[j-1,i-1];
    end
end %S(1,:) ...prvy vrchol atd
D=zeros(r*c);
for i=1:r*c
    for j=1:r*c
        D(i,j)=norm(S(i,:)-S(j,:));
    end
end
m=D(1,r*c);
vzd=unique(D(D<(m/2+1)& D~=0));

```

```

mm=size(vzd,1);

index=0;
for i=1:mm
    w=0;
    while (w+2)*vzd(i)<=m
        w=w+1;
        index=index+1;
        nv(index)=vzd(i)*(w+1);
    end
end
nv=unique(nv);
mm=size(nv,2);
l=0;
for i=1:r*c
    for j=i+1:r*c
        pp=0;
        for k=1:mm
            if (D(i,j)~=nv(k))
                pp=pp+1;
            end
        end
        if(pp==mm)
            l=l+1;
            H(1,:)=[i,j];
        end
    end
end
end
end
%%%%%
%Uloha
[H,S,l]=gen(6,39);

kappa=200;
lambda=0.0001;
%tvorba matice A
A=zeros(6*39*2,1);
for i=1:l
    a=H(i,1);
    b=H(i,2);
    beta=sqrt(kappa)*(S(b,:)-S(a,:))*((norm(S(b,:)-S(a,:)))^(-2));
    A([2*a-1,2*a],i)=-beta;
    A([2*b-1,2*b],i)=beta;
end
A([1,2],:)=zeros(2,1);
A([14*2-1,28],:)=zeros(2,1);
A([26*2-1,52],:)=zeros(2,1);
A([39*2-1,78],:)=zeros(2,1);
b=zeros(6*39*2,1);

```

```

b(82:2:154,1)=1;
x0=zeros(1,1);

t=1/max(svd(A))^2;

A=sparse(A);
b=sparse(b);

G=A'*A;
h=A'*b;
%%%%%
%Vykreslovanie
function []=kresli(x4,H,S)
plot(S(:,1),S(:,2),'ro')
axis([-1,39,-0.2,5.2])
l=size(H,1);
hold on
for i=1:l
    if abs(x4(i))>0.00001
        hrubka=abs(x4(i))*9;
        a=H(i,1);
        b=H(i,2);
        sa=S(a,:);
        sb=S(b,:);
        xx=[sa(1),sb(1)];
        yy=[sa(2),sb(2)];
        plot(xx,yy,'b','LineWidth',hrubka)
    end
end
plot([1,37],[1,1],'g','LineWidth',5)
hold off
%%%%%
%PDMVB TTD
function [x,u]=pd(A,b,lambda)
%parametre
e=1e-13;
ef=1e-13;
alfa=0.01;
beta=0.5;
mu=5;
maxit=500;
maxls=20;

G=A'*A;
h=A'*b;
[~,n]=size(A);

j=ones(n,1);

```

```

%premenne
x=zeros(n,1);
u=0.2*j;
u1=j;
u2=2*j;
lj=lambda*j;

t=1e-5;
f1=x-u;
f2=-x-u;
gap=Inf;

for i=1:maxit
    t=max(t,2*n*mu/gap);
    iJ1=sparse(1:n,1:n,u1./f1);
    iJ2=sparse(1:n,1:n,u2./f2);
    Jm=iJ1-iJ2;
    Jp=iJ1+iJ2;
    iiJ=sparse(1:n,1:n,1./diag(Jp));
    s1=1/t*(1./f1);
    s2=1/t*(1./f2);
    w=2*(G*x-h);

    %vypocitanie smeru
    dx=(2*G-Jp+Jm*iiJ*Jm)\((-w+s1-s2-Jm*iiJ*(lj+s1+s2)); %newton
    du=iiJ*(lj+s1+s2+Jm*dx);
    du1=-u1-s1+iJ1*(-dx+du);
    du2=-u2-s2+iJ2*(dx+du);

    rd=[w+u1-u2;lj-u1-u2];
    rc=[-u1.*f1-1/t; -u2.*f2-1/t];
    res=[rd;rc];
    %LINE SEARCH
    neg1 = (du1 < 0);
    neg2 = (du2 < 0);
    s = 1;
    if (any(neg1))
        s = min( s, min(-u1(neg1)./du1(neg1)) );
    end
    if (any(neg2))
        s = min( s, min(-u2(neg2)./du2(neg2)) );
    end
    s=0.99*s;
    for li = 1:maxls
        nx = x + s*dx;
        nu = u+s*du;
        nu1 = u1 + s*du1;
        nu2 = u2 + s*du2;
    end
end

```

```

        nf1 = nx-nu;
        nf2 = -nx-nu;
        %nove rezidua
        nrd = [2*(G*nx-h) + nu1 - nu2;lj-nu1-nu2];
        nrc = [-nu1.*nf1-1/t; -nu2.*nf2-1/t];
        nres = [nrd; nrc];

        if ( max(max(nf1),max(nf2)) < 0 && ...
            norm(nres) <= (1-alfa*s)*norm(res) )
            break;
        end
        s = beta*s;
    end
    % aktualizacia
    x = nx; u=nu; u1 = nu1; u2 = nu2; f1 = nf1; f2 = nf2;

    gap=-f1'*u1-f2'*u2;
    % stop
    if ((gap <= e)&&(norm(rd)<=ef))
        status = 'vyriesene';
        disp(status);
        return;
    end;
end
if (i >= maxit)
    status = 'dosiahnuty maxit';
    disp(status);
    return;
end
%%%%%
%Proximal grad. pre TTD
function [x1,x2,x3,x4]=PG(G,h,A,b,x0,lambda,t)
c=6000;
cas=c/4;
lambda=lambda/2; %ostatne je pre1/2*norm
%Proximal Gradient
x=x0;
k=0;
t1 = tic;
while (toc(t1)<cas)
    k=k+1;
    gg=G*x-h;
    [x]=proxnorm11(x-t*gg,t*lambda);
end
x1=x;
f1=norm(A*x-b)^2+2*lambda*sum(abs(x));
%Zrychleny PG
x=x0;
y=x0;

```

```

k=0;
t1 = tic;
while (toc(t1)<cas)
    k=k+1;
    xs=x;
    gg=G*y-h;
    [x]=proxnorm11(y-t*gg,t*lambda);
    y=x+((k-1)/(k+2))*(x-xs);
end
x2=x;
%DESCENT PG
x=x0;
y=x0;
k=0;
t1 = tic;
while (toc(t1)<cas)
    k=k+1;
    xs=x;
    gg=G*y-h;
    [z]=proxnorm11(y-t*gg,t*lambda);
    fs=norm(A*xs-b)^2+2*lambda*sum(abs(xs));
    fn=norm(A*z-b)^2+2*lambda*sum(abs(z));
    if(fn<=fs)
        x=z;
    end
    v=xs+((k+1)/2)*(z-xs);
    y=(k/(k+2))*x+(2/(k+2))*v;
end
x3=x;
%NAS PG
x=x0;
y=x0;
k=0;
t1 = tic;
while (toc(t1)<cas)
    k=k+1;
    xs=x;
    gg=G*y-h;
    [z]=proxnorm11(y-t*gg,t*lambda);
    fs=norm(A*xs-b)^2+2*lambda*sum(abs(xs));
    fn=norm(A*z-b)^2+2*lambda*sum(abs(z));
    if(fn<=fs)
        x=z;
    else
        gg=G*xs-h;
        x=proxnorm11(xs-t*gg,t*lambda);
    end
    y=x+((k-1)/(k+2))*(x-xs);
end

```